ҚАЗАҚСТАН РЕСПУБЛИКАСЫ
БІЛІМ ЖӘНЕ ҒЫЛЫМ МИНИСТРЛІГІ

Б.БӨРІБАЕВ, Г.А.МАДЬЯРОВА

Web-технологиялар

ОҚУЛЫҚ

Қазақстан Республикасының Білім және ғылым министрлігі бекіткен

Алматы, 2011

УДК 004.7(075.8)
ББК 32.973-018.1 я 73
Б 79

Пікір берушілер:

У. А. Түкеев - т.ғ.д., профессор
(әл-Фараби атындағы ҚазҰУ-ң «Ақпараттық жүйелер» кафедрасы);
А. Ж. Сейкетов - профессоры, т.ғ.д.
(«Ақпараттық технологиялар» университеті).

Б.Бөрібаев, Г.А. Мадьярова
Б 79 Web-технологиялар. Оқулық – Алматы, ЖШС РПБК «Дәуір» 2011 ж., 319 бет.

ISBN 978-601-217-201-0

Оқулық қазіргі кездегі web-сайттар құруға арналған кең қолданыстағы HTML, CSS, JavaScript, Perl және PHP тілдерін оқып-үйренуге арналған. Оқулықта программалау тілдерінен көптеген мысалдар келтіріліп, оларды орындау жолдары қарастырылады. Әрбір тарау бақылау сұрақтары мен тапсырмалармен толықтырылған. Оқулықты студенттермен қатар осы тілдерді өздігінен үйренушілер де пайдалана алады.

ISBN 978-601-217-201-0

УДК 004.7(075.8)
ББК 32.973-018.1 я 73

КІРІСПЕ

Қазіргі кезде Интернет желісінің кең өріс алуына байланысты web-сайттар жасау жұмыстары өзекті іске айналды. Әрбір мекеме немесе жеке тұлға өзі не бизнесі жайлы жарнамалық мәліметтерді жылдам құрастыратын, өзгертетін деңгейде болуы тиіс. Бұл оқулықтың мақсаты – студенттерді қарапайым сайттар жасауға үйрету болып табылады. Мұнда HTML, CSS, JavaScript, Perl және Php тілдерін пайдаланудың негізгі технологиялары қарастырылған.

Бірінші тарауда web-технологиялардың қолдану орталары мен негіздері қарастырылады.

Екінші тарауда web-қосымшаларын құру технологиялары сипатталып, HTML гипермәтінді белгілеу тілінің мүмкіндіктері келтірілген және оның web-парақтар жасай алатын басқа тілдермен байланысу жолдары жайлы айтылады.

Үшінші тарау стильдердің сатылы кестелерін (Cascade Style Sheets) қолдану мәселелері туралы жазылған, онда CSS сипаттамаларының синтаксисі, стильдерді құжат ішінде немесе сыртында беру, CSS атрибуттарының жиі қолданылатын атрибуттары, элементтерді позициялау тәсілдеріне мысалдар келтірілген.

Төртінші тарау динамикалық web-парақтар жасай алатын JavaScript программалау тіліне арналып, World Wide Web ортасында осы тілді пайдалану жолдарынан мысалдар мен программалық кодтар берілген.

Бесінші тарауда желілердегі сервер жағынан әр түрлі іс-әрекеттер атқара алатын Perl тілінің мүмкіндіктері және PHP тіліне қысқаша мағлұматтар берілген.

Барлық материалдар практикалық есептер мен мысалдар арқылы түсіндіріледі. Көптеген мысалдарды тікелей Internet Explorer программасы арқылы орындап тексеруге болады. Бұл оқулық web-сайттар жасауға арналған мемлекеттік тілде шыққан алғашқы кітаптардың бірі болғандықтан, пікірлеріңіз бен ұсыныстарыңызды редакцияға хабарлауларыңызды сұраймыз және олар болашақтағы оқулықтардың сапасын жақсартады деген үміттеміз.

1 WEB-ТЕХНОЛОГИЯЛАРДЫ ҚОЛДАНУ ОРТАЛАРЫ

1.1 Интернеттегі клиент-серверлік архитектура

Бірнеше компьютерлердің келісімді жұмысын қамтамасыз ету үшін оларды өзара байланыстыру қажет. Ол үшін компьютерлердің арасында өзара ақпараттық байланыс ұйымдастыру қажеттігі туындайды.

Әр түрлі компьютерлердің арасында өзара ақпараттық байланыс ұйымдастыру келесі мәселелерді шешуге мүмкіндік береді:

- ақпараттарды өте үлкен қашықтықтарға тарату (жүздеген, мыңдаған километрге);
- өте бағалы, құнды ақпараттық, программалық және аппараттық ресурстарды бірнеше компьютерлердің ортақ пайдалану; Бұл ресурстар - өте қуатты үрдісор, сиымдылығы мол жинақтауыш, жоғары өнімді баспа құралдары, мәліметтер базасы, программалық жабдықтар және т.б. болуы мүмкін.
- мәлімет жинақтауыштармен жұмыс істегенде компьютерлер арасында өзара ақпарат алмасу;
- ауқымды жобамен бірігіп жұмыс істегенде, пайдаланушы адамдарға жалпы мәліметтердің соңғы көшірмесін кез келген уақытта алып отыру.

Компьютерлер арасындағы байланысты ұйымдастырудың негізгі үш тәсілі бар:

1. қатар орналасқан екі компьютерді арнайы кабель көмегімен олардың коммуникациялық порттары арқылы біріктіру;
2. модем арқылы сым немесе спутниктік байланыс торабы көмегімен бір компьютерден екіншісіне мәліметтер беру;
3. компьютерлерді компьютерлік желіге біріктіру.

Көп жағдайларда екі компьютер арасында байланыс ұйымдастырғанда бір компьютерге ресурстарды жеткізуші (программалар, мәліметтер және т.б.) рөлі, ал екіншісіне осы ресурстарды қолданушы рөлі бекітіледі. Мұндай жағдайда бірінші компьютер **сервер,** ал екіншісі - **клиент** немесе жұмыс станциясы деп аталады.

Сервер – сыртқы жад көлемі өте үлкен, өнімділігі өте жоғары компьютер. Ол бірігіп пайдаланатын бағасы жағынан өте қымбат ресурстарды үлестіруді басқару арқылы басқа компьютерлерге қызмет көрсетуді қамтамасыз етеді.

Клиент – сервердің қызметтерін пайдалана алатын кез келген компьютер.

Кейбір жағдайларда бір компьютер сервер әрі клиент те бола алады. Ол компьютер өз ресурстарын және сақталып отырған мәліметтерін басқа компьютерлерге береді және сонымен қатар олардың ресурстары мен мәліметтерін пайдалана алады.

Клиент деп пайдаланушы атынан серверден қызмет түрлерін алатын, қолданбалы программаларды да айтуға болады.

Компьютерлік желі дегеніміз – бір-бірімен мәлімет алмаса алатын, кем дегенде екі компьютердің байланыс жасауына арналған ақпарат өңдеудің тармақталған жүйесі.

Басқаша айтқанда желі деп дербес компьютерлердің, принтер, модем, факсимильдік аппарат тәрізді есептеу құрылғыларының бір-бірімен байланысқан жиынын айтады. Компьютерлік желі әрбір қызметкерге басқалармен мәлімет алмасып құрылғыларды ортақ пайдалануға, қашықта орналасқан қуатты компьютерлердегі мәліметтер базасымен қатынас құруға және тұтынушылармен тұрақты байланыс жасауға мүмкіндік береді.

Желі құрамына кіретін компьютерлер мынадай жұмыстар атқарады:

- желімен қатынас құруды ұйымдастыру;
- олардың арасында мәлімет алмасуды басқару;
- желі тұтынушыларына есептеу құрылғыларын пайдалануға беріп, оларға әртүрлі қызмет көрсету.

Компьютерлік желілерді түйіндер (компьютерлер және желілік жабдықтар) мен оларды байланыстыратын тараулардың (байланыс арналары) жиыны ретінде қарастырады. Желі тарауы – қатар орналасқан екі түйінді қосатын жол. Түйіндер **шеткі** (тек бір тараудың соңында орналасқан), **аралық** (біреуден көп тараулардың шеттерінде орналасқан) және **шектес** (мұндай түйіндер басқа түйіндері жоқ бір ғана жолмен қосылған) болып бөлінеді.

Интернет сөзі Interconnected networks (байланысқан жүйелер) терминінен шыққан, яғни техникалық көзқараспен – кіші және ірі желілердің бірігуі, ал кең мағынасында - бір-бірімен мәліметтер алмасатын жер жүзіндегі миллиондаған компьютерлер арасында бөлінген ақпараттық кеңістік деуге болады. Интернет – бұл қайталанбас жетістіктерді толығымен біріктіретін технология болып табылады. Пайдаланушылар көп жағдайда Интернетті желінің ақпараттық құрамы ретінде түсінеді. Интернет ең күшті және тәуелсіз ақпарат қоры, байланыстың сенімді және жедел тәсілі, жер жүзіндегі миллиондаған адамдардың шығармашылық түрде өзін өзі көрсету және ақпараттық технологияларды дамыту негізі болып табылады.

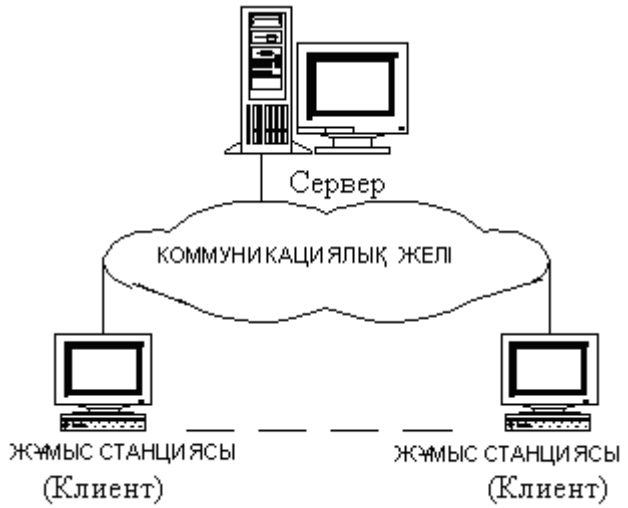
Интернетке қосылған барлық компьютерлерді екі түрге бөледі: серверлер және клиенттер. Бір компьютерде серверді де клиентті де орнату мағынасында бөлу онша қатаң жүргізілмейді. Жергілікті компьютерде Web-сервер орнатылуы мүмкін және осыған қарамастан, дәл осы компьютерде браузермен және пошталық клиентпен де жұмыс жасауға болады.

Сервердің басты міндеті – сервиске қайсыбір клиент сұраныс жібермейінше әр кезде жұмыс жасап және күту жағдайында болу. Серверде сұраныстардың көптігінен оның жұмысы баяулап және белгілі бір сұраныстарға қызмет көрсетуді тежейді. Серверге сұраныс белгілі бір хаттама шегінде болады. Хаттама дегеніміз – бұл желіде компьютерлер арасында байланысты қамтамасыз ететін стандарттар жиыны. Серверлік программалар клиенттік программаларға қызмет көрсету үшін компьютердің аппаратты ресурстарын қолданады. Клиент-программа сұраныс құрып, оны

желі арқылы белгілі бір адреске жібереді және алдын ала белгіленген хаттама арқылы сервер-программамен өзара байланысады. Бір компьютерде бірнеше серверлік программаларды орнатуға болады. Әрбір сервер-программа үшін өзіндік клиент-программасы болады. Осылай, Web-клиент Web-серверге, пошталық клиент – пошталық серверге хабар береді және т.с.с. Серверлік программа әрқашан сұранысты орындауға дайын болу керек, сол себепті сервер-программа жұмыс жасайтын компьютерлерге сенімділікке және өнімділігіне қатысты жоғары талаптар қойылады. Клиенттік компьютердің жұмысының тұрақтылығы бір адамның жұмысына әсер ететіндіктен, олардың жұмысына сенімділігіне байланысты аз талаптар қойылады, ал аппаратты сервердің жұмысының сенімділігіне байланысты көптеген клиенттердің жұмысының жүргізілуі тәуелді болады. Жоғарыда көрсетілген тәсіл (клиент-серверлік архитектура) дербес компьютер пайдаланушысына өзінің жұмыс үстелінен Интернетке қосылған миллиондаған серверлердің ресурстарына қол жеткізуге мүмкіндік береді.

Желінің архитектурасы желінің негізгі элементтерін анықтайды, оның жалпы логикалық ұйымдастыруын, техникалық, программалық қамтамасыз етілуін, кодтау әдістерін сипаттайды. Архитектура сондай-ақ пайдаланушының жұмыс істеу принциптерін және интерфейсін анықтайды.

Клиент-серверлік архитектурада қызметтік әрекеттер қолданбалы программалар кешені арқылы жүзеге асырылады, оларға сәйкес түрлі қолданбалы үрдістер орындалады. Клиент-серверлік архитектураның құрылымы 1.1-суретте көрсетілген.



1.1- сурет. Клиент-серверлік архитектура

Клиент-серверлік архитектурада объектілердің төрт тобы бар: клиенттер, серверлер, деректер және желілік қызметтер. Клиенттер пайдаланушылардың жұмыс орнында, жүйелерде орналасқан. Деректер негізінен серверлерде сақталады. Желілік қызметтер ортақ қолданылатын

серверлер және деректер болып табылады. Желілік қызметтер деректерді өңдеу процедураларын басқарады.

Клиент-серверлік архитектураның мынадай артықшылықтары бар:

- жұмыс станцияларының саны көп желілерді ұйымдастыруға мүмкіндік береді;
- желілік әкімшілік етуді жеңілдететін пайдаланушылардың есептік жазбаларын, қауіпсіздікті және желіге қатынауды орталықтан басқаруды қамтамасыз етеді;
- желілік қорларға тиімді қатынауға мүмкіндік береді;
- пайдаланушылар бір ғана құпия сөз арқылы пайдалану құқығы бар барлық қорларға қол жеткізе алады.

Клиент-серверлік архитектураның артықшылықтарымен қатар, кемшіліктері де бар:

- сервер бұзылған кезде желі жұмыс жасамай қалады, кем дегенде желілік қорлар жоғалады;
- әкімшілік ету үшін жоғары мамандандырылған адам қажет;
- желілер мен желілік құрылғылары бағасы жағынан қымбатқа түседі.

Желінің архитектурасын таңдау

Желінің архитектурасын таңдау желінің тағайындалуына, жұмыс станцияларының санына және онда орындалатын жұмысқа байланысты.

Біррангілі желіні таңдау керек, егер:

- пайдаланушылардың саны 10-нан аспаса;
- барлық машиналар бір-біріне жақын орналасса;
- қаржылай мүмкіндік жоғары болмаса;
- ДБ сервері, факс-сервер немесе басқа арнайы серверді қажеттілігі болмаса;
- орталықтырылған әкімшілік етудің қажеті болмаса немесе оған мүмкіндік болмаса.

Клиент-серверлік желіні таңдау қажет, егер:

- пайдаланушылардың саны 10-нан артық болса;
- орталықтандырылған басқару, қауіпсіздік, қорларды басқару немесе қорға көшірме жасау қажет болса;
- арнайы сервер қажет болса;
- ауқымды желіге қатынау қажет болса;
- пайдаланушылар деңгейінде қорларды бөлу қажет болса.

1.2 Компьютерлік желілер

Барлық желілер бір-біріне ұқсас болғанмен, олар екі түрге бөлінеді:

- біррангілі;
- ерекшеленген сервер негізіндегі.

Мүмкіндіктері әр түрлі болатындықтан, біррангілі желілер мен ерекшеленген сервері бар желілер арасында принципиалды айырмашылық бар. Желінің түрін таңдау көптеген факторларға байланысты, мысалы:

- кәсіпорын көлемі;
- қауіпсіздіктің қажетті дәрежесі;
- бизнес түрі;
- әкімшілік қолдаудың ыңғайлылығы;
- желілік трафиктің көлемі;
- желілік қолданушылардың мұқтаждығы;
- қаржыландыру деңгейі.

Біррангілі желілер

Біррангілі желіде барлық компьютерлердің құқығы бірдей болады, яғни компьютерлер арасында иерархия және ерекшеленген сервер жоқ. Әдетте әрбір компьютер сервердің де, клиенттің де рөлін атқара береді; басқаша айтқанда барлық желі үшін жауапты жеке компьютер жоқ. Пайдаланушы өз компьютеріндегі ортақ пайдалануға болатын мәліметті анықтауды өзі шешеді.

Әдетте біррангілі желілердегі компьютер саны 10-нан артық болмайды. Пайдаланушылардың ұжымы кішкене болғандықтан біррангілі желіні жұмысшы топ деп те атайды.

Әрбір компьютер бірмезгілде сервердің де, клиенттің де рөлін атқара алатын болғандықтан, өте қуатты орталық серверді және басқа да күрделі желілер үшін керекті құрылғыларды орнатудың қажеттілігі жоқ. Сондықтан сервер негізіндегі желілерге қарағанда біррангілі желілердің бағасы да арзан болады.

Сонымен қатар біррангілі желілерде желілік программалаық жабдықтардың қауіпсіздігі мен олардың жұмыс өнімділігіне қойылатын талаптар да ерекшеленген сервері бар желілерге қарағанда әлдеқайда төмендеу болады. Ерекшеленген сервері бар желілерде серверлер тек сервер қызметін атқарады да, клиент немесе жұмысшы станциясы қызметін атқара алмайды.

Microsoft Windows NT Workstation, Microsoft Windows for Workgroups және Microsoft Windows сияқты операциялық жүйелерде біррангілі желілерді қолдау енгізілген. Сондықтан осы операциялық жүйелер орнатылған компьютерлерде біррангілі желіні ұйымдастыру үшін қосымша программалық жабдықтың қажеті жоқ.

Ерекшеленген сервері бар желілер

Біррангілі желіге 10 компьютерден артық қосылса, онда ол қойылған мақсатты шеше алмайды. Сондықтан көптеген желілердің конфигурациясы бөлек болады – олар ерекшеленген сервер негізінде жұмыс істейді. Ерекшеленген сервер деп тек сервердің рөлін атқаратын, ал клиент немесе жұмыс станциясы рөлінде пайдаланылмайтын серверді айтады. Сервер желілік клиенттерден түсетін сұранысты жедел өңдеу және файлдар мен

каталогтардың қорғалуын арттыруға арналған. Сервер негізіндегі желілер өндірістік стандарттар қатарына жатады.

Желілер өлшемі және желілік трафик көлемі ұлғайғанда сервер сандарын да көбейткен дұрыс. Пайдаланушылардың қазіргі заманғы талаптарына сай келу үшін үлкен желілерді мамандандырылған, яғни бір бағытқа лайықталған етіп жасайды.

Желілердің жіктелуі

Желілерді ұйымдастыру әдісі бойынша *нақты* және *жасанды* желілер болып бөлінеді.

Жасанды желілер (псевдосети) компьютерлерді тізбектелген немесе параллель порттар арқылы байланыстыра алады. Мұндай желілерді нуль-модем (модем пайдаланылмайды) бойынша байланыс деп те атайды. Біріктірудің өзін нуль-модемдік деп атайды. Жасанды желілерді ақпараттарды бір компьютерден екінші компьютерге жөнелту үшін пайдаланады. MS DOS пен Windows операциялық жүйелерінде нуль-модемдік байланысты жүзеге асыру үшін арнайы программалар бар.

Негізгі кемшілігі – мәліметтерді жөнелту жылдамдығы төмен және тек екі компьютерді ғана байланыстыра алады.

Нақты желілер компьютерлерді коммутациялық және мәліметтерді жөнелтудің физикалық ортасының арнайы құрылғылары көмегімен байланыстырады.

Негізгі кемшілігі – қосымша құрылғылардың қажеттілігінде болып саналады.

Компьютерлік желілерді келесі объект топтары бойынша жіктеуге болады:

- Аймақтық таралымы;
- Ведомство құрамында болуы;
- Ақпаратты жөнелту жылдамдығы;
- Жөнелту ортасының түрлері;
- Топологиясы;
- Компьютерлердің өзара байланысының ұйымдастырылуы

Желілердің аймақтық таралымы бойынша жіктелуі

Жергілікті желі, LocaI Area Networк (LAN). Ережеге сай бір немесе бірнеше ғимараттарда жинақы орналасқан кәсіпорынның компьютерлерін біріктіреді. Жергілікті желінің көлемі аздаған километрден аспайды. Салыстырмалы түрде коммуникациялық құрылғылардың арзандығы және мәліметтер алмасу мен қарапайым алгоритмдерді қолдануда компьютерлер арасының аздаған арақашықтығын, жаңа жоғары сапалы байланыс сымдарының пайдалануын, экономикалық жағынан ақтайды. Қазіргі жергілікті желілердің өткізгіштік мүмкіндігі 1000 Мбиг/с-қа жетеді. Желілік ресурстарды қарауға кететін уақыт жұмыс станцияларының жергілікті ресурстарын қарауға кететін уақытпен пара-пар. Мәлімет алмасудың жоғары сапалылығы желінің Пайдаланушылар

тұтынуына кең көлемде қызмет көрсетеді, яғни жеке компьютерде іске асыру қымбатқа түсетін: файлдық қызмет, баспа, факс, электрондық поштаға, сканер, мәліметтер қоры және басқа да қызметтерге мүмкіндік береді. Байланыс арналары желінің компьютерлерімен бір уақытта қолданысқа түсе алады. Бұл жағдайда компьютерлік мәліметтер алмасуда байланыс сымдарына түсетін жүктеменің теңсіздігін түзететін мәліметтер алмасудың казіргі әдістері қолданылуы мүмкін (пакеттерді коммутациялау туралы алда айтамыз). Қосылатын компьютерлердің санына және сымның ұзындығына, олардың қолданатын технологиясына қатаң шектеу қойылатындықтан жергілікгі желілердің көлемділігі нашар болады.

Бүкіләлемдік желілер, Wide Area Network (WAN). Бір-бірінен әжептеуір қашықтықта орналасқан компьютерлерді біріктіреді. Жалпы жағдайда мұндай компьютер жер шарының кез келген нүктесінде болуы мүмкін. Бұл жағдайда байланыс сымын әр компьютерге жалғау экономикалық жағынан мүмкін емес. WAN желілерді ұйымдастыруда бұрыннан бар байланыс сымдары қолданылады, мысалы телефон сымдары. Бұл сымдар компьютерлік мәлімет алмасуда жақсы қолданыс тапты. Қымбат аппаратураны және арнайы күрделі алгоритмдер мен мәлімет алмасу процедураларын қажет ететін мұндай байланыс сымдарының сапасы әрине мардымсыз. Мәлімет алмасу жылдамдығы LAN желілерге қарағанда, анағұрлым төмен, көрсететін қызмет саны аз, байланыс арналарын қолдануға дәрменсіз, байланыс уақытында жеке дара компьютерлерге қолданылатын байланыс сымы. Бүкіләлемдік желілердің көлемділігі жақсы. Қосымша компьютерлердің қосылуы іс жүзінде тұтас желіге еш әсер етпейді.

Бүкіләлемдік және жергілікті есептеуіш желілердің ерекшеліктері

Жергілікті желілердің, бүкіләлемдік желілерден негізгі айырмашылығы сапалы байланыс сымдарының қолданылуында болып табылады. Басқа айырмашылықтар жұмыс істеу барысында туындайды. Бұған мысал ретінде дәстүрлі байланыс сымдарының бағасына жетіп қалған талшықты-оптикалық (оптоволоконный) техниканың қолданылуын айтуға болады. Мәліметтер алмасу жылдамдығы айтарлықтай өсіп, жергілікті желіде атқарылатын істер енді бүкіләлемдік желіде де кеңінен қолданыла бастады. Мұнда технологиялардың керісінше енгізілуі де байқалады. Мысалы бүкіләлемдік желілердің транспорттық технологиясы жергілікті желіде де қолданылады. Бөлінетін байланыс сымдарымен қатар жергілікті желі стандарттары жекелеген байланыс сымдарының жұмыстарын да қолдайды. Бүкіләлемдік ашық желілерге арналған ақпараттарды рұқсатсыз ашудан қорғау әдісі жергілікті желілерде де кең қолданыс тапқан. Бұл қажеттілік ертеректе бой түзеген жергілікті желілердің бүкіләлемдік желілерде қолданылуы арқылы бірігуінен туындап отыр. Жергілікті желіні пайдалануға іс жүзінде бүкіләлемдік желіні қолданушылардың кез келгенінің мүмкіндігі бар. Бүкіләлемдік желілердің жергілікті желілер әлеміне өтуі соңғы кездері жан-жақты дамуда, яғни бүкүләлемдік желілердің жұмысын іске асыру мақсатында жергілікті желілерге қойылатын талаптарды білдіретін

Интранет-технологиясы түсінігі пайда болды. Жергілікті желілердің көлемі тек аймақтық себептермен анықталатын болды.

Ведомство құрамында болуына байланысты:

- ведомствалық – бір ұйым құрамына кіреді және соның аумағына орналасады;
- мемлекеттік – мемлекеттік құрылымдарда пайдаланылатын желілер.

Ақпараттарды жөнелту жылдамдығы бойынша:

- төменгі жылдамдықты (10 Мбит/с-қа дейін);
- орташа жылдамдықты (100 Мбит/с-қа дейін);
- жоғары жылдамдықты (100 М бит/с -тен жоғары)

Жөнелту ортасының түрлері бойынша:

- сым арқылы: коаксиальды, оптикалық-талшықты, қос ширатпа;
- сымсыз: радиоарна арқылы, инфрақызыл диапазонда ақпарат жөнелту.

Желінің көлеміне қарай жіктелуі

Жұмыс топтарының жергілікті желілері (Локальные сети рабочих групп) бір операциялық жүйенің басқаруымен жұмыс істейтін аздаған компьютерлерді біріктіреді. Желіде желілік қызметтерді атқаратын (мысалы файлдық сервер, баспа сервері, факс сервері) бір компьютер белгіленеді.

Бөлімдердің жергілікті желілері (Локалъные сети отделов) жеке бір бөлімнің компьютерлерін біріктіре алады. Компьютер сандары жұмыс топтарына қарағанда бірнеше есе көп болуы мүмкін. Желілік қызметтер жеке белгіленген компьютерлік серверлерге бөліне алады. Жергілікті желілердің екеуі де базалық технологияның біреуін пайдаланады. Бұл желілерді классикалық жергілікті желілер деп те атауға болады. Аймақтық жағынан олар бір-екі ғимаратты, яғни аздаған аумақты алады.

Кампустер желісі бірнеше майда желілерді бір үлкен желіге біріктіруді көздейді. Мұндай желілер үлкен аймақты алып жатады. Желілерді біріктіруде аппараттық және программалық қамсыздандыру, әр түрлі технологияларды интеграциялау мәселелерін шешу керек. Желінің едәуір бөлігі жұмыс топтары мен бөлімдер желілерінің айналасында жергіліктендірілген. Кампустер желісі жекелеген желілердің байланысын, қымбатқа түсетін жалпы желілік ресурстарға мүмкіндікті қамтамасыз ету мәселелерін шешеді. Біріктіру барысында бүкіләлемдік байланыстыру қолданылмайды.

Барлық уақытта да адамзаттың алдында ақпарат алмасу (тарату, беру) мәселесі болып отырды. Компьютердің пайда болуымен адамның қолы ақпаратты жинау мен өңдеудің қуатты құралына жетті және осыған байланысты ақпаратты тарату мәселесінің маңызы арта түсті. Мәліметтерді таратудың дәстүрлі арналары (телефон, телеграф, пошта, радио және т.б.) күннен-күнге геометриялық прогрессияда өсіп келе жатқан ақпараттың үлкен ағымын таратуға барлық уақытта дәрменді бола бермейді. Сондықтан компьютерлердің пайда болуымен қатар, олардың арасында ақпараттарды тасымалдау мәселесі де туды. Қазіргі уақытта әр түрлі есептеуіш машиналар

арасындағы байланысты ұйымдастыру кезеңі кез келген компьютерлік жүйенің қажетті және жеткілікті шарты болып табылады.

Желіні құру принциптері

Желі – бір-бірімен байланысқан программалық және аппараттық компоненттердің күрделі жүйесі. Аппараттық құралдардың ішінде компьютерлерді және коммуникациялық құралдарды бөліп қарастыруға болады. Программалық компоненттер желілік қосымшалар мен операциялық жүйелерден тұрады. Қазіргі уақытта желіде әр түрлі типтегі компьютерлер мен олардың сипаттамалары желінің мүмкіндіктерін анықтайды. Бірақ кейінгі кездері коммуникациялық құралдар да (кабельдік жүйелер, қайталамалар, көпірлер, бағыттамалар т.б.) аз роль ойнамайтын болды. Олардың қиындықтарын ескеріп, құнын және басқа да сипаттамаларын желінің жұмыс қабілетін қамтамасыз етуде арнайы жұмыстарды шешетін кейбір құрылғыларды компьютер деп те атауға болады. Нәтижелі жұмыс істеу үшін арнайы операциялық жүйелер қолданылады, олардың дербес операциялық жүйелерден айырмашылығы сол, желідегі компьютер жұмысын басқаруда арнайы тапсырмаларды шешуге арналған.

Желілік қосымшалар – бұл желілік операциялық жүйелер мүмкіндіктерін кеңейтетін қолданбалы программалар жиыны болып табылады. Солардың ішінде пошталық программаларды, ұжымдық жұмыс жүйесін, желілік мәліметтер қорын т.б. атауға болады. Желілік операциялық жүйелердің даму барысында желі қосымшаларының кейбір функциялары операциялық жүйенің кәдімгі функцияларына айналады. Желіге қосылған барлық құрылғыларды үш функционалдық топқа бөлуге болады:

1. жұмыс станциясы;
2. желі серверлері;
3. коммуникациялық тораптар.

Жұмыс станциясы (workstation) - бұл желіні пайдаланушылар жұмыс атқаратын, желіге қосылған дербес компьютер. Әрбір жұмыс станциясы өз операциялық жүйесін қолданады және өздерінің жергілікті файлдарын өңдейді. Әйтсе де бұл жағдайда қолданушының желі ресурстарын пайдалануына мүмкіндігі бар. Жұмыс станциясының үш типін айтуға болады:

- жергілікті дискілі жұмыс станциясы;
- дискісіз жұмыс станциясы;
- шалғайдағы жұмыс станциясы;

Дискілі жұмыс станциясында (қатты немесе жұмсақ) операциялық жүйе осы жергілікті дискіден жүктеледі. Дискісіз станция үшін операциялық жүйе файлдық сервер дискісінен жүктеледі. Мұндай мүмкіндік дискісіз станцияның желілік адаптеріне орналастырылған арнайы микросхема арқылы қамтамасыз етіледі.

Шалғайдағы жұмыс станциясы – бұл жергілікті желіге телекоммуникациялық арналар байланысы (мысалы телефон желісі) арқылы косылған станция

Желі сервері – бұл, желіге қосылған және желіні қолданушыларға белгілі бір қызмет жасайтын, мысалы ортақ қолданылатын мәліметтерді сақтау, баспаға беру, МҚБЖ-не деген сұранысты өңдеу, т.б. жұмыстарды атқаратын компьютер.

Файлдық сервер **–** желіні қолданушылардың мәліметтерін сақтайтын және осы мәліметтермен олардың жұмыс істеуіне мүмкіндік беретін компьютер. Осыған орай бұл компьютерде үлкен дискілі кеңістік болады. Файлдық сервер пайдаланушылардың мәліметтерді бір уақытта қолдануын қамтамасыз етеді. Сонымен қатар келесі қызметтерді атқарады:

- мәліметтерді сақтау;
- мәліметтерді архивтеу;
- әр түрлі пайдаланушылар жұмыс атқаратын мәліметтерді өзгерту келісімі;
- мәліметтерді жеткізу.

Мәліметтер қорының сервері **–** мәліметтер қоры файлдарын басқаратын, өңдейтін және сақтау функцияларын орындайтын компьютер. (МҚ) Мәліметтер қоры сервері келесі функцияларды орындайды:

- мәліметтер қорының тұтастығын, толықтығын, көкейтестілігін корғай отырып сақтау;
- МҚ сұраныстарын қабылдау және өңдеу, сондай-ақ нәтижелерді жұмыс орнына өңдеуге жіберу;
- МҚ қолдануға автоматтандырылған мүмкіндіктерді қамтамасыз ету, Пайдаланушылар есебін және енгізу жүйелерін қолдау, Пайдаланушылар мүмкіндіктерін шектеу;
- басқа орында орналасқан, таратылған МҚ қолдау, МҚ басқа серверлерімен байланысы.

Қолданбалы программалар сервері— Пайдаланушылардың қолданбалы программаларын орындауға арналған компьютер.

Коммуникациялық сервер **–** енгізу-шығарудың кезекті порттарына, жергілікті желінің қолданушыларына көмескі мүмкіндік беретін құрылғы, немесе компьютер. Коммуникациялық сервер арқылы модемді сервер порттарының біріне жалғап, бөлшектенген модем жасауға болады. Коммуникациялық серверге қосылған қолданушы бұл модеммен тіпті модем жұмыс орнына жалғанбаған күнде де жұмыс істей алады.

Мүмкіндік сервері **–** шалғайдағы тапсырмаларды өңдей алатын белгіленген компьютер. Шалғайдағы жұмыс орнындағы программалар осы серверде орындалады. Шалғайдағы жұмыс орны клавиатурадан қолданушы енгізген командаларды қабылдайды, ал орындалған тапсырма нәтижесі, қайтарылады.

Факс сервер **–** жергілікті желіні пайдаланушыларға мәліметтерді жіберу және факсимилді мәліметтерді қабылдауды іске асыратын құрылғы немесе компьютер.

Мәліметтердің қосымша көшірме сервері **–** жұмыс станцияларында және файлдық серверлер орналасқан мәліметтер көшірмесін кайта қалпына келтіруді, сақтауды және құруды шешетін құрылғы немесе компьютер.

Мұндай сервер ретінде файлдық серверлердің бірі қолданылуы мүмкін. Айта кету керек аталған сервер типтерінің бәрі бір ғана белгіленген компьютерде функция атқара алады.

Желінің коммуніикациялық тораптарына келесі құрылғылар жатады:

- қайталамалар (повторитель);
- коммутаторлар (көпірлер, мосты);
- бағыттамалар (машрутизаторлар);
- шлюздер;

Желінің беріктігі, станциялар арасының алшақтығы біріншіден мәлімет алмасу ортасының физикалық сипаттамасымен анықталады (каоксилді кабель, қос ширатпа (витая пара) т.б.). Мәліметтер алмасуда арақашықтықты шектейтін сигналдардың өшуі кез келген ортада болып тұрады. Мұндай шектеулерді болдырмай және желіні кеңейту үшін арнайы құрылғылар-қайталамалар, көпірлер және коммутаторлар орнатылады. Кеңейту құрылғылары енбейтін желі бөлігін *желі сегменті* деп атау қабылданған.

Қайталама (повторитель) – өзі келген сигналды күшейтуші немесе регенарациялайтын құрылғы. Қайталама пакетті бір сегментпен қабылдап, оны қалған барлығына береді. Бұл жағдайда қайталама өзіне жалғанған сегменттерді ажыратпайды. Әр уақытта барлық қайталама арқылы байланысқан сегменттерде тек екі станцияның арасындағы мәлімет алмасу қамтамасыз етіледі.

Коммутатор немесе көпір – бұл да қайталама сияқты бірнеше сегментті біріктіретін құрылғы. Қайталамадан айырмашылығы көпір өзіне жалғанған сегменттерді ажыратады, екінші сыңары үшін мәлімет алмасудың бірнеше үрдісін қаматамсыз етеді.

Маршрутизатор – мәлімет алмасудың бір хаттамасы бойынша бірдей және әр түрлі типтегі желілерді біріктіруші құрылғы. Маршрутизатор тапсырылған адресті талдайды және мәліметтерді тура таңдалынған маршрут бойынша бағыттайды..

Шлюз – мәлімет алмасудың әр түрлі хаттамаларын қолданатын, әр түрлі желі объектілері арасындағы мәлімет алмасуды ұйымдастыратын құрылғы.

Бақылау сұрақтары:

1. Компьютерлер арасында өзара байланысты орнату қандай мәселелерді шешеді?
2. Компьютерлер арасындағы байланысты ұйымдастырудың қандай тәсілдері бар?
3. Сервер, клиент деген не?
4. Компьютерлік желілер деп нені айтамыз?
5. Желі құрамына кіретін компьютерлер қандай жұмыс атқарады?
6. Желі тарауы деген не?
7. Түйіндердің қандай түрлері бар?
8. Интернет деген қандай терминнен шыққан?
9. Сервердің басты міндеті қандай?
10. Хаттама деген не?
11. Клиент-серверлік архитектураға сипаттама беріңіз.
12. Компьютерлік желілер жіктелуін сипаттаңыз

13. Файлдық сервер деген не?
14. Қайталама (повторитель) деп нені айтады?
15. Коммутатор немесе көпір деген не?
16. Маршрутизатор деген не?
17. Шлюз деген не?

1.3 Әр түрлі деңгейлі тораптардағы технологиялар

Жергілікті желілер компьютерлермен басқа да құрылғыларды тек шектелген аймақтарда ғана біріктіре алатыны белгілі. Үлкен мекемелер мен бірнеше бөлімшелері бар мекемелер өздерінің жергілікті желілерін бір ірі желіге біріктіру қажеттілігі туындайды. Ауқымды желілер (Wide area network - WAN) – бірнеше жергілікті желілерден құрылған желілер. Сонымен, ауқымды желіні үлкен территорияға таралған жергілікті желілердің жиыны деп айтуға да болады.

Ауқымды желілердің біріктірілген желілерден (internetwork) айырмашылығы неде? Біріктірілген желі үлкен территорияны қамти алмайды. Ол ірі мекеменің бір ғимаратында ғана жұмыс істей алуы мүмкін және жергілікті желіні кеңейту үшін ғана немесе желінің өткізгіштік қабілетін сақтай алатын желілерде немесе бірегей желі ішінде әр түрлі архитектуралы желілердің арасында байланыс орнату үшін пайдаланылуы мүмкін. Ал ауқымды желі біріктірілген желі сияқты сол құрылғыларды пайдалана отырып, географиялық тұрғыдан алғанда жан-жаққа таралған жергілікті желілерді біріктіруді мақсат етеді.

Географикалық тұрғыдан алғанда жан-жаққа таралған жергілікті желілерді біріктіру үшін мәліметтердің үлкен қашықтыққа орын алмасуы тек арнайы технологияларды пайдалану арқылы ғана жүзеге асырылады. Сонымен қатар, сигнал белгіленген пунктке еш өзгеріссіз жету керек. Мәліметтерді үлкен қашықтыққа тасымалдау үшін бірнеше әр түрлі технология пайдаланылуы мүмкін.

Телефондық желі

Телефондық желінің қолданыста болғанына 100 жылдан асып кетті. Телефондық желі - бұл коммутациялық желі болып табылады. Сондықтан, телефондық желіде байланыс орнатқанда ол әр түрлі траекториялармен жүзеге асырылуы мүмкін. Мысалы, бір нөмірге телефон соғып отырғанда телефондық байланыс кезіндегі дауыстық сигналдың тек бір ғана маршруттпен қозғалуы міндетті шарт емес.

Телефондық желіні көбінесе жалпылай пайдаланылатын коммутациялық телефондық желі (Public Switched Telephone Network - PSTN) деп те атайды. Алғашқыда қарапайым телефондық желіде коммутацияны техникалық мамандар коммутациялық тақта көмегімен жүзеге асыратын. Уақыт өте мұндай коммутацияны электромеханикалық коммутациялар алмастырды; ал қазір кәдімгі телефондық желілерде коммутация сандық құрылғылар көмегімен жүзеге асырылады.

Мәліметтерді бастапқы пункттен белгіленген жерге тасымалдауда телефондық желі бірнеше мүмкіндіктерді ұсынады. Олар –

коммутацияланатын құрамалар, ерекшеленген бағыттар және әр түрлі коммутацияланатын пакеттер технологиялары. Кәдімгі телефондық желімен мәліметтерді жіберудің ең жаңа технологиясы – сандық абоненттік бағыт немесе жол (Digital Subscriber Line - DSL). Бұл технология телефондық бағыттар бойынша сандық байланысты ретке келтіруге мүмкіндік береді және ATM мен кадрларды ретрансляциялау сияқты мәліметтерді тасымалдау тәсілдеріне шамасы келмейтін мекемелар мен қарапайым қолданушылар үшін (Интернет желісіне өте үлкен жылдамдықпен қосылу үшін) таптырмайтын технология болып табылады.

Кәдімгі телефондық желі географиялық тұрғыда жан-жаққа таралған жергілікті желілерді біріктірудің бірнеше нұсқаларымен қамтамасыз ете алады. Мысал ретінде модем көмегімен коммутациялық қосылуды (dial-up) және ерекшеленген бағыт бойынша қосылуды қарастыруға болады.

Ауқымды желілер өте үлкен қашықтықтарға созылған болса да, қандайда бір ғимаратта орналасқан WAN- жабдыққа жергілікті желінің иесі жауап береді. WAN- жабдықты қалыпты (жұмысшы) жағдайда ұстау жергілікті желі және оның жабдықтарын сондай жағдайда ұстаумен бірдей өте маңызды мәселе болып табылады.

Асинхронды және синхронды модемдік қосылулар

Коммутацияланатын қосылулардың ішіндегі ең қарапайым және арзан әдісі – екі компьютер арасындағы байланысты орналастыру үшін модемді пайдалану. Компьютерлер сандық информациялармен жұмыс істейтін болғандықтан, цифрлы мәліметтерді аналогтық жолмен беру үшін түрлендіру қажет болады.

Асинхронды модем мәліметтерді кәдімгі аналогтық дауыстық телефондық жолмен беруге арналған, және көбіміз осы асинхронды модемдермен өте таныспыз. Олардың көмегімен алыста отырған қызметкерлер желілік сервермен коммутациялық байланыс орната алады, ал миллиондаған Пайдаланушылар Интернетке шыға алады.

Асинхронды модем қабылдаушы модемге мәліметтер пакеті қай жерден басталатынын және қай жерде аяқталатынын хабарлау үшін старттық және стоптық биттерді пайдаланады. Старттық бит пен стоптық бит арасында көптеген модемдер мәліметтердің телефон арқылы дұрыс жеткен жетпегенін тексереді.

Синхронды модемдер старттық және стоптық биттерді пайдаланбайды; олардың орнына жіберуші және қабылдаушы модемдер жіберу уақытын орнату үшін синхрондау биттерін пайдаланады. Синхрондық модемдер старттық және стоптық биттерді пайдаланбағандықтан, олар асинхронды модемдерге қарағанда мәліметтерді өте үлкен жылдамдықтармен жөнелте алады. Бірақ та, синхронды модемдер кәдімгі телефондық желілердегі байланысқа арналмағандықтан, олар тек қана ерекшеленген жолдарда пайдаланылады.

Ерекшеленген жолдар (бағыттар) екі желі арасында кәдімгі телефондық байланыс немесе басқа технология көмегімен тұрақты байланыс орнатуды

қамтамасыз етеді. Ерекшеленген жолдар көбінесе цифрлы болып келеді және аналогты жолдарға қарағанда үлкен өткізгіштік белдеумен қамтамасыз етілген; дауыстық қосылуларға тән шу және интерференцияларға аз бейімделген.

Мәліметтерді беруге арналған сандық жолдар ақпараттарды цифр арқылы беру жолдарынан (Digital data servise - DDS) және мәліметтерді әр түрлі жылдамдықтарда беруге арналған әрқилы жолдар типін қамтитын T-Carrier жүйесінен тұрады. DDS жолы өткізгіштік белдеуін 56 Кбит/с-қа дейін қамтамасыз етеді және желіде тұрақты дуплексті қосылуды (мәліметтер бір уақытта берілуі және қабылдануы) қолдап отырады.

Қазіргі кезде DDS жолымен қосылу басқа WAN-технологиялармен жылдам алмасуда. Өте арзан T-Carrier жүйесімен және басқа да қосылулармен, мысалы, абоненттік цифрлы жолдар (DSL) арқылы қосылулар және кеңбелдеулі кабельді қосылулармен салыстырғанда DDS жолын ескірген деп те есептеуге болады.

T-Carrier – көптеген арналардан құралған жоғары жылдамдықты цифрлы жолдардан тұратын телекоммуникациялық технология. Бұл да кәдімгі телефондық желінің ұсынып отырған байланыс түрінің бірі. T-Carrier жүйесін дауысты, бейне және мәліметтерді жоғары жылдамдықта беруге пайдалануға болады. Көптеген арналардың бірігуі арқасында әр түрлі элементтерді, мысалы, дауыс және мәліметтерді бір жолмен бір уақытта беруге болады.

Коммутацияланатын желілер

Мәліметтерді әр түрлі жергілікті желілер арасында беруге арналған WAN-технологиясының тағы бір альтернативі ретінде коммутацияланатын желілер технологиясын айта аламыз**.** Коммутацияланатын желіде жергілікті желі (немесе бірнеше географиялық тұрғыда жан-жақта орналасқан жергілікті желілер) коммутацияланатын желі қызметінің провайдері немесе кәдімгі телефон желісі көмегімен ауқымды желіге қосыла алады. Жергілікті желінің мәліметтері WAN-қосылуы арқылы өтеді де, коммутацияланатын желіге беріледі, содан кейін белгіленген пунктке қарай жылжиды. Бұл көптеген пайдаланушылар коммутацияланатын желіге өз мәліметтерін бір мезгілде бере алады дегенді білдіреді.

Коммутацияланатын желілер жеңіл кеңейтіледі, өйткені мекемелер оларға өздерінің қосымша бөлімшелерін кез келген уақытта қоса алады. Коммутацияланатын желілер екі түрге бөлінеді: тізбектік коммутацияланатын және пакеттік коммутацияланатын желілер.

Тізбектік коммутацияланатын желілерде жіберуші мен қабылдаушы арасында байланыс сеансы кезінде тізбек орнатылады (жіберуші және қабылдаушы құрылғыларды қосуға пайдаланылатын барлық телефондық жолдар тізбек бола алады). Бұл осы жолдар берілген байланыс сеансы үшін ерекшеленеді деген сөз. Сеанс біткеннен соң, тізбек бірден «ажырайды» да, басқа байланыс сеансына босайды. Коммутацияланатын тізбектер екі түрге бөлінеді: ISDN және DSL.

Пакеттік коммутацияланатын ауқымды желілерде мәліметтер пакеттерге бөлінеді. Пакеттік коммутациялауда пакеттердің өлшемдерінің кіші болуы мәліметтердің жылдам және тиімді жетуін қамтамасыз етеді.

Негізінде қазіргі пакеттері коммутацияланатын ауқымды желілер абоненттер трафиктері мен абоненттер жабдықтарының кез келген түрін бере алуы керек - бұған әр түрлі желілердің конфергенция процесі мүмкіндік береді. Яғни, ауқымды желілер келесі кешенді қызметтер көрсетуі тиіс: жергілікті желілер пакеттерін беру, мини-компьютерлер мен мэйнфреймдердің пакеттерін беру, факс алмасу, офистік АТС-тардың трафигін беру, қалалық, қалааралық және халықаралық телефондық желілерге шығу, бейне конференциялар өткізу үшін бейне көрсетулермен алмасу, кассалық аппараттардың, банкоматтардың трафиктерін беру және т.б., және т.с.с.

Әрбір пакеттің өз меншігінде бақылау ақпараты болады және олар желіде тәуелсіз коммутацияланады. Бұл пакеттер ауқымды желіде әр түрлі маршруттармен қозғалатынын және белгіленген пунктке әрқайсысы өз бетінше жететіндігін білдіреді. Пакеттердің тізбектегі орны пакеттің тақырыбына енгізіледі де, қабылдаушы құрылғы мәліметтерді дұрыс ретпен қою үшін осыны пайдаланады.

Пакеттерді коммутациялаудың көптеген технологиялары бар, соның ішінде біздің қарастыратынымыз: X.25, кадрларды ретрансляциялау және ATM.

X.25 стандарты бастапқыда географикалық тұрғыда жан-жаққа таралған мэйнфреймдерді қосу үшін жасалған. Қазіргі заманғы талаптар бойынша X.25 стандарты (1975 жылы бекітілген) баяу жұмыс істейтіндер қатарына жатады, өйткені ол қателерді тексеру процедурасына өте көп уақыт жібереді (оны жасаған кездегі телефондық жолдардың сапасы төмендігіне байланысты).

Кадрларды ретрасляциялау технологиясы (Frame Relay) X.25 хаттамасы негізінде пайда болған және X.25-ке қарағанда жылдам, өйткені оның жұмысын баяулатқан қателерді тексеру функцияларының кейбіреуін қабыл алмады. Кадрларды ретрансляциялау хаттамасы мәліметтерді берудің OSI деңгейінде жұмыс істейді және ауқымды желінің әр түрлі пунктері арасында байланыс сеансын ұйымдастыру үшін тұрақты виртуальді тізбектерді пайдаланады.

Frame Relay хаттамасының ерекшелігінің бірі, онда әр түрлі өлшемдегі пакеттер орналаса береді. Әрине, мәліметтер тасқынына айтарлықтай әсер етпегенімен, жалпы өткізгіштік қабілетіне қандайда бір кідірту енгізеді.

Мәліметтерді берудің асинхронды режимі (ATM) мәліметтерді, дауысты, бейне материалдарды жалпы қолданудағы желі бойынша беру үшін 53 битті пакеттерді пайдаланатын, пакетті коммутацияға негізделген WAN – технология. ATM-ның хаттамалар жиыны OSI моделінің мәліметтерді беру арнасы деңгейі және физикалық деңгейде жұмыс істейтін хаттамалардан тұрады. ATM желілері мультиплекстелген ақпараттармен жұмыс істейді,

яғни бірнеше ақпараттық арналар бір мәліметтер тасқынына бірігуі мүмкін. Жергілікті желілерде пайдаланылатын АТМ коммутаторлары жергілікті желінің эмуляция режимінде жұмыс істейді.

Интернет желісін құрудың негізгі принциптері

Есептеуіш құралдардың байланыс процестері жекелеген фирмалар мен кәсіпорындар арасында едәуір өсіп келеді. Қазіргі кезде интеграциялау мен жаһандандыру тенденциялары компьютерлік технологиялар айналасында да өз көріністерін бере бастады. Біраз аймақты қамтитын, коммуникациялық құралдар арқылы байланысқан есептеуіш машиналар ауқымды компьютерлік желілер деген атқа ие болды. Соңғы екі-үш онжылдықта бұл желілердің аппараттық және программалық жасақталуы мен ұйымдастыру түрлері шапшаң дамып, көптеген өзгерістер болып отыр. Желілер арасында әлемге танымалдары ретінде SPRINT желісін, FIDO коммерциялық емес компьютерлік желісін S.W.I.F.T. халықаралық есеп жүйелерін атауға болады. Бірақ соңғы жылдары компьютерлік желілер арасынан суырылып шығып алға озғаны көлемі мен функционалдық мүмкіндіктері жағынан Интернет желісі. Интернет - бұл бүкіләлемдік ақпараттық инфрақұрылым. Интернет географиялық орналасуына қарамастан пайдаланушылар мен компьютерлер арасындағы байланыс ортасы, сондай-ақ ақпарат таратудың механизмі болып табылады. Интернетті ақпараттық технологиялардың дамуы мен зерттелуінің ұзақ мерзімді анағұрлым ұтымды инвестицияларының бірі деуге болады. Алғашқыда Интернет әр түрлі типтегі компьютерлік желілерді біріктіру үшін құрылды. Қазіргі уақытта Интернет телекоммуникация мен компьютерлерді қолдану аймағына байланысты ғана емес, сондай-ақ тұтастай қоғамға қызмет етіп отыр.

Провайдерлер және олардың желілері

Провайдер – бұл ұйымдар мен жеке тұлғаларға Internet қызметтерін ұсынатын мекеме. Провайдер ретінде жекеменшік арнайы маманданған фирмалар да, ірі телефон мекемелері де қызмет істей алады. Әдетте провайдерлер біркелкі қызметтер жиынтығын ұсынады, бірақ олардың қызметі түріне және сапасына қарай ерекшеленетіндіктен, төлемақысы да әр түрлі болады. Өз қажетімізге лайықты провайдерді таңдай отырып, бірнеше шартты ескерген абзал:

- мәліметтерді жеткізу жылдамдығы – уақыт бірлігінде модем арқылы өтетін ақпараттың биттер саны. Ұсынылатын жылдамдық провайдер жабдықтарының техникалық мүмкіндіктеріне байланысты болады.
- қосылым түрі. Коммутацияланатын желі бойынша модем арқылы қосылу – бұл Internet-ке қосылудың ең сенімді, бірақ ең баяу түрі. Қосылымның бұл түрі көптеген Internet-те жұмыс істеу жағдайларын қамтамасыз етеді, бірақ ақпараттың үлкен көлемін (дыбыстық файлдар, жан бітірілген сызбалық файлдар, бейне, интерактивті ойындар) жеткізуде қиындықтар туындауы мүмкін.

- кабельді модем, DSL, жерсерігі арқылы кең жолақты қосылым болып табылады. Қосылымның мұндай түрі жоғарғы жылдамдықпен байланысуға мүмкіндік беріп, шынайы уақыт режімінде аудио және бейне файлдарын жеткізуді қамтамасыз етеді. Қызметтер құны осы аталған факторларға және өзіміз пайдаланғымыз келетін тарифтік жоспарға байланысты болады.

Интернетке қосыла отырып, біз интернет-провайдерлердің қызметтерін ISP (Internet Service Provider – Интернет қызметін жеткізуші) пайдаланамыз. ISP – бұл өзіндік желісі бар арнайы ұйым (магистральды деп аталады), оған клиенттердің көптеген саны қосылады. Провайдердің желісі ғаламның кез келген нүктесімен байланысуды қамтамасыз ететін жер жүзінің басқа да желілерімен байланысуы мүмкін. Қалыпты жағдайда ISP-провайдерлер – бұл белгілі бір аймақтарда өзіндік орналасу нүктесі (POP - Point of Presence) бар ірі мекемелер, бұл нүктелерде клиенттерінің Интернетке қосылуын қамтамасыз етуге арналған провайдердің аппараттық қамтамасыз етілуі жүзеге асады. Ірі провайдердің әртүрлі қалаларда өзінің орналасу нүктесі мен мыңдаған клиенттері болады. Бірнеше қалаларда орналасу нүктелері бар провайдерлермен қатар, бір қалада орналасу нүктесі бар провайдерлерді де атап көрсетуге болады. Телефон арнасы арқылы ISP мен байланысуды ұйымдастыру былайша жүзеге асады: пайдаланушы провайдерге хабарласады және модем жинақтарының ішіндегі провайдер модемдерінің бірімен байланыс орнатады (модемді пул деп атауға болады). Қолданушы өзінің ISP-не қосылғаннан кейін, ол оның желісінің бір бөлігі болып табылады. Провайдер өзінің серверінде клиенттерге әртүрлі қызмет көрсете алады: электрондық почта (e-mail), желілер жаңалықтары (Usenet) және т. б. Провайдердің магистральды желісін көбіне тіректі желі немесе бэкбоун деп атайды (ағыл. Backbone – қырат). Провайдер желілері көптеген клиенттерге қызмет көрсететіндіктен, оның жоғары жылдамдықты желісі болуы және жоғары трафикті қамтамасыз етуі керек (желі бойымен берілетін мәліметтер көлемі). Өзінің барлық орналасу нүктелерін біріктіру үшін, провайдер ірі коммуникациялық мекемелерден жоғары жылдамдықты арналарды жалға ала алады, сонымен қатар, өзінің арналарын тарта алады. Ірі коммуникациялық мекемелердің өздерінің жоғары жылдамдықты арналары бар.

Провайдерлердің желілерін біріктіру

Кейбір провайдерлердің клиенттері, мысалы, ISP-A бір бірімен өздерінің жеке желілері арқылы өзара байланысады, ал басқа ISP-B мекемеснің клиенттері өздерінің, бірақ егер ISP-A және ISP-B желілерінің арасында байланыс болмаса, онда A мекемесі мен B мекемесінің клиенттері бір бірімен байланыса алмайды. Өздерінің клиенттерін бір желіде біріктіру мақсатында A және B әр қалада желілік кіруді (NAP - Network Access Points) қамтамасыз ететін нүктелер арқылы өз араларында тікелей байланысты орнатады. Осылайша, басқа провайдерлердің магистральды желілеріне қосылу құрылады, нәтижесінде жоғары деңгейлі көптеген желілердің бірігуі болады.

Интернетте жүздеген ірі интернет-провайдерлер орналасады және олардың магистральды желілері NAP арқылы әр түрлі қалаларда жасалады, және мәліметтердің үлкен ағыны NAP-түйіннің әр түрлі желілері арқылы таралады.

Үлкен және кіші желілердің бірігуі (Интернетті құрайтын) негізінде шартты келісімдер жатады. Әрбір клиенттің белгілі бір ISP пен өзінің компьютерін немесе жергілікті желісін провайдер желісіне қосу туралы келісім шарты бар. Кейбір ISP-A провайдерлердің клиенттері ISP-A желісіне қосылу туралы келісім құрайды, өз кезегінде ISP-A ISP-B мен желілерін біріктіру туралы келіседі және солай жалғаса береді..

Провайдерлер желілерінің иерархиясы

Әр түрлі елдерде халықаралық, ұлттық және аймақтық болып бөлінетін жүздеген провайдерлер бар.

Аймақтық провайдерлердің желілері (екіншілік) ұлттық провайдерлердің желілерімен (біріншілік) жоғары жылдамдықты каналдар арқылы байланысады. Мысалы, АҚШ-та T1стандартты мәліметтерді беру жылдамдығы 1,544 Мбит/с арна немесе жылдамдығы 44,74 Мбит/с жететін Т3 арнасы бар.

Қолданушылардың көпшілігінің провайдерге қарағанда желіге жоғарғы жылдамдықпен кіру мүмкіндіктері аз болады. Абонентпен провайдердің бөлу нүктесі арасындағы болатын байланыс технологиясы технологияның соңғы милясы деп аталады. Бұл атау шартты түрде, ал іс жүзінде осы айтылған қашықтық бір миляға сәйкес келуі міндетті емес. Интернетке жіберу соңғы миля технологиясы әртүрлі бола алады, бірақ “кіру жылдамдығы жоғары болған сайын, каналдарды қолдану ақысы жоғары болады” деген принцип әмбебап болып келеді. Бар инфраструктураны қолдануға мүмкіндік беретін көптеген технологиялар бар, олар - Интернетке кіру мүмкіндігіне арналған телефонды линиялар, кабельді теледидар желілері және басқалар болып саналады.

Бақылау сұрақтары:

1. Ауқымды желі деген не?
2. Телефондық желі деген не?
3. Асинхронды және синхронды модемдік қосылулар қалай орындалады?
4. Интернет желісін құру принциптері.
5. Интернеттің даму тарихы.
6. Интернет дамуындағы құжаттандырудың рөлі қандай ?
7. Әр түрлі тораптардағы технологияларға сипаттама беріңіз.
8. Провайдер деген не?
9. Интернеттергі ISP, POP, NAP сипаттама беріңіз.
10. Интернеттегі соңғы миля деген түсінігі нені білдіреді?

1.4 Интернетке ақпаратты жіберу

Қазіргі заманғы Интернет - жер шарының кез келген нүктесінде тұрған адамдардың өзара ақпарат алмасуына мүмкіндік беретін және кез келген қажет ақпараттарды жылдам тауып беріп, өз мәліметтерді жариялауға болатын өте күрделі және жоғарғы технологиялық жүйе. Интернет көмегімен өзімізге қажетті жұмысты табуға, танысуға, қызықты тақырыптарды талқылауға болады.

1968 жылы ARPAnet желісі пайда болып, біріншіден барлық компьютерлер «тең» дәрежеде жұмыс істеді, екіншіден негізгі желілік хаттама ретінде IP (Интернет хаттама) қабылданды.

Интернетпен байлансудың ең кең тараған және арзан тәсілі – модем және телефон жүйесі арқылы байланысу. Осыған байланысты Интернетке қосудың қызмет көрсету көлемдері мен бағасы жағынан бір-бірінен ерекшеленетін үш түрі бар:

1. ***пошталық*** – кез келген Интернетті қолданушымен тек электронды пошта арқылы алмасуға рұқсат береді, ең арзан түрі;
2. on-line тәртібінде ***сеанстік*** («тікелей байланыста») – сұхбаттық тәртіптегі жұмыс – жүйесінің барлық мүмкіндігі тікелей сеанс уақытында;
3. ***тура*** (жеке), ең қымбат тұратын – барлық мүмкіншілігі бар байланыс кез келген уақытта.

Сеанс тәртібіндегі жұмыс кезінде Интернетке кіру көбінесе провайдерлерден (англ. provide — рұқсат беру, қамтамасыз ету), Интернеттің кейбір бөліктеріне кіруге және оның пайдаланушыларына алуан түрлі қызметтерді жеткізіп беретін фирмалардан сатып алынады.

Интернеттің кейбір маршрутизаторлардың көмегімен өзара байланысатын учаскелерді әр түрлі архитектурадағы жүйелерді ұсынады. Жіберілетін мәліметтер, пакеттер деп аталатын кішігірім тиісті бөліктерге бөлінеді. Әрбір басқа пакеттерге қатысы жоқ пакет жүйе бойынша орналастырылады.

Интернеттегі жүйелер бір-бірімен шексіз байланыста болады, себебі мәлімет беруге қатысатын барлық компьютерлер TCP/IP коммуникациясының біркелкі хаттамасын пайдаланады.

TCP/IP хаттамасы – жүйедегі мәліметтерді жіберудің әр түрлі аспектілерін анықтайтын әрқайсысы жеке екі хаттама болып табылады.

- TCP хаттамасы (Transmission Control Protocol) – қателері бар пакеттерді автоматты түрде қайтадан жіберетін, мәліметтерді жіберетін басқару хаттамасы; бұл хаттама жіберілген ақпараттарды пакеттерге бөлу, қабылдаушы пакеттеріндегі ақпараттардың дұрыс қалпына келтіруге жауап береді;
- IP хаттамасы (Internet Protocol) – жүйеаралық өзара әрекеттердің хаттамасы. Ол пакетті адресіне жіберу және қажет адреске беру жолында тұрған бірнеше жүйелерден өту мүмкіндігін жауап беретін хаттама.

TCP/IP хаттамасы бойынша ақпараттарды жіберу схемасы мынадай: TCP хаттамасы ақпаратты пакеттерге бөліп, оларды нөмірлейді; одан соң IP

хаттамасының көмегімен барлық пакеттер қабылдаушыға тапсырылады, сол жерде барлық пакеттердің қабылданғаны жайында TCP хаттамасының көмегімен тексеріледі; барлық пакеттер қабылданып болған соң TCP хаттамасы оларды қажетті тәртіп бойынша орналастырады және бір тұтас етіп жинайды.

Әрбір Интернетке қосылған компьютерлердің бағалары бірдей, өте сирек кездесетін екі адресі: сандық IP-адрес (цифровой IP-адрес) және символды домендік адресі (символический доменный адрес) болады. Адрестерді тағайындау мынадай схема бойынша жүргізіледі: Жүйелі ақпараттық Орталық халықаралық ұйымы жергілікті жүйелердің иелеріне адрестердің топтамасын береді, ал қалғандары нақты адрестерді бөледі.

IP-компьютердің адресінің ұзындығы 4 байт. Әдетте бірінші және екінші байттар жүйе адресін, ал үшінші байт жүйе бөлімшесінің адресын, ал төртіншісі бөлімше жүйенің тармағының адресін анықтайды. IP-ге ыңғайлы бөлу үшін адрес нүктелермен бөлініп 0-ден 255-ке дейінгі төрт таңбалы сандармен жазылады, мысалы: 145.38.6.135. Жүйенің адресі – 145.38; жүйе бөлімшесінің адресі – 6; компьютердің жүйедегі адресі – 135.

Сандық адрестен домендік адрестің айырмашылығы оның символикалығында және оны адам есте жақсы сақтайды. Домендік адрестің мысалы, basa.lek.nur.kz. бұл жердегі домен basa – компьютердің анық атауы, өз IP-адресі бар, домен lek – өзі компьютерге берілген топтың атауы, домен nur lek – доменіне атау берген өте ірі топтың атауы, және т.с.с. Домендік адрес мәліметтер жіберу процесінде IP-адресіне өзгереді.

TCP/IP хаттамасының стектері

TCP/IP-Итернетте қолданылатын әр түрлі деңгейдегі желілік хатттамалардың жиынының, стек жинақталған аты.

TCP/IP ерекшелігі:

– программаның және ақпараттың жабдықтардан тәуелсіз өңделетін ашық стандартты хаттамалар;

– өткізудің физикалық ортасынан тәуелсіз;

– пайдаланушылар қызметін таратуға арналған жоғары деңгейлі стандартты хаттамалар.

TCP/IP хаттамасының стектері 4 деңгейге бөлінеді: қолданбалы, транспорттық, желі аралық, физикалық және каналдық.

Мәліметтер пакетпен жіберіледі. Пакеттердің басы және соңы болады, оған қызметші ақпараттар жазылады. Өте жоғары деңгейдегі мәліметтер кіші деңгейдегі пакеттерге қосылады.

TCP/IP стегі белгілі бір хаттамаларды ғана қолдануды көрсетпейді, ол қолжетімді деңгейде мәліметтерді өткізу ортасы мен физикалық ортасын көрсетеді. Өткізу ортасында қолжетімді деңгейде IP модульді интерфейс болуы шарт, ол IP – пакеттерді өткізуді қамтамасыз етеді.

IP хаттама мәліметтер блогын бір IP-адрестен екіншісіне жеткізеді. Бір немесе бірнеше хаттамалардың функциясын жүргізетін программа модульдер деп аталады, мысалы: «IP-модуль» «TCP-модулі». IP модулі IP-

пакеті төменгі деңгейден қабылданған кезде, оны тағайындалған IP адреске тексереді.

– егер IP-пакет осы компьютерге бағытталса, онда ондағы мәліметтер жоғары деңгейдегі модульдерге өңдеуге жіберіледі.

– егер IP-пакеттің тағайындалған адресі бөтен болса, онда IP модуль екі шешім қабылдайды: бірінші – IP пакетті жою, екінші – ары қарай тағайындалған адреске жіберу, ал маршрутты анықтауды маршрутизаторлар бар желі шекарасында IP-пакеттерді фрагменттерге бөледі, содан соң қабылдаушы компьютерге жинақталады.

Егер IP модуль қандай да себептермен IP пакетті жеткізе алмаса, ол жойылады. Бұл кезде IP модуль компьютерге IP-пакеттің қателігі туралы хабарлама жібереді, ондай хабарламалар ICMP (Internet Control Message Protocol- IP хаттамасының кеңейтілуі, қате туралы немесе тексеру туралы хабарламаларды жіберуге мүмкіндік береді) көмегімен жеткізіледі. Одан басқа IP-пакеттерге жеткізгені туралы, дұрыстығын реттеуге мүмкіндіктері жоқ. Бұл мәселе транспорт деңгейге жүктелген.

DNS-сервер

Domain Name System (DNS)- «домендік жүйе аты» болып аударылады. DNS-домендік жүйе аты қолданушыларға интернетте жұмыс істеуді оңайлатады, оған хостың цифрлық адресін сақтайды. Компьютерге цифрлар жиынымен жұмыс істеу оңай, алыстатып машинаның IP-адресінің домендік атын енгізу арқылы жұмыс істейді. DNS стандартында жазылған әлемдік желі ресурстарының адрестері бір-бірінен нүкте арқылы ажыратылған бірнеше құраушылардан тұрады. Бұл элементтер «домендік» деп аталады. Домен интернеттің логикалық деңгейі, яғни өз аты бар желі станцияларымен басқарылатын желі ресурстарының тобы. DNS құраушы адрестері – «бірінші деңгейлі домен» жеке мемлекет территориясының ауқымды географиялық зонасы. Мысалы АҚШ «US», Қазақстан «kz», Ресей «ru» т.с.с..

Екінші деңгейлі доменге (банк, университет, қалалық муниципалды қызмет желілері) еркін ат береді. Үшінші деңгейлі домен екінші деңгейлі доменнің құраушыларының бөлігі, олар кез келген атты қолданады. Хостың IP-адресінен басқа домендік аты болады. Домен аттары нүктемен бөлінген символдық өрістен тұрады. Оң жағынан соңғы өрісі екінші деңгейлі доменді көрсетеді, соңғы сол жақтағы өрісі хостың аты. **Мысалы crypt. iae. nsk. Su.** Crypt-хост, iae-домен, nsk ішкі доменде, ол Su доменінде.

DNS қызметі әрбір сервер бөлігі зонаға жауап беретін серверлердің иеархиялық құрылымын құрайды, яғни домендік атау бұтағы өз бөлігіне сәйкес мәліметтер қорымен сұрауларына жауап береді. Бұл кезде жоғары бұтақ серверлері төменгі бұтақ серверлерімен байланысты қамтамасыз етеді. Домен мен зона айырмасын түсіну маңызды болып саналады. Домен домендік атау бұтағының ішкі бұтағы. Зона – DNS-сервер жауап беретін бұтақ бөлігі. Мысалы, vvsv.ru доменінде келесі ішкі домендер бар: cts, admin, labs. DNS сервер әкімшісі vvsu.ru және admin. vvsu.ru,vvsu.ru үш зонаға бөлінеді.

Прокси-сервер клиент-компьютерлер мен серверлер арасында дәнекер болып қызмет атқаратын сервер. Пайдаланушыға ақпараттарды беруден бұрын оның бұл ақпаратқа рұқсаты бар немесе жоқ екенін тексереді.

1.5 WEB-технологиялардың негізі

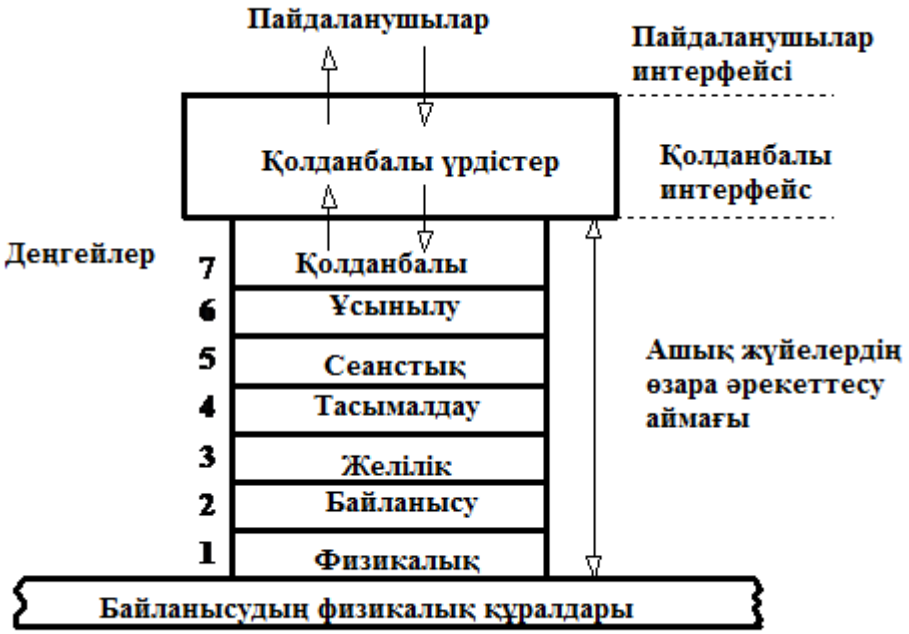
OSI моделінің қолданбалы деңгейінің хаттамалары

Мәліметтерді тасымалдау ісін стандарттау жұмысы Халықаралық стандарттар институтының (ISO - International Standards Organization) техникалық ұсыныстарына байланысты жүргізіліп, желі параметрлерін сәйкестендіру ісі OSI (ашық жүйелердің әрекеттесу моделі – Model of Open System Interconnections) деп аталған модель негізінде жасалып шықты. Осы ISO/OSI моделіне сәйкес мәлімет алмасу схемасы 7 деңгейге бөлініп қарастырылады:

1-физикалық, 2-каналдық, 3-желілік, 4-транспорттық, 5-сеанстық, 6-мәліметтерді ұсыну деңгейі, 7-қолданбалы деңгей. Бұл деңгейлердің жоғарғысы қолданбалы деңгей, ал ең төменгісі–физикалық деңгей болып саналады.

Қолданбалы – тұтынушының есептеу жүйесімен әрекеттесуін атқаратын ең жоғарғы деңгей. **Физикалық** – құрылғылар арасында сигналдар алмасуын қамтамасыз ететін ең төменгі деңгей.

ISO/OSI моделінде әр түрлі құрылықтардағы компьютерлер арасындағы байланыстың жеті деңгейі арқылы мәлімет тасымалданады (1.2-сурет).



1.2-сурет. OSI моделі

1. *Қолданбалы деңгейде* тұтынушы өз компьютеріндегі программалар көмегімен жіберілетін құжатын дайындайды.
2. *Ұсынылу деңгейінде* тұтынушы компьютеріндегі операциялық жүйе мәліметтің қайда жазылып тұрғанын анықтап, оның келесі деңгеймен әрекеттесуін қадағалайды.
3. *Сеанстық деңгейде* тұтынушы компьютері жергілікті немесе ауқымды желімен байланысады.
4. *Тасымалдау (транспорт) деңгейінде* жөнелтілетін құжат қолданылатын желі хаттамалары талаптарына сәйкес тасымалдауға ыңғайлы форматқа (пакеттерге) түрлендіріледі.
5. *Желілік деңгейде* мәліметтің тасымалдану маршруты анықталып, пакеттер адреспен толықтырылады. Енді олар бір-бірінен тәуелсіз күйде жеке-жеке жеткізіле береді.
6. *Байланысу деңгейінде* желілік деңгейден алынған мәлімет модем көмегімен физикалық деңгейге қажет нақты сигналдарға түрлендіріліп, оны модульдеу ісі атқарылады.
7. Мәліметтерді нақты түрде тасымалдау соңғы *физикалық деңгейде* орындалады. Мұнда мәлімет тек биттер түрінде өрнектеліп тасымалданады. Мәліметті қабылдап алу тағы да екінші тұтынушы компьютерінде осы деңгейлердің кері бағытта жұмыс істеуі арқылы биттерді қайтадан нақты құжатқа түрлендіру жолымен атқарылады.

Әрбір деңгей жоғарғы деңгейдегі модель арқылы анықталған өз қызметтерін (функцияларын) атқарып, мәліметті келесі деңгейге беріп отырады. Осылай компьютерлер әрекеттесуінің әрбір деңгейінің өзіне ғана тән хаттамасы, яғни мәлімет алмасу ережесі болады.

Хаттама – бір деңгей аймағындағы жүйелер арасында мәлімет алмасу әрекеттерін анықтау ережелері.

Интерфейс – төменгі деңгейдің жоғарғы деңгейге бере алатын функциялар жиыны.

Хаттамалар стегі (тізімі) – бұл жүйелерді байланыстыруды ұйымдастыруға қажетті әр түрлі деңгейдегі хаттамалар жиыны.

Сонымен **хаттама** *деп тасымалданатын мәліметтің форматтары мен олармен орындалатын іс-әрекеттерді анықтайтын ережелер жиынын айтады.* Ол байланыстың атқарылу тәсілдерін және арналардағы кедергілерді азайту жолдарын анықтап, екі компьютердің арасында мәліметті мүлтіксіз тасымалдау ісін жүзеге асыруды қамтамасыз етеді. Желідегі стандарт ұғымы да осы бекітілген хаттамадан немесе солардың жиынынан тұрады.

Қолданбалы деңгейде пайдаланушыларға өңделген ақпаратты беру қажет. Бұл жүйелік және пайдаланушының программалық қамтамасыз етуі арқылы жүзеге асырылады.

Қолданбалы деңгей қосымшалардың жүйеге қатынауына жауап береді. Бұл деңгейдің міндеттері – файлдардың орнын ауыстыру, пошталық хабарламалармен алмасу және желіні басқару.

Жоғарғы үш деңгейдің (қолданбалы, ұсынылу, сеанстық) кеңінен таралған хаттамалары:

- FTP (File Transfer Protocol) файлдарды беру хаттамасы;
- TFTP (Trivial File Transfer Protocol) файлдарды жіберудің қарапайым хаттамасы;
- X.400 электрондық пошта;
- Telnet қашықтағы терминалмен жұмыс;
- SMTP (Simple Mail Transfer Protocol) пошталық алмасудың қарапайым хаттамасы;
- CMIP (Common Management Information Protocol) ақпаратты басқарудың жалпы хаттамасы;
- SLIP IP (Serial Line IP) тізбекті сызықтар үшін. Деректерді әрбір нысан бойынша жіберудің тізбекті хаттамасы;
- SNMP (Simple Network Management Protocol) желілік басқарудың қарапайым хаттамасы;
- FTAM (File Transfer, Access and Management) файлдарды жіберу, қатынау және басқарудың хаттамасы.

Гипермәтін және Web-парақтар

WWW (World Wide Web) – қызметі гипермәтіндік ақпараттар алмасуына арналған. Жоба 1989 жылы ұсынылған. 1993 жылы алғашқы браузер пайда болды. WWW «клиент-сервер» схемасы бойынша құралған.

Егер WWW-клиент-сервер орнатылмаған болса, онда алыстағы терминал режимінде жұмыс істеуге болады. WWW үшін ең тиімді программалық интерфейсі болып ms explorer, netscape және басқалары болып табылады.

HTML тілінде құжат дайындау үшін кез келген мәтіндік редактор жарайды (UNIX, ME, MS-DOS т.б.)

Гипермәтінді дайындағанда HTML тілін немесе көптеген программалық құралдарды қолдануға болады, ол сіздің құжатты қажетті форматқа түрлендірсе болғаны. Гипермәтіндегі құжаттар бір бірімен сөз жиындарымен байланысады. Қолданушыға ол құжаттың қайда тұрғанын білмесе де болады.

WWW серверіне сілтемелер http: сөзінен басталады. Гипермәтін алыстағы серверде сақталған статьяларға түсініктемелер беруді жүзеге асырады.

Гипермәтін мәтіндік қана емес, графиктік, дыбыстық болғандықтан гиперорта (hupermedia) термині қолданылады. WWW арнайы индекс құжаттарында кілттік сөз бойынша іздеуге болады. WWW әр түрлі форматты қолданады және ақпараттық кеңістікке қол жетімді.

Браузер (Inter Explorer, Opera . . .) мультихаттамалы клиент және HTML интерпретаторы. Типті интерпретатор секілді клиент (тәг) командаларға тәуілді әр түрлі функцияны орындайды. Бұл функциялар шеңберіне мәтінді

экранға орналастыру, сервермен ақпарат алмасу, графиктік мәтін т.б. енеді. HTTP сервері файл алуда клиент сұрауларын өңдейді. Гитермәтін деген не?

Гипермәтін – ASCII символдарынан тұрады. Қашықтық пен қосылған сілтемелерді орындау үшін сөз-белгілер қолданылады. Гипермәтін негізі – HTML элементтері. Мұндай элементке аты, атрибуты, мәтін және гипермәтін енеді.

HTTP – серверді басқа программалармен әсерлестіру мақсатында үшін құрылған. Гипермәтіндік ақпараттың жүйе идеясы мынадай: қолданушы құжаттарды өз қалауы бойынша көре алады, кітап оқыған сияқты тізбекті түрде емес. Сондықтан Т. Нельсон гипермәтінді сызықтық емес деп анықтады. Мұның бәрі арнайы байланыс механизмін гипермәтіндік сілтемелерге қолдану арқылы, мысалы жай мәтінде «келесі – алдыңғы» типі болса, гипермәтінде өте көп сілтеме жасауға болады. Гипермәтін бойынша мамандардың жақсы көретіні библия, «Help» жүйесі. Алғаш қарағанда қарапайым, бірақ іс жүзінде күрделі статикалық, динамикалық, контексті сілтемелер жасалынады.

HTML идеясы – гипермәтінді жүйені құру мәселелерін арнайы бейнемен басқару құралдарының жүйені құру мәселелерін арнайы бейнемен басқару құралдарының сәтті шешімі болып табылады.

Гипермәтіндік тілді өңдеуге екі фактор әсер етті: гипермәтіндік жүйені интерфейсі саласында зерттеу және үлестірілген желіде гипермәтіндік мәліметтер қорын қарапайым және жылдам әдіспен қамтамасыз ету.

Жай гипермәтіндік жүйелер арнайы программа құралдарымен гипермәтіндік байланыс тұрғызады. Гипермәтіндік сілтемелердің өзі арнайы форматта сақталып немесе арнайы файлдарды құрайды. Мұндай әдіс жергілікті жүйе үшін тиімді, тек үлестірілген түрлі компьютерлік платформа жиынына арналмаған. HTML-да гипермәтіндік сілтемелер құжат денесіне енгізіліп, оның бөлігі ретінде сақталады. WWW құжаттары – бұл қарапайым ASCII-файлдары, оны кез келген мәтіндік редакторда дайындауға болады.

HTTP (Hyper Text Transfer Protocol) хаттамасы гипермәтіндік құжаттарды алмастыруға арналған және оның спецификациясын есепке алады. Өзара әсер процесінде клиент желі қорының жаңа адресін қабылдайды, қосылған графикті сұрайды, параметрлерді қабылдайды және өткізеді. HTTP басқару ASCII командасы түрінде жүргізіледі. Гипермәтіндік мәліметтер қорын өңдеушілер хаттама элементтерімен сыртқы есептеу программаларын қолданғанда немесе сыртқы ақпараттық қорларға қол жетімді болғанда кездеседі.

HTTP сұрақ/жауапқа негізделген. Сұрайтын программалар (клиент) қызмет көрсетуші программалар (сервермен) байланыс орнатады және сұрауды серверге келесі формада жүргізіледі: сұраулар әдісі, URL, хаттама нұсқасы. MIME ұқсас хабарламалар клиент туралы сұрау ақпараттарын басқарады. Сервер статус жолындағы хабарламаларға жауап береді. Бұл программа бір мезгілде клиент те, сервер де бола алатындығын айта кетуге болады.

FTP қызметі

FTP (**File Transfer Protocol** – файлдарды тасымалдау хаттамасы. Қашықтағы компьютерден екілік және мәтіндік файлдарды алуға болады. Программалаушылардың бір-бірімен программа, сурет, мәтін (Word, Excel құжаттары, программа, т.с.с.) алмасуы үшін өте қажетті қызмет түрі. Бұл қызметті FTP-серверлер атқарады. Олар кез келген форматтағы файлдарды тасымалдай алады.

Интернет серверлерінде файлдар түрінде сақтаулы құжаттар көп. Оларды қабылдап, экранға шақыру үшін FTP қызмет хаттамасы пайдаланылады.

Файлдар барлық компьютер серверлерінде сақтала бермейді. Оларды сақтап, FTP арқылы қызмет көрсететін арнайы компьютерлер (файлдарды сақтау қоймалары) бар. Оларды FTP-серверлер деп атайды.

FTP хаттамасын іске қосу үшін Internet Exploler терезесінің Адрес өрісіне файл серверінің адресін, мысалы, ftp://ftp.relcom.ru/ сияқты адресті енгізу керек. Экранға кәдімгі компьютердегідей каталогтар мен файлдар көрінетін Сервер беті шығады. Каталогты шертіп, ішкі каталогқа өтуге, файлды шертіп, мазмұнын экранға шығаруға болады. Кейбір серверлерде файл архив түрінде көрінеді. Оны ашу үшін архивтеуші программаны пайдаланған дұрыс.

Электрондық пошта хаттамалары

SMTP (Simple Mail Transfer Protocol) – электрондық поштаның арнайы (E-mail) хаттамасы. Электрондық пошта **SMTP**, **POP3** сияқты екі хаттама арқылы жұмыс атқарады. Алғашқысы мәлімет ішінде гиперсілтеме қолдануға мүмкіндік береді, хаттарды жақын жердегі серверге жөнелтеді, ал **POP3** – алушы маңындағы серверге келіп түскен мәліметті керекті компьютерге жіберуді қамтамасыз етеді.

Telnet – компьютердің желідегі басқа компьютердің терминалы ретінде тіркелетін қызмет түрі болып табылады. Қашықтағы компьютерден программаларды іске қосып, оқу орындарының кітапхана түбіртегтерін көріп, олармен өз компьютеріміздегідей жұмыс істеуге мүмкіндік береді (өңдеу және түзету әрекеттерімен қоса).

Telnet – интернеттің ең ескі ақпараттық технологиясы. Ол бір жарым мың ресми желі материалдарының ішіндегі RFC (Request For Comments) деп аталатын, үш ондыққа енетін стандарттар санына енеді. Telnet көмегімен кез келген портқа жүгініп және қарауға болады, оның жауабын көруге болады.

Telnet келесі үштіктен тұрады:

1. қолданушының Telnet-интерфейсі;
2. Telnet үрдісі;
3. Telnet хаттамасы.

Бұл үштік алыстағы компьютер ресурстарына қол жеткізу үшін желілік терминалды жүргізеді және сипаттауды қамтамасыз етеді.

Telnet хаттамасы. Telnet хаттама ретінде TCP транспорт хаттамасының қосымшасы болады.

Telnet-тің негізінде үш іргелі идея жатыр:

1. NVT (Network Virtual Terminal - Желіге виртуалды терминал) концепциясы;
2. Келісім-шарт опциясы (өзара әсер параметрлерін келістіру);
3. «терминал -үрдіс» байланыс симметриясы.

Telnet байланыс программасын орнатқанда, ол нақты терминал құрылғыларымен жұмыс істейді және бұл программаға қызмет көрсеткенде спецификация ақпаратының алмасуы үшін терминал құрылғыларының жұмысының ережелері немесе Желілік виртуалды терминал (Network Virtual Terminal) қолданылады. Қысқарту мақсатында мұны NVT спецификациясы дейміз. NVT – бұл нақты физикалық терминал құрылғысының мүмкіндіктерін кеңінен қолданылатын стандартты сипаттау табылады.

USENET жаңалықтар тобы немесе телеконференциялар – кез келген тақырыпта мәлімет жинайтын пікір таластыру топтары болып саналады. Пікір таластар электрондық пошта арқылы жүргізіледі. **Usenet –** желі бойынша таралып орналасқан пікір таластыру клубы, телеконференциялар, жаңалықтар тобы (News). Бұл қызмет түрі NNTP (Net News Transfer Protocol) хаттамасы арқылы жүргізіледі. Электрондық поштадан айырмашылығы – Usenet клиенті өз мәліметтерін жеке адресатқа емес, өзі білмейтін ортадағы басқа топқа, яғни телеконференция абоненттеріне жібереді. Конференцияның барлық қатысушыларының құқықтары бірдей болып саналады, сондықтан әрқайсысы қойылған мәселе жайында өз пікірін ашық айта алады. Әрбір телеконференция бір тақырыпқа (ғылымға, өнерге, спортқа, демалысқа, т.с.с.) арналады.

IRC (**Internet Relay Chat** - Интернет арқылы тікелей хабарласу - чат)– мұнда мәлімет алмасу нақты уақытта тікелей жүргізіледі.

IRC қызметі нақты уақыт кезеңінде бірнеше кісінің тікелей хабарласуы үшін қажет. Кейде *IRC* қызметін *чат-конференция* немесе жай *чат* деп атайды. *IRC* жүйесіне қатынасып сөйлесу тек бір канал аралығында және оған бірнеше адам ғана қатынаса алады. Бір IRC серверіне қосылып бірден әңгіме жүргізуге болады. Әңгіме тікелей пернеден сөздерді теру арқылы жүргізіледі. Чаттын телеконференциядан айырмашылығы жауап бірден келеді.

IP-телефония – бұл Интернетті немесе кез келген IP-желісі арқылы телефонмен сөйлесу технологиясы.

Жұмыс режимі:

- телефон - телефон;
- компьютер - телефон.

IP-телефония ретінде қолданыла алатын көптеген алуан түрлі бағдарламалар бар. IP-телефония Internet арқылы дауысты жеткізуге мүмкіндік береді.

Қазіргі таңда бұл программалар ішіндегі ең танымалы Skype (http://www.skype.com.) болып табылады. Skype программасын орнату компьютер жөнінде қандайда бір артық білімдерді талап етпейді.

Арнайы бейімдеуіштер көмегімен Skype терминалы ретінде кәдімгі қалалық телефонды пайдалануға немесе кіретін қоңырауларды ұялы телефонға қабылдап алуға болады. Skype программасымен сәйкестендірілетін DECT-телефон да пайда болды. Қарапайым да арзан шешім – неғұрлым ыңғайлырақ болуы үшін USB-портқа арнайы IP-телефонды қосуға болады.

WAP хаттамасы негізіндегі мобильді интернет

WAP (Wireless Application Protocol - желісіз қосымшалар хаттамасы) – бұл хаттама немесе техникалық стандарт, осы тәсіл арқылы ақпарат Интернет желісінен шағын ұялы телефон дисплейіне беріледі. Бұл технология ұялы телефоннан Internet желісінің мобильді ресурстарына қосылу мүмкіндігіне ие болуға, WML немесе XHTML формаларындағы ұялы телефон мүмкіндіктеріне арнайы икемделген ақпаратты алуға мүмкіндік береді.

Бақылау сұрақтары:

1. Интернетке қосудың қандай түрлері бар?
2. Интернеттегі адрес туралы анықтама беріңіз. Адресдың қандай түрлері бар?
3. Интернетте мәлімет жіберу әрекеті қалай жүзеге асады?
4. TCP/IP хаттамасының стектеріне сипаттама беріңіз.
5. DNS нені білдіреді?
6. ISO/OSI модельдеріне сипаттама беріңіз.
7. Мәлімет алмасудың схемасының әр деңгейіне сипаттама беріңіз.
8. ISO/OSI моделіне сәйкес мәлімет алмасудың схемасын талдаңыз.
9. Хаттама, интерфейс, хаттамалар стегіне анықтама беріңіз.
10. Гипермәтін деген не?
11. HTTP хаттамасы не үшін қажет?
12. FTP қызметіне сипаттама беріңіз.
13. SMTP хаттамасы қандай мүмкіндіктер береді?
14. POP3 хаттамасының міндеті қандай?
15. Telnet қызметі қандай мүмкіндіктер береді?
16. USENET қандай мүмкіндіктер береді?
17. IRC қызметіне сипаттама беріңіз.
18. Телеконференция мен чаттың айырмашылығы неде?
19. IP-телефония қызметі қандай мүмкіндік береді?
20. Skype бағдарламасы қандай қызмет атқарады?
21. WAP хаттамасына сипаттама беріңіз.

2 WEB ҚОСЫМШАЛАРЫН ҚҰРУ ТЕХНОЛОГИЯЛАРЫ

Интернеттің бар мәліметтерінің, яғни барлық Web-құжаттарының бір ортақ қасиеті – олардың барлығы да HTML тілінде жазылған. HTML тілінде Web-құжаттарды жасау программалауға ұқсас болғанмен, ол қарапайым программалау тілі емес. HTML – гипермәтіндік белгілеу тілі. Ол кәдімгі мәтіндерді Web-парақтар түрінде бейнелеуге арналған ережелер жиынын анықтайды.

Интернеттің қазіргі дамуы 90 - жылдар басында компьютерлер арасында мәлімет алмасудың жаңа *хаттамасы (protocol)* пайда болғаннан кейін басталды. Бұл хаттама *HTTP (HyperText Transfer Protocol – гипермәтінді тасымалдау хаттамасы)* деп аталған болатын. Осы хаттамамен қатар HTTP серверлерінің кеңейтілген желілері болып табылатын Интернет арқылы файлдар тасымалдай алатын *World Wide Web* қызмет бабы (*WWW* немесе тек *Web)* пайда болды.

Бұл файлдардың басым көпшілігі *Web-парақтар* түрінде – *HTML (Hyper-Text Markup Language – гипермәтінді белгілеу тілі)* тілінде жазылған арнаулы файлдар түрінде болады. Осы файлдарды HTTP серверлерінде (*Web*-тораптарында) орналастыру жолымен *Web*-парақтар қалың көпшілік пайдаланатындай түрде Интернетте жарияланады. *Web*-парақтар мазмұны әр түрлі бола береді және олар көптеген тақырыптарды қамти алады, бірақ олардың бәрінің де негізгі жариялану, яғни жазылу тілі *HTML* болып табылады. Осындай *HTML* құжаттарының бәрінің де файл аттарының кеңейтілуі (тіркеуі) *. HTM немесе *. HTML болуы тиіс.

HTML тілі World Wide Web қызмет бабымен бірге дами отырып, Web-парақтарының ең жақсы деген мүмкіндіктерін жүзеге асырып, оны кең пайдалану жолдарымен толықтырылып отырылды. Ол World Wide Web жүйесінің негізі бола отырып, оның өте кең тарауына себепші болды. World Wide Web сөзі қазақ тіліне *кеңейтілген бүкіләлемдік өрмек* болып аударылады. HTML тілінің мағынасы мен атқаратын қызметін оның атынан анықтауға болады.

Гипермәтін – қосымша элементтерді басқару мақсатында ішіне арнаулы код, яғни екпінді элемент (anchor) орналасқан мәтін. Ол – мәтін ішіне сурет, дыбыс енгізу, мәтінді безендіру, пішімдеу (форматтау) ісін орындайтын немесе осы құжаттың басқа бөлігіне сілтемесі бар алғашқы нүкте ретінде қарастырылатын белгіленген сөз. Сөзді ерекшелеп белгілеу дегеніміз – келесі көрсетілетін құжат бөлігі қалай бейнеленетінін анықтайтын айрықша кодты осы сөз ішіне енгізу. Гипермәтін экранда белгіленіп ерекшеленген қарапайым сөз ретінде тұрады, егер курсорды сол сөзге жеткізіп, тышқанды шертсек (ENTER пернесін бассақ), онда сонымен байланысты (ол сілтеп тұрған) басқа құжат ашылады. Ол құжаттар мәліметтер ішіндегі басқа парақтарда немесе Интернет жүйесіндегі басқа компьютерде орналасып, бейнежазба, сурет, жазылған дыбыс күйінде болуы мүмкін.

Гипермәтінді экранға шығарып бейнелеу үшін *броузер (browsers)* деп ат-алатын арнайы көрсету программалары қолданылады. Броузер арнайы

командалармен – тәгтермен толықтырылған мәтіндік құжатты қабылдап алып, оның тақырыптарын экранға үлкен әріптермен, ал жай мәтіндерін кішірек таңбалармен жазады, оның ішіндегі суреттерді де адрестеріне сәйкес басқа немесе осы компьютерден оқып экранда көрсетеді. Ең кең тараған броузерлерге Windows операциялық жүйесімен бірге қойылатын Internet Explorer және жеке қолданылатын Netscape Navigator программасы жатады. Осы екеуі (олардың бұрынғы нұсқаларын қоса есептегенде) бүгінгі қолданылып жүрген броузерлердің 90 %-ын құрайды.

Сонымен, web-құжат дегеніміз тәгтермен толықтырылған мәтіндік файл болып табылады, оның мәтіндерін бір-бірімен байланыстыра отырып белгілеуге мүмкіндік беретін HTML тілі. Оның дұрыс нәтиже алуды қамтамасыз ететін өз заңдылықтары мен ережелері бар. Web-құжаттың алғашқы нұсқасын Блокнот сияқты редакторлардың бірінде дайындап алып, броузер арқылы экранда көреміз. Егер оны түзету керек болса, броузер арқылы мәтін нұсқасын экранға шығарып түрлендіру қажет. Web-құжаттарды жасап, оны көрнекті түрге келтіріп безендіретін мамандарды web-шебер немесе web-дизайнерлер деп атайды. Шағын көлемді web-құжатты HTML тілінің негізін білетін кез келген студент жасай алады.

Негізінде гипермәтіндік web-құжаттар жасаудың екі тәсілі бар, олар: HTML тілін пайдалану және арнаулы HTML-редакторларды қолдану. Соңғы тәсіл «экранда не көрсең, соны аласың» деген WYSIWYG принципімен диалог режимінде істейтін Netscape Editor, Hot Dog, Front Page сияқты программалар арқылы гипермәтін жасауды жүзеге асырады, олар құжаттың ішкі құрылымына араласпай, тек меню командалары немесе батырмалар арқылы керекті құжат бейнесін қалыптастырады. Дегенмен, HTML тілі ең қысқа, әрі жылдам істейтін тәсіл болып есептеледі, редакторлар арқылы жасалған Web-парақтың ең соңғы нұсқасына өзгертулер енгізу де HTML тілі арқылы орындалады.

2.1 HTML тілінің атқаратын қызметі

Web-парақтар экранда ықшам түрде безендіріліп көрсетілгенмен, HTML тілі мәтіндерді пішімдеп (форматтап) көрсететін тілге жатпайды. Өйткені әрбір тұтынушы әр түрлі компьютерлерді пайдаланады. Сол себепті жаңа ғана зауыттан шыққан бір компьютердің Windows жүйесінде жұмыс істей алатын броузері бар болса, екінші бір тұтынушы компьютері тек MS DOS жүйесінде жұмыс істейтін ескі броузерді пайдалануы мүмкін. Бұл екеуінің көрсету мүмкіндіктері әр түрлі болғандықтан, бір файл екеуінде екі түрлі болып көрсетіледі.

Құжаттарды әрбір тұтынушының әр түрлі құрылғыларда және әр түрлі броузер программалармен көретіндіктерін ескерсек, HTML тілін мәтіндерді пішімдеу (форматтау) тәсілдерін жазуға арналған тіл деп айтуға болмайды. Ол Интернеттегі мәтін бөліктерінің атқаратын қызметін анықтап, соларды әрбір тұтынушыға бейімдеп жеткізе алатын *құжатты функционалды түрде белгілейтін тіл* болып табылады.

Мысалы, егер мәтін тақырыбын бейнелеу керек болса, онда HTML коды оны тақырып ретінде көрсетуге тырысады. Тақырыптың белгілеу коды алынған соң, оны броузер-программа өз мүмкіндігін пайдаланып, үлкейтіп ірі әріптермен жазуы ықтимал немесе тек экран жолдарының ортасына жылжытып қана көрсетуіне де болады. Ал егер бұл құжат мәтіні дыбыс синтезаторы арқылы берілетін болса, онда тақырып қаттырақ шығатын дауыс арқылы айтылып, одан соң аздап үзіліс жасалуы да мүмкін.

HTML тілінде мәтінді форматтау тәсілдерінің көптеген мүмкіндіктері бар, бірақ жалпы тұрғыдан алғанда құжаттың мәтіні мен оны безендіріп көрсету жолдарының айырмашылығы сақталып отырады. Мысалы, HTML тілінің соңғы [HTML 4.0] нұсқасында мәтінді форматтау командаларын пайдалану ұсынылмаған.

2.2 HTML командалары

HTML тілінің бастапқы берілген символдық мәтінді белгілейтін командалары *белгі* немесе *тәг* (tag) деп аталады. Тәг символдар тізбегінен тұрады. Кез келген тәг «кіші» (<) символынан басталады да, «үлкен» (>) символымен аяқталады. Осындай қос символ тізбегі *бұрыштық жақшалар* деп те аталады. Ашылатын бұрыштық жақшадан соң команда аты болып табылатын ағылшын тіліндегі *түйінді сөз – тәг* орналасады.

HTML тіліндегі әрбір тәг бір арнаулы қызмет атқарады. Тәгтің ағылшын тіліндегі аты оның қызметімен сәйкес келеді. Олардың жазылуында әріптер регистрі ешбір рөл атқармайды, бас әріпті де, кіші әріптерді де қатар қолдана беруге рұқсат етілген. Бірақ тәг атауларын мәтіннен айыру мақсатында оларды бас әріппен жазу қалыптасқан.

HTML тілінің бір тәгі әдетте құжаттың белгілі бір бөлігіне, мысалы бір азат жолқа ғана әсер етеді. Осыған орай керекті сөз тіркестерін қоршап тұрған екі тәг қатар қолданылады: бірі азат жолты – *ашады*, екіншісі – *жабады*. Ашатын тәг белгілі бір әсер ету ісін бастайды, ал жабатын тәг – сол әсерді аяқтайды. Жабу тәгтері қиғаш сызық символымен (/) басталуы тиіс.

Кейбір тәгтер өз жазылу орнына ғана әсерін тигізеді. Мұндайда жабу тәгі қажет болмай қалады да, тек жалқы тәг жазылады (мысалы, сызық жүргізу). Егер қате кетіп, тәг ретінде HTML тілінде қолданылмайтын түйінді сөз жазылып кетсе, онда оның ешбір әсері болмайды.

Броузер-программа арқылы құжат экранда көрсетілген шақта тәгтердің өздері бейнеленбей, тек керекті мәтін немесе суреттер көрініп, тәгтердің сол құжат мәтініне тигізетін әсері ғана білініп тұрады.

2.2.1 Тәг атрибуттары

Көбінесе ашылу тәгтерінің тигізетін әсерлерін айқындайтын немесе түрлендіретін олардың атрибуттары (сипаттамалары) болады. *Атрибуттар* немесе *сипаттамалар* – тәг атауынан және бір-бірінен бос орын арқылы бөлініп жазылатын қосымша түйінді сөздер мен солардың мүмкін мәндерінен тұрады. Көптеген атрибуттар оның *мәнін* жазуды талап етеді. Атрибут мәні оның түйінді сөзінен теңдік белгісі (=) арқылы бөлініп жазылады. Атрибут

мәні қостырнақшаға алынып жазылуы тиіс, бірақ кейде қостырнақшаны жазбаса да болады. Жабылу тәгтерінің ешқашанда атрибуттары болмайды.

HTML тәгтеріне мысалдар:

```
<HTML>
 <HEAD>  Жалпы тақырып  <TITLE> Терезе тақырыбы  </TITLE>
 </HEAD>
 <BODY> <H1> Ең ірі мәтін </H1> <BR>
    <H2> Кішілеу мәтін</H2>  <HR>
    <P> <H3> Бірінші азат жол </H3>
   HTML құжаты сол құжаттың негізгі мәтінінен және белгілеу тәгтерінен
   тұратын қарапайым символдар жиыны болып табылады. Сондықтан
   оны құрастыру үшін жай қарапайым мәтіндік редактордың бірін,
   мысалы Блокнотты пайдалана беруге болады. BR тәгі жаңа жолға
   көшуді қамтамасыз етеді. P тәгі жаңа азат жол  жасайды.  <HR>  <H3>
   Екінші азат жол </H3>
    <P> Азат жол үшін жабу тәгі қажет емес. HR тәгі көлденең  сызық
   сызады.  <HR>
 </BODY>
</HTML>
```

HTML тәгтерінің қосарланып жазылуы:

```
<HTML>    ...
</HTML>
<B>          ...
</B>
<HEAD>    ...
</HEAD>
<H3>         ...
</H3>
<LI>               ...
</LI>
```

HTML тәгтерінің жалқы жазылуы:

```
<BR>         ...
<HR>               ...
<FRAME>  ...
```

HTML тәгтерінің атрибуттарымен бірге жазылуы:

```
<BODY  BGCOLOR="YELLOW"  TEXT="BLUE" >      ...
<HR  COLOR=RED   SIZE=16   WIDTH=100%>   ...
<HR>  <A  HREF="FISH.JPG">  балық  </A>                  ...
<HR>  <IMG  SRC="DOG.JPG">
<P> <IMG  SRC="FISH.JPG"  WIDTH=500  HEIGIT=250>  ...
```

Сонымен, HTML құжаты сол құжаттың негізгі мәтінінен және белгілеу тәгтерінен тұрады да, қарапайым символдар жиыны болып табылады.

Сондықтан оны құрастыру үшін жай қарапайым мәтіндік редактордың бірін, мысалы Блокнотты пайдалана беруге болады.

Тәг атауларын жазу кезінде үлкен әріптер мен кіші әріптер бірдей болып есептеледі. Ашылу тәгінің жалпы жазылу ережесі, яғни синтаксисі төмендегідей:

<тәг_аты [атрибуттары]>

Тік жақшаға алынған тізбекті жазбай кетуге де болады, яғни көптеген тәгтер атрибутсыз жазыла береді. Ашылу тәгінен соң, негізгі құжат мәтіні жазылады да, соңында ешқашанда атрибуты болмайтын жабылу тәгі орналасады, оның синтаксисі:

</тәг_аты>

Ал атрибуттың жазылу ережесі:

атрибут_аты [= “мәні”]

Атрибуттың мәнін жазу міндетті емес, кейде оның тек аты ғана әсер ете алады, *тәг_аты, атрибут_аты* деген сөздердегі сызықша олардың тек бір сөзден тұратынын көрсетеді, яғни тәг аты және атрибут аты екі сөзден құрастырылуы мүмкін емес.

Экранға шығарылатын мәтін тізбегі кез келген әріптен, цифрлардан, тыныс белгілерінен және арнайы таңбалардан да (@, #, $, %) құрастырыла береді. Бірақ тәгтер жазуда арнайы мағынасы бар мына символдарды: <, >, &, “ жай мәтін таңбалары ретінде теру керек болса, онда ерекше символдық тізбектер – ескейп-тізбектер (&-тізбектер) қолданылады. Созылмайтын бос орын таңбасы да осы тізбек арқылы өрнектеледі, өйткені оны бірнеше рет теру web-құжаттағы сөздер арасындағы бос аралықты үлкейту тәсілі болып табылады. Ал, әдеттегі бірнеше босорын таңбасын броузер тек бір ғана босорын таңбасы ретінде ғана қабылдайды.

Осы ескейп-тізбектер символдары амперсенд (&) таңбасымен басталып, нүктелі үтір таңбасымен (;) аяқталады.

1- кесте

&-символдар тізбегі	Бейнелейтін символы
<;	< символы
>	> символы
&	& символы
"	“(қостырнақша) символы
	созылмайтын босорын символы

Ескейп-тізбектерді тек төменгі регистрде (кіші әріптер) теру керек, оларды " немесе & түрінде жазуға болмайды.

HTML тілінің көмегімен қарапайым WEB-парақтар жасап үйрену үшін компьютердегі Блокнот (Notepad) және Internet Explorer программаларымен жұмыс істей білу жеткілікті.

2.3. HTML-құжатты дайындау

HTML-құжат – бұл аты аты.htm түріндегі қарапайым мәтіндік құжат. Мынадай HTML - құжатты Блокнотта теріп шығыңыздар:

```
<html>
  <head> Менің алғашқы парағым
         <title> 1-ші мысал </title>
  </head>
 <body>
   <H1>  Сәлем!  </H1>
   Бұл HTML-құжаттың ең қарапайым мысалы.
   <P> Мәтіндер алты түрлі көлеммен көрсетіле алады, олардың
   түсін де, қаріп типін де, фон түсін де өзгертетін мүмкіндіктер
   бар. Мәтін ішіне суреттер, дыбыстық әуендер орналастыруға
   да болады.
   </P>
  </body>
</html>
```

Енді осы терілген құжатты, мысалы, **Айна.htm** деген атпен дискіде сақтау керек. Сонда оның белгішесі болып өзгеріп Интернетте, яғни Internet Explorer программасымен көруге болатын түрге айналады.

Бұл **Айна.htm** файлын бір мезетте Internet Explorer-де және Блокнотта ашып, оларды түрлендіре отырып, қатарластыра көруге болады. Ол үшін файлды Internet Explorer-де ашқаннан соң, *Түр – HTML түрінде (Вид – В виде HTML)* командаларын орындау қажет. Сонда файлдың алғашқы мәтіні *Блокнотта* ашылып, оны түрлендіріп өзгерту мүмкіндігіне ие боламыз. Қажетті өзгертулер енгізіп оны қайта дискіге жазып сақтап қою қажет. Осы өзгертулердің HTML-құжатта іске асқанын көру үшін, қайта Internet Explorer-ді ашып *Түр – Жаңалау (Вид – Обновить)* командасын орындау керек немесе *Саймандар тақтасындағы* осы командаға сәйкес батырманы басу қажет.

Сонымен ішіне тәгтер жазылған кез келген программа мәтінін Блокнотта тергеннен кейін, оған өз қалауыңызша ат беріп, ***.htm** (* – кез келген ат) түрінде сақтау керек. Тергеніңіздің нәтижесін экранда көру үшін, оның атын тышқанмен екі шерту керек немесе Internet Explorer-де ашу қажет.

Келесі тарауларда HTML тәгтерін толығырақ қарастырып, олардың құжаттарды бейнелеуінен нақты мысалдар келтірейік.

Бақылау сұрақтары:

1. Web-парақ дегеніміз не және олар қалай жасалады?
2. HTML тілі қандай қызмет атқарады?
3. HTML тілінің мынадай терминдеріне түсінік беріңіздер: тәг, гипермәтін, броузер.
4. HTML тәгтерінің қандай түрлері бар? Олардың жазылу синтаксисі қандай?
5. Тәг атрибуттары не үшін қолданылады? Олардың жазылу ережелерінен мысалдар келтіріңіздер.
6. Ескейп-тізбектер қандай мақсатта қолданылады, қалай жазылады?
7. HTML құжаттары қалай дайындалады? Оларды қалай өзгертуге болады?

2.3 HTML тілінің негізгі тәгтері

2.3.1 HTML құжатының құрылымын анықтау тәгтері

HTML құжатының кез келгені <HTML> тәгінен басталып, соған сәйкес </HTML> түріндегі жабылу тәгімен аяқталады. Осы екеуінің ортасында құжаттың тақырыптық бөлігі мен тұлғасы болып келетін негізгі бөлігі орналасады.

Құжаттың тақырыптық бөлігі <HEAD> және </HEAD> тәгтерінің ортасында тұрады да, жалпы құжат туралы мәлімет береді. Әдетте, бұл бөлікте <TITLE> ... </TITLE> тәгтерімен шектелетін құжаттың терезе маңдайшасында тұратын ресми атауы орналасады. Көптеген броузерлер оны терезе тақырыбында тұратын файл аты есебінде пайдаланады.

Осы құжатты принтер арқылы шығарғанда, броузер оны әр парақтың сол жақ жоғарғы бұрышына жазып отырады. Атаудың өте ұзын болмағаны дұрыс, әдетте ол 64 символдан аспауы керек.

Жазылатын мәтін құжаттың негізгі тұлғасы деп аталатын <BODY> ... </BODY> тәгтерінің ортасына жазылады. Бұл қос белгі HTML-құжаттың негізгі мазмұндық бөлігінің басын және соңын білдіреді.

Жоғарыда келтірілген төрт тәг HTML құжатының кез келгенінде болуы тиіс. Бірақ <HTML>, <TITLE> тәгтерін жазбай кетсе де болады, дегенмен HTML тілінің құрылымы олардың толық болуын талап етеді. Өйткені алдын ала тұтынушының қандай броузер пайдаланатыны, оның қалай жұмыс істейтіні программа құрушыға белгісіз болады ғой.

Түсініктемелер. Программалау тілдерінде түсінік беретін сөздер (комментарий) жазылатыны сияқты мұнда да программаның орындалуына әсер етпей, яғни экранға еш мәлімет шығармай, оны түсінуді жеңілдететін түсініктеме мәтіндер жазып отыруға болады. Түсініктеме мәтін <COMMENT>...</COMMENT> тәгтері ортасында орналасады.

HTML тілінің комментарийлерін тәг жазбай-ақ, арнайы символдардан <!-- кейін жалғастырып жазуға болады. Түсініктеме мәтін соңына --> символдары жазылуы тиіс. Түсінік мәтін «үлкен» таңбасынан (>) өзге кез келген символдардан құрастырыла береді.

<!-- мынау түсініктеме мәтін тізбегі -->

Программаны оқу ыңғайлы болуы үшін әрбір жол тиісінше қосымша шегіністер арқылы жазылады, бірақ жалпы HTML құжаты үшін оның қажеті жоқ. Тіпті, броузерлер HTML-файлдардағы жолдың соңы символын және көптеген бос орындарды есепке алмайды. Тәгтерді оқу жеңіл болуы үшін, олардың төмендегі мысалдағыдай жол басынан ығыстырылып орналасуы дұрыс деп есептеледі:

```
<html>
  <head> Менің алғашқы парағым
     <title> 1- мысал </title>
  </head>
   <body> <H1> Сәлем! </H1>
```

```
    <P>  Бұл HTML – құжаттың ең қарапайым
мысалы.</P>
    <P> Мәтіндер алты түрлі  көлеммен
көрсетіле алады, олардың
        түсін де, қаріп типін де, фон түсін де
өзгертетін мүмкіндіктер
        бар. Мәтін ішіне суреттер
орналастыруға да болады.
    </P>
  </body>
</html>
```

Жоғарыдағы программалық мәтіннің нәтижесі

Сәлем!

Бұл HTML – құжаттың ең қарапайым мысалы.

Мәтіндер алты түрлі көлеммен көрсетіле алады, олардың түсін де, қаріп типін де, фон түсін де өзгертетін мүмкіндіктер бар. Мәтін ішіне суреттер орналастыруға да болады.

2.3.2 HTML тілінің негізгі тәгтері

<H1>...</H1> – <H6> ... </H6>

<Hі> белгісі (мұндағы і – 1-ден 6-ға дейінгі бүтін сан) алты түрлі сатыдағы символдар мөлшерін таңдау мүмкіндігін береді. Бірінші сатыдағы тақырып – ең ірісі, алтыншы сатыдағы – ең кішісі. Бұл тәг көрсетілмесе, экранға <H3> мөлшеріне сәйкес мәтін шығады.

<P> ... </P> немесе жалғыз **<P>**

Бұндай қос белгі азат жолты сипаттайды. Егер ол жабылмаса, келесі азат жолтың басы алдыңғы азат жолтың соңы екенін білдіреді. Жалпы <P> және </P> белгілерінің арасына жазылғандардың барлығы бір азат жол ретінде қабылданады.

<Hі> және <P> белгілерінің қосымша **ALIGN** (ағылшынның “туралау” деген сөзі) атрибуты болуы мүмкін. Мысалы:

```
<H1  ALIGN=CENTER> Тақырыпты ортаға жылжыту </H1>
```

немесе

```
<P ALIGN=RIGHT>Азат жолты оң жақ шетке туралау түрі </P>
```

Осыларды төмендегі 2-1 мысалда қарастырайық:

```
<html>  <head> <title> 2-1 мысал </title> </head>
  <body> <H1  ALIGN=CENTER> Сәлем! </H1>
     <H2> Бұл HTML-құжаттың сәл күрделірек мысалы </H2>
     <P> Енді біз азат жолты, тек сол жақ шетке ғана туралап жазбай,
     <P ALIGN=CENTER> ортаға қарай немесе </P>
     <P ALIGN=RIGHT> оң жақ шетке де туралауға болатындығын
      білеміз.
```

```
  </body>
</html>
```

Осы программалық мәтін нәтижесі

Сәлем!

Бұл HTML-құжаттың сәл күрделірек мысалы

Енді біз азат жолты, тек сол жақ шетке ғана туралап жазбай,

ортаға қарай немесе

оң жақ шетке де туралауға болатындығын білеміз.

Бұдан былай қарапайым HTML-құжатты осы мысалдарда көрсетілгендей құрастыруға мүмкіндік бар.

Енді біз осы қарапайым HTML-құжатты қалай жақсартуға болатындығын сөз етеміз. Жаңа жолға көшу белгісінен бастайық.

 тәгі азат жолты бөлмей, келесі сөзді жаңа жолға көшіру керек болған жағдайда қолданылады. Ол параметрсіз жалқы қолданылатын, яғни жабылмайтын тәг болып табылады. Мысалы:

```
...<BR> Абай Құнанбаев <BR> Мұхтар Әуезов <BR>
Сәкен Сейфуллин...
```

Бұл жолдар экранға мынадай мәліметтер шығарады:

Абай Құнанбаев
Мұхтар Әуезов
Сәкен Сейфуллин

Ал мына жолдар:

```
<P> Мына жолдар екі-үш қатарға
      бөлініп жазылғанымен
    олар бір-екі жолға бірге жазылады </P>
```

экранға мынадай мәлімет шығарады:

Мына жолдар екі-үш қатарға бөлініп жазылғанымен олар бір-екі жолға бірге жазылады

Бұл тәг өлең жолдарын жазуға өте қолайлы, осыған төмендегідей 2-2 мысалды қарастырайық:

2 - кесте

HTML тілінде терілуі	Оның экрандағы нәтижесі
<html> <head> <title> 2-2 мысал </title> </head> <body><H1 ALIGN=CENTER> Өлең </H1> <H2 ALIGN=CENTER> Мұқағали </H2> <P> Көзің қайда көшеден мені іздеген, Сөзің қайда екеуміз егіз деген. Терезеңнің алдына келіп тұрмын,	**Өлең** **Мұқағали** Көзің қайда көшеден мені іздеген, Сөзің қайда екеуміз егіз деген. Терезеңнің алдына келіп тұрмын, Көгершіндей қысты күн жем іздеген. **Абай** Айттым сәлем, Қаламқас Саған құрбан мал мен бас. Сағынғаннан сені ойлап,

  Кептердей қысты күні жем іздеген. <H2 ALIGN=CENTER> Абай </H2> <P> Айттым сәлем, Қаламқас Саған құрбан мал мен бас. Сағынғаннан сені ойлап, Келер көзге ыстық жас.<P> Көзімнің қарасы, Көңілімнің санасы. Бітпейді іштегі, Ғашықтық жарасы. </body> </html>	Келер көзге ыстық жас. Көзімнің қарасы, Көңілімнің санасы. Бітпейді іштегі, Ғашықтық жарасы.

<HR> тәгі экран бетінде көлденең сызық жүргізеді. Ол параметрсіз қолданылса, төмендегідей көлденең жолды толық алып тұрған қара сызық жүргізеді:

Ал параметр арқылы оның түсін (COLOR=”түс”), ұзындығын (WIDTH=n% пайызбен, экран еніне байланысты сызықтың пайызбен берілген ұзындығын анықтайды) және қалыңдығын (SIZE=n пиксель, яғни нүктелер саны) өзгертуге болады. Төменде бірнеше көлденең сызықтар салудан мысал келтірілген:

```
<HTML>  <HEAD> <title>Сызықтар</title> </HEAD>
<BODY>
  <H1> Көлденең сызықтар жиыны </H1>
  <HR COLOR=RED       SIZE=2   WIDTH=100%><BR>
  <HR COLOR=GREEN  SIZE=4   WIDTH=50%><BR>
  <HR COLOR=BLUE     SIZE=8   WIDTH=25%><BR>
  <HR COLOR=BLACK    SIZE=16  WIDTH=12%><BR>
</BODY>
</HTML>
```

Бұл жолдардың нәтижесі төмендегідей (сызықтар түсі әр түрлі):

2.3.3 Құжаттың негізгі бөлігі

Құжаттың тақырыптан кейінгі негізгі бөлігі <BODY> ... </BODY> тәгтерінің ортасына орналасады. Мұнда көптеген атрибуттар, яғни параметрлер болады. Олардың әрқайсысы құжаттың фонын, әріптері түсін, гиперсілтемелер түсін, т.б. анықтайды. Бұл тәгтің негізгі атрибуттары: **BACKGROUND, BGCOLOR, TEXT, LINK, VLINK** және **ALINK.** Олар төмендегі түрде жазылады:

<BODY 1-параметр=мәні 2-параметр=мәні 3-параметр=мәні ...>
құжаттың негізгі тұлғасы ...
</BODY>

Параметрлер тізімін толық берудің қажеті жоқ, көбінесе олардың бірде біреуі болмауы да мүмкін, мұндайда олардың алдын ала (үнсіз) келісім бойынша бекітілген мәндері қолданылады.

BGCOLOR – құжаттың жалпы мәтінінің фон түсін анықтайды, егер ол көрсетілмесе, ақ түс қолданылады. Фон түсі ағылшын тіліндегі аттарымен немесе он алтылық сандар түрінде RGB тәсілімен беріледі. Олар жайында кейінірек айтылады. Мысалы:

<body bgcolor=”yellow”> мұнда фон сары түсті болады.

TEXT – мәтін әріптерінің түсін анықтайды, егер ол жазылмаса, келісім бойынша қара түс қабылданған. Фон түсін өзгерткенде соған үйлесімді символдар түсі бекітіледі. Бұл да ағылшын тіліндегі аттарымен немесе он алтылық сандар түрінде RGB тәсілімен беріледі.

LINK – гипермәтіндік сілтеме ретінде қабылданған сөз тіркесінің түсін белгілейді. Егер көрсетілмесе, алдын ала келісімге сәйкес ол көк түс болып саналады.

VLINK – пайдаланылған гипермәтіндік сілтеме түсін анықтайды. Келісім бойынша ол қызылқоңыр түс болып саналады.

ALINK – гипермәтіндік сілтемені курсор көрсетіп тұрған кездегі оның түсін анықтайды. Бұл параметр өте сирек өзгертіледі.

BACKGROUND – мәтіннің фонында бірнеше рет қайталанып, тұсқағаз (обои) ретінде орналасатын суретті анықтайды. Сурет файлының типі **gif** немесе **jpg** болуы тиіс. Суреттің адресі көрсетілсе, ол Интернет желісінен тауып алынады. Мысалы:

<body bgcolor=lightyellow text=red link=purple vlink=maroon alink=fuschia background="kbtu.jpg">

2.3.4 Құжаттардағы түстерді анықтау

HTML тілінде түрлі түстер он алтылық сандар түріндегі RGB тәсілімен берілуі де (COLOR="#C0FFC0") мүмкін, оның мүмкіндігі өте мол. Мұндағы алғашқы екі он алтылық цифр (C0) қызыл түс бөлігін, келесі екі цифр (FF) – жасыл түс бөлігін, соңғы екі цифр көк түс (C0) бөлігін анықтайды. Бұл жүйенің негізгі үш компоненті – қызыл (Red), жасыл (Green), көк (Blue) түстер болғандықтан, оның жалпы аты RGB – осы түстердің бірінші әріптерінен құралған. Әр компонентке он алтылық санау жүйесінің 00-ден FF-ке (ондық санау жүйесінің 0-ден 255-ке дейінгі сандар) дейінгі саны сәйкес келеді. Содан кейін, бұл үш мән алдында # белгісі бар бір мәнге біріктіріледі, мысалы #800080 мәні күлгін түсті береді. Барлық түстерді он алтылық санау жүйесі бойынша бір-бірінен ажырату қиын болғандықтан, HTML тіліндегі түстердің ағылшын тіліндегі атаулары жиі қолданылады. Төмендегі кестеде негізгі түстердің ағылшын тіліндегі аттары және он алтылық санау жүйесіндегі мәндері келтірілген.

3 - кесте Негізгі түстердің RGB форматында жазылу кодтары

Түс атауы	RGB коды	Түс атауы	RGB коды
Black (қара)	"#000000"	Green (жасыл)	"#008000"
Silver (күміс түсті)	"#C0C0C0"	Lime (лимон түсті)	"#00FF00"
Gray (сұр)	"#808080"	Olive (олиф түсті)	"#808000"
White (ақ)	"#FFFFFF"	Yellow (сары)	"#FFFF00"
Maroon (қызыл күрең)	"#800000"	Navy (қара көк)	"#000080"
Red (қызыл)	"#FF0000"	Blue (көк)	"#0000FF"
Purple (күлгін)	"#800080"	Teal (жасыл көк)	"#008080"
Fuchsia (қызғыш, фуксия)	"#FF00FF"	Aqua (ақшыл көк)	"#00FFFF"

Мысалы: <body bgcolor = white text = black link = red vlink = maroon alink = fuschia background = "face.jpg">.

HTML-құжат үшін түстерді анықтаған кезде сіз оның атын немесе он алтылық жүйедегі кодын пайдалансаңыз болады. Мысалы, төмендегі жолдардың қызметі бірдей:

<BODY BGCOLOR=”#FFFFFF”>
<BODY BGCOLOR=”WHITE”>

2.3.5 Мәтіндерді түрлендіру тәгі

FONT тәгі HTML-дағы мәтіннің сыртқа бейнесін түрлендіреді. Ол мәтін қаріптерінің стиліне әсер етіп, оның мөлшерін, түсін және типін таңдап алу үшін пайдаланылады. Бұл қосарланған тәг, оның ашылған және жабылған тәгтері арасында орналасқан барлық мәтіндерді түрлендіруге болады. Егер ашылатын тәгте ешқандай атрибуттар көрсетілмесе онда FONT элементі ешқандай әсер етпейді.

FONT элементінің кез келген мәтін үшін қолдануға болатын FACE (гарнитура, тип), SIZE (мәтін көлемі) және COLOR (қаріп түсі) атрибуттары арқылы құжаттағы мәтіннің сыртқы түрін өзгертуге болады.

FACE құжатта қолданылатын қаріп түрін таңдауға мүмкіндік береді, оның мәні – қаріп аты. Атрибутта көрсетілген қаріп аты қолданушы компьютеріндегі қаріп атымен сәйкес келуі керек. Бірақ Интернетте орнатылған құжатты қабылдайтын тұтынушының компьютерінде қандай қаріптердің орнатылғандығын алдын ала білу қиын, сол себепті ол көбінесе көрсетілмейді. Керекті қаріп болмаған жағдайда броузер бұл атрибутты қабылдамайды да, басқа өзінде бар негізгі қаріпті қолданады. Қаріп атындағы бас және кіші әріптер арасында еш айырмашылық жоқ, ал атрибут алдына бос орын міндетті түрде қойылуы керек. Төменде қаріпті таңдаудан 2-3 мысалы келтірілген.

<HTML><HEAD><TITLE>2-3 мысал. Қаріп түрін таңдау</TITLE> </HEAD>

```
  <BODY>
    <FONT FACE=”KZ Arial”> Бұл жерде басқа қаріп пайдаланылған</FONT>
  </BODY>
 </HTML>
```

Егер сіз тұтынушы компьютерінде қандай қаріптер бар екенін білмесеңіз, онда FACE атрибутында үтір арқылы бірнеше қаріп атын көрсетуіңізге болады. Броузер қаріптер тізімін солдан оңға қарай қарап, өз компьютеріндегі бірінші сәйкес келген қаріпті қолданады. Төменде бірнеше қаріпті қолдану мысалы көрсетілген.

```
<HTML> <HEAD> <TITLE> Қаріп таңдау мысалы </TITLE> </HEAD>
 <BODY>
   <FONT FACE=”Verdana”, “Arial”, “Helvetica”> Бұл қаріп түрін таңдау.
   </FONT>
 </BODY>
</HTML>
```

Бұл мысалда негізгі қаріп ретінде Verdana көрсетілген, ол болмаған жағдайда Arial, Helvetica қаріптерін қолдануға болады.

SIZE – бұл элемент мәтіндегі әріптің көлемін (мөлшерін) тағайындайды, яғни символдардың биіктігін таңдауға мүмкіндік береді. Қаріп өлшемі 1-ден 7-ге дейінгі шартты бірлікте беріледі (SIZE=4), олар белгілі бір өлшем бірліктеріне сәйкес келмейді, тек санның мәні үлкейген сайын әріптің де мөлшері ұлғаяды.

4 - кесте

HTML программасы үзіндісі	Оның нәтижесі
 size=1 size=2 size=3 size=4 size=5 size=6 size=7 	size=1 size=2 size=3 size=4 size=5 size=6 size=7

Егер сан көрсетілмесе, келісім бойынша ол 3-ке тең болып саналады. *Size* атрибутын екі түрлі әдіспен қолдануға болады: қаріптің абсолютті өлшемін көрсету, мысалы, SIZE=5 немесе салыстырмалы өлшемді көрсету SIZE=+2. Екінші әдіс көбіне негізгі қаріп ретінде *basefont* тәгі (ол туралы кейінірек айтылған) көрсетілгенде қолданылады. Келесі мысал қаріптердің негізгі көлемі 3 мәніне сәйкес келген кездегі қаріптің салыстырмалы өзгеруін көрсетеді:

```
size=-2  size=-1  size = +1  size=+2  size=+3  size=+4
```

Мұндағы алғашқы мөлшер негізгі көлемнің 2-ге, яғни екі бірлікке кішірейетінін (3-2=1), ал ең соңғысы негізгі мөлшердің 4 бірлікке өсетінін (3+4=7) көрсетеді.

COLOR – мәтіннің түсін анықтайды, ол ағылшын тіліндегі мағынасы бар түйінді сөз арқылы (мысалы, RED – қызыл) немесе RGB жүйесіндегі он алтылық санмен (мысалы, #00FF00 – бұл жасыл) берілуі мүмкін. Төменде құжат түстерін анықтауға арналған мысалдар келтірілген.

```
<HTML>
  <HEAD> <TITLE>Түстер таңдауға мысал</TITLE> </HEAD>
 <BODY>
   <FONT COLOR=”#FF0000”> Бұл мәтіннің түсі қызыл</FONT><BR>
   <FONT COLOR= “GREEN”>Бұл мәтіннің түсі жасыл</ FONT><BR >
    Алдарыңызда <FONT COLOR="red" FACE="ARIAL" SIZE="3">
    үшінші мөлшермен arial типімен жазылған қызыл түсті әріптер
   </FONT>
 </BODY>
</HTML>
```

Нәтижесі:

Бұл мәтіннің түсі қызыл
Бұл мәтіннің түсі жасыл
Алдарыңызда **үшінші мөлшермен arial типімен жазылған қызыл түсті әріптер**

Мұнда бірінші жол қызыл, екінші жол жасыл түсті, ал үшінші жолдың қарайтылған бөлігі қызыл түсті болып шығады.

Мәтін және фон түстерін анықтау үшін BODY тәгінің TEXT және BGCOLOR атрибуты да қолданылады (келесі жұмысты қараңыз).

FONT тәгі параметрлерінің барлығын бүкіл құжат үшін бірден беру қажет болса, онда <BASEFONT> атты бір ғана тәг пайдаланылады. Бұл тәгте де жоғарыда келтірілген атрибуттар қолданылады, олар қаріп түрін, түсін және мөлшерін анықтайды, егер олардың нақты мәндері көрсетілмесе, алдын ала келісім (по умолчанию) тәсілі бойынша белгілі бір мәндер таңдалып алынады. Мысалы:

```
<HTML  <HEAD> <TITLE> Қаріп түрін басқару </TITLE> </HEAD>
  <BODY>
    <BASEFONT  SIZE=4  FACE=” KZ Arial”>
    <P>  Бұл мәтін стандарттан тыс әріп түрін пайдаланады.
    <P><FONT SIZE=2  FACE=”KZ Times New Roman” COLOR=”GREEN”>
        Бұл мәтін әріптері ұсақтау және ол басқа қаріп түрі мен басқа
        түсті қолданады. </FONT>
  </BODY>
</HTML>
```

немесе

```
<HTML <HEAD> <H2> Қаріп типін, мөлшерін басқару </H2> </HEAD>
```

```
<BODY>
  <BASEFONT SIZE=4  FACE=”KZ Arial”>
    Негізгі қаріп Kz Arial типінде төртінші өлшеммен жазылған
  <P> <FONT SIZE=-2 FACE= ”KZ Times New Roman” COLOR=”GREEN”>
  Бұл мәтін алдыңғыдан екі өлшемге ұсақтау және ол басқа қаріп типі мен жасыл түсті қолданады. </FONT>
 </BODY>
</HTML>
```

Тапсырмалар:

1. Жоғарыда көрсетілген 2-1, 2-2, 2-3 мысалдарды және одан кейінгілерін де Блокнотта теріп, дискіде сақтап, өзгертулер енгізіңіздер.
2. Студенттік кеңестің кезекті мәжілісі болатындығы жайлы хабарлама жазып, құжаттың нәтижесін Internet Explorer-де көргеннен кейін:
   а) тақырып және бөлім аттарын ортаға жылжытып қойыңыз;
   ә) бірінші азат жолтың түсін – қызыл, екінші азат жолты – көк, ал үшінші азат жолты – жасыл етіп, мәтін фонын сары түске өзгертіңіздер;
   б) әр азат жолтан кейін көлденең сызықтар сызып, олардың түстерін және ендері мен қалыңдықтарын қалауларыңызша алыңыздар.
3. Топ студенттерінің тізімін жазыңыздар. Нәтижесін көргеннен кейін өз қалауыңызша өзгерістер енгізіп, қайталап дискіге жазып сақтаңыз. Өзгерістеріңіздің HTML-құжатқа енгендігін тексеріңіз.
4. Жуық арада болатын мерекеге байланысты, достарыңызға құттықтау жазыңыз. Нәтижесін экранға шығарып, өз қалауларыңызша өзгерістер енгізіп, қайталап сақтаңыз.

Бақылау сұрақтары:

1. HTML құжатының құрылымы қандай?
2. HTML тілінің негізгі тәгтерін, олардың жазылуын көрсетіңіздер.
3. HTML құжаты ішінде түсініктеме беру үшін не істеу керек?
4. Тақырыптар мәтіндерінің көлемі қалай өзгертіледі?
5. ALIGN атрибуты не үшін қолданылады және ол қандай тәгтермен бірге беріле алады?
6. Өлең жолдарын жазу кезінде қандай тәг жиі қолданылады?
7. HTML тіліндегі азат жол ерекшеліктері неде? Жаңа азат жол ашу (жабу) үшін қолданылатын тәгті жазып көрсетіңіз.
8. Көлденең сызық жүргізу тәгінің атрибуттарын көрсетіңіздер.
9. Қаріптер параметрлерін өзгерту атрибуттары қандай?
10. <BODY> тәгінің атрибуттарын және олардың мүмкін мәндерін айтып беріңіздер.
11. Мәтін түсін, фон түсін қалай беруге болады?
12. <FONT> тәгінің SIZE атрибутының мәндері қалай жазылады?
13. Егер сіз көрсеткен қаріп типі тұтынушы компьютерінде болмаған жағдайда не істеуге болады?

2.4 ҚҰЖАТТЫ ФОРМАТТАУ

HTML-құжаттың үзінділерін экранда бейнелеу тәсілдерін өзгертіп, гиперсілтемелер мен суреттерді пайдалануды меңгеру және оның қаріптерін безендіру істері оларды *форматтау* деп аталады.

2.4.1 Қаріптерді форматтау

HTML-да қаріптерді форматтаудың екі жолы бар. Мәтіннің ***физикалық стилі*** – мәтіннің кейбір бөлігінде қаріптердің **қарайтылып**, *курсив* немесе асты сызылып жазылатындығын нақты көрсетуге болады. ***Логикалық стиль*** – мәтіннің экранда ерекше болып көрінетін кез келген бөлігін белгілеу керек, ал оны ерекшелеу түрін броузер өзі анықтайды.

2.4.1.1 Физикалық стиль

Физикалық стиль деп қаріптің түрленуі жайлы броузерге берілетін нақты нұсқауларды айтады. Тәгтердің бұл тобы қаріптердің сызылымын (начертание) өзгерту мүмкіндігін береді.

<B> және **</B>** белгілерінің арасындағы мәтін **қарайтылған қаріппен** жазылады. **<I>** және **</I>** белгілерінің арасындағы мәтін *курсив қаріппен*, ал **<U>** және **</U>** белгілерінің арасындағы мәтін асты сызылған қаріппен жазылады. **<TT>** және **</TT>** белгілерінің қызметі ерекше. Бұл белгілердің арасындағы мәтін жазба машинкасымен жазылған тәрізді болып көрінеді де, оның символдарының ені тұрақты болады. Мысалы:

```
<HTML <HEAD><H3> Қаріп типін,түрін, мөлшерін басқару </H3></HEAD>
  <BODY> <P> <B> қарайтылған қаріп түрі </B>
      <P> <I> қисайтылған курсивпен жазылған қаріп түрі </I>
      <P>  <U> асты сызылған қаріп түрі </U>
    <P>  <S>  белінен сызылған қаріп түрі </S>
    <P> <TT>  жазба машинкасындағы сияқты </TT>
  </BODY> </HTML>
```

Мұның экрандағы нәтижесі:

Қаріп типін,түрін, мөлшерін басқару
қарайтылған қаріп түрі
қисайтылған курсивпен жазылған қаріп түрі
асты сызылған қаріп түрі
~~белінен сызылған қаріп түрі~~
жазба машинкасындағы сияқты

2.4.1.2 Логикалық стильдер

Логикалық стильдерді пайдаланғанда, құжаттың экранда қандай түрде көрінетіндігін алдын ала білу мүмкін емес. Логикалық стильді әр броузер әр түрлі етіп қабылдайды. Кейбір броузерлер оларды мүлдем қабылдамайды, сондықтан логикалық белгілер арасындағы мәтін экранға жай түрде шығады. Көп тараған логикалық стиль белгілері:

<EM> – </EM> ағылшынның *emphasis – акцент* деген сөзінен, ол *курсив* қаріптерге ұқсас болып келеді.

<STRONG> – </STRONG> ағылшынның **strong emphasis – ерекше ақцент** деген сөз, ол **қарайтылған** қаріп түрінде көрінеді**.**

<CODE> – </CODE> бастапқы мәтін бөліктері үшін пайдалануға негізделген.

<SAMP> – </SAMP> ағылшынның *sample – нұсқау (үлгі)* деген сөзі. Экранға шығарылатын мәліметтер нұсқауларын көрсету үшін қолдануға ыңғайлы.

<KBD> – </KBD> ағылшынның *keyboard – пернетақта (клавиатура)* сөзі. Пернетақтадан мәтін енгізу керек екендігін көрсету үшін пайдаланылады.

<VAR> – </VAR> ағылшынның *variable – айнымалы* сөзі. Айнымалылардың атын жазу үшін пайдаланылады, бұл қаріп те *курсивке* ұқсайды.

Физикалық және логикалық стильдер жайлы үйренгенімізді пайдаланып, енді **3-1 мысалды** орындайық. Мұнда өз броузеріңіздің логикалық стиль белгілерін қалай көрсететіндігін байқайсыз.

```
<HTML>  <HEAD>  <TITLE> 3-1 мысал </TITLE>  </HEAD>
<BODY> <CENTER> <H3>Мәтін фрагменттерін қаріптермен
белгілеу</H3> <HR> <P>  Енді біз мәтін бөліктерін әр
түрлі етіп белгілеуге болатындығын білеміз.
<P> Сонымен қатар, бірнеше логикалық стильдер бар:
<P><EM>EM – ағылшынның emphasis – акцент деген
сөзінен, курсив тәрізді, </EM>
<P><STRONG> STRONG – ағылшынның strong emphasis –
ерекше акцент деген  сөзінен, </STRONG><BR>
<CODE>CODE – бастапқы мәтін бөліктерін көрсету
үшін</CODE> <BR> <SAMP>SAMP – ағылшынның sample – үлгі
деген сөзінен,</SAMP> <BR> <KBD>KBD – ағылшынның
keyboard – пернетақта (клавиатура) деген сөзінен,
</KBD> <BR>
<VAR>VAR – ағылшынның variable – айнымалы деген
сөзінен </VAR> <HR> </CENTER>
</BODY>
</HTML>
```

Мұның экрандағы нәтижесі:

Мәтін бөліктерін қаріптермен белгілеу

Сонымен қатар, бірнеше логикалық стильдер бар:

EM – ағылшынның emphasis – акцент деген сөзінен, курсив тәрізді
STRONG – ағылшынның strong emphasis – ерекше акцент деген сөзінен,
CODE – бастапқы мәтін бөліктерін көрсету үшін
SAMP – ағылшынның sample – үлгі деген сөзінен
KBD – ағылшынның keyboard – пернетақта (клавиатура) деген сөзінен
VAR – ағылшынның variable – айнымалы деген сөзінен

2.4.1.3 Жоғарғы және төменгі индекстер

Көбінесе формула элементтеріндегі дәрежелерді немесе индекстерді жазу кезінде символдарды жоғары немесе төмен ығыстырып жазу керек болады. Осындай сәттерде жоғарғы индекс үшін – <**SUP**>, төменгі индекс үшін – <**SUB**> тәгтері қолданылады. Мысалы, H_2O сөз тіркесін жазу үшін: **H₂ O** жолдарын, ал $E=mc^2$ жолдарын жазу керек болса, **E = mc²** тәгтерін енгізу керек.

2.4.2 Гиперсілтеме бойынша ауысу

Форматтау тәсілдерін меңгеру үшін Блокнот және Internet Explorer программаларынан өзге Интернеттен немесе кітаптардан сканер арқылы көшіріліп алынған суреттік файлдар: жан-жануарлар (мысық, ит) суреттері – cat01.jpg, cat02.jpg, dog01.jpg, dog02.jpg, dog03.jpg, Қазақстан елтаңбасы (гербі) – gerbRK.gif мен туы – znamjaRK.gif болуы тиіс. Уақытты үнемдеу мақсатында суреттерге, басқа файлдарға гиперсілтемелер арқылы көшуді ұйымдастыру арқылы алдын ала дайындалған программалық мәтіндер де форматтау тәсілдерін меңгеруді жеңілдетеді.

Web-парақтардың басқа Web-парақтарға қатысты сілтемелерінің болуы – World Wide Web жүйесінің ең тартымды ерекшеліктерінің бірі. HTML құжаттарында гипермәтіндік сілтемелерді құру өте жеңіл. Ол үшін атрибуты, яғни параметрі бар <A ...> және </A> тәгтері пайдаланылады. Жалпы сілтемелер жасаған кезде мынадай ережелерді есте сақтаған жөн.

1.Гиперсілтемелерді құрған кезде **HREF=**”...” атрибуты міндетті түрде қажет. Оның мәні сол сілтеме көрсетіп тұрған файл атынан немесе оның Интернеттегі URL-адресінен тұрады. Сілтеме мәтін <A> мен </A> тәгтерінің арасына орналасады. Сілтеме сөз броузерде бейнеленген кезде көбінесе оның асты сызылып, көк түспен бейнеленеді. Сол сілтеме сөздің үстіне курсорды қойып тышқанды шерту көрсетілген файлға немесе желі бойындағы URL-адреске көшуді қамтамасыз етеді. Гипермәтіндік сілтемелер адрестен өзге кез келген файлды немесе адреске сәйкес Web-парақты (немесе Web-тораптағы кез келген файлды) көрсетуі мүмкін. Мысалы, мына жол:

<AHREF=”C:\МОИ ДОКУМЕНТЫ\СЕРІК.HTM”> *файл* </A>

экранға көк түспен боялған ***файл*** сөзін бейнелейді, сол сөзді тышқанмен шерту *Менің құжаттарым* (оны көрсетпесек, ағымдағы) бумасы ішіндегі СЕРІК.HTM файлына көшу әрекетін орындайды. Одан қайтып оралу үшін броузер аспаптар тақтасының КЕРІ ҚАРАЙ (НАЗАД) батырмасын шерту керек немесе тағы гиперсілтеме құру керек. Төмендегі жол:

<A HREF="dog.jpg"> *ит* </A>

экранға *ит* сөзін шығарады, оны шерту ағымдағы бумадағы сурет салынған **dog.jpg** файлына көшу ісін атқарады.

Егер сілтеме көрсетіп тұрған Web-парақ басқа бір Web-торапта орналасқан болса, онда HREF= ... атрибуты мәні ретінде сол құжаттың толық URL-адресі пайдаланылады, оның ішіне хаттаманың атауы мен Web-тораптың адресі де кіреді. Осындай сілтемелер *сыртқы сілтеме* деп аталады. Мысалы, мынадай жол:

<A HREF=”http://www.microsoft.com> *Microsoft* </A>

экранға ***Microsoft*** сөзін шығарып, оны шерту Интернет желісіндегі Microsoft мекемесының web-парағының алғашқы бетіне көшуді қамтамасыз етеді. Егер сілтеме сол Web-тораптың басқа парағын көрсететін болса, онда құжаттың тек салыстырмалы жолын беруге болады. Мұндайда *ішкі сілтеме* құрылады. Ішкі сілтемелерді пайдаланған қолайлырақ, себебі Web-торапты басқа серверге ауыстырған кезде, жеке құжаттар адресіне өзгеріс енгізу талап етілмейді.

2. Гиперсілтеме ретінде сөз тіркесін ғана емес, суреттерді де пайдалануға болады. Суреттер мәтін сияқты сілтеме бола алады. Бұл үшін <IMG ...> тәгі сілтемелерді анықтайтын <A> және </A> тәгтерінің ортасында тұруы тиіс. Суреттік сілтеме көк түсті қоршаудың ішіне орналастырылады. Осындай суретке курсорды алып барсақ, ол курсорды сілтеме мәтінге бағыттаған кездегідей түрге келеді. Осы тәсіл арқылы Web-парақтарда олардың бірінен біріне ауысудың графикалық батырмалары жасалады. Мысалы:

<A HREF="dog.htm"><IMG SRC="dog.gif"></A>

жолында **<IMG SRC="dog.gif">** тіркесі гиперсілтеме ретінде кішкене ит суретін береді де, оны шерту иттер туралы мәлімет беретін dog.htm файлына ауыстырады (IMG туралы кейінірек айтылған).

3. Егер файл басына емес, оның мәтінінің ішіндегі басқа бір қажетті орынға көшу керек болса, онда сол орынға алдын ала анкер (белгіленген сөз) орнатамыз. Сонда гипермәтіндік сілтемелер сол парақ ішіндегі анкері бар белгілі бір орынды көрсетуі мүмкін. Анкер құруда да <A> және </A> тәгтері қолданылады, бірақ мұндайда HREF= ”#....” атрибутына қосымша NAME= ”...” атрибутын пайдалану керек. Ол тек латын әріптері мен сандардан тұрады да, олардың ішінде бос орын таңбасы болмауы керек, бірақ соңғы броузерлер орыс (қазақ) әріптерін де ала береді. Мысалы:

<A HREF="2.htm#AAA">AAA анкеріне көшу</A>
<A NAME="AAA">Көшу орындалды</A>

жолдары 2.htm файлындағы AAA сөзіне көшіреді. Ал егер көріп отырған мәтін ішіндегі басқа бір 1.1 тіркесіне көшу үшін:

<A HREF="#1.1">1.1 анкеріне көшу</A>
<A NAME="1.1"> </A>
қатарларын жазу қажет.

Ескерту. Орнатылған анкерге сілтеме жасау үшін, URL адресінен кейін # таңбасымен бөлінген анкер атауы көрсетілуі керек. Бір құжат ішіндегі анкер алдына адрес жазылмайды. Мысалы (3-1 мысал):

...
<H2>3-1 мысал</H2>
<HR> <A HREF="#3.1">3.1-ге көшу </A >
<A Name="3.1"> </A>

 <H3> 1.1. Бірінші азат жол</H3>
Егер файл басы емес, оның мәтінінің ішіндегі басқа бір қажетті орынға көшу керек болса, онда сол орынға алдын ала анкер орнатамыз. Сонда гипермәтіндік сілтемелер сол парақ ішіндегі анкері бар белгілі бір орынды көрсетуі мүмкін.
<HR> <H3> 2.1. Екінші азат жол </H3>
Егер сілтеме сол Web-тораптың басқа парағын көрсететін болса, онда құжаттың тек салыстырмалы жолын беруге болады. Мұндайда ішкі сілтеме құрылады.
<HR> <H3> 3.1. Үшінші азат жол</H3>
Мұндайда HREF= ”#....” атрибутына қосымша NAME= ”...” атрибутын пайдалану керек. Ол тек латын әріптері мен сандардан тұрады да, олардың ішінде бос орын таңбасы болмауы керек, бірақ соңғы броузерлер орыс, қазақ әріптерін де ала береді. <HR> ...

2.4.3 HTML-құжатқа суреттер енгізу

Суреттік бейнелер Web-парақтарды әшекейлеп безендіру кезінде маңызды рөл атқарады. Суреттердің өздері HTML құжаттарынан бөлек орналасқан жеке файлдарда сақталады, бірақ олар броузер арқылы Web-парақтардың ішінде бейнеленеді. Суреттерді бейнелеу ережелерін былай беруге болады.

Суреттерді құжаттардың ішіне орналастыру үшін <IMG ...> жеке, яғни жабылмайтын жалқы тәг қолданылады.

Бұл тәгте міндетті түрде SRC= «...» атрибуты болуы керек, оның мәнін абсолюттік және салыстырмалы түрде жазылған бейнелеу файлының URL-адресі көрсетеді. Құжатты экранға шығарған кезде ол міндетті түрде құрамындағы суреттермен бейнеленеді және ол <IMG ...> тәгі тұрған орыннан көрінеді. Мысалы, мына жол:

<IMG SRC= "fish.jpg">

экранға **fish.jpg** файлындағы балық суретін шығарады.
Ескерту: Қазіргі кездегі броузерлер тек қана **gif, jpg, png** типтердегі суреттік файлдарды ғана пайдаланады.

Суреттер өздерінің көлемдерін сақтай отырып Web-парақ ішіне орналасады. Егер суретті ықшамдап бейнелеу кезінде оның масштабын өзгерту қажеттігі туса, суреттің қажетті көлемін WIDTH= (ені) және HEIGHT=(биіктігі) атрибуттары көмегімен беруге болады. Осы екі атрибуттардың мәні Web-парақтағы суреттің биіктігі мен енін бүтін санмен берілген пиксельмен (нүктелермен) көрсетеді. Төмендегі жол:

<IMG SRC="fish.jpg"WIDTH=500 HEIGIT=250>

суретті 500x250 нүктелерден тұратын төртбұрышқа орналастырады.

Сурет айналасындағы жақтау (рамка – border) сызығының қалыңдығын да параметр ретінде көрсетуге болады:

Border = Пиксельдер саны

Жақтау тек әдемілік үшін ғана емес, суретті <A …> тәгінің ішінде гиперсілтеме ретінде пайдаланғанда, ол бір рет шертілген соң жақтау сызығының түсі өзгеріп, оның қолданылғаны белгілі болып тұрады. Енді гиперсілтеме ретінде суретті пайдаланудан бір мысал келтірейік:

```
<HTML> <BODY>
<A HREF = "dog.htm"> <IMG SRC = "dog.gif" WIDTH = 87
HEIGHT=100 BORDER=2> </A>
</BODY> </HTML>
```

Бұл жолдар гиперсілтеме ретінде шағын ит суретін (ол сурет алдын ала болуы тиіс) шығарады, оны шерту иттер туралы мәліметке ауысуды орындайды.

Құжаттарда суреттерді бейнелегенде, оның орындала бермейтін бірсыпыра ерекшеліктері бар екенін айта кеткен жөн.

Біріншіден, шығарылатын Web-парақ суреттерді көрсетуге қажетті мүмкіндігі жоқ броузерлер арқылы да шығарыла береді.

Екіншіден, көбінесе тұтынушылар құжатты желі арқылы тез қабылдау үшін суреттерді бейнелейтін командаңы алып тастайды. Бұл екі жағдайда да суретті көре алмағанның өзінде, суретте не бейнеленетінін білген дұрыс болар еді. Бұл мақсат үшін суретті сипаттайтын қосымша мәтіндерді қолданады.

Қосымша мәтін суреттің мүмкіндігінше толық мәтін түрінде сипатталады. Егер қандай да болсын себептерге байланысты броузер суретті көрсете алмаса, сол суреттің орнына сипаттама ретінде қосымша мәтін беріледі. Ол мәтін <IMG ...> тәгі арқылы ALT= “...” арнайы атрибутының мәнімен беріледі. Мысалы:

```
<html>
  <BODY> <IMG SRC="nan.gif"  alt="нан суреті"> </BODY>
</html>
```

Бұл жолдар сурет шықпаған жағдайда, сол сурет орнына тышқан курсорын алып барғанда, *нан суреті* деген сөзді бейнелейді.

<IMG ...> тәгінің атрибуттарының толық кестесін келтіре кетейік.

5 – кесте. Сурет атрибуттары

Атрибут	Жазылу форматы	Атқаратын қызметі
ALT	<IMG SRC="DOG.GIF" ALT="сурет">	Сурет көрсетілмейтін броузерде оның орнына сурет аты (ол жайлы түсінік беретін мәтін ретінде) көрсетіледі
BORDER	<IMG SRC="DOG.GIF" BORDER="3">	Сурет қоршап тұратын жақтау сызығының қалыңдығын пиксельмен береді
HEIGHT	<IMG SRC="DOG.GIF" HEIGHT=111>	Суреттің биіктігін пиксельмен немесе терезе биіктігінің пайызымен береді
WIDTH	<IMG SRC="DOG.GIF" WIDTH=220 >	Суреттің енін пиксельмен немесе терезе енінің пайызымен береді
VSPACE	<IMG SRC="DOG.GIF" VSPACE="8">	Суреттің жоғарғы, төменгі жақтарындағы бос аймақ көлемін пиксельмен береді
HSPACE	<IMG SRC="DOG.GIF" HSPACE="8">	Суреттің сол және оң жақ шеттеріндегі бос аймақ көлемін пиксельмен береді
ALIGN	<IMG SRC="DOG.GIF" ALIGN=TOP>	Мәтінге байланысты суреттің орналасуын көрсетеді, төмендегі мәндердің бірін қабылдайды: TOP – жоғары (мәтін суреттің жоғарғы жағында), MIDDLE - ортада, BOTTON – төмен, LEFT – сол жақта (сурет жолдың сол жақ шетінде), RIGHT – оң жақта (сурет жолдың оң жақ шетінде)

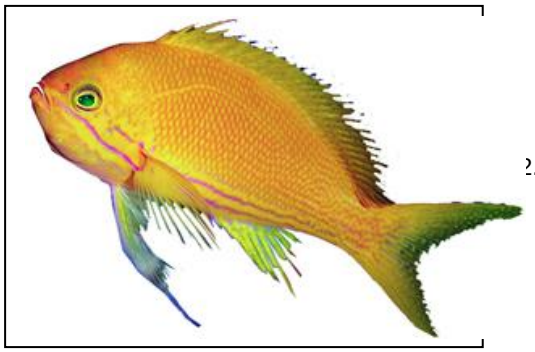
Суреттер маңына оған түсінік беретін мәтін жазылғанда, оны суретке байланысты жоғары немесе төмен жылжытатын мүмкіндіктер және суретті беттің сол немесе оң жақ шетіне жылжыту **Align** атрибуты арқылы беріледі. Оны туралау атрибуттары деп атайды, олар:

Align=”bottom” – мәтін **суреттің** төменгі жағында;

Align=”left” – сурет жолдың сол жағында;
Align=”middle” – мәтін суреттің ортасында;
Align=”right” – сурет жолдың оң жағында;
Align=”top” – мәтін суреттің жоғарғы жағында орналасады.
Мысалы:

<IMG SRC="fish.jpg" border=2 align="middle">

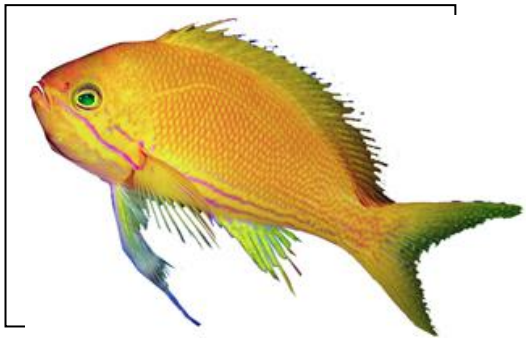
Мұнда мәтін сурет ортасында орналасады.



2.1- сурет. Балық бейнесі

<IMG SRC="fish.jpg" border=2 align="top">

2.2- сурет. Бұл да балық суреті



Мұнда мәтін суреттің жоғарғы жағында орналасады.

<IMG SRC="fish.jpg" border=2 align="right">
Мұнда сурет жолдың оң жағында орналасады.

2.4.4 Құжаттарда сырғымалы жолдарды ұйымдастыру

<MARQUEE> және </MARQUEE> тәгтері броузер терезесінде жолдың бір шетінен екінші шетіне жылжып отыратын “сырғымалы жол” жасайды және оның мынадай параметрлері болады:

<MARQUEE [ALIGN=”align”] [BEHAVIOR=”behavior”]
[BGCOLOR= ”#rrggbb”] [DIRECTION=”direction”]
[HEIGHT=”integer”] [HSPACE=”integer”]
[LOOP=”integer”] [SCROLLAMOUNT=”integer”]
[SCROLLDELLAY=”integer”] [VSPACE=”integer”]
[WIDTH=”integer”]> Кез келген мәтін
</MARQUEE>

Осылардың бірсыпырасының мағыналары мен жазылу түрі төмендегідей болып келеді.

ALIGN – "сырғымалы" мәтінді жолдың жоғарғы шетіне, ортасына немесе төменгі шетіне туралап орналастыру тәсілін береді және де ол мына мәндердің біреуін қабылдайды: TOP, MIDDLE, BOTTOM.

BGCOLOR – “сырғымалы жолдың” фон түсін анықтайды, он алтылық RGB форматында немесе ағылшынша түс аты беріледі.

DIRECTION – жолдық жылжу, яғни сырғу бағытын анықтайды, оның мүмкін мәндері: LEFT, RIGHT және оның мәні көрсетілмеген жағдайда, келісім бойынша LEFT мәні автоматты түрде іске қосылады.

HEIGHT – “сырғымалы жолдың” биіктігін пиксель (нүктелер) арқылы анықтайтын бүтін сан, оны пайызбен де (%) көрсетуге болады.

LOOP - “сырғымалы жолдардың” қайталану санын анықтайтын бүтін сан, *INFINITE* (шексіздік) мәнін қабылдауы да мүмкін.

SCROLLAMOUNT – жылжудың бір қадамында мәтіннің қанша пиксельге жылжитынын анықтайтын бүтін сан.

SCROLLDELAY – екі сырғудың арасындағы интервалды миллисекундпен көрсететін бүтін сан.

WIDTH – экрандағы “сырғымалы жолдың” енін пиксель арқылы анықтайтын бүтін сан, оны пайызбен де (%) көрсетуге болады.

Енді келесі мысалды Блокнотта теріп**,** 3-2 мысал.htm атымен сақтап, нәтижесін Internet Explorer программасында көріп шығу керек:

```
<HTML> <HEAD> <TITLE> 3-2 мысал </TITLE> </HEAD>
  <BODY  text=red>
    <CENTER>
     <H2> Сырғымалы жолдар </H2>  <HR>
     <H3> <MARQUEE  BGCOLOR= ”yellow”   DIRECTION = ”RIGHT”
        SCROLLAMOUNT = ”10”   SCROLLDELAY=”200”  WIDTH=”90%”>
         Бұл бірінші сырғымалы жол
            </MARQUEE>
       <P> <MARQUEE   BGCOLOR= ”Green”   DIRECTION = ”LEFT”
       HEIGHT=30  SCROLLAMOUNT=”10”   SCROLLDELAY=”100”
            WIDTH=”90%”>
          Бұл екінші сырғымалы жол  </MARQUEE>  </H3> <HR>
    </CENTER>
 </BODY>
</HTML>
```

Тапсырмалар:

1. Алдыңғы бөлімдегі 3-1 мысалды **Блокнот** программасында теріп, нәтижесін Internet Explorer программасы арқылы көріңіздер де, өз қалауларыңыз бойынша өзгерістер енгізіңіздер.
2. Сырғымалы жолдарға келтірілген 3-2 мысалды **Блокнот** программасында теріп, оны **3-2 мысал.htm** деп сақтап алып, нәтижесін Internet Explorer программасы арқылы көріңіздер де, келесі өзгерістер енгізіңіздер:

а) сырғымалы жолдардың биіктігін ауыстыру;

ә) бірінші сырғымалы жолдағы мәтін түрін қызылмен, екінші жолдағы мәтін түрін – көкпен бояу;

б) азат жолтан кейінгі көлденең сызықтарды әр түрлі түстерге бояу, оның қалыңдығы мен ұзындығын өз қалауларыңызша өзгерту керек.

3. Алдын ала дайындалған **lab3.htm** файлын (төменде келтірілген) ашып келесі өзгертулерді енгізіңіз:

а) Тақырып және бөлім аттарын ортаға қойыңыз;

ә) Бірінші азат жолтың түсін қызыл, екінші азат жолтың түсін – көк, үшінші азат жолтың түсін жасыл етіп өзгертіңіз;

б) Әр азат жолтан кейін көлденең сызықтар сызып, олардың түстерін және ендерін қалауларыңызша өзгертіңіз;

в) Мәтіннің фонын сұр түске өзгертіңіз.

4. **lab3.htm** файлын ашып келесі өзгертулерді енгізіңіз:

а) pr4.htm, pr5.htm, pr6.htm, pr7.htm құжаттарына және Orantang.jpg, Popugay.jpg, Monky.jpg суреттеріне гиперсілтеме жасаңыз.

ЕСКЕРТУ: *lab3.htm, pr4.htm, pr5.htm, pr6.htm, pr7.htm* құжаттары және *Orantang.jpg, Popugay.jpg, Monky.jpg* суреттері алдын-ала дайындалып

компьютердің ішіндегі *Мои документы* немесе басқа бір жұмыс бумасының ішінде орналасуы тиіс. Олардың орнына 2-1 мысал, 2-2 мысал, тағы сол сияқты дайындалған файлдарды қолдануға болады.

Алдын ала дайындалған web-құжат құрылымымен танысуға арналған **lab3.htm** лабораториялық жұмыс мәтіні:

```
<HTML> <HEAD> <TITLE> Құжаттың негізгі бөлімдері </TITLE>
        </HEAD>
<BODY> <H1> Басты тақырып</H1> <H2>Ішкі тақырыпша</H2>
  <HR> <A HREF="#3.1">Үшінші азат жолқа көшу</A> <br>
  <H3> 1.1 Бірінші азат жол </H3> Осы жолдар бірнеше жолға
          жазылғанымен
          құжатта олар
          бірге жазылады
  <HR> <H3> 2.1 Екінші азат жол </H3>
 <P> Азат жол үшін жабу тәгі қажет емес. <HR>  <A  Name="3.1">  </A>
 <H3>3.1 Үшінші азат жол</H3> <P>Келесі жолға көшу тәгінен гөрі азат
 жол басы тәгі алдында бос орын үлкенірек болады. <HR>
 Көлденең сызықтан кейінгі мәтін  <BR> екіге бөлінген.
 <P> <H3  align=center><B>Бірнеше сызықтар салайық. </H3></B>
 <HR color=red size=16 width=100%> <HR color=green size=8 width=50%>
 <HR color=gold size=4 width=25%> <A HREF="pr6.htm"> pr6 мәтініне </A>
 ауысып, кейін оралайық. <A HREF="DOG.JPG"> Ит суретіне </A> ауысып,
 кейін оралайық. <HR> <A HREF="tu2.gif"> Ту суретіне </A> ауысып, кейін
 оралайық. <HR> <A HREF="cat01.jpg"> Мысық суретіне </A> ауысып,
 кейін оралайық. <HR> <A HREF="dog03.jpg"> Ит суретіне </A> ауысып,
 кейін оралайық. <HR> <A HREF="fish.jpg"> Балық суретіне </A> ауысып,
 кейін оралайық. <P> <IMG SRC="fish.jpg" WIDTH=500 HEIGIT=250
 border=2 align="bottom">  <P> <H3 ALIGN=center> Енді өздеріңіз осындай
 суреттерге сілтеме жасаудан мысал келтіріңіздер </H3>
</BODY>
</HTML>
```

Бақылау сұрақтары

1. HTML құжатын форматтау дегеніміз не?
2. Физикалық стиль жасауға қажет тәгтер тізімін келтіріңіздер. Олардың атрибуттары бола ма?
3. Логикалық стильдер дегеніміз не? Кең тараған логикалық стиль тәгтерін көрсетіңіздер.
4. Жоғарғы және төменгі индекстер қандай тәгтер арқылы жазылады?
5. Гиперсілтеме түсін қалай өзгертеміз, қарастырылған, курсор көрсетіп тұрған сілтемелер түрін басқаша ету қалай орындалады?
6. Мәтін фоны ретінде суретті қалай пайдалануға болады?
7. Түрлі түстердің RGB-кодтары дегеніміз не? Сарғыш жасыл түстің RGB-кодын қалай есептеп шығаруға болады?
8. Гиперсілтеме қалай құрылады?

9. <A HREF= ... > атрибуты мәні ретінде қандай сөз тіркестерін алуға болады?
10. Ішкі сілтемелер дегеніміз не?
11. Орнатылған анкерге сілтеме жасау үшін не істеу керек?
12. HTML-құжатқа сурет орналастыру үшін қандай тәг қолданылады? Оның атрибуттарын атап шығыңыздар.
13. Суреттердің көлемін қалай өзгертуге болады?
14. Пиксель ұғымын қалай түсінесіздер?
15. Сырғымалы жолдар қалай ұйымдастырылады?
16. Суреттік бейнелерді сырғытуға бола ма?
17. Сырғымалы жолдар жасау тәгінің қандай атрибуттары бар, олардың мәндері неге тең бола алады?
18. Сырғымалы жолды екі бағытқа да кезектестіре отырып жылжытуға бола ма?
19. Сырғыту жылдамдығын қалай өзгертеміз?
20. Жолды жоғары, төмен сырғытуға бола ма?

2.5. Құжат ішіне тізімдер орналастыру

HTML тілі мәтін азат жолтарының сыртқы пішімін толық анықтауға мүмкіндік береді. Азат жол элементтерін үлкейтуге, кішірейтуге, қалыңдатып немесе қисайтып, астын сызып жазуға, тізім түрінде белгілеуге болады. Енді тізімдер жасайтын мүмкіндіктердің бірсыпырасын қарастырайық. Жалпы тізім элементтері арнайы тәгтермен қоршалып, экранға шығарылғанда олардың сол жақтарында тізім белгілері орналасады.

2.5.1 Нөмірленбеген тізімдер

<UL> және </UL> (unordered list – реттелмеген тізім) тәгтері арасында орналасқан мәтіндер нөмірленбейді, бірақ белгіленіп жазылған тізімдер ретінде қарастырылады. Мұнда тізімнің әрбір жаңа элементін <LI> (list – тізім) белгісінен бастап жазу қажет. Мысалы, экранда жасыл түсті әріптермен терілген мынадай тізім жасау үшін:

- Сәуле;
- Мақсат;
- Данияр;
- Ержан

төмендегідей түрде HTML тілі мәтінін Блокнотта теріп, Internet Explorer-де 4-1 мысалды көріп шығу керек:

```
<HTML> <HEAD> <TITLE> 4-1 мысал </TITLE> </HEAD>
 <BODY  text= green>
    <H2 ALIGN = CENTER> Белгіленген тізім жолдары </H2>  <HR>
     <UL>
       <LI> Сәуле;
       <LI> Мақсат;
       <LI> Данияр;
       <LI> Ержан
     </UL>
     <HR>
```

```
 </BODY>
</HTML>
```

<LI> белгісі үшін жабылу тәгінің қажет емес екенін байқаған боларсыздар.

<UL> тәгінің атрибуты type = disc | circle | square

олар белгі маркерінің сыртқы пішінін өзгертіп, сәйкесінше дөңгелек | шеңбер | квадрат бейнесінде көрсете алады.

2.5.2 <OL> тәгі арқылы жазылатын нөмірленген тізімдер

Нөмірленген тізімдер алдыңғы белгіленген тізім тәрізді шығарылады, олар тек <OL> және </OL> (ordered list – реттелген тізім) тәгтерімен қоршалады да, нәтижесінде тізім нөмірі ретінде бүтін сандар жазылады. Жоғарыда келтірілген мысалды аздап түрлендіріп, төмендегі 4-2 мысалда оларды нөмірлеп жазып шығайық:

```
<HTML> <HEAD> <TITLE> 4-2 мысал </TITLE> </HEAD>
 <BODY  text= green>
  <H2 ALIGN = CENTER> Нөмірленген тізім жолдары </H2>  <HR>
  <OL>
     <LI> Сәуле;
     <LI> Мақсат;
     <LI> Данияр;
     <LI> Ержан
  </OL>
 <HR>
</BODY>
</HTML>
```

Нөмірленген тізім жолдары

1. Сәуле;
2. Мақсат;
3. Данияр;
4. Ержан

Бұл HTML тәгтері жұмысы нәтижесінде мынадай тізім шығарылады:

<UL> тәгінің екі атрибуты бар, олар:

start = нөмір

type = 1 | A | a | I | i

бұлардың алғашқысы нөмірлеуді біріншіден емес, көрсетілген кез келген нөмірден бастайды, ал екіншісі тізімді нөмірлеу жүйесін төмендегі кестедегідей түрде анықтайды.

6 - кесте. **Type** атрибуты мәніне қарай тізімдерді нөмірлеу түрлері

type мәні	стилі	тізім нөмірленуі
1	Араб цифрлары (келісім бойынша)	1, 2, 3, 4, ...
A	Латын алфавитінің бас әріптері	A, B, C, D, …
a	Латын алфавитінің кіші әріптері	a, b, c, d, …
I	Рим цифрлары (жоғарғы регистр)	I, II, III, IV, …
i	Рим цифрлары (төменгі регистр)	i, ii, iii, iv,…

2.5.3 <DL> тәгі арқылы жазылатын анықтау тізімдері

Анықтау тізімдері <DL> (definition list) тәгі арқылы жазылады да, алдыңғылардан аздап өзгешелеу болып келеді. Мұнда <LI> тәгі орнына <DT> (definition term – анықталатын термин) және <DD> (definition definition – анықтаманың анықтамасы) белгілері жазылады. Осыларды 4-3 мысал ретінде қарастырып, программа мәтінінен үзінді келтірейік:

```
<HTML> <HEAD> <TITLE> 4-3 мысал </TITLE> </HEAD>
 <BODY  text= green> <H3 ALIGN = CENTER>
   Анықтау  тізімдері </H3>
 <DL>
   <DT>HTML
       <DD>
```

HTML (HyperText Markup Language) термині – “гипермәтіндік белгілеу тілі” деген сөз. Оның алғашқы нұсқасын Европадағы элементар бөліктер физикасы лабораториясының қызметкері Тим Бернерс-Ли жасап шығарған болатын.

```
   <DT>HTML құжаты
      <DD> Тіркелу аты  *.htm (Unix жүйелеріндегі файлдарда
 *.html) болып келген мәтіндік файл.
</DL>  </BODY>  </HTML>
```

Осы программа жұмысы нәтижесінде экранға мынадай мәлімет шығады:

HTML
HTML (HyperText Markup Language) термині – “гипермәтіндік белгілеу тілі” деген сөз. Оның алғашқы нұсқасын Европадағы элементар бөліктер физикасы лабораториясының қызметкері Тим Бернерс-Ли жасап шығарған болатын.

HTML құжаты
Тіркелу аты *.htm (Unix жүйелеріндегі файлдарда *.html) болып келген мәтіндік файл.

<LI> тәгі сияқты <DT> және <DD> тәгтерінің жабылмай, жалғыз жазылатынына назар аударыңыздар.

Егер терминдер анықтамасы қысқаша түсіндірілетін болса, онда <DL COMPACT> белгісін пайдалануға болады. Енді төмендегідей 4-4 мысалды программа үзіндісі түрінде қарастырайық:

```
<HTML> <HEAD> <TITLE> 4-4 мысал </TITLE> </HEAD>
    <BODY  text= green> <HR> <H3 ALIGN=CENTER>
      Қысқаша анықтамалар </H3>
      <DL COMPACT>
        <DT>АБАЙ
<DD> XIX ғасырдағы қазақ ақыны, әрі ағартушысы
        <DT> ҚҰРМАНҒАЗЫ
           <DD> XIX ғасырдағы қазақ сазгері, атақты күйші
```

```
        <DT> МАХАМБЕТ
          <DD>XVIII ғасырдағы қазақ ақыны, әрі батыры – жыр
              семсері
        </DL>
      </BODY>
    </HTML>
```

Бұл программа экранға мынадай мәтін шығарады:

АБАЙ	XIX ғасырдағы қазақ ақыны, әрі ағартушысы
ҚҰРМАНҒАЗЫ	XIX ғасырдағы қазақ сазгері, атақты күйші
МАХАМБЕТ	XVIII ғасырдағы қазақ ақыны, әрі батыры – жыр семсері

2.5.4 Қабатталған тізімдер

Кез келген тізімдегі элемент ішіне басқа да тізім түрлері енгізіліп, қабаттасқан тізімдер құрастырылуы да мүмкін. Бірақ мұндай қабатталған тізімдерді жиі қолдансақ, ол мәтінді экранға көп сатылы түрде шығарып, оның ұзындығын арттырып жібереді. Сондықтан оларды тек керек кезінде ғана қолданған дұрыс деп саналады.

Қабатталып орналасқан тізімдерді мәтін мазмұндары мен әр түрлі жоспарлар дайындауда қолданған тиімді болып табылады. Осындай тізімдер ұйымдастыруды қысқаша мынадай 4-5 мысалмен көрсетейік:

```
<HTML> <HEAD> <TITLE> 4-5 мысал </TITLE> </HEAD>
  <BODY>
    <H1 ALIGN=center> HTML бірнеше тізім түрлерін ұйымдастыра алады
</H1>
<DL>
  <DT> Нөмірленбеген тізімдер
<DD> Нөмірленбеген тізім элементтері сол жақтан арнайы таңбамен
белгіленіп, мәтін аздап оңға таман жылжып орналасады:
<UL>
   <LI> 1 элемент
   <LI> 2 элемент
   <LI> 3 элемент
</UL>
  <DT> Нөмірленген тізім жолдары
<DD> Нөмірленген тізім элементтері сол жақтан нөмірлер арқылы
       белгіленіп орналасады:
<OL>
 <LI> 1 элемент
 <LI> 2 элемент
 <LI> 3 элемент
</OL>
  <DT> Анықтау тізімдері
     <DD> Мұндай тізімдер алдыңғы екеуінен күрделірек, бірақ оқуға ыңғайлы
болады.
```

<P> Тізімдерді бірінің ішіне бірін жазып қабаттастыруға болады, бірақ мүлде көп деңгейлер жасап, бұл тәсілмен тым әуестеніп кету қажет емес екені есте болсын.

<P > Тізімдегі бір элемент ішінде бірнеше азат жолтар тұруы мүмкін.

Ондай азат жолтар сол жақ шеттен бірдей қашықтыққа ығысып орналасады.
</P>

</DL>

</BODY>

</HTML>

Осы программа нәтижесі:

HTML бірнеше тізім түрлерін ұйымдастыра алады

Нөмірленбеген тізімдер

Нөмірленбеген тізім элементтері сол жақтан арнайы таңбамен белгіленіп, мәтін аздап оңға таман жылжып орналасады:

- 1 элемент
- 2 элемент
- 3 элемент

Нөмірленген тізім жолдары

Нөмірленген тізім элементтері сол жақтан нөмірлер арқылы белгіленіп орналасады:

1. 1 элемент
2. 2 элемент
3. 3 элемент

Анықтау тізімдері

Мұндай тізімдер алдыңғы екеуінен күрделірек, бірақ оқуға ыңғайлы болады. Тізімдерді бірінің ішіне бірін жазып қабаттастыруға болады, бірақ мүлде көп деңгейлер жасап, бұл тәсілмен тым әуестеніп кету қажет емес екені есте болсын. Тізімдегі бір элемент ішінде бірнеше азат жолтар тұруы мүмкін. Ондай азат жолтар сол жақ шеттен бірдей қашықтыққа ығысып орналасады.

Тапсырмалар:

1. Осы бөлімдегі 4-2 мысалды теріп алып, оның нәтижесін Internet Explorer-да қарап әр қатарға өз топтарындағы студенттер аттарын енгізу керек.

3.Осы бөлімдегі 4-3 мысалды алып, нәтижесін қарап, әр қатарға өз топтарындағы студенттер фамилиясын тізіп жазып, онан кейін қалыңдық-тары және түстері әр түрлі болып келген, ұзындықтары да өзгермелі көлденең сызықтар жүргізу қажет.

4. Жоғарыда келтірілген 4-3 мысалды алып, нәтижесін көріп, қосымша жолдар енгізіп, инженер, дәрігер, бухгалтер мамандықтарына түсінік беріңіздер.

5.Қысқаша анықтама беретін 4-5 мысалды алып, нәтижесін көріп, әр жолға топ оқушыларының аттарын жазып және оларға “оқу озаты”, “дарынды бала”, “көрікті бойжеткен” сияқты қысқаша анықтамалар беріп, қосымша мінездемелер жазу қажет.

Бақылау сұрақтары

1. Нөмірленбеген тізім тәгтері дегеніміз не?
2. Нөмірленетін тізім тәгтері қандай? Олардың атрибуттары ше?
3. Нөмірді 5 санынан бастау үшін не істеу керек?

4. Нөмірлерді a), b), c), … деген түрде жасауға бола ма? Болса, қалай?
5. Анықтау тізімдері қалай құрылады? Олар қысқаша құрыла ала ма?
6. Қабатталған тізімдер қалай құрылады?

2.6 КЕСТЕЛЕР ТҰРҒЫЗУ

Кесте тұрғызу <TABLE> және </TABLE> тәгтері көмегімен орындалады, оның әрбір жолын анықтау – <TR> және </TR> тәгтері арқылы, ал сол жолдардағы бағаналар – <TD> және </TD> немесе <TH> және </TH> тәгтері арқылы анықталады. <TD> және <TH> тәгтерінің жұмысы ұқсас, бірақ <TH> тәгтерімен қоршалған мәтін қарайтылған бағана тақырыптары болып табылады да, <TD> тәгтерімен одан кейінгі қарапайым бағаналар жазылады.

Кесте тақырыбы <CAPTION> және </CAPTION> тәгтерімен қоршалып жазылады. Жалпы кестені толық анықтау ережесі төмендегі үлгімен беріледі:

```
<TABLE ALIGN="center" BGCOLOR="#rrggbb" BORDER="integer"
        BORDERCOLOR="#rrggbb"  WIDTH="integer"> .....
</TABLE>
```

Бірақ кесте тұрғызу кезінде олардың кейбірі қолданылмауы да мүмкін. Енді осы кесте тәгі атрибуттарының атқаратын жұмысына тоқталайық:

ALIGN атрибуты кестенің шет жақтарға туралануын анықтайды (көрсетілмесе, келісім бойынша сол жақ шетке). ALIGN мәні – қос тырнақша ішіндегі сөз – мына сөздердің біріне сәйкес келуі тиіс: LEFT (сол жақ шетке), CENTER (ортаға), RIGHT (оң жақ шетке).

BGCOLOR кесте торының ішкі фон түсін тағайындайды (он алтылық RGB форматындағы сан немесе ағылшын тіліндегі белгіленген түс атауы).

BORDER – бүтін сан, кесте жақтаулары сызығының пиксельмен берілген қалыңдығы. Егер BORDER берілмесе, жақтау сызықтары көрсетілмейді.

BORDERCOLOR жақтау сызықтарының түсін тағайындайды (он алтылық RGB форматындағы сан немесе ағылшын тіліндегі белгіленген түс атауы), BORDER атрибутымен бірге қолданылады.

WIDTH – кесте енін анықтайтын бүтін сан, оның мәні пиксельмен немесе пайызбен (%) беріледі.

Кесте тақырыбы <CAPTION> тәгімен төмендегі ережеге сәйкес беріледі:

```
<CAPTION  ALIGN="top"> .....  </CAPTION>
```

Мұндағы атрибуттардың атқаратын қызметі мынадай болады.

ALIGN атрибуты кесте тақырыбын шет жақтарға туралау кезінде оның мәні LEFT, CENTER (көрсетілмесе, келісім бойынша осы мән қабылданады), RIGHT сөздерінің біріне сәйкес келуі тиіс. Ал егер ол тақырыпты вертикаль бағытта кестенің жоғарғы және төменгі жақтарына орналастыруы қажет болса, онда BOTTOM – жоғарыда (келісім бойынша осы мән қабылданады), TOP – төменде сөздерінің бірін мән ретінде қабылдай алады.

Кесте жолы **<TR>** және **</TR>** тәгтерімен қоршалып тұрады, бұлардың алғашқысының мынадай бірсыпыра атрибуттары болуы мүмкін:

```
<TR  ALIGN="center"  BGCOLOR="#rrggbb"
BORDERCOLOR="#rrggbb">  Кесте жолы… </TR>
```

Енді **<TR>** тәгінің осы атрибуттарына тоқталайық.

ALIGN – жол шеттерін туралау. Оның мүмкін мәндері LEFT (келісім бойынша), CENTER, RIGHT.

BGCOLOR жолдың ішкі фон түсін анықтайды (он алтылық RGB форматындағы сан немесе ағылшын тіліндегі белгіленген түс атауы).

BORDERCOLOR – жол жақтауларының түсі (он алтылық RGB форматындағы сан немесе ағылшын тіліндегі белгіленген түс атауы). Бұл атрибут <TABLE> тәгінің BORDER атрибутының мәні нөлге тең болмағанда ғана қолданылады.

ROWSPAN – бірнеше жолдарды біріктіріп, бір-ақ жол етіп жазу кезінде берілетін біріктірілетін жолдар саны.

Кесте жолындағы бағаналар (ұялар) **<TD>... </TD>** және **<TH>... </TH>** тәгтерімен төмендегі ережеге сәйкес анықталады:

```
<TD немесе TH  ALIGN="right"  BACKGROUND="url"
   [BGCOLOR="#rrggbb"  BORDERCOLOR="#rrggbb"] >
      Бағана… </TD немесе  /TH>
```

<TD> және <TH> тәгтерінің атрибуттары қызметтері:

ALIGN – мәтінді горизонталь бағытта туралау атрибуты. Оның мүмкін мәндері: LEFT, CENTER (келісім бойынша), RIGHT.

BGCOLOR – фон түсін анықтау атрибуты (он алтылық RGB форматындағы сан немесе ағылшын тіліндегі белгіленген түс атауы).

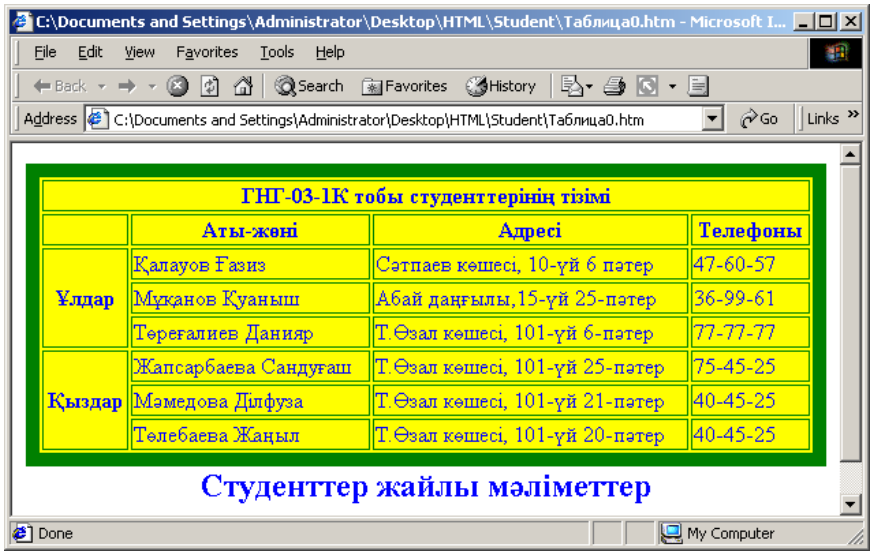
BORDERCOLOR – ұя жақтауларының түсі атрибуты (он алтылық RGB форматындағы сан немесе ағылшын тіліндегі түс атауы). Бұл атрибут <TABLE> тәгінің BORDER атрибутының мәні нөлге тең болмағанда ғана қолданылады.

COLSPAN – бағаналар тақырыбына арналған біріктірілген жол (ұя) ішінде орналасатын бағаналар саны.

Бір студенттер тобы жайлы кесте құрып, мысал келтірейік (2.3-сурет).

```
<html> <body text=blue> <table border=10 bordercolor=green
                  width=100%  align=center  bgcolor=”yellow”>
  <caption  align=bottom> <H2>Студенттер жайлы мәліметтер </H2>
  </caption>
  <tr> <th colspan=4> ГНГ-03-1К тобы студенттерінің тізімі </th> </tr>
  <tr> <th>  </th> <th>Аты-жөні </th> <th> Адресі </th>
     <th> Телефоны </th> </tr>
  <tr> <th rowspan=3>Ұлдар </th> <td>  Қалауов Ғазиз</td> <td>
     Сәтпаев көшесі, 10-үй 6 пәтер </td> <td> 47-60-57 </td> </tr>
   <tr><td>Мұқанов Қуаныш</td><td>Абай даңғылы,15-үй 25-пәтер</td>
     <td> 36-99-61 </td> </tr>
   <tr><td>Төреғалиев Данияр</td><td>Т.Өзал көшесі, 101-үй 6-пәтер</td>
     <td> 77-77-77 </td> </tr>
    <tr> <th rowspan=3>Қыздар</th>
     <td>Жапсарбаева Сандуғаш</td><td> Т.Өзал көшесі, 101-үй 25-пәтер
     </td> <td> 75-45-25 </td> </tr>
    <tr> <td> Мәмедова Ділфуза</td><td>  Т.Өзал көшесі, 101-үй 21-пәтер
```

```
    </td> <td> 40-45-25 </td> </tr>
   <tr> <td> Төлебаева Жаңыл</td><td>  Т.Өзал көшесі, 101-үй 20-пәтер
    </td> <td> 40-45-25 </td> </tr>
 </table>
</body>
</html>
```



2.3-сурет. Кестеге енгізілген мәліметтердің экрандағы көрінісі

Тапсырмалар:

1. Төмендегі мысалды теріп, нәтижесін қарап шығыңыздар да, оған бірсыпыра өзгерістер енгізіңіздер:

```
<TABLE   ALIGN=CENTER  BORDER=3    WIDTH="80%"
            BGCOLOR=YELLOW   BORDERCOLOR=BLUE>
     <CAPTION> <H2> Кесте тақырыбы </H2></CAPTION>
     <TR><TD> кестенің бірінші  торы </TD>
           <TD> кестенің екінші  торы </TD> </TR>
      <TR><TD> кестенің бірінші  торы </TD>
            <TD> кестенің екінші  торы </TD></TR>
</TABLE>
```

а) 1-жолдың фонын ақшыл (<TR BGCOLOR=SILVER>), сұр түске түрлендіріңіздер.

б) жақтаулар (бордюрді) қалыңдығын, ұялардағы мәтінді өзгертіңіздер, т.с.с.

2) Келесі мысалды теріп, нәтижесін қарап шығыңыздар да, оған бірсыпыра өзгерістер енгізіңіздер:

```
<TABLE  ALIGN="RIGHT"   BORDER="3"  BORDERCOLOR="Blue"
          WIDTH="80%">
    <CAPTION ALIGN="RIGHT">
      Тақырып кестенің оң жақ жоғарғы бұрышына шығады
     </CAPTION>
     <TR>  <TH COLSPAN="3"> Бағана тақырыбы </TH> </TR>
     <TR ALIGN="RIGHT" BGCOLOR="yellow">
```

```
    <TH> 1 бағана </TH> <TH> 2 бағана </TH> <TH> 3 бағана </TH>
  </TR>
  <TR> <TD> 1 бағана мәліметтері </TD>
  <TD> 2 бағана мәліметтері </TD>
  <TD> 3 бағана мәліметтері </TD>
  </TR>
</TABLE>
```

а) Кестенің екі жолына да тағы бір бағана қосыңыздар;

ә) Тағы екі жол қосыңыздар;

б) Тордағы мәтін сөздерін, кесте жолының фонын, жеке ұялардың да ішкі фонын өзгертіңіздер;

в) Тақырып түсін өзгертіңіздер, т.с.с.

г) Кесте алдында, тақырыптан соң және кесте соңында түрлі түсті көлденең сызықтар жүргізіңіздер.

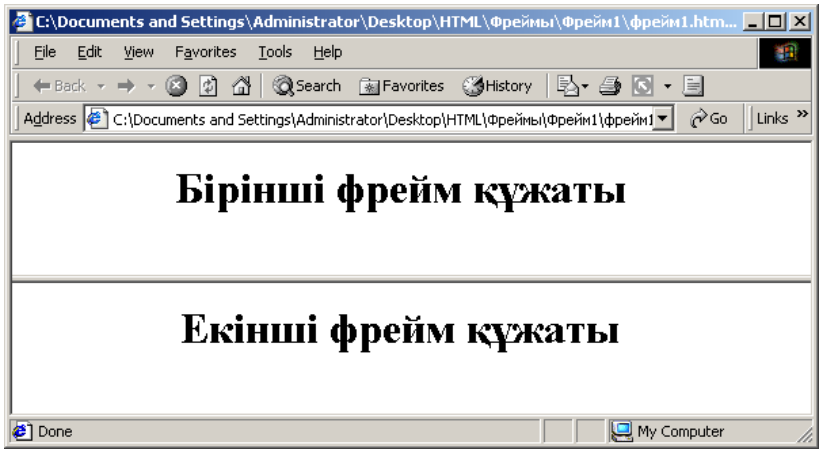
2.7 Бір WEB-парақта фреймдер арқылы бірнеше құжаттарды орналастыру

HTML тілі броузер терезелерін бірнеше бөліктерге бөлу мүмкіндігін береді және олардың әрқайсысында жеке web-құжаттар бейнеленеді. Осындай бөліктер *фрейм* немесе *кадр* деп аталады. Мұнда әрбір фрейм экрандағы жеке тіктөртбұрышты аймақты алып тұрады. Әр фрейм ішінде бір-бірінен тәуелсіз құжат орналастыра аламыз. Мысалы, экранды екі фреймге бөліп, сол жағына Netscape фирмасының, ал оң жағына Microsoft фирмасының web-сайттарының алғашқы парақтарын шығарып салыстыра отырып көруге болады.

2.7.1 Фреймдер құру тәсілдері

Фреймдер құру үшін <FRAMESET> және <FRAME> тәгтері қолданылады да, мұнда әдеттегідей <BODY> тәгі пайдаланылмайды. <FRAMESET> тәгі броузер терезесіндегі фреймдердің көлемдері мен олардың орналасу тәртібін сипаттайды, <FRAME> тәгінде фреймдердің әрқайсысына шақырылатын құжаттар аттары көрсетіледі. Екі фреймнен тұратын экран құрайтын программа бөлігінен мысал келтірейік:

```
<html>
   <frameset rows="50%,50%">
        <frame src="1 файл.htm">
        <frame src="2 файл.htm">
   </frameset>
</html>
```



2.4- сурет. Фреймдердің экрандағы көрінісі

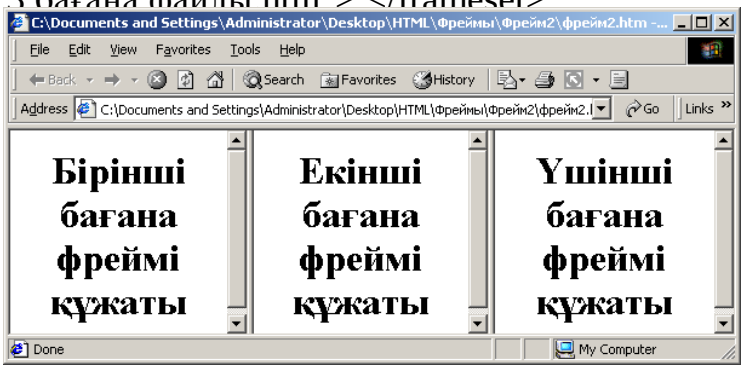
Мұндағы екі фрейм жолдар (rows) бойынша көлденеңнен бірінің астына бірі орналасады, олар экранды 50 %-дан бөліп алады. Үстіңгі фреймде "1 файл.htm" құжаты ашылады да, төменгісінде – "2 файл.htm" орналасады (2.4 сурет).

<FRAMESET> тәгінің ROWS=... (қатар) атрибуты терезені горизонталь – көлденең бағыт бойынша бөледі де, COLS= ... (бағана) атрибуты тіке – вертикаль бағытта фреймдерге бөледі. Егер осы атрибуттың екеуі де берілсе, терезе тіке және көлденең төртбұрыштардан тұратын торларға бөлінеді. Атрибуттардың мәндері терезе бөліктерінің көлемдерін (биіктігін немесе енін) анықтайды. Әр бағанаға (жолға) арналған параметрлер пиксель өлшем бірлігі бойынша немесе пайыздармен (%) үтірлер арқылы бөлініп беріледі. Соңғы параметр ретінде қалған көлемді автоматты түрде толық алып тұратын (*) жұлдызша белгісін де пайдалануға болады. Осындай фрейм үшін барлық қалған бос кеңістіктер бөлініп беріледі.

<FRAME> тәгінде фреймдерге шақырылатын құжаттарды анықтайтын SRC=... атрибуты болуы керек. Қалған атрибуттар фреймдер арасындағы бөлу сызықтары параметрлерін және оның басқа кейбір қасиеттерін реттеу мүмкіндігін береді.

Экранды үш тік орналасқан тең көлемді фреймдерге бөлу тәгтері төмендегідей болып жазылады:

```
<frameset cols="33%,33%,* ">
  <frame src="1 бағана файлы.htm">
  <frame src="2 бағана файлы.htm">
  <frame src="3 бағана файлы.htm"> </frameset>
```

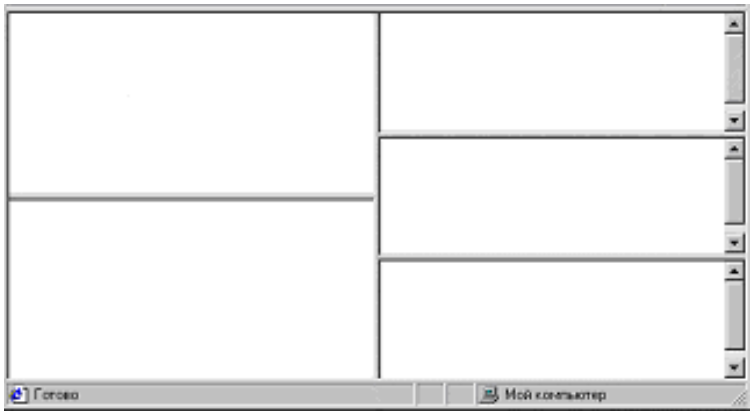


2.5- сурет. Фреймдердің баған түріндегі көрінісі

Осы тәгтер жұмысы нәтижесі 2.5- суретте көрсетілген. Енді бес тордан – алғашқы бағанасы биіктіктері бірдей екі жолдан, ал екінші бағанасы – бірдей үш жолдан тұратын фреймдер тұрғызатын программа бөлігін келтірейік.

```
<frameset cols="50%,50%">
    <frameset rows="50%,* "> <frame src="1 top.htm"> <frame
src="2 top.htm">
    </frameset>
    <frameset rows="33%,33%,* "> <frame src="3 top.htm"> <frame src="4
top.htm">
      <frame src="5 top.htm">
    </frameset>
</frameset>
```

Бұл программаның нәтижесі төменде көрсетілген (2.6-сурет).



2.6- сурет. Бірнеше фреймдердің орналасуы

Әрбір фрейм көлемі, яғни ұзындығы немесе ені абсолютті түрде пиксельмен, экран мөлшерінің пайызымен немесе салыстырмалы i* (i – бүтін сан) түрінде беріледі. Егер екі-үш мән көрсетіліп, олар әр түрлі бірліктерде берілсе, броузер алдымен абсолюттік пиксель түріндегі мәнді тағайындайды, сонан соң қалғандарын пропорционал түрде алады. * белгісі 1* деген белгімен парапар. Экранды фреймдерге бөлу солдан оңға және жоғарыдан төмен қарай жүргізіледі.

Экранды үш тік орналасқан фреймдерге бөлейік: екіншісінің ені 250 пиксель (мөлшері берілген сурет еніне тең), біріншісі – қалған көлемнің 25%-ын, ал үшіншісі – 75%-ын алатындай ету үшін мынадай атрибуттар мәнін жазу керек

```
<FRAMESET cols = ”1*,250,3*”>
  .....
</FRAMESET>
```

Келесі мысалда 2x3 мөлшерлі төртбұрыштар жиынынан тұратын фреймдер жасалады.

```
<frameset rows=”30%,70%” cols=”33%,34%,33%”>
  .....
</frameset>
```

Келесі мысалда терезе биіктігі 1000 пиксельге тең деп қабылданған. Бірінші фреймге жалпы биіктіктің 30%-ы (300 пиксель) берілген. Екіншісіне

– 400 пиксель, ал қалған екеуіне 300 пиксель тағайындалған, төртінші фрейм биіктігі “2*” түрінде, үшіншісінің биіктігі – * деп анықталған, сондықтан үшінші фреймнің – 100, төртінші фреймнің – 200 пиксель биіктігі болуы тиіс.

```
<FRAMESET rows=”30%, 400, *, 2*” >
  .....
</FRAMESET>
```

Егер бір фреймге абсолюттік мән нақты берілген кезде қалғандарынан белгілі бір көлем артылып қалып, немесе жетпей қалып жататын жағдай туындаса, онда броузер сол қалған фрейм көлемдерін пропорционал күйде көбейтеді немесе азайтады. FRAMESET тәгі бір-бірімен қабаттасып жазыла береді. Келесі мысалда сыртқы FRAMESET терезені үш тең бағаналарға бөледі. Ал ішкі FRAMESET екінші бағананы биіктіктері әр түрлі екі жолға бөліп тұр.

```
<FRAMESET cols=”33%,33%,34%” >
  …бірінші бағана фреймі ішкі мәліметі …
<FRAMESET rows=”40%,50%” >
  … екінші бағанадағы бірінші жол фреймі ішкі мәліметі …
  … екінші бағанадағы екінші жол фреймі ішкі мәліметі..…
</FRAMESET>
  …үшінші бағана фреймі ішкі мәліметі …
</FRAMESET>
```

SRC атрибуты фрейм ішіне орналасатын мәтінді анықтайды. Ол фреймнің ішкі мәтіні сыртқы фрейм жазылған құжатта болмауы керек. Мысалы:

```
<FRAMESET cols=”33%, 33%, 33%”>
  <FRAMESET rows=”*, 200”>
    <FRAME src=”1_фрейм_мазмұны.htm”>
    <FRAME src=”2_фрейм_мазмұны.gif”>
  </FRAMESET>
    <FRAME> src=”3_фрейм_мазмұны.htm”>
    <FRAME> src=”4_фрейм_мазмұны.htm”>
</FRAMESET>
```

Бұл тәгтер жұмысының нәтижесінде төмендегі суреттегідей төртбұрышты фреймдер жиыны экранға шығады (2.7-сурет).

2.7- сурет. Фреймдердің тіктөртбұрыш түріндегі көрінісі

2.7.2 <FRAMESET> … </FRAMESET> тәгтері

Фрейм құру <FRAMESET> тәгінен басталады. Мұнда <BODY>тәгі қолданылмайды.

Бұл тәгтер фреймдер орналасатын төртбұрышты контейнерлерді, яғни мәлімет орналасатын екі жағында да ашылу және жабылу шекаралық жақтаулары бар тәгтерді анықтайды. Олардың пайдалануға болатын атрибуттарымен бірге жазылу синтаксисі:

<FRAMESET [COLS="col1,col2,col3,..."] [ROWS="row1,row2,row3, .,,"] [FRAMEBORDER="1 немесе 0"] [FRAMESPACING="integer"] >

.....

</FRAMESET>

<FRAMESET> тәгінің *келесі атрибуттары* бар:

COLS – фрейм бағаналарының пайызбен (%), пиксельмен немесе салыстырмалы бірлікте (*) берілген ендері. Мысалы, COLS= "25%, 100,*" үш бағанадан тұратын фреймдерді анықтайды, біріншісінің ені – терезенің 25 пайызы, екіншісінің ені – 100 пиксель, ал үшіншісінің ені – броузер терезесінің қалған бөлігі.

Жақтау сызықтарын анықтайтын үш атрибут бар, олар – BORDER, FRAMEBORDER және BORDERCOLOR.

FRAMEBORDER – фрейм жақтауларын бөліп тұратын сызық сызады. Ол 1-ге тең болса – дөңес сызық, 0 болса – жай сызық сызылады (мәні берілмесе, келісім бойынша, 1 болып есептеледі).

FRAMESPACING – фреймдер арасындағы бос кеңістік мөлшерін пиксельмен беретін бүтін сан.

BORDER – барлық фреймдер жақтаулары сызығының қалыңдығын пиксельмен анықтайды. Мысалы: <FRAMESET BORDER="10"> . Егер оның мәні 0 болса, жақтау сызығы болмайды. Бұл атрибут көрсетілмесе, келісім бойынша, оның мәні 5 болып қабылданады. Жалпы оның мәні бестен жоғары болғаны дұрыс, кейбір броузерлер 5-тен төменгі мәнді сызып көрсетпейді. Фреймдердің ішкі бөліктері жақтаулары да FRAMEBORDER және FRAMESPACING атрибуттары мәнімен анықталады, яғни фрейм жасау кезінде бұларды сыртқы тәгте бір рет көрсету жеткілікті.

BORDERCOLOR атрибутының мәні стандартты түс атауларына немесе түстің он алтылық жүйедегі RGB-мәндеріне сәйкес келеді.Мысалы: <FRAMESET BORDERCOLOR="red" ROWS="*,*">

ROWS – фрейм жолдарының пайызбен (%), пиксельмен немесе салыстырмалы бірлікте (*) берілген биіктігін анықтайды. Мысалы, ROWS="25 %,100,*" үш көлденең орналасқан фреймдерді анықтайды, біріншісінің биіктігі – терезе ұзындығының 25 пайызы, екіншісінің биіктігі – 100 пиксель, ал үшіншісінікі – терезенің қалған бөлігі.

Төмендегі мысалда үш фреймнен тұратын терезе көрсетілген: биіктігі 100 пиксель, ені терезе енімен бірдей тақырып жазу фреймі және оның төменгі жағында екі фрейм орналасқан. Төменгі сол жақтағы фрейм терезе енінің 20

пайызын алып тұрады, оң жақтағы фрейм – терезенің қалған бөлігінде орналасқан.

```
<FRAMESET ROWS="100,*">
<FRAME NAME="Тақырып" NORESIZE SCROLLING="NO"
        SRC="heading.htm">
<FRAMESET COLS="20%,*">
<FRAME NAME="мазмұны" SRC="contents.htm">
<FRAME MAME="нәтижесі" SRC="results.htm">
</FRAMESET>
</FRAMESET>
```

2.7.3 <FRAME> тәгі

<FRAME> тәгі жеке бір фрейм қасиеттерін анықтайды. Ол FRAMESET контейнерлік – екі жақты тәгтер ішінде орналасатын, жабылмайтын жалқы тәг болып табылады. Мысалы:

```
<FRAMESET ROWS="*, 2*">
 <FRAME> ...
 <FRAME> ...
</FRAMESET>
```

<FRAME> тәгтерінің саны <FRAMESET> тәгінде ашылған кадрлар санына тең болуы тиіс. Жоғарыдағы мысалда екі фрейм ашылған, бірақ әлі тәгтер атрибуттары толтырылмаған.

<FRAME> тәгінің фрейм қасиеттерін анықтайтын алты атрибуты бар: NAME, MARGINWIDTH, MARGINHEIGHT, SRC, NORESIZE және SCROLLING. Олардың қолданылу ережелері төмендегідей:

```
<FRAME SRC="URL" NAME="терезе_аты"
 SCROLLING=yes|no|auto MARGINWIDTH="мәні"
 MARGINHEIGHT="мәні" NORESIZE>
```

Барлық атрибуттарды пайдалану міндетті емес. Көбінесе тек SRC атрибуты ғана пайдаланылады.

SRC= ... осы фреймде орналасатын гипермәтіндік құжат атын, яғни фреймде көрсетілетін файлдың URL-адресін анықтайды. Үйрену кезінде бұл файл әдетте FRAMESET тәгі жазылған бумада орналасқан кәдімгі HTML құжаты болады. Мысалы,

```
<FRAME SRC="м1.htm">
```

м1.htm құжатында HTML, HEAD, BODY, т.с.с. тәгтер болуы тиіс немесе ол суреттік файл да бола алады (орман.gif). Егер SRC атрибуты көрсетілмесе, фрейм бос тұрады.

MARGINWIDTH=n атрибуты фреймнің сол және оң жақтарындағы пиксель санымен берілген бос кеңістік мөлшерін анықтайды.

MARGINHEIGHT=n, фреймнің жоғары және төмен жақтарындағы пиксельмен берілген бос кеңістік мөлшерін анықтайды. Бұл екеуінің мәні әрқашанда абсолютті түрде пиксельмен беріледі. Мысалы:

```
<FRAME MARGINHEIGHT="5" MARGINWIDTH="6">
```

кадрдың жоғары және төмен жағында 5 пиксельден, ал сол және оң жақтарында 6 пиксельден орын қалдырады. Әдетте бос кеңістік 1-ден 6-ға дейінгі мәндерді қабылдайды.

SCROLLING=... фрейм бойынша жылжу жолақтары. Егер жылжу жолағы болуы керек болса – SCROLLING=YES , ал олар болмайтын болса – SCROLLING=NO, ал егер жолақтардың болуын/болмауын броузер анықтайтын жағдайда – SCROLLING= AUTO деп жазылады. Егер бұл атрибут көрсетілмесе де, келісім бойынша AUTO мәні қабылданады. Мысалы:

<FRAME SCROLLING=yes>

NORESIZE атрибуты фрейм көлемі тұрақты болып, оның өзгермейтінін көрсетеді. Мысалы: <FRAME NORESIZE>

BORDERCOLOR атрибуты <FRAMESET> және <FRAME> тәгтерінің екеуінде де қолданыла береді. Оның мәні стандартты түс атауларына немесе түстің он алтылық жүйедегі RGB-мәндеріне сәйкес келеді. Мысалы:

```
<FRAMESET BORDERCOLOR="red"  ROWS="*,*">
 <FRAME  SRC="M1.htm"  BORDERCOLOR="#FF00FF">
 <FRAME SRC="M2.htm">
</FRAMESET>
```

Мұнда <FRAMESET> тәгінің BORDERCOLOR атрибуты жақтауларға қызыл түс тағайындайды, ал <FRAME> тәгінің осындай атрибуты күлгін түс (#FF00FF) береді. Осылардың нәтижесінде екінші кадр түсі толық анықталмай қалған, сондықтан оның бірінші кадрмен жанасатын шекарасы күлгін түсті, ал басқа жағы – қызыл түсті болып қалады.

Егер екі кадрдың әрқайсысында өз BORDERCOLOR атрибуттары болса, онда ол екеуі де әсерін жояды. Олардың жақтаулары сыртқы FRAMESET тәгінің BORDERCOLOR атрибуты бекіткен түспен боялады.

2.7.4 Фреймдер арқылы сілтемелер ұйымдастыру

Енді фреймдерді пайдаланып, олардың бірінен біріне ауысу мүмкіндіктерін қарастырайық. Фреймдерді шығарудағы ең негізгі ой – олардың ішкі мазмұндарын автоматты түрде ауыстырып отыру болатын.

Фреймге гиперсілтеме жасау үшін оған ат қойылуы керек. Ат қою <FRAME> тәгінің NAME атрибуты арқылы жүзеге асырылады.

NAME= frame1 – фреймге, яғни кадрға frame1 деген ат қойылады.

TARGET= frame1 – frame1 деген аты бар фреймге гиперсілтеме жасалатындығын көрсетеді. Ол <A> тәгінің атрибуты ретінде беріледі.

Мысалы, <FRAME NAME="frame1"> жолы алдыңғы ашылған фреймге "frame1" деген ат қояды, бұдан кейін осы фреймге гипермәтіндік сілтеме жасау былай орындалады:

```
<A HREF="first.html"  TARGET="frame1">бірінші фрейм </A>
```

мұнда *бірінші фрейм* сөзін шерткенде, осы frame1 кадрында алғашқы орналасқан файл, мысалы M1.htm файлы, first.htm файлымен алмастырылады.

Егер TARGET атрибуты жазылмай қалса, онда first.html файлы алғашқы сілтеме жазылған кадрға шығады. Сол себепті TARGET атрибуты HREF атрибутында көрсетілген файлды белгілі бір кадрға орналастыру үшін керек. Осы бір кадрда орналасқан файлды екінші бір кадр арқылы басқарып ауыстыру тәсілі сайттардың бірінен біріне көшу ісін жүзеге асыра алады. Төменгі кестеде TARGET атрибутының фреймдер жұмысын басқаратын кейбір тұрақты мәндері көрсетілген. Олар төменгі сызықшадан басталады, сондықтан басқа сөздермен шатастыра алмаймыз.

2.1-кесте TARGET атрибутының фреймдер жұмысын басқаратын кейбір тұрақты мәндері

Мәні	**Атқаратын қызметі**
_blank	Көрсетілген файлды ат қойылмаған жаңа терезеге жүктейді
_self	Көрсетілген файлды сілтеме жасалған фреймге жүктейді
_parent	Көрсетілген файлды фреймдерді ашатын түпкі кадрға жүктейді; егер ондай кадр анықталмаған болса, онда оның әсері алдыңғы _self қызметімен бірдей болады
_top	Көрсетілген файлды кадрлар құрылымы біріктірілген толық терезеге жүктейді

Әрбір кадрге ат берілуі тиіс, әйтпесе оған сілтеме жасауға болмайды. Сондықтан ішкі мазмұны өзгертілетін етіп жоспарланған кадрлардың барлығына нақты ат қойылуы тиіс. Кадрлар аттары әріптен немесе цифрдан басталуы керек, өйткені жоғарыдағы кестеде көрсетілген мағынасы алдын ала берілген кадр аттары төменгі сызықшадан басталады. Мысалы, бір STUDENT деген бума ашып, оның ішіне төменде көрсетілген тәгтері бар frames.html файлын жазыңыздар:

```
<HTML>
  <FRAMESET ROWS="*,*">
   <FRAMESET COLS="*,*">
<FRAME SRC="frame1.html" NAME="fr1"> <FRAME SRC="frame2.html" NAME="fr2">
</FRAMESET>
 <FRAME SRC="frame3.html" NAME="fr3">
</FRAMESET> </HTML>
```

Мұнан кейін frame1.html, frame2.html, frame3.html атты файлдар ашып, оларға төмендегідей мәліметтер жазыңыздар:

frame1.html:

```
<HTML><BODY BGCOLOR=white>
<H1>1 фрейм</H1>
<A HREF=frame2.html  TARGET="fr3"> 2 кадрға сілтеме</A>
</BODY></HTML>
```

frame2.html:

```
<HTML><BODY BGCOLOR=red TEXT=yellow>
<H1>2 фрейм</H1>
<A HREF=frame3.html TARGET=_top>3 фрейм толық терезеде</A>
</BODY></HTML>
```

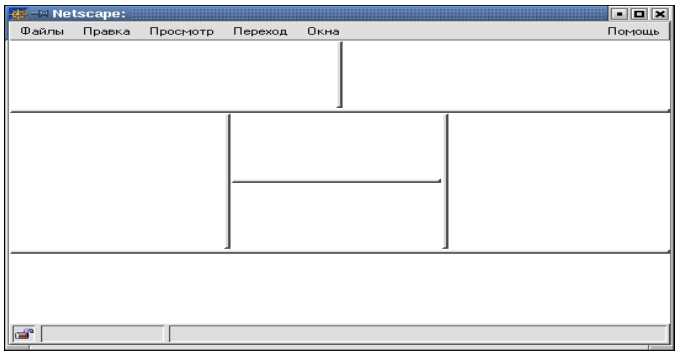
frame3.html:

```
<HTML><BODY BGCOLOR=purple>
<H1>3 фрейм</H1>
<A HREF=frame1.html  TARGET=_self>1 фрейм осы терезеге</A>
</BODY></HTML>
```

Енді frames.html файлын екі шертіп, оның нәтижесін қарап шығыңыздар.

Тапсырмалар:

1.Frame1.html файлын ашып, FRAMESET тәгі арқылы броузер терезесін 2.8- суреттегідей кадрларға бөліңіздер.



2.8- сурет

2.Index.html деген файл ашып, терезені төрт фреймге бөліңіздер (екі жол және екі бағана), оларға алдын ала ішіне HTML тәгтерімен толықтырылған мәлімет жазылған мынадай файлдардың мәліметін орналастырыңыздар: list.html, table.html, image.html және first.html. Одан кейін:

а) list.html файлы орналасқан фреймде міндетті түрде жылжу жолақтары болатын болсын;

ә) table.html файлы орналасқан фреймнің мөлшерін өзгермейтін (NORESIZE) етіңіздер;

б) first.html файлын былай етіп өзгерту керек: image.html файлына жасалған сілтемені таңдағанда, ол first.html файлы тұрған фреймге жүктелетін болсын;

в) image.html файлындағы first.html файлына арналған сурет түріндегі сілтемені шерткенде, first.html файлы жаңа терезеде ашылатын болсын.

Бақылау сұрақтары

1. Фреймдер дегеніміз не? Көлденең орналасқан фреймдер қалай жасалады? Тік орналасқан фреймдер ше?
2. <FRAMESET> тәгінің атрибуттары қандай?
3. Бір фрейм ішінде орналасатын web-парақ аты қалай көрсетіледі?
4. <FRAME> тәгінің қандай атрибуттары болады? Олардың атқаратын қызметі қандай?
5. Аты көрсетілген фреймге қалай сілтеме жасауға болады?

6. Фреймдер шекаралық сызықтарының түсін қалай өзгертеді?
7. TARGET атрибуты қандай қызмет атқарады? Ол қандай тәгпен бірге орналасады?

2.8 HTML тілінің мультимедиялық мүмкіндіктері

Броузерлердің (Netscape Navigator, Internet Explorer) мультимедиялық файлдарды іске қосатын қосымша программалық модульдері бар. Олар LiveAudio (WAV, AU, AIFF және MIDI форматтарындағы дыбыстық файлдар), Live3D (VRML), LiveVideo (AVI бейнелік файлдары) және QuickTime (дыбыс пен мәтін қосылатын MIDI түріндегі MOV форматындағы файлдар) тәрізді технологияларды сүйемелдей алады.

2.8.1 <EMBED> тәгін пайдалану

Жоғарыдағы айтылған мультимедиалық мүмкіндіктердің бәрін web-құжатқа <EMBED> тәгі арқылы енгізуге болады. Оны жабу міндетті емес. Бұл тәгті пайдалану кезінде дыбыс ойнайтын құралдың басқару элементтерін сурет тәрізді көрсету мүмкіндігі бар. <EMBED> тәгінің бірсыпыра параметрлерін келтіре кетейік:

2.2- кесте. <EMBED> тәгінің параметрлері

Атрибут	Атқаратын қызметі
1	2
SRC	Іске қосылатын объектінің толық маршруты көрсетілген файлының аты. Бұл параметр **міндетті түрде** жазылуы тиіс.
PALETTE	Іске қосылатын объектінің палитралары типтері (түстер палитрасы), мәндері: foreground, background
TYPE	Объект файлының типі
HIDDEN = false | true	Объектінің экранда көрсетілу типі, мәндері: false – объект көрсетілмейді (келісім бойынша), true – объект көрсетіледі
CONTROLS= console | smallconsole	console – терезеге дыбысты басқару тақтасы шығарылады, оның ойнату, тоқтату, дыбысты реттеу тәрізді батырмалары көрсетіледі; smallconsole – терезеге шағын басқару тақтасы шығарылады, оның да ойнату, тоқтату, дыбысты реттеу тәрізді батырмалары көрсетіледі;
HEIGHT=n	дыбысты басқару тақтасының немесе бейнефильм кадрының пиксельмен берілген биіктігі;
WIDTH=n	дыбысты басқару тақтасының немесе бейнефильм кадрының пиксельмен берілген ені;
LOOP= true | false | n	true – аудио не бейнефайл тұрақты ойналады; false – аудио не бейнефайл бір рет ойналады; n – аудио не бейнефайл n рет ойналады.

Бұл тәгтің басқа атрибуттары <IMG> тәгінің атрибуттарына ұқсас (жалпы HTML құжаты ортасына объект енгізу тәсілі бірдей десе болады),

сондықтан NAME, ALT, BORDER, ALIGN, HSPACE, VSPACE сияқты атрибуттарды қарастырмаймыз.

Мультимедиялық мүмкіндіктерді сүйемелдей алмайтын басқа броузерлер үшін <NOEMBED> элементі көрсетіледі де, оның ашылу және жабылу тәгтері арасында түсініктеме мәтін жазылады.

Мысалдар:

<EMBED SRC=nature.wav>

тәгі nature.wav музыкалық файлын фон ретінде тыңдауға мүмкіндік береді. Құжатқа PDF (Adobe Portable Document Format) форматындағы JS.PDF файлын енгізу үшін мынадай жолды жазу керек:

<EMBED SRC=js.pdf width=500 border=0 align=left>

2.8.2 Фондық дыбыс

Web-құжатты жүктеген соң, оны көру барысында өзіңіз ұнататын әнді тыңдап отырғыңыз келсе, <BGSOUND> тәгін төмендегідей түрде жазып пайдалануға болады:

<BGSOUND SRC=”ән аты” LOOP= “қайталану саны”>

Мысалы, “Ақбақай.mp3” әуендік файлын 3 рет қайталап тыңдау үшін, мынадай жол енгізілуі тиіс:

<BGSOUND SRC=”Ақбақай.mp3” LOOP=3>

Егер бір әуен шексіз түрде қайталанып айтылғанын қаласаңыз, соңғы атрибут LOOP=INFINITE деп жазылуы қажет. Әрине, әуен орнына жасалған web-сайт жайлы түсініктеме мәтін немесе оны қалай түрлендіруге болатыны жайлы кеңестерді де тыңдауға болатыны түсінікті шығар.

Әуенді web-құжат ашылғанда, бірден тыңдамай, өзіңіз қалаған сәтте ғана іске қосу үшін, дыбыстық файлға суреттік файлдарды шығару кезіндегі сияқты төмендегідей түрде гиперсілтеме жасау керек.

<A HREF=”Ақбақай.mp3”> өлең </A>

Мұнда экранға “өлең” атты гиперсілтеме шығып тұрады, соны тышқанмен шерткен кезде ғана ән ойнала бастайды.

2.8.3 Бейнекадрды пайдалану

Web-парақ ішіне бейнекадрды қою жай сурет орналастыру тәрізді орындалады. Мұны жүзеге асыру тәгінен мысал келтірейік:

<IMG DYNSRC=”FILE1.AVI” START=OPENFILE LOOP=2>

Мұнда DYNSRC атрибуты бейнефайлдың атын береді, ал START параметрі бейнероликті іске қосу тәсілін тағайындайды. Егер оның мәні OPENFILE болса – бейнефайл бірден жүктеледі, ал MOUSEOVER болса – онда тышқан курсорын сол роликтің бірінші кадрын іске қосатын суретте шерткенде барып жүктеледі. LOOP параметрі бейнефайлды қайталау санын береді.

Бейнефайлдарды гиперсілтеме арқылы да көрсетуге болады:

<A HREF=”VIDEO.MOV”> Бейнефильм көру (1,3MB)</A>

Гиперсілтемені шерткенде, броузер файл типін оқып, бейнефильмді көрсететін плейерге қосымша құрылғы іске қосылады. Қандай плейер қолданылатыны тұтынушы броузерінің құрамы мен құрылымына

(конфигурациясына) байланысты болады. Көбінесе бейнефильм басқару тақтасы бар жеке терезеде көрсетіледі.

<EMBED> тәгі арқылы бейнефайлды көру кезінде web-парақтағы плейерге қатысты құрылғы ізделеді де, іске қосылады. Мысалы:
<EMBED SRC=”cool.mov” AUTOPLAY=false WIDTH=160 HEIGHT=120>
</EMBED>
Мұндағы AUTOPLAY=false атрибуты бейнефильм PLAY батырмасын шерткен соң, іске қосылатынын көрсетеді. Ол бірден іске қосылуы үшін AUTOPLAY=true болуы тиіс. WIDTH=160 HEIGHT=120 атрибуттары бейнекадрдың ені мен биіктігін береді.

Бақылау сұрақтары:
1. <EMBED> тәгінің атқаратын қызметі және атрибуттары.
2. Бір әуенді бірнеше рет қайталау үшін не істеу керек? Шексіз қайталау үшін ше?
3. Фондық дыбыс қалай іске қосылады?
4. Web-парақ ішіне бейнекадрды қою қалай орындалады? Оны бірнеше рет қайталап көру үшін қандай атрибут қолданылады?

2.9. HTML құжаттарындағы формалар

HTML формалары web-құжат жариялаушылар мен оқырмандар арасында ақпарат алмасуға мүмкіндік жасайды. Бұған дейін біз web-құжаттарды тек экранға шығару әдістерін талқылап келген болсақ, енді, керісінше оларға мәлімет енгізу әрекеттерін қарастырамыз. Формалар арқылы тұтынушыдан жалпы мәтін түрінде ақпарат сұрап алуға, "иә/жоқ" деген жауаптың бірін таңдауға немесе бірнеше жолдың біріне тоқтауға болады.

Формаларды әр түрлі мақсаттарда қолдану мүмкіндігі бар. Мысал ретінде, сайтқа кірген оқырмандардың ол туралы өз пікірлерін формаға енгізуі туралы айтуға болады, жалпы HTML формаларын пайдалану аймағы алуан түрлі болып келеді.

2.9.1. <FORM> тәгі

Әрбір форма осы тәгпен басталады. Оның қолданылатын форманы өңдеу программасын (скрипт) және мәліметтерді жөнелту әдісін көрсететін екі атрибутын анықтап алу қажет.

2.3-кесте. FORM> тәгінің атрибуттары

Атрибуты	Атқаратын қызметі
ACTION	Форма мәліметтерін қабылдап алып, оны өңдейтін URL-ды анықтайды. Егер бұл атрибут анықталмаса, онда мәліметтер форма орналасқан web-парақ адресіне жіберіледі
METHOD	Форманы өңдеу программасына (скрипт) қалайша ақпарат жөнелтілетінін көрсетеді. Әдетте, оның мәні POST болады, мұндайда форма мәліметі URL-дан бөлек жеке жөнелтіледі. Ал оның мәні GET болса, онда мәлімет URL-мен бірге жіберіледі

Мысал:
<FORM METHOD="post" ACTION="/cgi-bin/comment_script">

…
</FORM>

Бұл мысалда броузерге мынадай нұсқау берілген: толтырылған форманы *post* әдісін қолдана отырып жөнелтіп, оны web-құжат орналасқан сервердің cgi-bin каталогында орналасқан *comment_script* скрипті көмегімен өңдеу керек екендігі көрсетілген. Бір web-парақта орналасатын формалар саны шектелмейді, бірақ та бір форманың екінші бір форманың ішіне кіріп кетпеуін қадағалап отыру керек.

2.9.2. Форма жасау тәгтері

HTML тілінде формадағы әр түрлі өріс типтерін жасау үшін <TEXT-AREA>, <SELECT> және <INPUT> сияқты үш түрлі тәг қолданылады. Олардың кез келген саны <FORM> ... </FORM> тәрізді қосарланған тәгтер ішінде, яғни контейнерде орналаса алады.

Төмендегі кестеде солардың қысқаша сипаттамалары көрсетілген, кейінірек олар толық қарастырылады.

2.4-кесте. Тәгтердің қысқаша сипаттамалары

Тәг	Атқаратын қызметі
<TEXTAREA>	Кез келген тұтынушы бірнеше жолдан тұратын мәтіндік ақпарат енгізе алатын өрісті анықтайды
<INPUT>	Мәлімет енгізудің кейбір өзге түрлерін: бір жол мәтін енгізу, жалаушаларды көтеріп қою және түсіру (*check boxes*), ажыратып қосқышты таңдау (*radio buttons*) және мәліметтерді жөнелтуге немесе форманы тазартуға арналған батырманы басу сияқты әрекеттерді орындауды қамтамасыз етеді
<SELECT>	Тұтынушыға жылжымалы жолағы бар терезедегі немесе суырылып шығатын меню ішіндегі бір жолды таңдау мүмкіндігін береді

2.9.3 <TEXTAREA> тәгі

Бұл тәг бірнеше жолдан тұратын мәтіндік ақпарат енгізуге арналған өріс құру үшін қажет. <TEXTAREA> …</TEXTAREA> тәгтері арасына, алдын ала келісім бойынша, енгізу өрісіне шығарылатын кез келген мәтін жолдарын орналастыруға болады. Бұл тәгтің атрибуттары:

NAME – өріс атын анықтайды, міндетті түрде болуы тиіс;

ROWS – өріс биіктігі бойынша орналасатын жолдар санын береді;

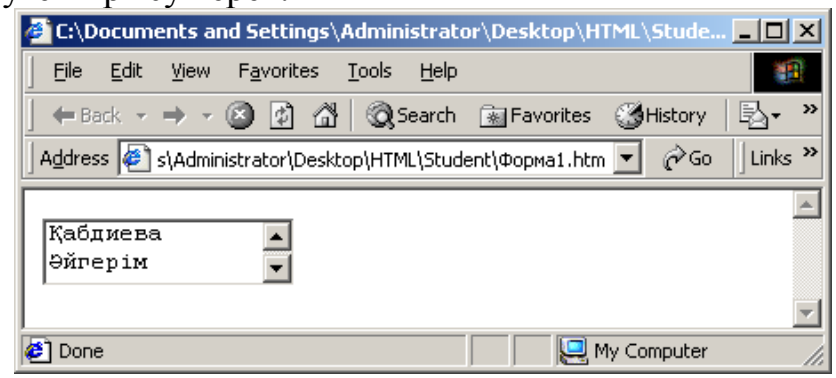
COLS – өріс енін, яғни жол ұзындығын символдар санымен береді.

Мысалы, әр жолына 15 символ мәтін сиятын, екі жолдан тұратын өріс былай жасалады **:**

```
<HTML>
 <BODY>
  <FORM>
     <TEXTAREA Name="Студент" ROWS=2 COLS=15>
        Қабдиева Әйгерім
     </TEXTAREA>
  </FORM>
 </BODY>
</HTML>
```

Мұның нәтижесі 2.9-суретте көрсетілген.

ROWS және COLS атрибуттары арқылы кез келген көлемдегі өріс құруға болады. Бұл атрибуттар міндетті түрде қажет болмағанмен, олардың алдын ала келісім бойынша бекітілген белгілі бір мәні жоқ (кез келген броузер үшін бұл мәндер әр түрлі), сондықтан олардың мәнін әрқашанда ашық көрсетуге тырысу керек.



2.9- сурет. Мәтіндік ақпараттан тұратын өріс

2.9.4 SELECT тәгі

Бұл тәг суырылып шығатын меню немесе айналдыру жолағы бар опциялар (командалар) тізімін құру үшін қолданылады. Опциялар тізімі мен меню пункттері SELECT контейнерінің ішінде орналасады. <TEXTAREA> тәгі сияқты <SELECT> тәгінде де берілген атты анықтайтын NAME атрибуты міндетті түрде болуы тиіс. Опциялар саны SIZE атрибутында көрсетіледі. Төменгі кестеде <SELECT> тәгінің атрибуттарының атқаратын қызметтері жайлы айтылған.

2.5- кесте. <SELECT> тәгі атрибуттарының сипаттамасы

Атрибут	Атқаратын қызметі
NAME	Ақпарат атын анықтайды
SIZE	Таңдау опциялары үшін терезенің биіктігін (вертикаль көлемін) анықтайды. Егер атрибут көрсетілмесе немесе оның мәні 1-ге тең болса, онда қалқымалы опциялар тізімі шығады. Егер оның мәні 1-ден артық болса, онда опциялар айналдыру жолағы бар терезеде көрінеді. Ал егер атрибут мәні тізім элементтерінің нақты санынан артық болса, онда бос жолдар қосылады. Оларды таңдаған кезде, бос жолдар қайтарылады.
MULTIPLE	Бірден бірнеше опциялар таңдауға мүмкіндік береді.

Опциялар тізімі <OPTION> тәгі арқылы <SELECT> контейнеріне кіргізіледі. Бұл тәгтің екі атрибуты болады, олар төмендегі кестеде сипатталған.

2.6- кесте. Атрибуттар сипаттамасы

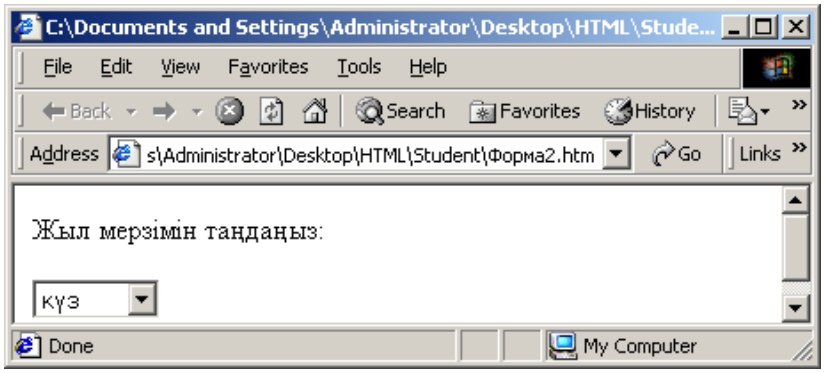
Атрибут	Атқаратын қызметі
VALUE	Тұтынушы опцияны таңдаған жағдайда, өңдеу программасына (скриптіге) қайтарылатын мәнді көрсетеді
SELECTED	Алдын ала келісім бойынша таңдалған опцияны көрсетеді

Мысал:

Төменде ұқсас формалары бар екі HTML құжаттарының мәтіндері көрсетілген. Екінші құжатта тізімнен алдын ала бір элемент таңдалып алынған (SELECTED атрибуты қосылған).

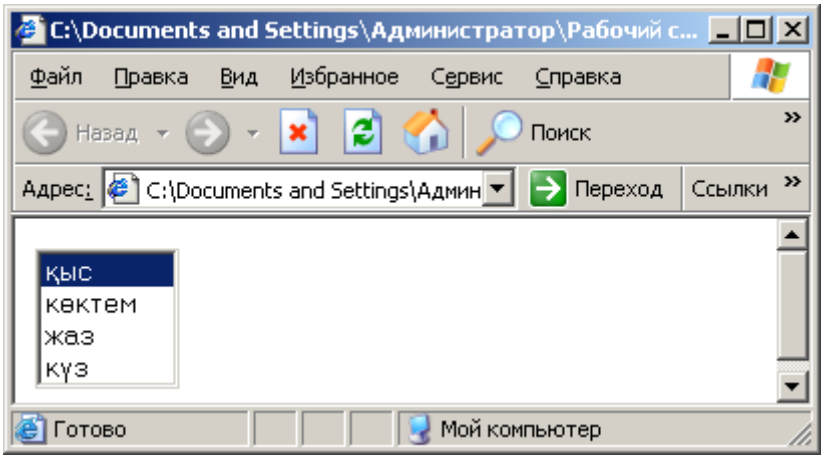
<HTML>

```
  <BODY>
   Жыл мерзімін таңдаңыз:
   <FORM>
     <SELECT NAME=YEAR>
      <OPTION SELECTED VALUE="winter"> қыс
      <OPTION VALUE="spring" > көктем
      <OPTION VALUE="summer"> жаз
      <OPTION VALUE="autumn"> күз
     </SELECT>
   </FORM>
    </BODY>
</HTML>
```



2.10- сурет. Тізімнен бір элементтің таңдап алынуы

```
<HTML>
 <BODY>
    Жыл мезгілін таңдаңыз:
     <FORM>
      <SELECT  MULTIPLE  NAME="YEAR">
         <OPTION  SELECTED  VALUE="WINTER"> қыс
         <OPTION VALUE="spring"> көктем
         <OPTION VALUE="summer "> жаз
         <OPTION VALUE="autumn "> күз
      </SELECT>
     </FORM>
  </BODY>
</HTML>
```



2.11-сурет. Тізімнен бір элементтің таңдап алынуы

2.9.5 <INPUT> тәгі

<INPUT> тәгі <TEXTAREA> мен <SELECT> тәгтеріндей контейнер емес, жалқы тәг болып есептеледі. Ол әр түрлі тәсілдерді пайдаланып, ақпарат жинақтауға арналған. Сол тәсілдерге мәтіндік өрістер, пароль енгізу өрісі, ажыратып-қосқыштар, жалаушалар, мәліметтерді жөнелту (Submit) және формаларды тазарту батырмалары (Reset) жатады.

INPUT тәгінің төмендегі кестеде көрсетілгендей бірсыпыра атрибуттары болады.

2.7-кесте. Атрибуттар сипаттамасы

Атрибут	Атқаратын қызметі
NAMESIZE	Енгізу өрісінің символдар санымен берілген енін көрсетеді
MAXLENGTH	Өріске енгізуге болатын ең үлкен символдар санын (максимальды) анықтайды
VALUE	Мәтіндік өріс үшін алдын ала келісім бойынша шығарылатын мәтінді анықтайды. Жалаушалар мен ажыратып-қосқыштар үшін өңдеу программасына қайтарылатын мәнді көрсетеді. Формаларды жөнелту мен тазарту батырмалары үшін батырма үстіндегі жазуды анықтайды.
CHECKED	Жалауша немесе ажыратып-қосқыштың, алдын ала келісім бойынша, іске қосылған жағдайын орнатады.Басқа тәг түрлерінде <INPUT> қолданылмайды.
TYPE	Енгізу өрісі типін (түрін) орнатады.

2.9.6 Енгізу өрісі түрі, <TYPE> атрибуты

<INPUT> тәгінің <TYPE> атрибуты мәндері кестеде көрсетілген.

2.8- кесте. Атрибуттар сипаттамасы

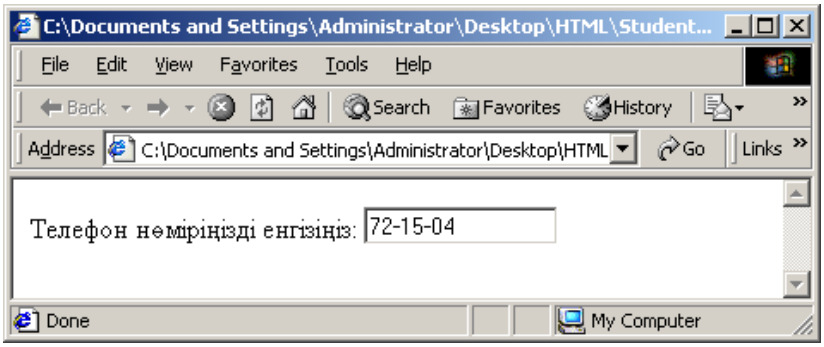
Атрибут	Атқаратын қызметі
1	2
TEXT	Мәтін енгізуге арналған өріс құрылады, ол бірнеше жолдан тұруы мүмкін. Бұл өріс типі үшін NAME (міндетті түрде),

	SIZE, VALUE және MAXLENGTH атрибуттары қолданылады. TYPE атрибуты көрсетілмесе, осы TEXT сөзі алдын ала келісім бойынша орнатылатын мән болып табылады.
PASSWORD	Пароль енгізілетін өріс құрылады, оған терілген таңбалар көрсетілмей, жұлдызшалармен ауыстырылып көрсетіледі. Бұл өріс типінің параметрлерін айқындау үшін SIZE, VALUE және MAXLENGTH атрибуттары қолданылады.
CHECKBOX	Шағын квадрат ішіне жалауша (✔) орналастыруға арналған өрістер құрылады, ол бірнеше нұсқаны таңдауға мүмкіндік береді. Бұл өріс типі үшін NAME, VALUE және CHECKED (жалаушаның алдын ала бекітілген мәнін анықтайды) атрибуттары қолданылады. Мысалы: Сіз қандай ОС-тарды пайдаланасыз? ☐ MS DOS ☐ Windows 98 ☑ Windows 2000 ☑ Windows XP
RADIO	Бірнеше батырмалар жасап, олардың ішінен біреуін таңдауға мүмкіндік береді. NAME, VALUE және CHECKED атрибуттары қолданылады. Мысалы, төмендегідей: Сіздің ең жақсы ұнататын операциялық жүйеңіз қандай? ○ MS DOS ○ Windows 98 ◉ Windows 2000 ○ Windows XP
RESET	Форманы тазартуға арналған батырма жасауға мүмкіндік береді. VALUE атрибуты бұл жерде осы батырмаға ат беру үшін қолданылуы мүмкін (алдын ала келісім бойынша батырмаға RESET деген ат беріледі)
SUBMIT	Енгізілген мәліметтерді серверге жөнелтіп, скрипт-программа арқылы оларды өндеуді жүзеге асыратын батырма жасау үшін қолданылады. VALUE атрибутында бұл батырманың аты көрсетілуі мүмкін (келісім бойынша аты – SUBMIT QUERY).

Мысал:

Келесі формада TYPE атрибутының TEXT мәні қолданылады.

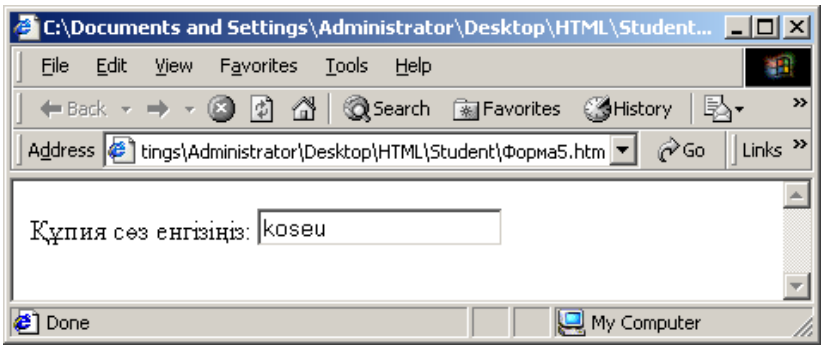
```
<HTML>
  <BODY>
    <FORM>   Телефон нөміріңізді енгізіңіз:
      <INPUT  TYPE="TEXT"  NAME="Phone"
                SIZE="15"    MAXLENGTH="15" >
    </FORM>
  </BODY>
</HTML>
```



2.12- сурет. Енгізу өрісі

TYPE атрибутының PASSWORD мәнінің қолданылуы.

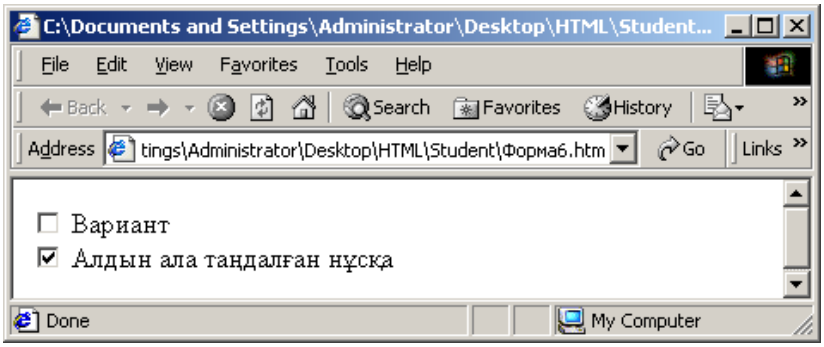
```
<HTML>
  <BODY>
    <FORM>  Құпия сөз енгізіңіз:
      <INPUT  TYPE=”PASSWORD” NAME=”SECRET_WORD”
              SIZE=”30”  MAXLENGTH=”30”>
    </FORM>
  </BODY>
</HTML>
```



2.13-сурет. Енгізу өрісі

TYPE атрибутының CHECKBOX мәнін қарастырайық.

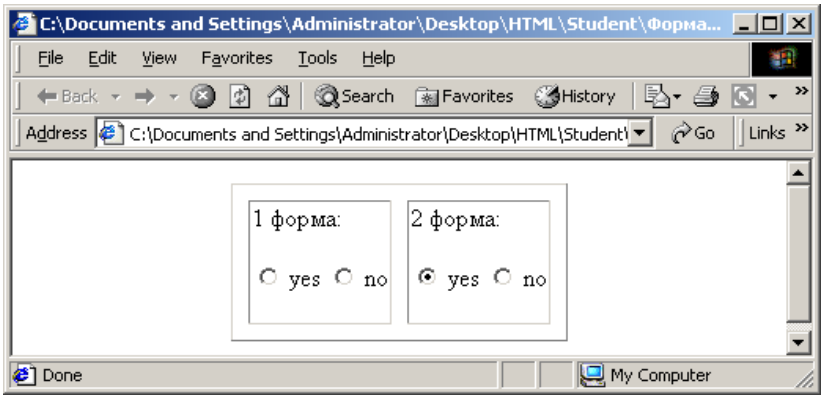
```
<HTML>
 <BODY>
   <FORM>
     <INPUT  TYPE="CHECKBOX"  NAME="CHECKBOX1”
             VALUE="checkbox_value1"> Вариант
              <BR>
     <INPUT  TYPE="checkbox"   NAME="checkbox2"
             VALUE="checkbox_value2"  CHECKED>
            Алдын  ала таңдалған нұсқа
   </FORM>
 </BODY>
</HTML>
```



2.14-сурет. Жалаушалар өрісі

Келесі мысалда батырмалар жасалады, екі-екіден батырмасы бар екі форма кестенің көршілес ұяшықтарында орналасқан.

```
<HTML>
 <BODY>
  <TABLE ALIGN=CENTER BORDER CELLSPACING=10>
   <TR>
    <TD>   1 форма:
     <FORM>
      <INPUT  TYPE="RADIO"  NAME="CHOICE"
              VALUE="CHOICE1"> yes
      <INPUT  TYPE="RADIO"   NAME="CHOICE"
              VALUE="CHOICE2"> no
     </FORM>
    </TD>
    <TD>   2 форма:
    <FORM>
      <INPUT  TYPE="RADIO" NAME="CHOICE"
              VALUE="CHOICE1" CHECKED> yes
      <INPUT  TYPE="RADIO" NAME="CHOICE"
              VALUE="CHOICE2"> no
   </FORM>
     </TD>
    </TR>
   </TABLE>
</BODY>        </HTML>
```



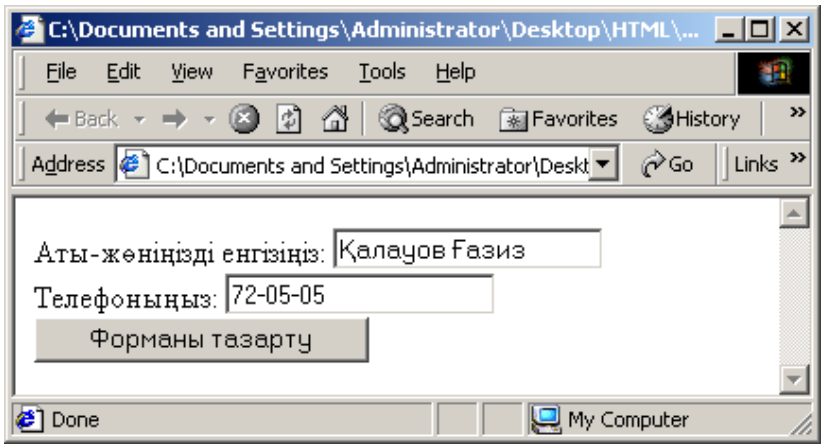
2.15- сурет. Екі батырмасы бар, екі форманың орналасуы

Келесі формада TYPE атрибутының RESET мәні, яғни енгізілген мәліметтерді тазарту жолы қолданылған, нәтижесі 2.16-суретте көрсетілген.

```
<HTML>
 <BODY>
   <FORM> Аты-жөніңізді енгізіңіз:
       <INPUT TYPE="text"> <BR>
       Телефоныңыз: <INPUT TYPE="text"> <BR>
      <INPUT TYPE="reset" VALUE="Форманы тазарту">
   </FORM>
 </BODY>
</HTML>
```

Енді TYPE атрибутының SUBMIT мәнін қолданайық (2.16-сурет).

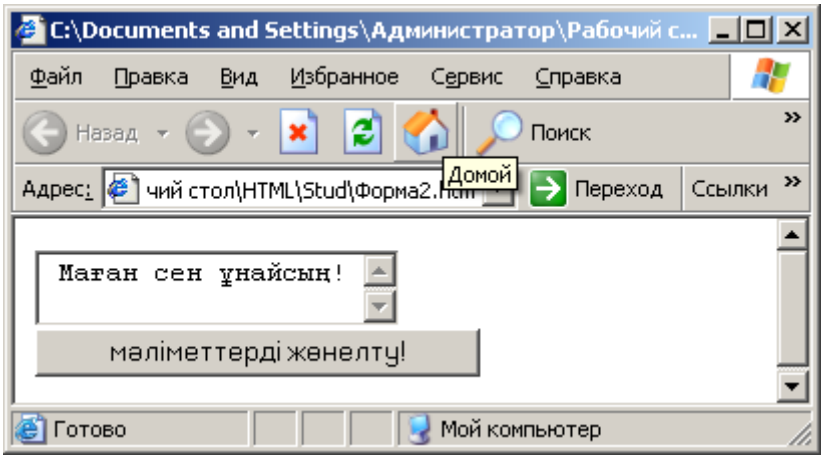
```
<HTML>
  <BODY>
   <FORM>
    <TEXTAREA> Маған сен ұнайсың! </TEXTAREA> <BR>
       <INPUT TYPE="submit" VALUE="мәліметтерді жөнелту!">
   </FORM>
  </BODY>  </HTML >
```



2.16-сурет. Батырманың орналасуы

2.9.7 Форма элементтерін стандарттан тыс қолдану

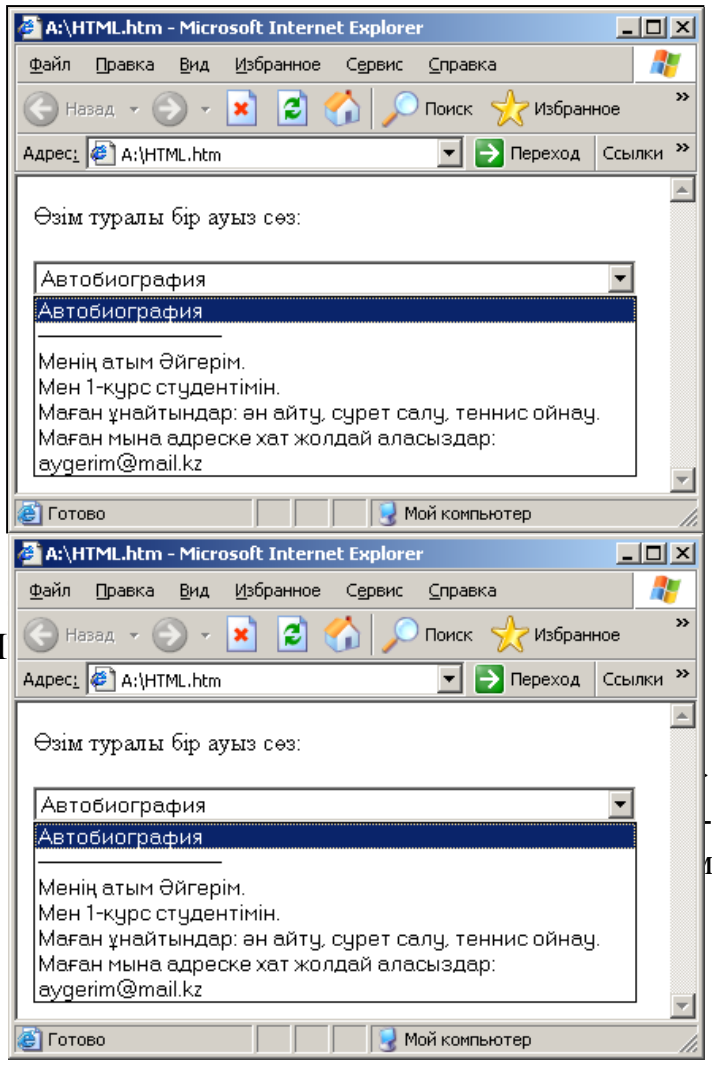
HTML тәгтерінің көбісі қолдану кезінде өздерінің негізгі қызметтерінен басқа мақсаттарда пайдаланылып жатады, мысалы, кестелер парақтарды белгілеу үшін қолданылады. Сондықтан форма элементтерін қолдану кезінде әр түрлі тәжірибе жасаудан тартынуға болмайды.



2.17 сурет. Мәліметтерді жөнелтуге мүмкіндік беретін батырма

Осындай мақсаттарда таңдау менюін пайдаланған ең ыңғайлы тәсіл болып табылады. Суырылып шығатын меню ақпараттық ағынға енгізілген мәтіндік алап (жол) қызметін атқара алады. Мысалы, таңдау менюін шағын әңгімелер енгізу үшін қолдануға болады (OPTION элементтеріне мәтін үзінділерін енгізу арқылы). Бұл мүмкіндіктерді қолдану оқырмандарды осы парақ мәліметтерімен екпінді түрде жұмыс жасауға шақырады. Төменде суырылып ашылатын тізімнің автор туралы шағын әңгіме орналастыру үшін қолданылғанын көрсететін HTML-құжаты келтірілген. 2.18-суретте мысалдағы “Автобиография” батырмасын басқаннан кейін қандай болатыны бейнеленген.

2.18- сурет. Программа нәтижесі



<H

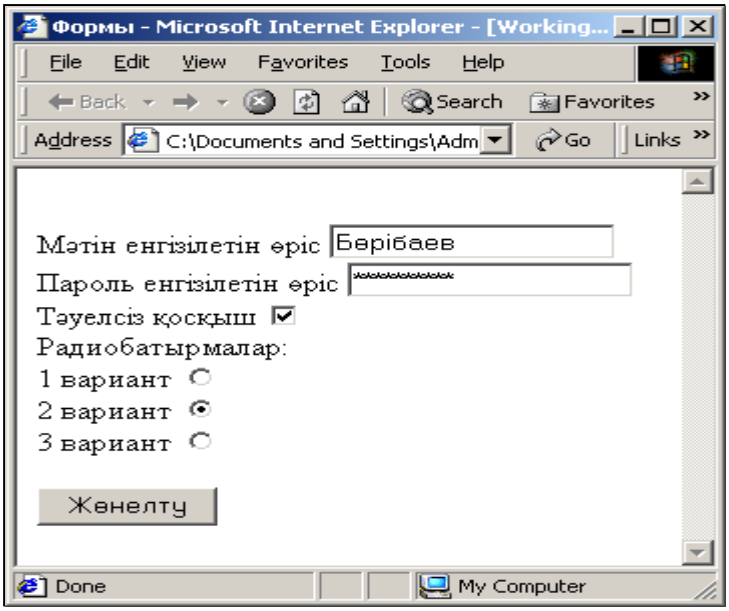
Автобиография

Әйгерім.

студентімін.

<OPTION VALUE="line4"> Маған ұнайтындар: ән айту, сурет салу, теннис ойнау.
<OPTION VALUE="line5"> Маған мына адреске хат жолдай аласыздар:
<OPTION VALUE="line6"> aygerim@mail.kz
</SELECT>
</FORM>
</BODY>
</HTML>

Енді формалардың бірсыпыра мүмкіндіктерін қамтитын тағы бір мысал келтірейік (2.19- сурет).



2.19- сурет.
Программа нәтижесі

<html> <head> <title>Формы</title> </head>
<body>
<form action="http://www.mysite.com/cgi-bin/test.exe" method="post">

Мәтін енгізілетін өріс <input type="text">

Пароль енгізілетін өріс <input type="password">

Тәуелсіз қосқыш <input type="checkbox" value=" checked">

Радиобатырмалар:

1 вариант <input type="radio" name="groupl" value="checkl">

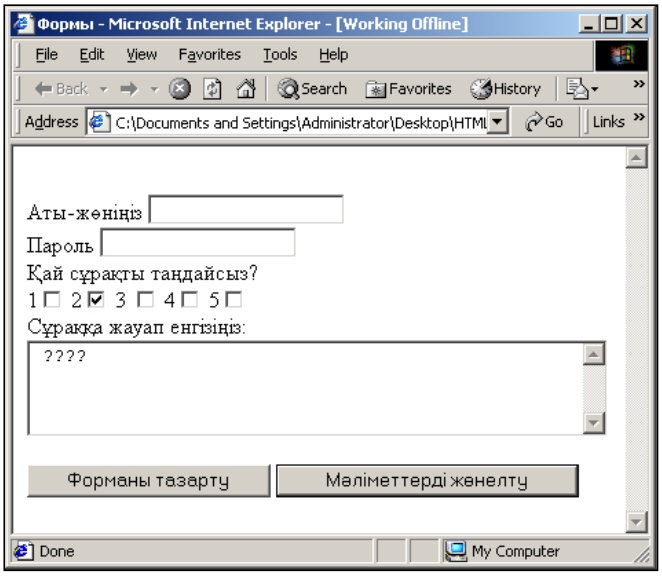
2 вариант <input type="radio" name="groupl" value="check2 checked>

3 вариант <input type="radio" name="groupl" value="check3">
<p> <input type="submit" value="Жөнелту">
</form>
</body>
</html>

Тапсырмалар:

Төмендегі 2.20-суретке сәйкес форма жасаңыздар. Оны безендіруге қойылатын талаптар:

1) Пароль өрісі 10 символдан аспауы тиіс және енгізілген мәліметтер көрсетілмейтін болсын.
2) Барлық сұрақтар нөмірлері ішінен тек біреуі ғана таңдалуы тиіс, алдын ала келісім бойынша бірінші сұрақ таңдалуы керек;
3) Сұрақ жауаптарына арналған өріс әрқайсысы алпыс символдан тұратын төрт жолды қамтуы тиіс, бастапқы мәні – "?????";
4) "Форманы тазарту" батырмасы форманың бастапқы түрін қайта қалпына келтіруі керек.



2.20- сурет.

Бақылау сұрақтары:

1. Формалардың атқаратын қызметі, орындалу тәгі және атрибуттары.
2. <TEXTAREA>, <SELECT>, <INPUT> <OPTION> тәгтерінің қызметі мен атрибуттары және олардың мүмкін мәндері.
3. Батырманы қалай жасауға болады? Қосқышты ше?
4. Енгізілген мәліметтерді тазалау қалай орындалады?

2.10 HTML тілін үйренуге арналған тапсырмалар

1-тапсырма. ***Қарапайым HTML файлын құру***

1. Өз жұмыс бумаларыңыздың ішінен жаңадан жасалған Web-құжаттарды сақтайтын STUDENT бумасын ашыңыздар.

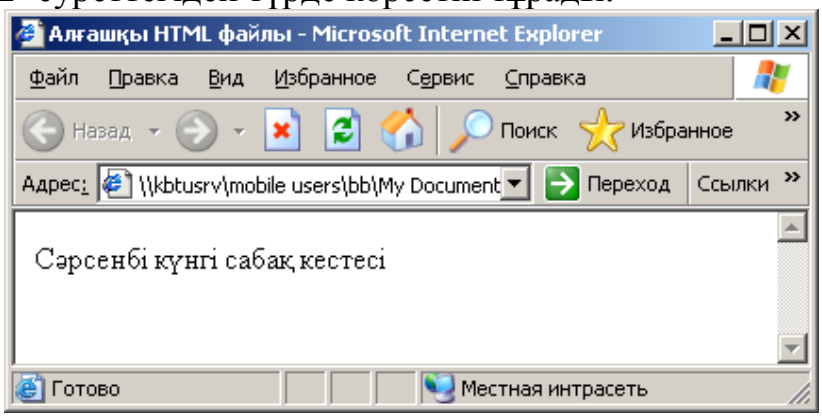
2. Блокнот редакторы терезесінде төменде көрсетілген қарапайым HTML файлының мәтінін теру керек:

```
<HTML>
 <HEAD>
  <TITLE> Алғашқы HTML файлы </TITLE>
 </HEAD>
 <BODY>
```

Сәрсенбі күнгі сабақ кестесі
</BODY>
</HTML>

3. Файлды RASP.HTM атымен STUDENT бумасында сақтап қойыңыздар да, Блокнот терезесін жауып тастаңыздар.

4. Жасалған Web-құжатты көру үшін сол **RASP.HTM** файлы белгішесін екі рет шертіңіздер, сонда *Microsoft Internet Explorer* броузері жазылған мәліметті 2.21- суреттегідей түрде көрсетіп тұрады.



2.21- сурет

5. Енді **Түр – HTML түрінде** (Вид – В виде) командасын орындау арқылы құжаттың алғашқы терілген HTML файлы мәтінін Блокнот терезесінде көруге болады. Осы мәтіннің екінші жолы соңына:

<H2> Менің алғашқы парағым </H2>

деген қосымша мәтін енгізейік.

6. Осылай түзетілген мәтінді **Файл – Сохранить** командасы арқылы қайта дискіге жазып қояйық та, *Блокнот* терезесін төменге жасырып, қайтадан *Тапсырмалар тақтасында* белгішесі көрініп тұрған **Алғашқы HTML файлы** атын шертіп, *Microsoft Internet Explorer* броузері терезесін ашайық.

7. Броузердің **Вид – Обновить** командасын орындап (немесе 

батырмасын шертіп) терезедегі мәліметтің өзгергеніне көз жеткізіңіздер. **Алғашқы HTML файлы** және **RASP.HTM** файлдарын жабыңыздар.

2- тапсырма. *Экрандағы мәтін орналасу түрін өзгерту.*

Келесі жолға көшу ісін атқаратын **
** тәгін қолданайық.

1. Бірінші тапсырманың 5–6 нөмірлі пункттерін орындау керек.

2. RASP.HTM файлын ашып, оның алғашқы мәтінін **Түр – HTML түрінде** (Вид – В виде HTML) командасы арқылы **Блокнот** терезесінде көрсету қажет.

3. HTML файлы мәтініндегі “Сәрсенбі күнгі”, “сабақ”, “кестесі” сөздерін жеке-жеке жолдарға жазайық:

<HTML>
<HEAD> <H2> Менің алғашқы парағым </H2>
TITLE> Алғашқы HTML файлы </TITLE>
</HEAD>

<BODY> Сәрсенбі күнгі
сабақ
кестесі
</BODY>
</HTML>

4. Өзгертілген осы мәтінді дискіге қайта сақтап, Блокнот программасын жасырып, Microsoft Internet Explorer терезесін ашыңыздар.

5. Microsoft Internet Explorer-де мәтін өзгерістерін бейнелеу үшін **F5** пернесін басу керек немесе **Вид – Обновить (Refresh)** командасын орындау қажет. Экрандағы бейне өзгерді ме? Жоқ.

Ескерту. Бұдан былай Web-құжатқа өзгеріс енгізген сайын екінші тапсырманың 4 – 5 пункттері орындалуы тиіс.

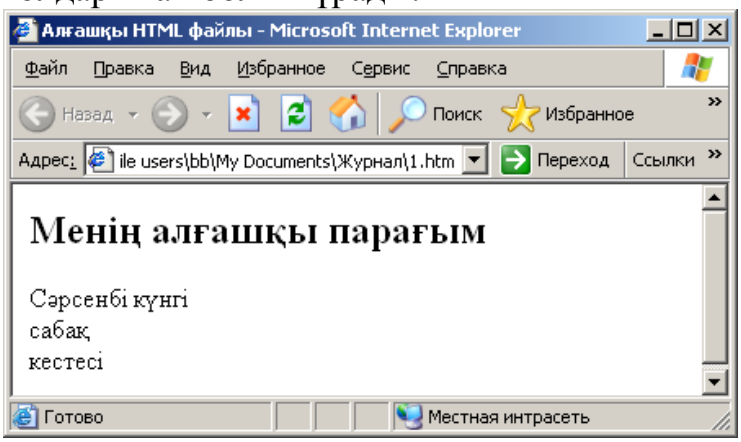
6. Енді HTML файлы мәтінінің негізгі бөлімінің керекті орындарына төмендегідей түрде
 тәгін енгізу керек:

**...<BODY> Сәрсенбі күнгі
 сабақ
 кестесі </BODY>...**

7. Мәтінді дискіде қайта сақтап, web-құжатты жаңартыңыздар, терезедегі мәлімет орналасуы 2.22-суреттегідей болып өзгереді. Енді HTML құжаты мен RASP.HTM файлын жауып тастаңыздар.

3-тапсырма. *Форматтаудың арнайы командалары*

Жаңа азат жол ашатын арнайы **<P>** тәгі бос жол енгізіп, жаңа азат жолты алдыңғы мәтін жолдарынан бөліп тұрады.

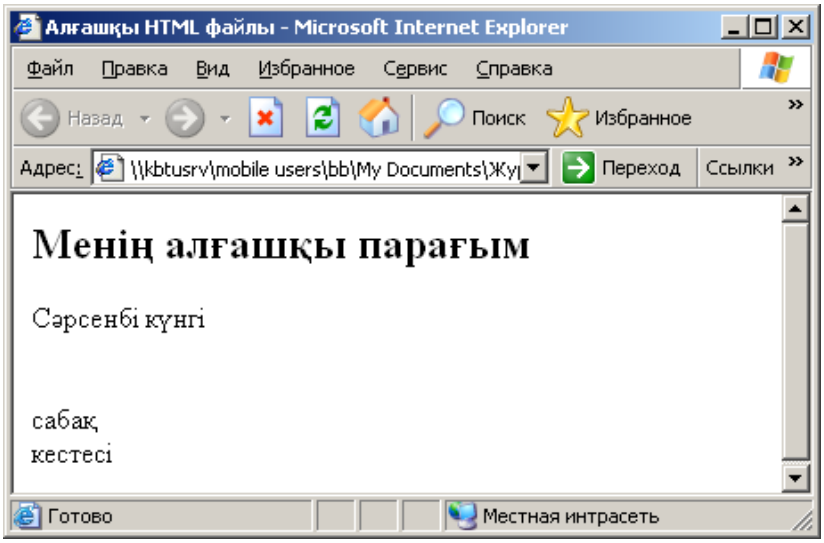


2.22-сурет

1. HTML файлы мәтінінің **<BODY> ... </BODY>** тәгтері ортасына төмендегідей өзгерістер енгізіңіздер:

**...<BODY> <P> Сәрсенбі күнгі </P>
 сабақ
 кестесі**
</BODY>...

2. Өзгертулерді **RASP.HTM** файлында қайта сақтап, *Microsoft Internet Explorer* броузері арқылы осы өзгертілген Web-құжатты қарап шығыңыздар. Экрандағы мәтін қандай өзгерістерге ұшырады? Жаңа құжат 2.23 - суреттегідей болып бейнеленуі тиіс.



2.23-сурет

4-тапсырма. *Мәтін бөліктерін ерекшелеу.*

1. **RASP.HTM** файлы мәтінінің **<BODY> ... </BODY>** тәгтері ортасына төмендегідей өзгерістер енгізіңіздер:

<BODY>
<B> Сәрсенбі күнгі </B> <I> сабақ </I> <U> кестесі </U>
</BODY>

2. Өзгертілген Web-құжатты экраннан көріп шығыңыздар (2.22-сурет). Мұнда сөздерді ерекшелеуді аралас түрде де қолдануға болады:

<I><B> Сәрсенбі күнгі </B></I> <I> сабақ </I> <U> кестесі </U>

Бірақ мұндайда тәгтерді араластыра жазудың мынадай ережелерін есте сақтаған жөн:

<Тәг-1> <Тәг-2> ... </Тәг-2> </Тәг-1> — дұрыс жазылған тізбек;
<Тәг-1> <Тәг-2> ... </Тәг-1> </Тәг-2> — қате жазылған тізбек.

5 -тапсырма. *Тақырып стильдерін қолдану*.

Тақырыптар көлемін өзгертіп жазудың алты түрлі тәгтері бар (<H1> ден <H6> -ға дейін). Әрбір тәг броузер параметрлеріне сәйкес мәтінге белгілі бір нақты стиль бере алады.

1. **RASP.HTM** файлының негізгі бөліміне төмендегідей өзгерістер енгізіңіздер**:**

... <BODY> <P> <H1> Сәрсенбі күнгі </H1> </P>
<H3> <I> сабақ </I> <U> кестесі </U> </H3>
</BODY>

2. Өзгертілген Web-құжатты экраннан көріп шығыңыздар.

6-тапсырма. *Қолданылып отырған қаріптің көлемін өзгерту*.

<FONT> тәгі ағымдағы мәтін ішіндегі кейбір жолдарда қолданылып отырған қаріптің көлемін 1-ден (ең кіші) 7-ге (ең ірі) дейін мөлшерде өзгерту мүмкіндігін береді..

1. **RASP.HTM** файлының негізгі бөліміне мынадай өзгерістер енгізіңіздер**:**

<BODY>

**<FONT SIZE=7> Сәрсенбі күнгі </FONT>
 сабақ кестесі**
</BODY>

2. Осы <FONT> тәгін пайдаланып, “сабақ кестесі” сөздерінің мөлшерін өз қалауларыңызша өзгертіп көріп шығыңыздар.

3. Мәтін бөліктерін жоғарыда айтылған ерекшелеу түрлерін және келесі жолға көшу, жаңа азат жол ашу тәгтерін пайдалана отырып бірнеше тәсілмен безендіріңіздер.

7-тапсырма. *Қаріп гарнитурасы (типі) мен түсін өзгерту.*

<FONT> тәгі мәтін қаріптерінің көлемін, типін және түсін өзгертуге мүмкіндік береді. Қаріп типін өзгерту осы тәгке **FACE** атрибутын қосу арқылы орындалады**.** Мысалы, мәтін бөлігін *KZ Arial* қарпімен жазу үшін**:** **<*FONT FACE="KZ ARIAL"*>** тәгін қолдану қажет**.**

Қаріп түсін беру үшін **<FONT>** тәгінің **COLOR="Green"** атрибутын жазу керек. Мұндағы **Green** сөзі орнына кез келген басқа түс атын немесе оның он алтылық жүйедегі кодын жазса болады.

1. RASP.HTM файлына мынадай өзгерістер енгізіңіздер:

```
... <BODY>
     <U><I><B><FONT COLOR="#FF0000" FACE="ARIAL"
      SIZE="7"> Сәрсенбі күнгі </FONT> </B></I></U> <BR>
    <FONT COLOR="GREEN" FACE="ARIAL" SIZE="4">
      сабақ кестесі
    </BODY>...
```

2. Өз қалауларыңыз бойынша құжат мәтіндерінің мөлшерін, түсін, гарнитурасын, стилін өзгерте отырып қарап шығыңыздар.

8 - тапсырма. *Мәтінді көлденең бағытта туралау.*

RASP.HTM файлына төмендегідей өзгерістер енгізіңіздер:

```
.... <BODY>
  <P ALIGN="CENTER"> <FONT COLOR="#008080" SIZE=7>
  <B> Сәрсенбі күнгі </B> </FONT> <BR>
  <FONT SIZE="6"> <I> сабақ кестесі </I> </FONT>
 </BODY> ...
```

Экранда мәтін бөліктері жол ортасына ығысады.

9-тапсырма. *Құжат фоны мен жалпы мәтіннің түсін беру.*

Құжат фоны мен жалпы мәтіннің түсін беру кезінде броузерлер алдын ала келісілген (по умолчанию) түстерді пайдаланады. Егер біз басқа түстерді пайдаланғымыз келсе, онда олардың бастапқы мәндері **<BODY>** тәгінде көрсетіледі. Осы тәгтің **BGCOLOR= ...** атрибуты құжаттың фон түсін, **TEXT=** ... атрибуты жалпы мәтін түсін анықтайды. **LINK=** ... және **VLINK=** ... атрибуттары әлі қаралмаған және қаралған гиперсілтемелік сөздер түсін тағайындайды.

RASP.HTM файлына төмендегідей өзгерістер енгізіңіздер:

```
...<BODY BGCOLOR="#FFFFCC" TEXT="#330066">
   <P ALIGN="CENTER"> <FONT COLOR="#008080" SIZE=7>
```

**<B> Сәрсенбі күнгі </B> </FONT>
**
<FONT SIZE=4> <I> сабақ кестесі </I> </FONT>
</BODY> ...

Экранда мәтін бөліктерінің аздап өзгеріске ұшырағандарын байқаған шығарсыздар.

10-тапсырма. *Web-парақ ішіне сурет орналастыру.*

<IMG> тәгі құжат ішіне сурет қою мүмкіндігін береді. Бұл тәг қосарланбай, жалғыз жазылады. Келесі мысалдағы *DOG.JPG* графикалық файлы осы Web-құжат орналасқан *STUDENT* бумасы ішінде жазылып тұр деп есептеледі.

RASP.HTM файлына төмендегідей өзгерістер енгізіңіздер:

... <BODY BGCOLOR="#FFFFFF" TEXT="#330066">
<P ALIGN="CENTER"> <FONT COLOR="#008080" SIZE=7>
**<B> Сәрсенбі күнгі </B> </FONT>
**
**<FONT SIZE=6> <I> сабақ кестесі </I> </FONT>
**
<IMG SRC="DOG.JPG">
</BODY>...

<IMG> тәгінің 5-*кестеде* көрсетілгендей бірсыпыра атрибуттары бар, олар **IMG** тәгінен кейін кез келген реттілікпен орналаса береді.

11-тапсырма. *Сурет атрибуттарын тағайындау*

ALT, BORDER, HEIGHT, WIDTH.атрибуттарын пайдаланып, өз қалауларыңызша RASP.HTM файлын өзгертіңіздер.

Ескерту. Суреттік файлдың көлемінің (байтпен берілген) өзгеруіне назар аударып отырыңыздар, өйткені ол Web-құжаттың Интернеттен компьютерге жүктелу уақытына қатты әсер етеді.

12-тапсырма. *Web-парақтарға фондық суреттер салу.*

Фондық сурет – мәтіндер астында (айналасында) тұсқағаз тәрізді орналасқан суреті бар графикалық файл. Броузерге фон ретінде тағайындалғанда, сурет көбейіп терезе аумағын толық алып тұрады.

Фон ретінде пайдаланылатын сурет **<BODY>** тәгінде көрсетіледі**.**

RASP.HTM файлының негізгі бөліміне төмендегідей өзгерістер енгізіңіздер:

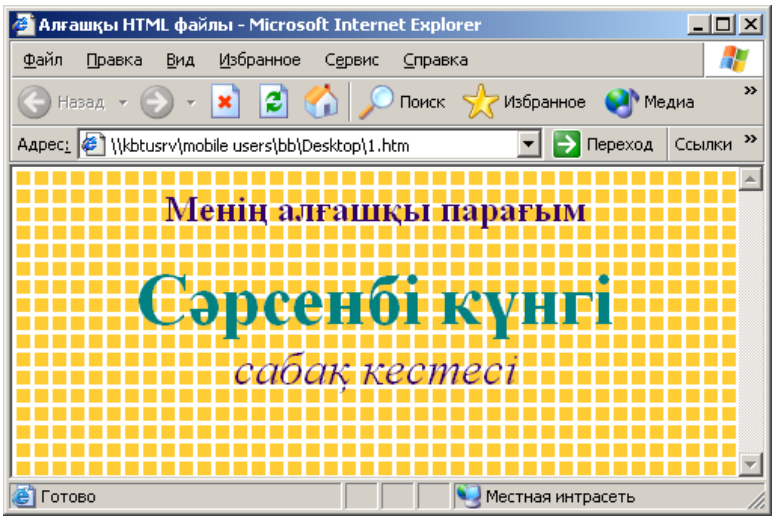
...<BODY BACKGROUND="fon1.GIF" TEXT="#330066">
<P ALIGN=CENTER>
<FONT COLOR="#008080" SIZE=7><B> Сәрсенбі күнгі </B>
**</FONT>
**
<FONT SIZE="6"> <I> сабақ кестесі </I> </FONT> </P>
</BODY>...

Осы файлдың жұмыс нәтижесі экранда төмендегі 2.24-суреттегідей болып бейнеленеді.

13-тапсырма. ***Кесте жасау***

Кесте тік бағыттағы бағаналар мен көлденең орналасқан жолдардан тұратын торлар жиыны түрінде қарастырылады. Кесте біртіндеп жолдар

тізбегі бойынша толтырылады (солдан оңға қарай жол соңына дейін, сонан соң келесі жолға көшу). Әрбір ұяға мәлімет енгізіледі. Бос ұя жасау үшін бос орын таңбалары енгізілуі тиіс.



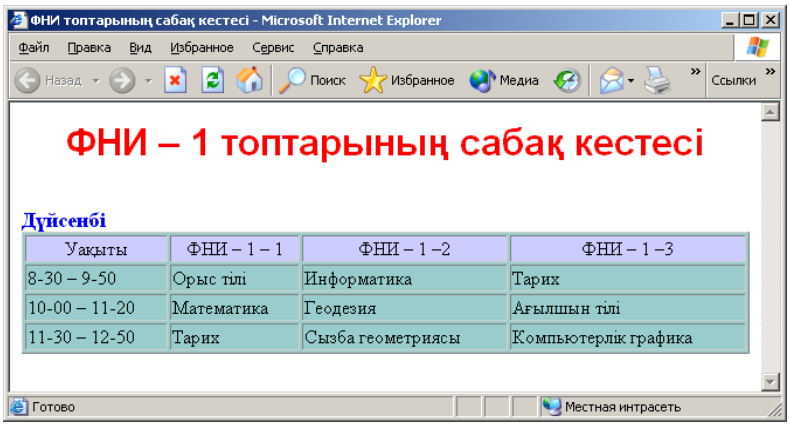
2.24- сурет

1. Блокнот программасын іске қосыңыздар.
2. Келесі мәтін жолдарын теріңіздер:

```
<HTML> <HEAD> <TITLE>ФНИ топтарының  сабақ кестесі
     </TITLE> </HEAD>
 <BODY> <P ALIGN=CENTER> <FONT COLOR="RED"
    SIZE="6" FACE="KZ ARIAL">
   <B> ФНИ – 1 топтарының сабақ кестесі </B>
   </FONT></P> <BR>
   <FONT COLOR="BLUE" SIZE="4" FACE="Times New Roman">
   <B> Дүйсенбі </B> </FONT> <BR>
    <TABLE BORDER="1"  WIDTH=100%  BGCOLOR="#99CCCC">
      <TR BGCOLOR="#CCCCFF" ALIGN=CENTER>
        <TD> Уақыты </TD> <TD> ФНИ-1-1</TD> <TD> ФНИ-1-2 </TD>
        <TD> ФНИ – 1 –3 </TD> </TR>
      <TR> <TD> 8-30 – 9-50 </TD><TD>Орыс тілі </TD>
         <TD> Информатика </TD>
         <TD> Тарих </TD> </TR>
      <TR> <TD>10-00 – 11-20</TD><TD>Математика</TD>
          <TD>Геодезия </TD> <TD>Ағылшын тілі </TD> </TR>
      <TR><TD>11-30 – 12-50</TD><TD>Тарих</TD>
          <TD>Сызба геометрия </TD>
          <TD> Компьютерлік графика </TD> </TR>
    </TABLE>
  </BODY>
</HTML>
```

3. Файлды ФНИ.HTM атымен сақтап, жауып тастаңыздар.
4. Сол файл белгішесін екі шертіп, Web-парақты *Microsoft Internet Explorer* броузер арқылы шығарғанда, экранда 2.25-суреттегідей бейнені көресіздер.

5. Енді осы тәсілмен *Сейсенбі* күнге арналған сабақ кестесін ары қарай жалғастырып жасап шығу керек. Ол үшін бос жолдар аралығында орналасқан программа бөлігін көшіріп, екінші бос жолдан кейін орналастыру керек. Сонан соң екінші бөліктегі «Дүйсенбі» сөзін «Сейсенбі» сөзіне алмастырып, сабақ кестесін де екінші күнге сәйкес өзгертіп шығу қажет. Осындай жолмен апта күндерінің бәріне де арналған сабақ кестесін жасап шығыңыздар.



Уақыты	ФНИ – 1 – 1	ФНИ – 1 –2	ФНИ – 1 –3
8-30 – 9-50	Орыс тілі	Информатика	Тарих
10-00 – 11-20	Математика	Геодезия	Ағылшын тілі
11-30 – 12-50	Тарих	Сызба геометриясы	Компьютерлік графика

2.25-сурет

14-тапсырма. *Гипермәтіндік байланыстар орнату*

HTML тілінің ең негізгі қасиеті ретінде оның басқа құжаттарға сілтеме жасау мүмкіндігін айтуға болады. Бір HTML файлынан мынадай басқа құжаттарға:

- алыста орналасқан компьютердегі HTML-файлға,
- ағымдағы HTML-құжаттың ішіндегі белгілі бір орынға,
- HTML-құжаты болмайтын кез келген файлға сілтемелер жасауға болады.

Сілтеме ретінде мәтінді немесе суретті пайдалануға болады.

Бір құжат аймағында сілтемелер жасау

Мұндай сілтемелер екі бөліктің болуын талап етеді, олар: белгі және сілтеме. Белгі сілтеме бойынша ауысып баратын осы құжат ішіндегі бір нүктені анықтайды. Сілтеме белгі атын пайдаланады. Броузердің алдын ала тағайындалатын параметрлеріне сәйкес сілтемелер басқа түспен ерекшеленіп тұрады немесе оның асты сызылады.

Сілтеме былай сипатталады:

<A HREF="#ДС"> Дүйсенбі </A>

Сілтеме арқылы ауысып баратын нүктедегі белгі **(ДС)** алдына # символы қойылады. Белгі атынан кейінгі **“>”** және **“<”** символдары арасына ауысу үшін тышқанмен шертілетін сілтеме сөз (“Дүйсенбі”) жазылады.

Осы **Дүйсенбі** гиперсілтеме сөзін шерткенде баратын нүктеде төмендегідей белгілер тұруы тиіс:

<A NAME="ДС"> Дүйсенбі **</A>**

1. Бұрынғы ФНИ.HTM файлы мәтінін Web-парақтың алдыңғы жағына жазылатын апта күндерінің аттары жазылған кестелермен толықтырайық. Ол үшін қалың қаріппен жазылған

...<FONT COLOR="RED" SIZE="6" FACE="KZ ARIAL">
<B> ФНИ – 1 топтарының сабақ кестесі </B>
</FONT></P>
 ...
жолдарынан кейін мынадай тәгтер тізбегін жазайық:

```
...<TABLE WIDTH=100%>
    <TR> <TD>Дүйсенбі</TD>  <TD> Сейсенбі </TD>
      <TD>Сәрсенбі</TD>  <TD>Бейсенбі</TD>
     <TD>Жұма</TD>  <TD>Сенбі</TD>
    </TR>
  </TABLE> <BR> ...
```

2. Бұл бөліктен кейін (RASP.HTM файлы) *Дүйсенбі* сөзіне сілтейтін белгі қоямыз, яғни <B> Дүйсенбі </B> </FONT> сөздері орнына:

... <B> <A NAME="ДС"> Дүйсенбі </A></B></FONT> ... сөздерін жазу керек.

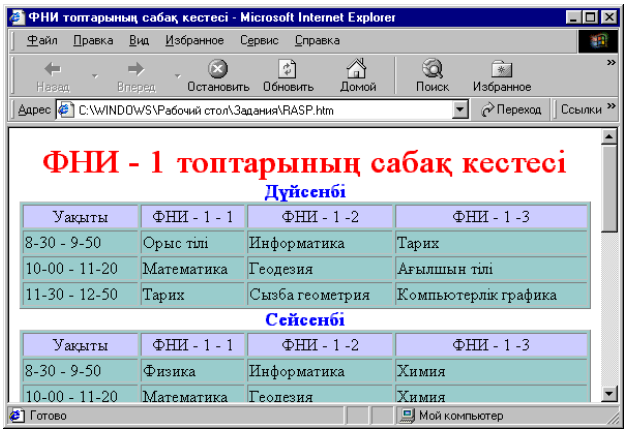
3. Енді кесте ішіндегі **Дүйсенбі** сөзіне гиперсілтеме жасаймыз, яғни <TABLE WIDTH=100%> <TR> <TD>Дүйсенбі</TD> <TD> Сейсенбі</TD> <TD> Сәрсенбі</TD>... сөздері орнына:

```
<TABLE WIDTH=100%>
 <TR> <TD> <A HREF= "#ДС">Дүйсенбі</A> </TD>
          <TD><A HREF= "#СС">Сейсенбі</A> </TD>
          <TD><A HREF= "#СР">Сәрсенбі</A> </TD> ...
```

сөздерін жазамыз.

4. Файлды дискіге жазып сақтаймыз.

5. Осының нәтижесінде алынған Web-парақты қарап шығыңыздар. Экранда 2.26-суреттегі көрініс бейнеленуі тиіс.



2.26- сурет

6. Енді Дүйсенбі сөзін, сонан соң Сейсенбі сөзін, т.с.с. күн аттарын тышқанмен таңдап шертіп көріңіздер. Гиперсілтемелердің дұрыс жұмыс істейтініне көз жеткізіңіздер.

15 - тапсырма. *Басқа HTML-құжатқа сілтеме жасау.*

Ерекшеленіп тұрған сілтеме сөздерді шерту арқылы басқа файлдарға да көшуге болады. Төмендегідей сілтеме жасайық:

<A HREF="RASP.HTM"> ФНИ **</A>**

Файл атынан (RASP.HTM) кейін **“>”** және **“<”**символдары арасында сілтеме сөз (“ФНИ”) орналасқан, осы сөзді шерту жолымен оның алдында аты көрсетілген файлға ауысуға мүмкіндік бар.

1. Броузер терезесіне RASP.HTM файлын жүктеңіздер.
2. Осы файлға төмендегідей өзгертулер енгізіңіздер:

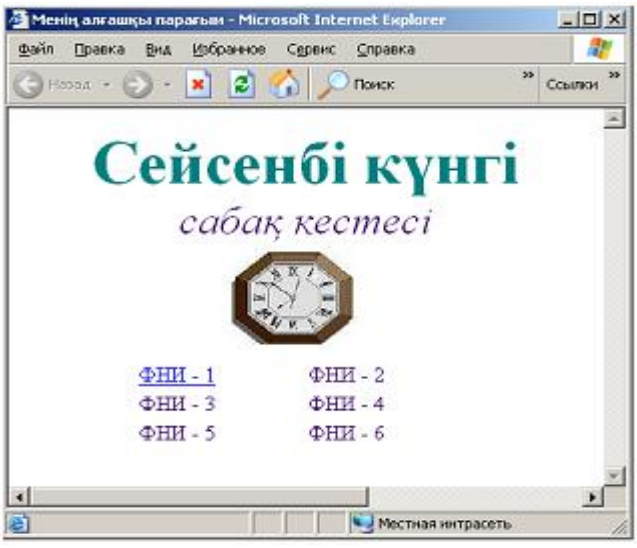
```
<HTML> <HEAD> <TITLE> Менің алғашқы парағым </TITLE></HEAD>
  <BODY BGCOLOR="#FFFFFF" TEXT="#330066"> <CENTER>
    <FONT COLOR= "#008080" SIZE="7"><B>Сейсенбі күнгі</B>
    </FONT><BR>
    <FONT SIZE="6"> <I> сабақ кестесі </I> </FONT><BR>
    <IMG  HSPACE=150  VSPACE=10  SRC="ЧАСЫ.JPG">
    <TABLE WIDTH=60%>
      <TR><TD><A HREF="RASP.HTM"> ФНИ - 1 </A></TD>
            <TD> ФНИ - 2</TD> </TR>
      <TR><TD> ФНИ - 3</TD> <TD> ФНИ - 4 </TD> </TR>
      <TR><TD> ФНИ - 5 </TD><TD> ФНИ - 6</TD>  </TR>
    </TABLE>
   </CENTER>
  </BODY>
</HTML>
```

3. Файлды сақтап, терезені жауып, оның нәтижесін броузер арқылы қарап шығыңыздар. Экрандағы көрініс 2.27-суреттегідей болуы тиіс.

ФНИ – 1 гиперсілтемесін шертіңіздер, сонда ФНИ-1.htm файлына ауысуларыңыз керек. *Артқа* (*Назад* – броузердің жоғарғы сол жақ бұрышында) батырмасын шертіңіздер, сонда RASP.HTM файлына қайтып ораласыздар.

16 -тапсырма. *Басқа HTML-құжатқа суретпен сілтеме жасау*

1. ФНИ.HTM файлының соңғы жағына алғашқы параққа – ***ФНИ топтарының сабақ кестесі*** (файл RASP.HTM) парағына сілтеме жасау қажет. Сілтеме ретінде төмендегідей түрде графикалық файлды пайдаланыңыздар:

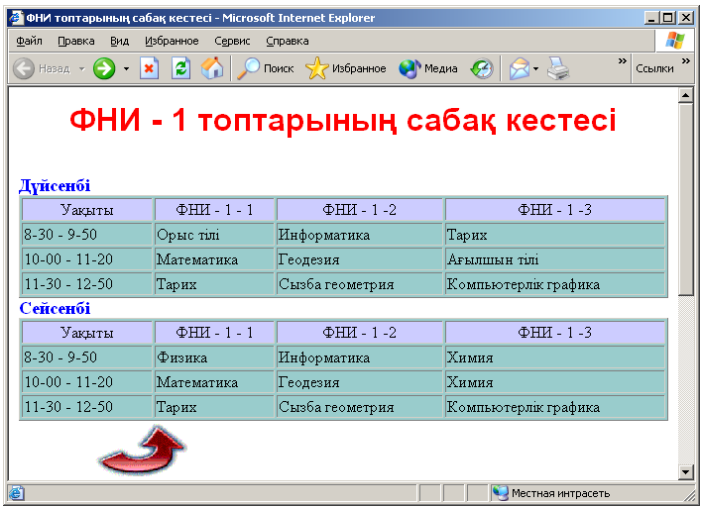


2.27-сурет

```
... </TR>
  </TABLE><BR>
   <CENTER>
    <A HREF="RASP.HTM"> <IMG SRC="HOME.GIF"
     BORDER="0"> </A>
   </CENTER>
</BODY>
</HTML>
```

2. Осының нәтижесінде алынған Web-парақты қарап шығыңыздар. Экранда 2.28-суреттегідей парақ болуы тиіс.

Мұнда сілтеме рөлін оң жаққа және жоғары бағытталған тіл сызық түріндегі сурет атқарады, ол HOME.GIF файлында жазылған.



2.28-сурет

17-тапсырма. *Соңғы өзіндік тапсырма.*

Өз топтарыңыз немесе отбасыңыз жайлы мәлімет беретін Web-парақ жасаңыз. Алғашқы параққа топ туралы, куратор мұғалімдеріңіз (ата-аналарыңыз) жайлы қысқаша әңгіме жазыңыздар. Бірге оқитын достарыңыз туралы мәліметті жеке парақтарға орналастырыңыздар. Алғашқы парақтан достарыңыз туралы мәлімет беретін парақтарға сілтемелер жасаңыздар. Достарыңыз туралы Web-парақтардан алғашқы параққа қайтатын сілтеме жасауды да ұмытпаңыздар.

Жақсы Web-парақты қалай жасауға болады?

1. Web-парақтардағы мәліметтердің орналасуы қарапайым әрі логикалық түрде қарап шығуға, оқуға ыңғайлы болуы тиіс. Парақтардағы мәліметтердің оларды қабылдауға жеңіл болуының бір тәсілі экран бетінде мәтін де, сурет те орналаспаған бірсыпыра бос орындардың қалдырылуы болып табылады. Парақта мәліметі көп болса, ол оқушыны жалықтырып жібереді.

2. Экрандағы ақпараттарды тізім немесе *кесте* түрінде жасауға тырысу керек, сонда маңызды мәліметтерді оңай тауып алуға болады.

3. Бір суреттен кейін бірден екінші сурет орналаспағаны дұрыс, олардың араларында мәтіндік ақпараттар берген дұрыс.

4. Ақпарат бөліктерге бөлініп берілсе, оларды оқу, түсіну жеңіл болады. Азат жолтардың да өте ұзын болмағаны дұрыс, олардың көлемділерін бірнеше шағын азат жолтарға бөлген абзал.

5. Егер Web-парақ көлемі үлкен болса, онда құжат бөліктеріне жылдам ауысуға мүмкіндік беретін сілтемелер жасау қажет. Кейде бір мәселеге арналған ақпараттарды тақырыптарға бөліп, оның мазмұнын негізгі бір параққа жазып, ал әр тақырыпты басқа беттерге сілтемелер арқылы орналастыру керек.

6. Суреттер мен графикалық бейнелерді пайдалану көптеген тұтынушыларды қызықтыруы мүмкін, бірақ суреттерді желі арқылы қабылдау ұзақ уақыт алатыны есте болуы керек. Егер бір әдемі суретті көру үшін 5 минуттай уақыт кететін болса, оның әдемілігі ешкімді де қызықтыра қоймас.

Web-парақты тексеріп тестіден өткізу

Серверге өз Web-парақтарыңызды орналастыру алдында оларды тест арқылы тексеріп алған жөн. Жасалған құжат алғашқы тексеруді, яғни "жергілікті тексеруді" өз қатты дискіңіздің аумағында өтуі тиіс. Тексеру кезінде әр түрлі броузерлерді пайдаланған абзал. Олардың бір-бірінен айырмашылығы әжептеуір болуы ықтимал.

Web-құжатты тексеруден өткізу кезінде мынадай жайттар есте болсын:

1. *Емле қателерін тексеру*. Мәтіннің жазылуын автоматты тексеру арқылы толық қарап шығу керек (ол үшін Microsoft Word редакторын пайдалануға болады, қазақша мәтінді тексеретін мүмкіндіктер де бар) немесе басқа кісіні қатесін тексеріп қарап шығуға шақырыңыздар.

2. *Навигацияны (ауысуларды) тексеру*. Әрбір парақта басқа құжаттарға, парақтарға ауысу мүмкіндіктері болуы тиіс, солардың дұрыс жұмыс істейтініне көз жеткізіңіздер.

3. *Сыртқы файлдармен қатынасу тәсілдерінің дұрыстығын тексеру*. Графикалық, дыбыстық және бейнефайлдарының өз орындарында тұрғанын қарап шыққан абзал, олардың сол орындарынан дұрыс жүктелетіндігіне көз жеткізіңіздер (файлдар адрестері дұрыс көрсетілуі тиіс). Графикалық мүмкіндігі жоқ броузерлерге арналған ауыспалы мәтіндік хабарламалардың бар екенін қарап шыққан артық болмайды.

4. *Құжаттың жүктелу уақытының шамадан тыс ұзақ болмағанына көз жеткізіңіздер*.

5. *Басқа кісілердің сіздер жасаған Web-парақтарды қарап шығуын қамтамасыз етіңіздер*. Сіздің құжатыңызбен таныс емес адамдардың парақтарыңызды бастан аяқ тексеріп шығуын өтіну керек. Кейде өздеріңіз ешқашан да таба алмайтын қателерді сыртқы көз жылдам байқайды.

3 СТИЛЬДЕРДІҢ САТЫЛЫ КЕСТЕЛЕРІ (CSS)

CSS – Cascading Style Sheets (Стильдердің сатылы кестелері) — гипермәтіндік мәліметтер мазмұнын олардың экрандағы бейнелену формасынан бөліп орындауға мүмкіндік беретін құрал болып табылады.

Қазіргі WWW-сайттарын безендіруді стильдердің сатылы кестелерінсіз көзге елестету қиын. Стильдердің сатылы кестелері мынадай мүмкіндіктер береді:

- WWW-парағын безендіруді оның мазмұнынан бөліп тәуелсіз орындау;
- Стандартты HTML тілі мүмкіндіктерін түбегейлі түрде кеңейту;

CSS технологиясы, мысалы, объектілерді абсолюттік орналастыру ісін орындай алады, яғни объектілерді экранға нақты координаталар арқылы қажетті орынға шығаруға болады. Мұндағы объектілік модельдің координатасын программада өзгерту экрандағы сол элементтің орнын ауыстыра алады. Олардың ара қашықтықтарын өзімізге ыңғайлы болып табылатын өлшемдердің бірі - *мм*, *см* немесе *дюйм*, *пункт* арқылы беруге де болады. CSS сценарийлермен қосыла отырып, гипермәтіндік парақтарды кәсіби программаларда жасалған мультимедиялық өнімдерден айнымайтын динамикалық әрі интерактивті деңгейде көрсетуге мүмкіндік береді. «Скриптер» немесе «сценарийлер» – гипер-мәтіндерді түрлендіре алатын арнайы программалау тілінде жазылған программалық кодтар. JavaScript – кодтарын HTML-мәтіндеріне енгізе отырып, құжаттарды түрлендіре алатын, көпшілікке арналған тіл. Сценарийлер енгізілген CSS статикалық HTML-тілін динамикалық жаңа күйге келтіретін қосымша мүмкіндіктер жиыны.

- CSS арқылы HTML-тілінің экрандағы элементтерді бейнелеудің көптеген тәсілдеріне жауапты тәгтер мен олардың атрибуттарын алып тастау жолымен файлдың жалпы көлемін кішірейте аламыз (мысалы, мынадай тәгтер: <font>, , <в>, <i>, <s>, <U> т. б. ; атрибуттар: align, color, bgcolor, size, width, height, т. с. с.);
- HTML тілінің тәгтері мен атрибуттарына қарағанда, CSS кестелері сайттың сыртқы түрін басқарудың жеңіл тәсілдерін береді;
- бір файлдағы стиль сипаттамаларын өзгерту арқылы жүздеген гипермәтіндік парақтардың сыртқы түрлерін бірден өзгерте аламыз.

Стильді бір тәгтің параметрлері ішіне жазуға болады. Ол үшін BODY ішіндегі кез келген тәгке (BODY тәгіне де) STYLE атрибуты қолданылады. Бұларды *"инлайндық" стиль кестелері* деп атайды.

STYLE атрибуты арқылы

- Жеке тәг стилі;
- Жеке HTML-файлы стилі;
- Бірнеше HTML-файлдарына арналған стиль;
- Аралас стильдерді пайдалану атқарылады.

3.1 Жеке бір тәг үшін жазылған стиль

Мысалы, <P> тәгі арқылы нақты бір азат жол қалай бейнеленетінін былай көрсете аламыз:

<P style="font-size:1.5cm;

color:green">

Бұл азат жолқа стильдік анықтау тәсілі қолданылып отыр.

Стиль style атрибутымен берілген. Мұнда браузерге азат жолты көлемі 1.5 сантиметр болатын жасыл әріптермен жазуға нұсқау берілген.

Стильді анықтау мынадай түрде жазылады:

сипаттамасы:шамасы;

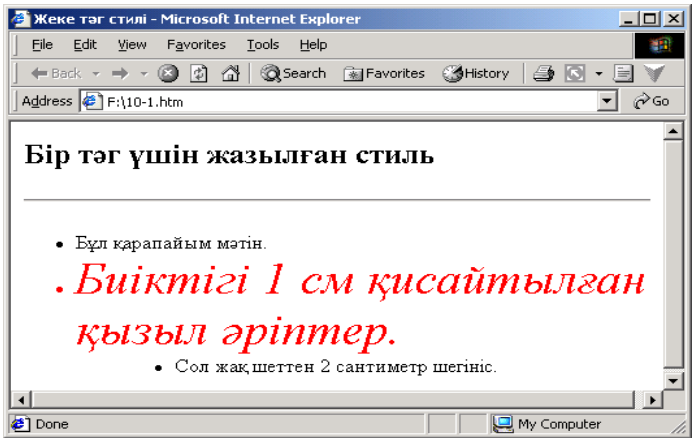
font-size:1.5cm;

color:green;

Әрбір анықтаулар бір-бірінен «;» символы арқылы бөлініп жазылады.

Style атрибутын оның параметрлері арқылы әрбір тәгке қолдана аламыз. Енді стильдер қолданылатын бір мысалды толығырақ қарастырайық:

```
<HTML>
 <HEAD>
  <TITLE> Жеке тәг стилі </TITLE>
 </HEAD>
 <BODY bgcolor=white text=black>
  <H2> Бір тәг үшін жазылған стиль </H2>
  <HR>
  <UL>
   <LI> Бұл қарапайым мәтін.
   <LI style="color:red; font-size:1cm;
    font-style:italic">
    Биіктігі 1 см қисайтылған қызыл әріптер.
   <LI style="margin-left:2cm">
    Сол жақ шеттен 2 сантиметр шегініс.
  </UL>
 </BODY>
</HTML>
```



3.1-сурет. Жеке тәг стилі

3.2 Жеке HTML-файлына арналған стиль

Стильді тек бір тәг үшін немесе бірнеше тәгтер үшін бір жазылған анықтаулар HTML-құжатының басынан соңына дейін әсер ететіндей жасауға болады. Мысалы, барлық тәгтер атауларын тізіп, стильдік анықтауларды құжаттың тақырып бөлігіне орналастыру қажет.

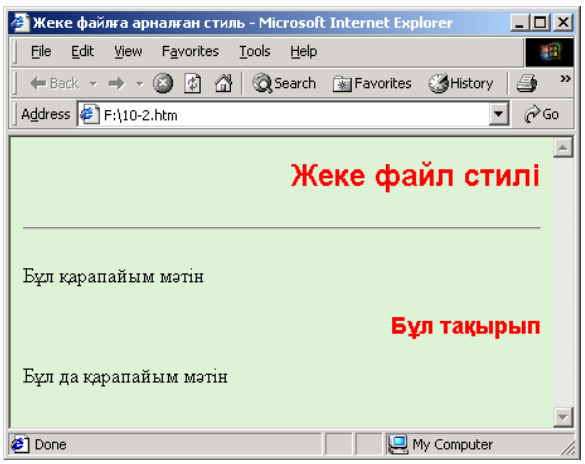
Стильдік анықтаулар немесе *селекторлар* мынадай блок ішіне жазылады <style>. . . </style> және HTML-комментарий ретінде жазылады.

Стильдік анықтау форматы:

Тәг аты (немесе бірнеше тәгтер аттары үтір арқылы бөлініп жазылады)
{
  сипаттама: мәні;
  …
сипаттама: мәні;
}

Мысалы:

```
<HTML>
 <HEAD>
  <TITLE> Жеке файлға арналған стиль </TITLE>
  <STYLE type="text/css">
   <!--
    H1,H2,H3,H4,H5,H6
    {
     text-align: right;
     color:red;
     font-family: "Arial Cyr",
     Geneva, sans-serif;
    }
   -->
  </STYLE>
 </HEAD>
 <BODY bgcolor=#DFF0D5 text=black>
  <H2> Жеке файл стилі </H2>
  <HR>
  <P> Бұл қарапайым мәтін
  <H3> Бұл тақырып </H3>
  <P> Бұл да қарапайым мәтін
 </BODY
</HTML>
```



3.2-сурет. Жеке файл стилі

Браузер мұндағы тақырыптарды жұмыр қаріппен (рубленый шрифт) қызыл түсте оң жақ шетке туралап орналастырады. Браузердің мұндай әрекетін мынадай кодтар атқарады:

```
<STYLE type="text/css">
  <!--
   H1,H2,H3,H4,H5,H6
   {
     text-align: right;
     color: red;
     font-family: "Arial Cyr", Geneva, sans-serif;
    }
  -->
</STYLE>
```

Бұл мысалда үш сипаттама былай берілген:

- text-align: right; – оң жақ шетке туралау;
- color: red; – түсі қызыл;
- font-family: "Arial Cyr", – жұмыр Geneva, sans-serif; қаріп (шрифт).

Тақырыптар Arial Cyr қарпімен берілуі тиіс (егер тұтынушы компьютерінде ол бар болса). Егер ол қаріп (шрифт) болмаса, браузер келесі көрсетілген Geneva, Helvetica қаріптерді немесе әйтеуір бір жұмыр шрифті («sans-serif») пайдаланады. Егер бір де бірі табылмаса, «үнсіз келісім бойынша қаріпті» (шрифтом по умолчанию), яғни көбінесе Times New Roman қарпін қолданады.

3.3 Бірнеше HTML-файлдарға арналған стиль

Әдетте бірнеше файлдарға арналған стильдер басқа бір жеке файлға бөлек жазылады. Мұндай файл типі (кеңейтілуі) css болып жазылады. Мысалы, style.css файлына мынадай стильдерді жазайық:

```
BODY {margin-left: 40px;}
Hl,H2,H3,H4,H5,H6
 {
  text-align: right;
```

```
  color: red;
  font-family: "Arial  Cyr", Geneva, sans-serif;
 }
```

Осы стильдерді іске қосу үшін HTML-файлдың тақырып <head>...</head> бөлігіне мынадай сілтеме орналастыру керек:

<LINK rel=stylesheet type="text/css“ href=style.css>

Осы стильдік файлға бірнеше HTML-құжаттар сілтеме жасай алады. Осы файлға бір өзгеріс енгізу ішкі сілтемелері бар ондаған парақтарға әсер етеді.

Мынадай стильдік анықтау

body {margin-left: 40px;}

барлық жолдар үшін сол жақ шеттен 40 пиксель шегініс береді. Міне, HTML тіліндегідей кесте қолданбай, өріс көрсетпей, стиль арқылы ғана осындай мүмкіндіктер жасауға болады екен.

Аралас стильдерді пайдалану

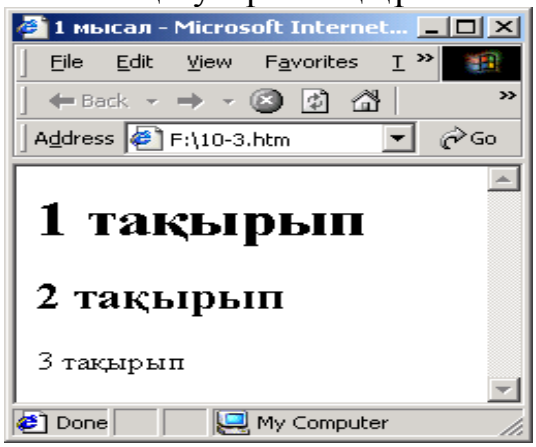
Сонымен HTML-кодтары үшін стильдерді пайдаланудың үш тәсілі бар екен:

- жеке тәг үшін анықтау;
- HTML-файлының тақырыбында анықтау;
- басқа CSS файлында стильді анықтау.

Енді осы тәсілдерді араластыра пайдаланып көрейік. Олардың қайсысы басым екенін мысалдар арқылы қарастырайық.

Бірнеше мысалдар келтірейік.

CSS анықтаулары жоқ құжат



3.3-сурет. Программа нәтижесі

Төмендегі мысалда тақырыптар қара түспен ақ фон арқылы жазылады.

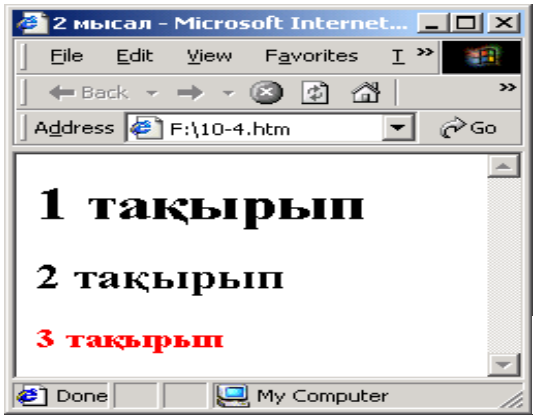
```
<HTML>
 <HEAD>  <TITLE> 1 мысал </TITLE>
 </HEAD>
 <BODY bgcolor=white
      text=black>
   <H1> 1 тақырып  </H1>
       <H2> 2 тақырып  </H2>
```

```
      <H3> 3 тақырып </H3>
 </BODY>
</HTML>
```

Жеке тәг үшін CSS анықтау

Төмендегі код алғашқы екі тақырыпты қара түспен, ал соңғысын қызыл түспен бейнелейді.

```
<HTML>
 <HEAD>
  <TITLE> 2 мысал </TITLE>
 </HEAD>
 <BODY bgcolor=white
      text=black>
  <H1>  1 тақырып </H1>
  <H2>  2 тақырып </H2>
  <H3 style="color:red">
       3 тақырып </H3>
 </BODY>
</HTML>
```



3.4-сурет. Программа нәтижесі

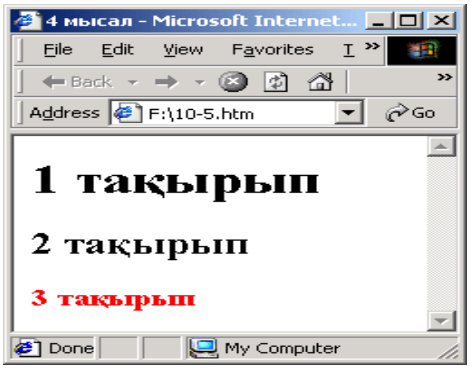
CSS-файлдағы CSS-анықтаулар prim.css файлының ішкі мәтіні:

```
H1,H2,H3
{
 color: green;
}
```

Енді осы файлды пайдаланатын HTML-құжат коды мен оның бейнесін қарастырайық.

```
<HTML>
 <HEAD>
  <!-- prim.css файлын іске қосу. - - >
<LINK rel=stylesheet type="text/css" href-prim.css>
   <STYLE type="text/css">
    <!--
     H1,H2,H3
```

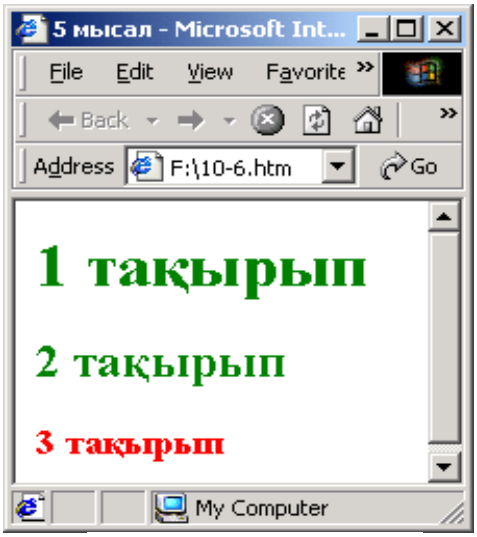
```
      {
        color: blue;
      }
    -->
  </STYLE>
<TITLE> 4 мысал </TITLE>
</HEAD>
<BODY bgcolor=white text=black>
   <H1> 1 тақырып </H1>
   <H2> 2 тақырып </H2>
   <H3 style="color:red">
       3 тақырып </H3>
</BODY>
</HTML>
```



3.5-сурет. Программа нәтижесі

<style> және <link> тәгтерін басқаша тәртіппен орналастырайық.

```
<HTML>
 <HEAD>
   <STYLE type="text/css">
   <!--
     H1,H2,H3
     {
       color: blue;
     }
   -->
   </STYLE>
   <!-- prim.css файлын іске қосу -->
   <LINK rel=stylesheet type="text/css"
        href=prim.css>
   <TITLE> 5 мысал </TITLE>
 </HEAD>
   <BODY bgcolor=white text=black>
     <H1> 1 тақырып </H1>
     <H2> 2 тақырып </H2>
     <H3 style="color:red"> 3 тақырып </H3>
   </BODY>
</HTML>
```



3.6-сурет. Программа нәтижесі

Сонымен, CSS арқылы мыналар атқарылады:

· өрістер, шегіністер, қаріп мөлшері (көлемі) және типі, мәтін түсі мен фоны, т.б. парақтың жекелеген элементтері (азат жолтар, сөздер, әріптер) үшін беріледі.

· жүздеген файлдардан тұратын толық сайт үшін оның безендірілуін HTML-кодқа тимей, тек бір ғана CSS файлын түзету арқылы өзгерту;

· HTML-құжаттың ішкі тәгтері санын азайтып, оның ішкі ақпараттық мазмұнын браузер экранының сыртқы түсінен бөліп жеке стильдер түрінде жазып шығу.

3.4 CSS қасиеттеріне шолу

Стильдік әр түрлі анықтаулар жазуда 70-тен аса оның түрлі қасиеттерін көрсетуге болады. Жалпы жұмыс атқару кезінде кітаптар соңында берілетін *анықтамаларды* пайдалану қажет.

Стильдегі оның қасиеттерін мынадай топтарға бөліп беру қалыптасқан:

- *қаріп (шрифт);*
- *түстер;*
- *мәтін;*
- *өрістер мен жақтаулар (поля и рамки);*
- *сыртқы түрлері.*

Пункттер мен пикалар – типографиялық өлшем бірліктері, олар қаріп көлемін немесе оның кеглін береді.

Word-та, мысалы, бұл параметр 8-ден 72 пт-ке дейін. Бұл кегльдің пунктпен берілген мөлшері. 1 типографиялық пт = 1/72дюйм= 0,375 мм. Бұл символдың өз мөлшері емес, оның “ұпай” («очко») мөлшері, яғни типографиядағы сол символды (литер) ойып орналастыратын матрица биіктігі. Оның мөлшері литер мөлшерінен үлкен. Мысалы, кеглі 10 шрифтің бас әрпінің көлемі 7 пункт шамасында болады.

Кітап шығаруда – мәтін көлемі 10 немесе 12 пт болады. Тақырыптар үшін – үлкенірек кегль, ал сілтемелер мен ескертпелер үшін кішірек (әдетте 8 пт) мәндері қолданылады.

Пика – үлкенірек өлшем бірлігі. 1 пика = 12 пт.

Пайыздық өлшем (процентный отсчет) негізгі мөлшерге байланысты болады. Шрифтер үшін – ағымдағы мән негізге алынады. Пайызды пайдалану ыңғайлы болып саналады. Егер программалаушы браузердегі шрифт көлемін стильдерде пайызбен беріп отырса, олар пропорционал түрде өзгереді.

font-family қаріп түрі (шрифт типі)

Бұл стильдік қасиет шрифтің гарнитурасы атын (мысалы, Arial) немесе оның топ атауын (родовое имя) көрсетеді:

serif – шығыңқы шрифт (с засечками – серифный);

sans-serif – жұмыр шрифт (без засечек – рубленый);

monospace – ені бірдей қаріп (моноширинный шрифт – символдарының ендері бірдей).

Гарнитура – бір қаріп символдарының сызылымдары жиыны.

Бұл – тіке және курсивтік сызылым, қарайтылуы (жирность), литер ені (қысыңқы, қалыпты, созылмалы) және кеглі әр түрлі болуы мүмкін деген сөз.

Шрифтер серифтік (шығыңқы - с засечками) и жұмыр (рубленые - без засечек) болады. Шығыңқы шрифт —Times гарнитурасы, жұмыр шрифт — Helvetica немесе Arial гарнитурасы.

Серифтік шрифтер жеңіл оқылады. Төменгі сериф көзге жылы көрінеді.
Жұмыр шрифтер тек тақырыптар үшін қажет.
Серифтік шрифт негізгі мәтін үшін, ал жұмыр — тақырыптар немесе майда ескертпелер үшін керек.
Тағы екі топ: *пропорционал және ені бірдей (моноширинные)* шрифтер бар.
Моноенді шрифтер баспа машинкасындай етіп басады. Олар мынадай тәгтерден пайда болады: <PRE>, <CODE>, <TT>, <SAMP>, <KBD>.
Кәдімгі мәтін пропорционал шрифтпен басылады. Оларда әр түрлі символдар ені әр түрлі, l – енсіз, ж – енді.
Пропорционал шрифт моноенді шрифтке қарағанда жеңіл оқылады. Пропорционал шрифт негізгі мәтін үшін, ал моноенді шрифт — программалар үшін қажет болып саналады
Times және Helvetica гарнитуралары — пропорционал.
Courier гарнитурасы - моноенді шрифт .
Кез келген графикалық операциялық жүйеде үш стандартты гарнитура бар, Windows үшін:
Times Roman – серифтік шрифт;
Arial – жұмыр шрифт;
Courier– моноенді шрифт.

Үнсіз келісім бойынша (по умолчанию) браузер қалыпты мәтін үшін Times Roman шрифтін, ал программалар үшін – Courier шрифтін (моноенді) пайдаланады. Сайт үшін өз гарнитураңды көрсету қажет емес, ондай шрифт компьютерде болмаса, ол иероглифке айналып кетеді. Жергілікті (локальды) гипермәтін үшін кез келген шрифт жарайды. Ал Интернет желісі үшін тек топты ғана көрсету жеткілікті шығар немесе шрифті қоса жіберіп, оны орнату ережесі де бірге берілуі тиіс. font-family қасиеті үшін бір емес, бірнеше шрифт көрсеткен абзал (үтір арқылы бөліп), мысалы:

```
H1,H2,H3,H4,H5,H6
{
  font-family: "Arial Cyr",  Geneva,  Helvetica,  sans-serif;
}
```

Браузер алдымен Arial Cyr шрифтін іздейді, таба алмаса - Geneva, соңынан - Helvetica, ешқайсысы да болмаса, әйтеуір бір жұмыр шрифт табады (sans-serif).
Егер шрифт аты бірнеше сөзден тұрса, ол міндетті түрде қос тырнақшаға алынады.
font-size
font-size қасиеті шрифтің абсолюттік немесе салыстырмалы мөлшерін береді. Салыстырмалы түрде бергенде, пайыздық өлшем қолданылады (ағымдағы шрифт - негізгі) немесе мынадай түйінді сөздер қою керек:
larger – ірілеу (крупнее);
smaller – шағындаум(мельче).

Абсолюттік мөлшерде келесі өлшем бірліктерінің бірін көрсету керек: in, cm, mm, px, pt, pc.

Түс

Стильдік қасиеттер түстерді тағайындаудың үш түрлі тәсілін береді:

· түйінді сөз, мысалы, white;
· он алтылық RGB-код, мысалы,
#eee5d8;
· ондық RGB-код, мысалы,
rgb(255,64,0)

color

Элемент түсін анықтайды.

background-color

Элемент шығарылатын маңайдағы фон түсін анықтайды.

мысалы, стиль үшін мынадай бөлік енгізуге болады:

```
<HTML>
  <HEAD>
    <!—CSS қасиеттерін қосу -->
    <STYLE type="text/css">
    <!--
      P
       {
         font-family: "Arial Cyr",Helvetica,sans-serif;
         font-size: 0.5cm;
         color: blue;
         background-color: yellow; }
    -->
    </STYLE>
  <TITLE>4 Мысал</TITLE>
  </HEAD>
```

Мәтін үшін letter-spacing

Әріптер арасындағы аралықты қосымша анықтау үшін қажет. *Normal* түйінді сөзін немесе нақты мән беруге болады, минус таңбалы мән де болады (символдар бірінің үстіне бірі жазылады). Пайыздық тәсіл қолданылмайды.

line-height

Жолдар аралығын тағайындайды (интерлиньяж). Абсолюттік биіктікті берсе де болады (14pt), жол аралығын (1.4) немесе жол биіктігінің пайызымен де (200%) көрсетіледі. Мысалы, жоларалық екі интервал:

line-height: 2

немесе line-height: 200%. normal сөзімен де берсе болады.

text-align

Мәтінді көлденең туралау тәсілін көрсетеді.

Келесі мәндер қолданылады:

left – сол жақ шетке туралау;

right – оң жақ шетке туралау;
center – ортаға туралау;
justify – екі шетін де туралау (по ширине).

Соңғы мәнді justify қолдану қиын, өйткені ұзын сөзді тасымалдауды браузер білмейді (мысалы, орысша немесе қазақша мәтіндерді).
Өрістер мен жақтаулар
border-style
Элементті қоршап тұратын жақтау сызықтардың түрін анықтайды. Мынадай мәндер бар:
none – жақтау жоқ (үнсіз келісім бойынша);
solid – кәдімгі тұтас сызық;
double – екі сызық;
groove – «қысылған» шекара;
ridge – «томпақ» шекара;
inset – «қысылған» элемент;
outset –«томпақ» элемент.

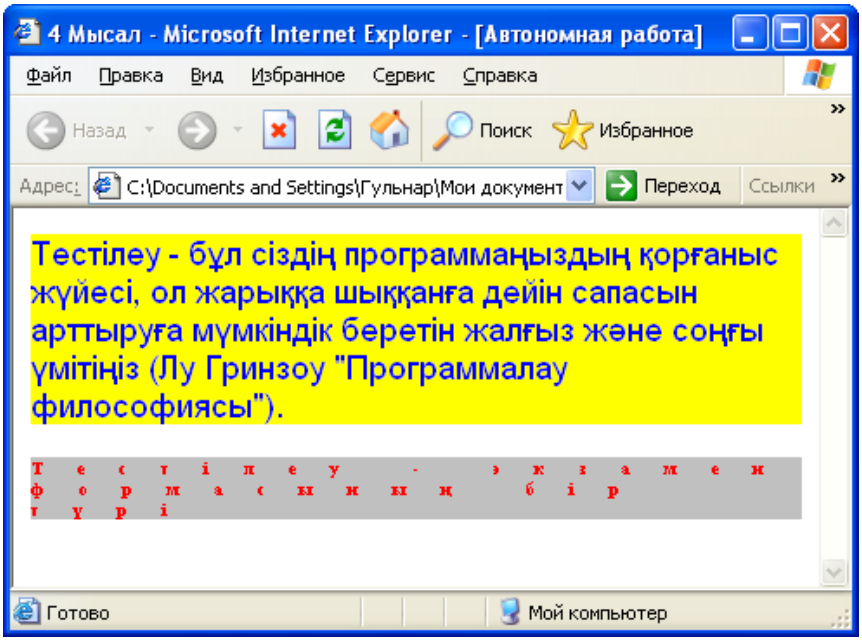
border-color
Жақтау сызығының түсін анықтайды. Бұл қасиет border-style берілсе ғана жұмыс істейді.
border-width
Жақтау сызығының қалыңдығын береді. Бұл қасиет те border-style берілсе ғана жұмыс істейді. Қалыңдығын сан бірліктерімен (пайыздық жазу жоқ) немесе келесі сөздермен беруге болады:
thin – жіңішке;
medium– орташа;
thick – жуан .
margin
Элементтегі блок шеттерінде бос қалатын өріс енін береді. Санмен, пайызбен (блок ені негізінде) немесе түйінді сөзбен – auto беріледі. Мәні Auto болса, тиімді мәнді браузер өзі анықтайды.

padding
Элемент пен жақтау арасындағы қашықтықты анықтайды. Санмен немесе пайызбен (от ширины элемента) беріледі.
Бір программа бөлігін толық келтірейік:

```
<HTML>
  <HEAD>
     <!—CSS қасиеттерін қосу -->
     <STYLE type="text/css">
     <!--
       P
        {
```

```
            font-family: "Arial Cyr",Helvetica,sans-serif;
            font-size: 0.5cm;
            color: blue;
            background-color: yellow; }
     -->
     </STYLE>
  <TITLE>4 Мысал</TITLE>
  </HEAD>
<BODY bgcolor=white text=black>
    <P> Тестілеу – бұл  сіздің программаңыздың қорғаныс жүйесі, ол жарыққа шыққанға дейін сапасын арттыруға мүмкіндік беретін жалғыз және соңғы үмітіңіз (Лу Гринзоу    "Программалау философиясы").
    <H6 style="color: red;background-color: silver;
                letter-spacing: 20px">
       Тестілеу – бұл емтихан формасының бір түрі
    </H6>
 </BODY>
</HTML>
```

Бұл программа нәтижесі келесі бетте орналасқан.



3.7-сурет. Программа нәтижесі

Бақылау сұрақтары:

1. CSS қысқартылған сөзі қалай жіктеледі?
2. CSS технологиясының атқаратын қызметін сипаттаңыз.
3. Тәгте стильді қандай атрибут арқылы беруге болады?
4. Төменде келтірілген жазулардың қайсысы азат жолтағы мәтіннің түсін қызыл етіп көрсетеді?

    а) <P color = red>;

б) <P color = #FF0000>;
в) <P color = rgb(255,0,0)>;
г) <P style ="color: red">;
д) <P style ="color: #FF0000">;
е) <P style="color:rgb(255,0,0)">.

5. Қандай арнайы HTML-блогында <?????>...</?????> стильдік сипаттамалар жазылады?
6. Төменде келтірілген стильдік сипаттамалардың қайсысы қатесіз жазылған?

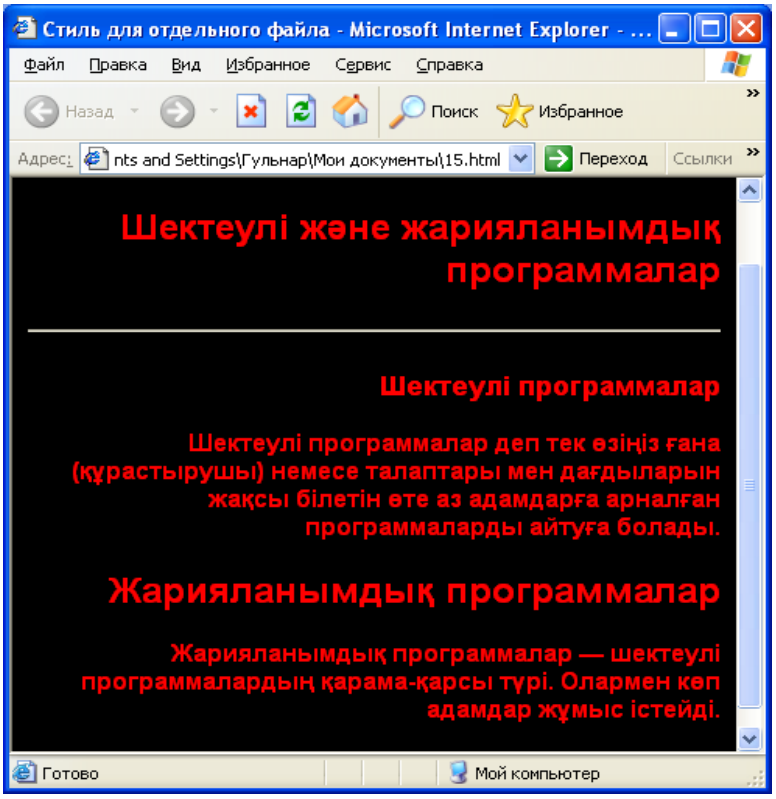
а) P {color = red; font-size=l cm};
б) P {color = red, font-size=l cm};
в) P {color:red font-size:1 cm};
г) P {color:red, font-size: 1cm};
д) P {color:red; font-size:1cm}.

7. Қандай тәгте файлға стильдік анықтаулармен сілтеме жазылады, ол қандай атрибутпен беріледі? Бұл атауларды сұрау белгілерінің орнына жазыңыз:
<???? rel = stylesheet type="text/css" ????=file.css>
8. Стильдік сипаттамалар берілуінің қандай тәсілдерін білесіз? Осы үш тәсілді салыстырыңыз, олардың әрқайсысының атқаратын қызметін көрсетіңіз.

Тапсырмалар

1. Жұмыстың мысалдарын мұқият меңгерген соң келесі сұрақтарға жауап беріңіз:
   - Қандай CSS-нұсқаулар басымдау: жеке тәгте жазылған ба, әлде HTML-программаның <HEAD> бөлімінде орналасқан ба?
   - Қандай нұсқаулар басымдау: HTML-программаның <HEAD> бөлімінде жазылған ба, әлде жеке CSS-файлында орналасқан және құжатпен <LINK> тәгі арқылы байланысқан ба? Нәтиже осы жазулардың орналасу ретіне тәуелді бола ма?

2. Стильдерді қолдана отырып, 3.5-суретте көрсетілген құжатты жоғарыда көрсетілген үш тәсіл арқылы дайындаңыз. Бұл құжатта:
   - Негізгі түстер: ақ фонда қара түс (оларды өз қалауыңыз бойынша әр тәсілді қолданғанда әр түрлі етіп өзгертіңіз);
   - Парақтағы барлық элементтердің оң жақтан және сол жақтан шегінісі 2 см;
   - Азат жолдар оң жақ шет бойынша тураланған;
   - Тақырыптар оң жақ бойынша тураланып, қызыл, рубленый қаріппен жазылған;
   - Қисайтылған форматтағы терминдер жасыл түспен жазылған.



3.8-сурет. Тапсырманы орындауға арналған сурет

```
BODY {margin-left:60px;}
Hl,H2,H3,H4,H5,H6
{
 text-align: right;
 color: red;
 font-family: "Palatino Linotype", Baltica, Times New Roman;
}


<HTML>
 <HEAD>
  <META http-equiv="Content-Type"
        content="text/html; charset=windows-1251">
  <TITLE>Стиль  для отдельного файла</TITLE>
  <STYLE type="text/css">
   <!--
    H1,H2,H3,H4
    {
     text-align: right;
     color:red;
     font-family: "Arial Cyr",Geneva,Helvetica,sans-serif;
    }
   -->
```

```
   </STYLE>
  </HEAD>
  <BODY bgcolor=#000000 text=black>
<H2> Шектеулі және жарияланымдық программалар </H2>
   <HR>
<H3> Шектеулі программалар </H3>
<h4> Шектеулі программалар деп тек өзіңіз ғана (құрастырушы) немесе
талаптары мен дағдыларын жақсы білетін
өте аз адамдарға арналған программаларды айтуға болады. </h4>
<H2> Жарияланымдық программалар </H2>
<h4> Жарияланымдық программалар — шектеулі программалардың қарама-
қарсы түрі. Олармен көп адамдар жұмыс істейді.
 <h4>
  </BODY
</HTML>
```

3.5 CSS құру негіздері

Стильдің мұра ретінде берілуі

HTML-кодтың құрылымы сатылы түрде болады. Барлық құжат <body>. . .</body> тәгтері ішінде орналасады. Ал <P> азат жолы ішінде <B> тәгімен белгіленген элементтер болуы мүмкін.

Стильдің мұра ретінде берілуі <body> тәгі үшін анықталған стильдің құжаттағы барлық тәгтерге әсер ететіндігін білдіреді. Сәйкесінше егер бір тәг үшін стиль құрылса, онда сол тәг ішіндегі басқа тәгтерге де осы стиль міндетті түрде беріледі.

Мысалы, <P> тәгі үшін келесі стиль берілген болсын:

```
P {color: red;
font-size: 14pt;
font-family: Arial,sans-serif}
```

Онда осы азат жол ішіне орналасқан <EM> тәгінің элементтері де экранда қызыл түспен, 14 пункт көлемде жұмыр қаріппен шығарылады.

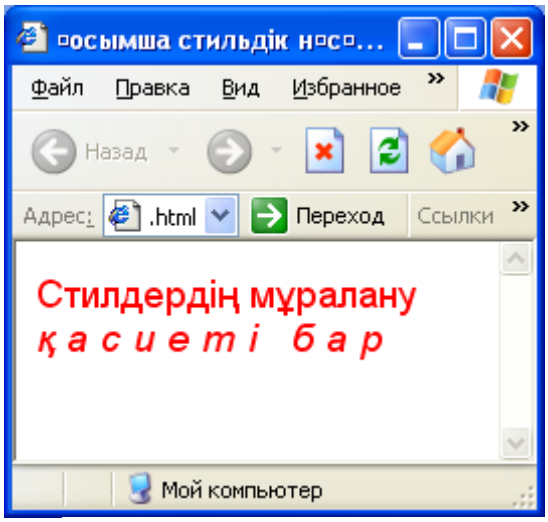
Мысалы:

```
<P>
Стильдік анықтаулардың <EM>мұралану</EM> қасиеті болады.
```

Егер <EM> тәгі үшін мәтін символдары арасын ашып жазу қажет болса, онда мұндай қасиет қосымша беріледі:

```
<HTML>
  <HEAD>
   <TITLE>Қосымша стильдік нұсқаулар
   </TITLE>
    <STYLE type="text/css">
         <!--
      P {color:red;font-size:14pt;
           font-family:Arial,sans-serif}
```

```
        -->
   </STYLE>
  </HEAD>
<BODY bgcolor=white text=black>
    <P>
      Стилдердің мұралану
      <EM style="letter-spacing: 6pt:">
         қасиеті бар
      </EM>
  </BODY>
</HTML>
```



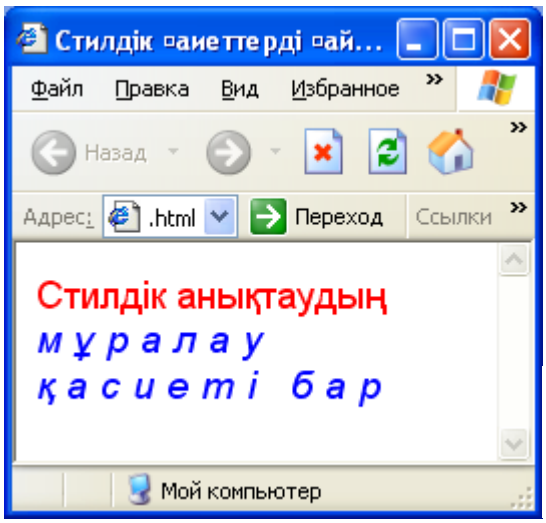
3.9-сурет. Программа нәтижесі

<EM> тәгінің ішкі мәтіні <P> тәгінің мәтіндері сияқты шығады (14 пункт, қызыл түсті, жұмыр қаріп), ал style="letter-spacing: 6pt" қосымша анықтаудың әсерінен сөздегі әріптердің арасы 6 пункт болып ашылып орналасады.

Сонымен ішкі қосымша тәгте жаңа стильдік анықтаулар кіргізіп қана қоймай, оның ата тегі болып саналатын сыртқы тәгтің де қасиеттерін өзгертуге болады.

Мысалы, келесі мәтіндегі “мұралау қасиеті бар” сөзі көк түске боялып тұрады.

```
<HTML>
 <HEAD>
  <TITLE>Стилдік қасиеттерді қайта анықтау
  </TITLE>
       <STYLE type="text/css">
   <!--
     P {color: red;font-size:14pt;
          font-family:Arial,sans-serif}
       -->
       </STYLE>
</HEAD>
<BODY bgcolor=white text=black>
 <P>
       Стилдік анықтаудың
       <EM style="letter-spacing:6pt;
     color:blue">мұралау қасиеті бар</EM>
</BODY>
</HTML>
```

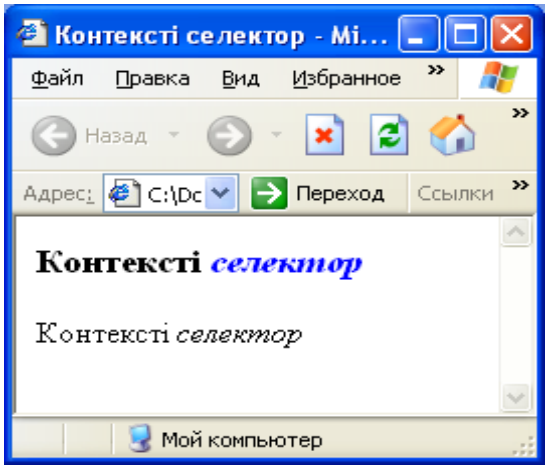


3.10-сурет. Программа нәтижесі

Контекстік селекторлар

Стильдік анықтауларды, тәгтердің кірістіріліп орналасу реттілігіне сәйкес орындалатындай етіп жазуға болады. Мысалы, <EM> тәгіндегі мәтін түсі ол <H3> тәгінің ішінде орналасқан жағдайда ғана көк болатындай етіп орналастырайық:

```
<HTML>
 <HEAD>
       <TITLE>Контексті селектор</TITLE>
       <STYLE type="text/css">
        <!--  H3 EM {color: blue}
        -->
       </STYLE>
 </HEAD>
 <BODY bgcolor=white text=black>
   <H3> Контексті <EM>селектор
</EM></H3>
   <P> Контексті <EM>селектор
</EM></P>
 </BODY>

</HTML>
```



3.11-сурет. Программа нәтижесі

Экранға «селектор» сөзі бірінші қатарда көк түспен (<H3> тәгінің ішінде), ал екінші қатарда қара түспен (<P> тәгінің ішінде) шығады. Стильдік анықтауларда үтір қойылмағандығына назар аударыңыз. Бұл контекстік анықтаулардың белгісі. Егер H3, EM { color: blue } түрінде жазсақ, онда <H3> тәгімен қатар <EM> тәгі де көк түске боялады, яғни үтір бірнеше тәгтер тобы үшін бірдей болатын ортақ стильді анықтайды.

Кластар

Стильдік анықтауларды тәгтерді көрсетпей-ақ жазуға болады. Мұндай жағдайда, берілген стилді нақты тәгке сәйкестендіру үшін, әрбір анықтауға арнайы атау меншіктелуі тиіс. Мұндай стильдік анықтаулар кластар деп аталады. Класс келесі түрде жазылады:

```
.аты
{ сипаттама: мәні;
  . . .
 сипаттама: мәні;
}
```

Басқаша айтқанда, анықтаулар әдеттегідей жазылады, бірақ тәгтердің аттары орнына .аты конструкциясы қолданылады.

Мысалы, def атты стильдік класс жазып шығайық:

```
.def
{color: red; font-size: 16pt;
font-family: Geneva, Helvetica, sans-serif;
border: solid 0.2cm blue }
```

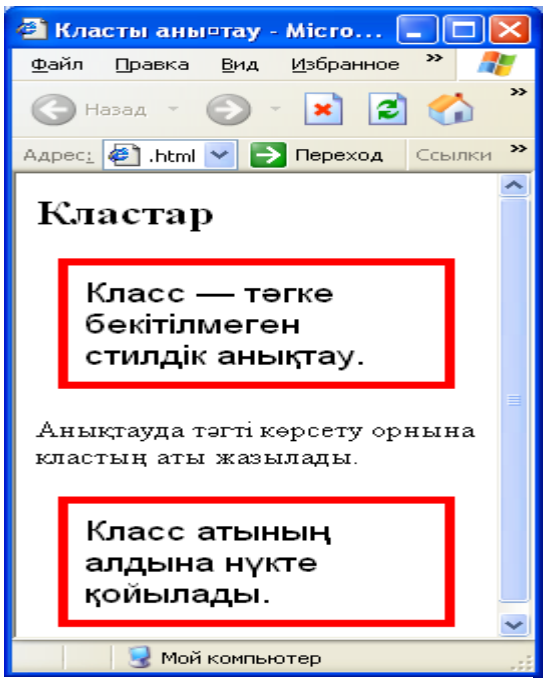
Мұнда border: solid 0.2cm blue қатары бір стиль-дік нұсқауда бірнеше параметрлердің қызметін жазуды көрсетеді (мұнда: стиль, жақтау қалыңдығы, және оның түсі). Мұндай біріктіруді border сияқты жалпыланды-рылған арнайы стильдік қасиеттерге қолдануға болады.

Стильдік класты тәгпен сәйкестендіру үшін class атрибуты қолданылады:

```
<P class=def > текст </P>
```

Бұл кодтың қалай жұмыс істейтінін қарастырайық.

```
<HTML>
  <HEAD>
   <TITLE>Класты анықтау</TITLE>
  <STYLE type="text/css">
  .def
   {font-family:Helvetica;font-size:14pt;
    border: solid 4pt red;padding: 6pt;
    margin-left:5%;margin-right:5% }
 </STYLE>
</HEAD>
 <BODY bgcolor=white text=black>
  <H2>Кластар</H2>
  <P class=def>
  Класс &#151; тәгке бекітілмеген стильді анықтау.
  <P> Анықтауда тәгті көрсету орнына кластың аты жазылады.
   <P class=def>
    Класс атының алдына нүкте қойылады.
 </BODY>
</HTML>
```

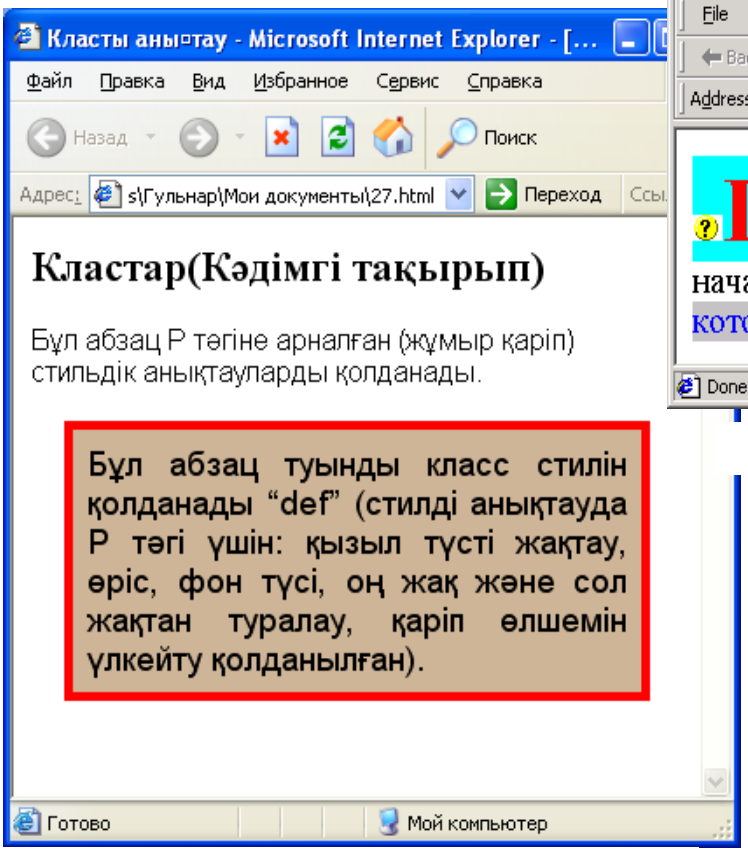


3.12-сурет. Класс стилін қолдану

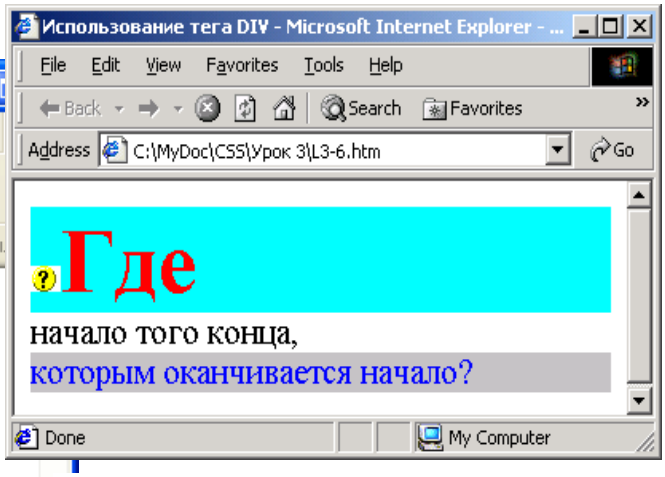
Құрылған стильдік анықтаулар негізінде кластар құруға болады. Келесі мысалда def класының негізі ретінде <P> тәгі үшін анықталған стиль алынып, оған қосымша жаңа қасиеттер қосылып жазылған:

```
<HTML>
<HEAD>
  <TITLE>Класты анықтау</TITLE>
  <STYLE type="text/css">
    P {font-family:Helvetica}
    P.def
      {text-align: justify;
            background:#CFB597;
            font-size:14pt;
       border:solid 4pt red;
       padding:6pt;
       margin-left:5%;margin-right:5%}
  </STYLE>
 </HEAD>
<BODY bgcolor=white text=black>
  <H2> Кластар(Кәдімгі тақырып)</H2>
  <P> Бұл азат жол P тәгіне арналған (жұмыр қаріп) стильдік анықтауларды қолданады.
  <P class=def>
Бұл азат жол туынды класс стилін қолданады
&#147;def&#148;
(стилді анықтауда P тәгі үшін: қызыл түсті жақтау, өріс, фон түсі, оң жақ және сол жақтан туралау, қаріп өлшемін үлкейту қолданылған).
```

</BODY>
</HTML>



3.13-сурет. Туынды класс стилін қолдану мысалы



3.14-сурет. Құжат бөлігін ерекшелеу

<DIV> және <SPAN> тәгтері

Бұл тәгтер CSS үшін маңызды болып табылады. Олар құжаттағы жеке бір бөліктерді ерекшелеп алып, оларға арнайы қасиеттер беру ісін атқарады. Ол үшін керекті элементтерді <DIV>. . .</DIV> немесе <SPAN>. . .</SPAN> тәгтері ішіне орналастыру керек.

Бұлардың айырмашылығы мынада: <DIV> блогынан соң браузер жаңа жолға көшіреді, ал <SPAN> блогынан кейін – бұрынғы жолда қала береміз. Сонымен, <SPAN> тәгін пайдалану - бір жолдағы сөздерге не символдарға жеке стильдік қасиеттер тағайындай алады. Осыларға мысал келтірейік.

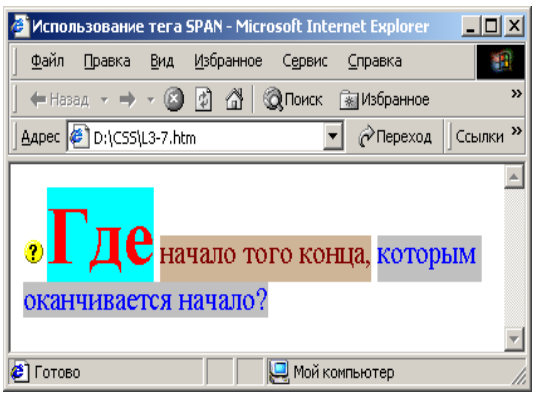
<DIV> тәгін пайдалану:

```
<HTML>
 <HEAD>
        <TITLE> DIV тәгін пайдалану </TITLE>
        <STYLE type="text/css">
   .area1
         { color:red; font-weight:bolder;
           font-size:40pt; background:aqua}
   .area2
```

```
 { color:black; background:#CFB597}
   .area3
          { color:blue;background:#C0C0C0}
        </STYLE>
 </HEAD>
<BODY bgcolor=white text=black>
  <DIV class=area1><IMG src=vopros.gif> Где</DIV>
  <DIV class=area2>начало того конца,</DIV>
  <DIV class=area3>которым оканчивается начало?
  </DIV>
</BODY>
</HTML>
```

<SPAN> тәгін пайдалану мысалы:

```
<HTML>
 <HEAD>
        <TITLE> Использование тега SPAN</TITLE>
        <STYLE type="text/css">
   .area1
          { color:red; font-weight:bolder;
            font-size:40pt; background:aqua}
   .area2
          { color:maroon; back-
ground:#CFB597;
      padding:6pt}
   .area3
          { color:blue;background:#C0C0C0;
      padding:6pt}
        </STYLE>
 </HEAD>
<BODY bgcolor=white text=black>
 <SPAN class=area1><IMG src=vopros.gif>Где</SPAN>
 <SPAN class=area2>начало того конца,</SPAN>
 <SPAN class=area3>которым оканчивается начало?
 </SPAN>
</BODY>
</HTML>
```



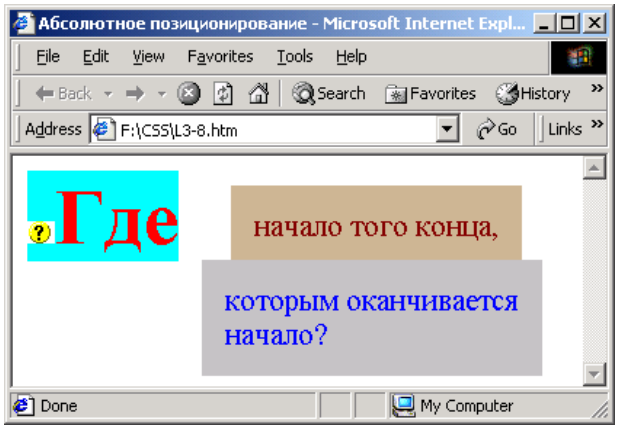
3.15-сурет. Құжат бөліктері стилі

Абсолюттік түрде орналастыру

CSS арқылы экрандағы элементтерді браузер терезесінің сол жақ жоғары бұрышынан бастап нақты координаталар бойынша орналастыра аламыз. Мұндай мүмкіндікке position стильдік қасиетінің absolute мәні арқылы қол жеткізуге болады. Координаталар пиксель бірлігімен left (горизонталь координата) және top (вертикаль координата) сөздері арқылы тағайындалады.

Енді мысалдар қарастырамыз.

```
<HTML>
 <HEAD>
  <TITLE> Абсолютное позиционирование</TITLE><STYLE type="text/css">
    .area1
          { position:absolute; top:10; left:10;
            color:red; font-weight:bolder;
            font-size:40pt; back-
ground:aqua}
          .area2
          { position:absolute; top:20;
left:150;
       color:maroon; back-
ground:#CFB597;
       padding:12pt}
          .area3
          { position:absolute; top:70;
left:130;
            color:blue; background:#C0C0C0;
            padding:12pt}
        </STYLE>
  </HEAD>
<BODY bgcolor = white text = black>
 <DIV class =area1><IMG src=vopros.gif>Где</DIV>
 <DIV class =area2>начало того конца,</DIV>
 <DIV class =area3>которым оканчивается начало?
 </DIV>
</BODY>
</HTML>
```



3.16-сурет. Құжат бөліктерін координаталар бойынша орналастыру

Мұнда браузер үш <DIV> тәгімен берілген элементтерді көрсетілген координаталар бойынша орналастырып шықты. Элементтер бірінің үстіне бірі шығып тұрғанын байқаған шығарсыңдар.

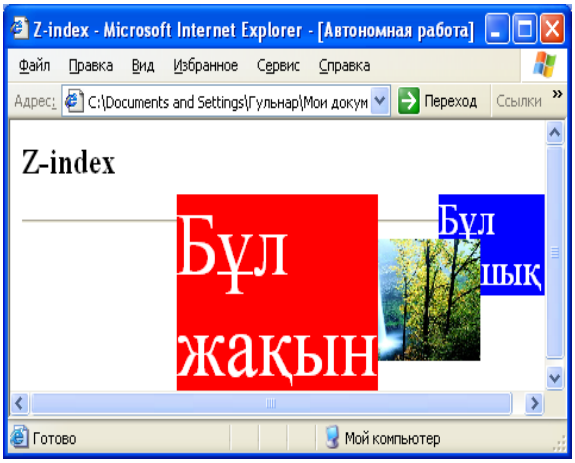
Ең үстіне, яғни бізге жақын түрде HTML-кодта ең соңғы көрсетілген элемент шығады. Егер <DIV> тәгтерінің орнын ауыстырсақ, онда элементтердің қабаттаса орналасу реттілігі де өзгеріске ұшырайды. Дегенмен, CSS элементтерді қабаттастыра орналасу реттілігін жылдам өзгертетін тағы бір мүмкіндікті - Z-index тәсілін де ұсынады.

Z-index

Бұл стильдік қасиет экрандағы элементтің қай деңгейде (қай қабатта) орналасуы керек екендігін анықтайды. Мұндағы негізгі деңгейдің нөмірі (элементтер стильдік анықтаусыз-ақ шығатын деңгей) нөлге тең болып саналады. Сәйкесінше, экран қабатын анықтайтын теріс Z-index мәні бар элементтер төменде (алыста), ал оң мәнді элементтер — жоғарыда (жақын) орналасады. Егер элементтердің Z-index мәндері бірдей болса, олар бір қабатта орналасады. Сонымен Z-index мәні үлкен элемент жоғары деңгейде орналасады екен.

Енді мысал қарастырайық.

```
<HTML>
 <HEAD>
       <TITLE> Z-index </TITLE>
 </HEAD>
 <BODY bgcolor=white text=black>
       <H2>Z-index</H2> <HR>
  <DIV style= "position:absolute; top:50;
      left:140; height:130; width:100;
      background:red; font-size:60;
      color:white; z-index:3">
         Бұл жақын </DIV>
       <DIV style="position:absolute; top:50;
      left:360; height:30; width:100;
      background:blue; font-size:30;
      color:white; z-index:1">
     Бұл қашық</DIV>
       <DIV style="position:absolute; top:80;
      left:270; width:125; z-index:2">
        <IMG src=11494[2].jpg width=125 height=82
       alt =""">
   </DIV>
 </BODY>
 </HTML>
```



3.17-сурет. Деңгей бойынша орналасуы

Сатылай орналастыру (каскадирование)

CSS көмегімен тұтынушылар экрандағы ақпаратты ыңғайлы түрде бейнелейтін мүмкіндік алады.

Тек жасалған стильдердің бір-бірімен өзара байланысын және тәгтер атрибуттарының жазылуын (олар да стильді анықтайды) браузердің өз стилімен (үнсіз келісім бойынша анықталатын стиль) шатастырып алмау керек.

Жоғарыда айтылғандай, программалаушы HTML-құжатта үш түрлі стиль пайдалана алады:

- Құрамдас (встроенный - inline). Style атрибуты арқылы тәг ішінде жазылған стиль, ол жеке тәг жұмысын бақылайды.
- Енгізілген (внедренный - embedded). HTML-файл тақырыбында <style>…</style> тәгтері арқылы жа-зылған стиль, ол жеке HTML-құжаттағы стильді бақылайды.
- Байланысқан (связанный - linked). Жеке CSS-файлда жазылған стиль.

Ол көптеген HTML-құжаттар стилін бақылай алады. Мұнда стильдік файлға сілтеме жасау үшіни HTML-құжаттың бас жағына <link> тәгі жазылады.

Бір құжатта осы айтылған үш стильдер механизмін де қолдануға болады. Бұған тәгтердің кәдімгі атрибуттары арқылы берілетін стильді және браузердің "келісім бойынша" тағайындалған өз стилін қосайық. Бұлар бір-бірімен қалай әрекеттесер екен, компьютер экранына құжатты шығарарда браузер қай стильді пайдаланады?

Құжатты шығарарда кәдімгі браузер ұстанатын ережелер *каскадты,* яғни *сатылы* деп аталады. *Бұл* браузер үшін ең негізгі стиль – құрамдас стиль деген сөз, одан кейін төмен қарай реті бойынша енгізілген және байланысқан стильдер қолданылады.

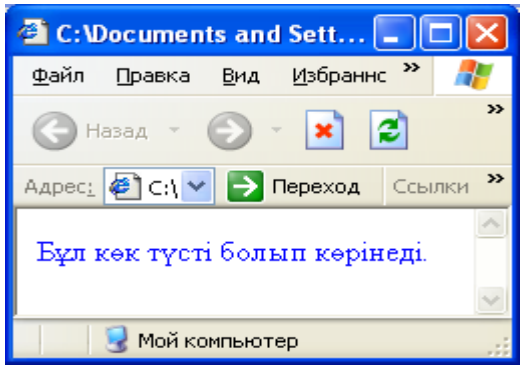
Сонымен браузер үшін енгізілген және байланысқан стильдер бірдей болып саналады, ол тек соңғы қолданылуға байланысты өзгереді.

Стильдік реттілік бойынша ең төменгі сатыда «үнсіз келісім бойынша» стиль тұрады. Оны браузер ешқандай стильдік нұсқау болмаған жағдайда қолданады. «Каскадтау - сатылау» ұғымына арғы тегінен берілу – мұралау (наследования) механизмі де жатады, ол бойынша өз стилі жоқ бала-тәг ата-тәгтің стилін қабылдайды.

Кәдімгі атрибуттар арқылы берілген стильдер осы сатылау ережесі бойынша жұмыс істейді.

1 -мысал.

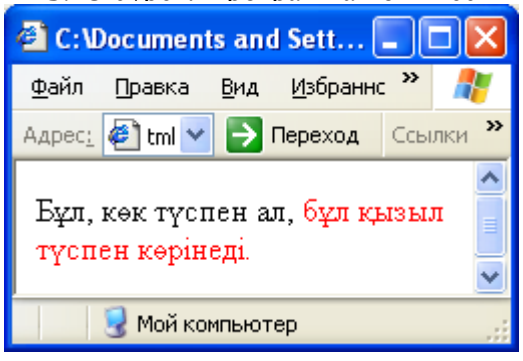
```
<FONT color=blye>
 <P style="color:red">
  Бұл көк түсті болып көрінеді.
 </P>
</FONT>
```



3.18-сурет. Программа нәтижесі

2- мысал.

```
<P style="color:blue">
```



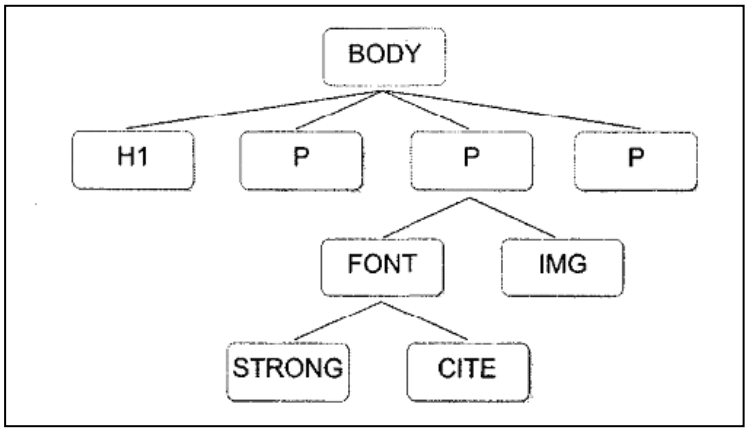
3.19-сурет. Программа нәтижесі

Бұл, көк түспен,ал
<FONT color=red>
бұл қызыл түспен көрінеді.
</FONT> </P>

Бақылау сұрақтары

1. «Иерархия» деген не?
2. Төменде келтірілген әрбір схемаға (3.20 және 3.21-суреттер) сәйкес келетін HTML-кодтарын жазыңыздар.

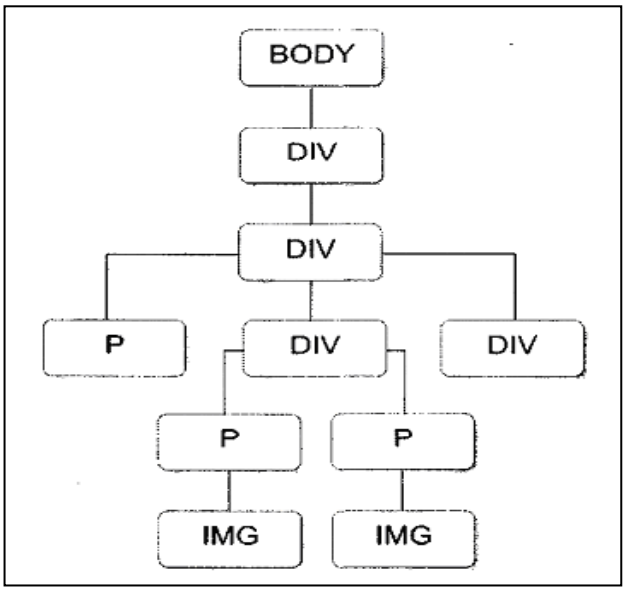
3.20-сурет. 1-схема



3. Стильдердің мұралану механизмін сипаттаңыздар.
4. Мұраланатын стильді алдын ала анықтау мүмкін бе? Егер мүмкін болса қалай?
5. <P> тәгі үшін **P** {color:red} стилі берілген. Келесі кодтарға сәйкес «күрделі» сөзі қандай түспен шығарылады:

а)<H1>Бұл күрделі мысал</H1>;
б)<H1>Это <SPAN style="Color:Blue">күрделі</SPAN> мысал.</H1>;
в)<P>Бұл күрделі мысал.</P>;
г)<P>Бұл<SPAN style="color:blue">күрделі</SPAN> мысал.</P>;
д)<P>Бұл<FONT color= blue>күрделі</SPAN> мысал.</P>;
е)<P>Это<SPAN style =”font-size:smaller”>күрделі</SPAN>мысал.</P>.

6. «Контекстік анықтау» деген не?
7. P EM {color:red} стилі берілген. Келесі кодтарға сәйкес «күрделі» сөзі қандай түспен шығарылады:
   а) <H1>Бұл күрделі мысал</H1>;
   б) <H1>Бұл <EM>күрделі </EM> мысал</H1>;
   в) <P>Бұл күрделі мысал.</P>;
   г) <P>Бұл <EM>күрделі</EM> мысал</P>?



3.21-сурет. 2-схема

8. Төменде көрсетілген әрбір программаның бұтақ тәріздес иерархиясын құрыңыздар.

1 Программа	2 Программа
<BODY> <DIV> … </DIV> <DIV> <DIV> <DIV> <DIV> … </DIV> </DIV> </DIV> <DIV> … </DIV> </DIV> <DIV> … </DIV> </BODY>	<BODY> <DIV> <DIV> … </DIV> <DIV> … <DIV> </DIV> <DIV> … </DIV> </DIV> <DIV> <DIV> … </DIV> </DIV> </DIV> </BODY>

9. P, EM {color:red} стилі берілген. Келесі кодтарға сәйкес «күрделі» сөзі қандай түспен шығарылады?:

а) <H1>Бұл күрделі мысал</H1>;
б) <H1>Бұл <EM>күрделі</EM> мысал</H1>;
в) <P>Бұл күрделі мысал.</P>;
г) <P>Бұл <EM>күрделі</EM> мысал</P>.

10. және тәгтерінің атқаратын қызметтерін сипаттаңыздар. Бұл тәгтерінің нәтижелерінің ұқсастықтары мен айырмашылықтары неде?

11.«Абсолютті орналастыру» деген не, ол қандай стильдік нұсқау арқылы беріледі?
12.Абсолютті орналастыру кезінде координата басы ретінде қай нүкте алынады?
13.Абсолютті орналастырылған объектілер қандай ереже бойынша бір-бірімен қабатталып орналасады?

14.z-index стильдік қасиеті қандай мәндерді қабылдай алады және ол объектінің экрандық бейнесіне қалай әсер етеді?
15.Сатылы стильдік нұсқаулар деген не?

Тапсырмалар:

1. Азат жол аралықтары 1 сантиметрге тең болатын және барлық азат жолтар оң жақ және сол жақ шетке тураланған парақ құрыңыз.

2. P.def жаңа стилін құрыңыз. Бұл стильдің азат жолтары ені бойынша тураланған, сол жақтан 2 см, оң жақтан 1 см шегініс болсын. Параққа P.def. азат жолынан басқа кәдімгі азат жолтарды да орналастырыңыздар.

3. Тауардың бұрынғы және кейінгі бағасын жазуға арналған стильді анықтаңыздар. Бұрынғы бағасы – сұр түсті, үсті сызылған. Жаңа бағасы – қызыл түсті, өлшемі басқа мәтінге қарағанда 50% үлкен болсын. Бұрынғы және кейінгі бағалары көрсетілген тауарлар тізімін құрыңыздар.
4. Екі стильді анықтаңыздар. Бірінші стильде:
   - әріптер ашық сұр фонға қоңыр түспен жазылсын;
   - элемент жақтауы мен мазмұнынның арасындағы қашықтық 0,5 см болсын;
   - мәтін оң және сол жақ шеті бойынша туралансын.

   Екінші стильде:
   - фон көгілдір болсын;
   - элемент жақтауы мен мазмұнының арасындағы қашықтық 0,5 см болсын;
   - элементтен оң жақ және сол шетке дейінгі қашықтық 1 см;
   - жұмырланған қаріп.

Екі бөлімнен тұратын құжат дайындаңыз. Бірінші бөлім бірінші стильмен, екінші бөлімі екінші стильмен анықталатын болсын. Мұралану көрініп тұруы үшін екінші бөлім бірінші бөлім ішіне ендірілуі тиіс. Екінші бөлімде қандай стильдік нұсқаулар мұраланады, ал қандайлары мұраланбайды?

5. .nb стилін құрыңыз: элемент жақтау ішінде орналассын, браузер терезесінің (оның өлшемінен қатыссыз) жартысын (ені бойынша) алып тұрсын, сол жақ шетіне орналассын, ал парақтың басқа элементтері осы элементтің оң жағынан «көмкеріп» (обтекали) тұрсын.
6. Оң және сол жағынан тік сызық (азат жол биіктігімен бірдей) жүргізілетін азат жолқа арналған стиль құрыңыздар.
7. Стильдерді қолдана отырып, мәтін екі бағанға шығатындай (газеттегідей) парақ құрыңыздар. Бұл кезде кестелерді қолданбаңыздар.
8. Экранда айналдыру сызғыштары бар екі аймақты тұрғызып, оған ақпараттық элементтерді орналастырыңыздар.
9. z-index қасиетін қолдана отырып, экранда бір-бірін жауып тұратын бірнеше объектілерді орналастырыңыздар.

3.6 Объектіні белгіленген орынға қою, z-index

3.6.1 Объектіні бір орынға қою

Объектіні бір орынға қою, яғни орналастыру – бұл браузер терезесіндегі элементтің орналасатын коор-динатасын басқару болып табылады. Ол үшін CSS ортасында position параметрі (қасиеті) қолда-нылады.

Бұл қасиет үш мәнді қабылдай алады: absolute (абсолюттік орналастыру), relative (салыстырмалы орналастыру) және static (статикалық орналастыру). Static мәні элементті сайтта ешқандай әрекетсіз қалыпты түрде орналастырады, сондықтан бұл мәнді қолданудың ешқандай қажеті жоқ. Ал absolute және relative мәндерін кейінірек қарастырамыз, алдымен HTML-кодының иерархиялық құрылымының бір маңызды сәтін атап өтейік.

3.6.2 Парақ коды сатылары (иерархиясы)

Иерархия — бұл қабаттасқан матрешкалар ойыншығына ұқсас, мәліметтердің бір-бірінің ішіне салынып жинақталған қапшықтарға салынғаны сияқты орналасуы. Парақтың барлық элемент-тері <body> блогының ішіне орналасады. Сондықтан <body> тәгі парақтағы барлық басқа элементтердің «анасы» болып табылады. Парақ коды суреттегідей болсын делік.

Мысалда <body> элементінің e0l және e02 тәрізді екі тікелей ішкі туынды элементтері (ұрпағы) бар. Өз кезегінде бұл элементтердің де екі-екіден өз ұрпақтары бар: сәйкесінше e011, 012 және e021, e022 .

```
<BODY id=e0>
  <DIV  id=e01>
     <DIV  id=e011>
     . . .
     </DIV>
     <DIV  id=e012>
       . . .
     </DIV>
   </DIV>
   <DIV  id=e02>
     <DIV  id=e021>
     . . .
     </DIV>
     <DIV  id=e022>
       . . .
     </DIV>
    </DIV>
<BODY>
```

3.6.3 Абсолюттік түрде орналастыру

Абсолютті орналастыру position:absolute стильдік нұсқауы арқылы беріледі. Бұл кезде элементтің бастапқы координатасы тікелей сыртқы элемент аймағының (анасының) жоғарғы сол жақ бұрышы болады (егер оның да орны абсолют немесе салыстыр-малы түрде анықталған болса). Егер

сыртқы элемент белгілі бір орынға қойылмаса, онда оның да сыртқы элементі (анасы) алынады. Егер барлық сыртқы элементтерде position нұсқауы жоқ болса, онда алғашқы басты нүкте ретінде <body> тәгінің экрандық бейнесінің сол жақ жоғарғы бұрышы, яғни құжаттың сол жақ жоғарғы бұрышы қабылданады.

Көлденең және вертикаль координаталар left және top параметрлері арқылы беріледі.

Төменде суретті (100, 50) нүктесіне орналастыратын мысал келтірілген. Браузер координаталар басы ретінде құжаттың сол жақ жоғарғы бұрышын алады.

```
<HTML>
 <HEAD>
  <TITLE>Абсолюттік орналастыру</TITLE>
 </HEAD>
 <BODY bgcolor=white  text=black>
       <H1>Абсолюттік орналастыру</H1>
       <P>
```

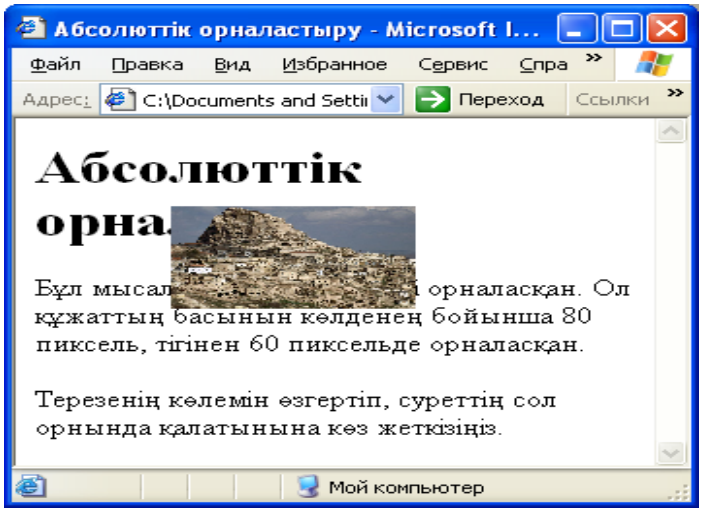
Бұл мысалда сурет абсолютті орналасқан.

Ол құжаттың басынын көлденең бойынша 80 пиксель, тігінен 60 пиксельде орналасқан.

```
<P>
```

Терезенің көлемін өзгертіп, суреттің сол орнында қалатынына көз жеткізіңіз.

```
 <IMG src=18166[1].jpg width=126 height=70
     border=0 alt=«Египет пирамидасы»
     style="position:absolute;left:80px;
     top:60px;">
</BODY>
</HTML>
```



3.22-сурет. Абсолюттік орналасу

Келесі мысалда екі сурет абсолютті түрде орналастырылған. Кодта оның әрқайсысы үшін (100, 50) координатасы көрсетілген, бірақ оның бірі үшін координаталар басы – құжаттың басы, ал екіншісі үшін – кестенің сол жақ жоғарғы бұрышы болады.

```
<HTML>
 <HEAD>
  <TITLE>Абсолюттік орналастыру </TITLE>
 </HEAD>
 <BODY bgcolor=white text=black>
       <H1> Абсолюттік орналастыру </H1>
       <P>
Екі сурет те (100,50) нүктесіне абсолютті орналастырылған.
Бір сурет үшін координаталар басы құжаттың басына, ал
екіншісі үшін кестенің басына сәйкес келеді.
<TABLE border=l cellspacing=0 cellpadding=10
       style="position:absolute; left:50px;
   top:300px;">
  <TR>
   <TD>
       Кесте абсалютті орналастырылған.
       <IMG  src=11494[2].jpg  width=85 height=80
      border=0 alt= " "                    style="position:absolute; left:100px;
      top:50px;">
   </TD>
  </TR>
 </TABLE>
 <IMG src=5465[1].jpg width=126  height=60
     border=0 alt=""
         style="position:absolute; left:100px;
     top:50px;">
</BODY>
</HTML>
```

3.23-сурет. Абсолюттік орналасу мысалы

Келтірілген мысалдар элементтердің абсолюттік орналастырылуы кәдімгі тізбекті форматтау процесінен тыс тұратынын көрсетеді. Браузер кодтардың жазылу ретін есепке алмайды, тек бастапқы координатаны анықтау үшін парақ элементтерінің бір-бірімен қабаттасу реттілігін ғана ескереді. Элементтер парақтағы басқа объектілер үстіне болса да, көрсетілген орынға шығарыла береді. Абсолюттік түрде орналастырылған элементтер кодтарының реттілігі объектілердің бірін-бірі жауып орналасуын анықтайды: «жоғарырақта» (беткі қабатта) коды кейінірек жазылған элемент тұрады.

Тағы мысалдар қарастырайық.

```
<HTML>
 <HEAD>
  <TITLE>Абсолютті орналастыру </TITLE>
 </HEAD>
 <BODY bgcolor=white  text=black>
   <H1>Абсолютті орналастыру</H1>
   <P>  Бұл мысалда екі сурет те абсолют орналастырылған.
 Коды бұрын орналастырғандықтан, екінші сурет бірінші суретті жауып тұр.
  <IMG src=111.jpg width=160 height=120
     border=0 alt=су гүлі
          style="position:absolute;
     left:l00px; top:150px;">
  <IMG src=12106[1].jpg width=140 height=150
     border=0 alt=Балық
          style="position:absolute;
           left:200px;top:250px;">
</BODY> </HTML>
```

3.24-сурет. Суреттердің абсолюттік орналасуы

3.6.4. Салыстырмалы орналастыру

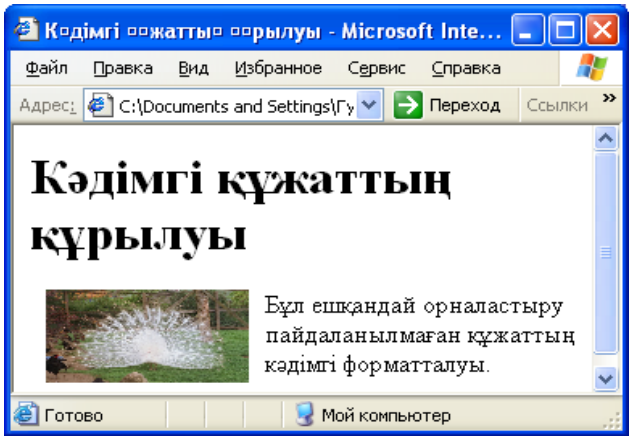
Салыстырмалы орналастыру position: relative стильдік нұсқауы арқылы беріледі. Мұндай элемент әдеттегі құжаттың тізбекті жазылуында тұра береді.

Координата басы ретінде элементті орны көрсетілмеген кездегі тұратын жері (координатасы) қабылданады.

Салыстырмалы орналастыру алгоритмін орындау кезінде бастапқыда браузер кәдімгі форматтауды орындай отырып, элементті параққа орналастырады, одан кейін left және top нұсқауларының мәндері бойынша оларды көрсетілген орынға жылжытады.

Төменде келтірілген мысалда құжат осы айтылған түрде форматталады (3.25-сурет).

```
<HTML>
 <HEAD>
  <TITLE>Кәдімгі құжаттың құрылуы
  </TITLE>
 </HEAD>
  <BODY bgcolor=white text=black>
   <H1>Кәдімгі құжаттың құрылуы</H1>
   <P>
   <IMG src=5465[1].jpg width=126 height=60
        border=0 alt=құс
        align=left hspace=10>
 Бұл ешқандай орналастыру пайдаланылмаған құжаттың кәдімгі форматталуы.<BR clear=left>
  </BODY>
</HTML>
```

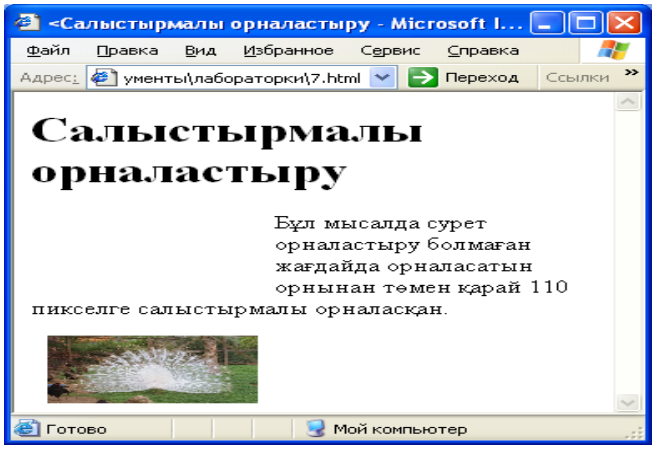


3.25-сурет. Құжаттың форматталуы

Келесі мысалда суретке салыстырмалы орналастыру қолданылған (2.26-сурет).

```
<HTML>
 <HEAD>
  <TITLE><Салыстырмалы орналастыру
  </TITLE>
 </HEAD>
 <BODY bgcolor=white text=black>
```

```
<H1> Салыстырмалы орналастыру </H1>
<P>
  <IMG src=5465[1].jpg width=126 height=60
      border=0 alt=Құс align=left
      hspace=10
      style="position:relative;
      left:0px; top:110px;">
```
Бұл мысалда сурет орналастыру болмаған жағдайда орналасатын орнынан төмен
қарай 110 пикселге салыстырмалы орналасқан. <BR clear=left>
```
 </BODY>
</HTML>
```



3.26-сурет. Салыстырмалы орналасу мысалы

Егер енді терезе көлемі өзгертілсе, онда суреттің экран бетінде түйінді нүктемен бірге – құжаттағы <img> тәгі орналасқан азат жол басымен бірге жылжитынын көруге болады. Азат жол басы оның алдындағы тақырыптың бір немесе екі экран жолына жазылуына байланысты өзгереді.

3.6.5 Аралас түрде орналастыру

Аралас түрде орналастырудың шартты түрде белгіленген төмендегідей төрт нұсқасын қарастырайық:

· absolute [relative] – абсолюттік орналастыру ішіндегі салыстырмалы орналастыру блогы;

· relative [absolute] – салыстырмалы орналастыру ішіндегі абсолютті орналастыру блогы;

· absolute [absolute] – абсолюттік орналастыру ішіндегі абсолютті орналастыру блогы;

· relative [relative] – салыстырмалы орналастыру ішіндегі салыстырмалы орналастыру блогы.

3.6.6 Absolute [relative]

Салыстырмалы түрде орналасқан суреттің коды абсолют түрде орналасқан кесте кодының ішіне орналасқан. Кесте құжат басынан оңға қарай 50 пикселге жылжытылған, осы қалып терезе көлемін өзгерткенде де сақталады. Сурет кесте ішіндегі «өз қалпынан» төмен қарай 100 пикселге

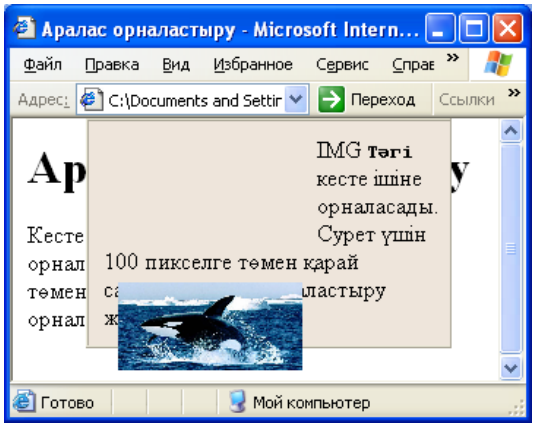
жылжытылған. Бірақ кесте ішіндегі мәтін сурет бұрынғы өз орнында тұрған сияқты болып орналасады.

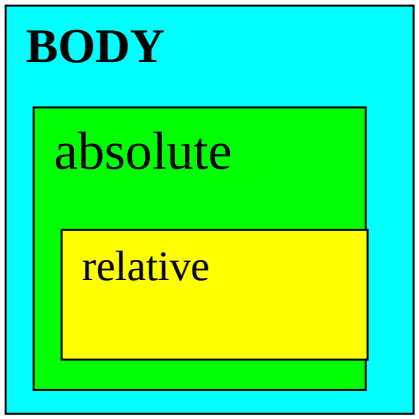
```
<HTML>
 <HEAD> <TITLE> Аралас орналастыру
      </TITLE>
 </HEAD>
 <BODY bgcolor=white text=black>
  <H1>Аралас орналастыру</H1>
<P>
```

Кесте (50,0) нүктесіне абсолют орналастырылған.

Кесте кодының ішіне төмен қарай 100 пикселге салыстырмалы орналастырылған сурет коды жазылған.

```
<P>
  <TABLE border=l cellspacing=0 cellpadding=10
     width=80% bgcolor=#EEE5DB
       style="position:absolute;
       left:50px;top:0px">
   <TR><TD>
    <P> <IMG src=12106[1].jpg width=126
      height=60 border=0 alt=Балық
      align=left hspace=10
     style="position:relative;
      left:0px; top:100px;">
IMG <TT><B>Тәгі</B></TT> кесте ішіне орналасады.
Сурет үшін 100 пикселге төмен қарай салыстырмалы орналастыру
жазылған.<BR clear=left>
</TD></TR>
</TABLE>
 </BODY>
</HTML>
```





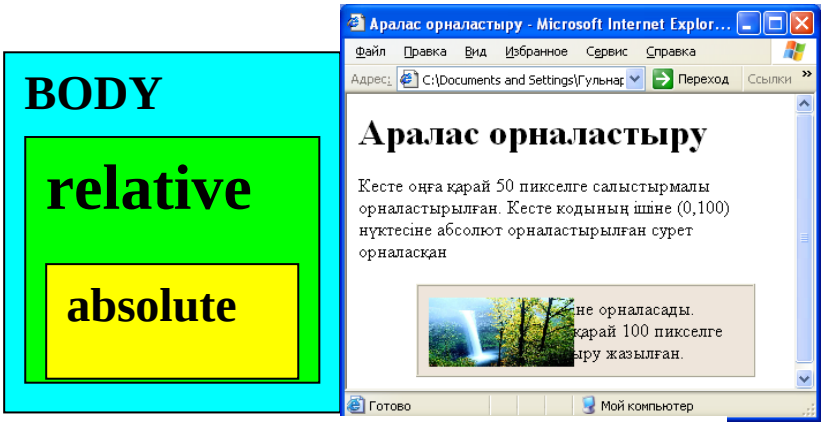
3.27-сурет. Орналасу реттілігі

3.6.7 Relative [absolute]

Екінші мысалда абсолютті орналасқан суреттің коды салыстырмалы орналасқан кесте кодының ішінде тұр. Кесте өзінің «қалыпты» орнынан оңға

қарай 50 пикселге жылжытылған. Сурет үшін координаталар басы кестенің сол жақ жоғарғы бұрышы болып табылады. Сурет осы қалпынан төмен қарай 100 пикселге ығысып орналасады. Осы сәтте кесте ішіндегі мәтін <img> тәгін есепке алмай форматталады.

```
<HTML>
 <HEAD>
  <TITLE>Аралас орналастыру</TITLE>
 </HEAD>
 <BODY bgcolor=white text=black>
  <H1> Аралас орналастыру </H1>
<P> Кесте оңға қарай 50 пикселге салыстырмалы орналастырылған.
Кесте кодының ішіне (0,100) нүктесіне абсолют орналастырылған сурет
орналасқан
 <P> <TABLE border=l cellspacing=0
    cellpadding=10 width=80% bgcolor=#EEE5DB
    style="position: relative;
    left :50px;top: 0px">
  <TR><TD>
   <P> <IMG src=11494[2].jpg
    width=126 height=60 border=0
        alt=Табиғат align=left hspace=10
        style="position:absolute;
    left:0px; top:l00px;">
IMG <TT><B>Тәгі</B></TT> кесте ішіне орналасады. Сурет үшін төмен
қарай 100 пикселге абсолют орналастыру жазылған.<BR clear=left>
        </TD></TR>
    </TABLE>
  </BODY>
</HTML>
```

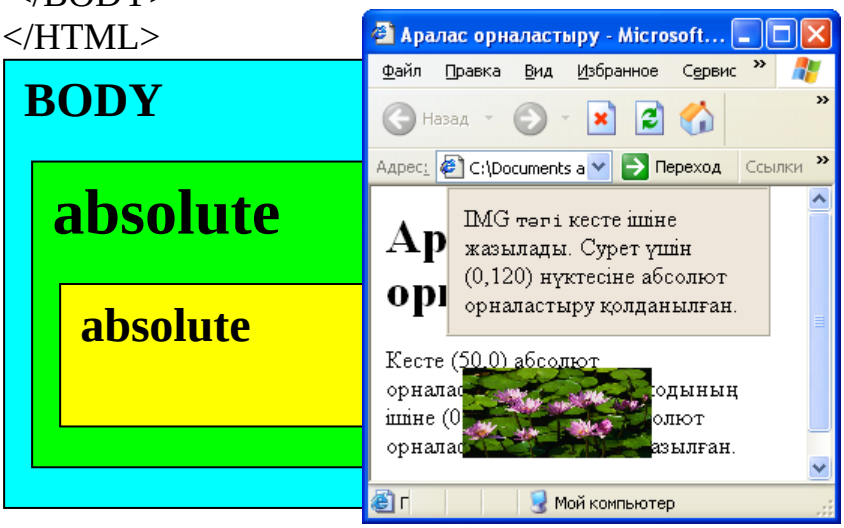


3.28-сурет. Аралас орналасу

3.6.8 Absolute [absolute]

Абсолютті түрде орналасқан суреттің коды абсолютті түрде орналасқан кестенің ішіне тұр. Мұнда терезе көлемін өзгерткенімен сурет өз қалпында тұр, өйткені кесте координаталары өзгеріссіз қалады.

```
<HTML>
 <HEAD>
  <TITLE>Аралас орналастыру</TITLE>
 </HEAD>
 <BODY bgcolor=white text=black>
  <H1>Аралас орналастыру</H1>
  <P> Кесте (50,0) абсолют орналастырылған. Кесте кодының ішіне (0,120)
нүктесіне абсолют орналастырылған сурет жазылған.
<P>   <TABLE border=l cellspacing=0
 cellpadding=10 width=80% bgcolor=#EEE5DB
 style="position:absolute;left:50px;top:0px">
<TR><TD>
 <P> <IMG src=111.jpg width=126 height=60
  border=0 alt=Гүлдер align=left
  hspace=10 style="position:absolute;
  left:0px; top:120px;">
       IMG <TT>тәгі<B></B></TT> кесте ішіне жазылады. Сурет үшін (0,120)
нүктесіне абсолют орналастыру қолданылған.<BR clear=left>
 </TD></TR>
</TABLE>
</BODY>
</HTML>
```



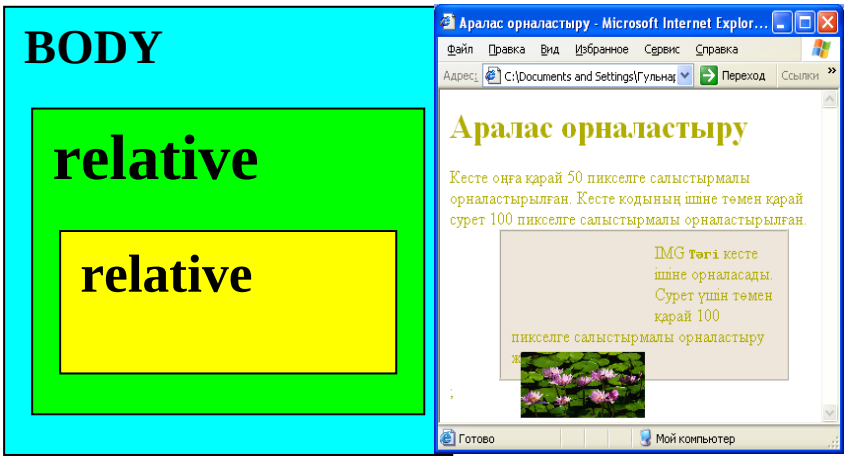
3.29-сурет Аралас орналасу

3.6.9 Relative [relative]

Салыстырмалы түрде орналасқан суреттің коды салыстырмалы түрде орналасқан кесте кодының ішінде тұр. Терезе көлемін өзгертуге байланысты

экрандағы кесте тұрған орын өзгергенде, суреттің орны да өзгереді: өйткені мұнда координаталар басы ығысады, ал сурет одан 100 пиксел төмен қарай орналасуға тиіс.

```
<HTML>
 <HEAD>
  <TITLE>Аралас орналастыру</TITLE>
 </HEAD>
  <BODY bgcolor=white  text=black">
   <H1>Аралас орналастыру</H1>
<P>Кесте оңға қарай 50 пикселге салыстырмалы орналастырылған.
Кесте кодының ішіне төмен қарай сурет 100 пикселге салыстырмалы орналастырылған.  </P>
<TABLE border=1 cellspacing=0
    cellpadding=10          width=80% bgcolor=#EEE5DB
        style="position:relative;
    left:50px;top:0px">
   <TR><TD>
    <P> <IMG src=111.jpg width=126
    height=60 border=0 alt=Гүлдер
    align=left hspace=10
        style="position:relative;
    left:0px; top:100px;">
IMG <TT><B>Тәгі</B></TT> кесте ішіне орналасады. Сурет үшін төмен қарай 100 пикселге салыстырмалы орналастыру жазылған.<BR clear=left>
   </TD></TR>
  </TABLE>          </BODY>
</HTML>
```



3.30-сурет.Аралас орналасу

3.6.10 Z-index

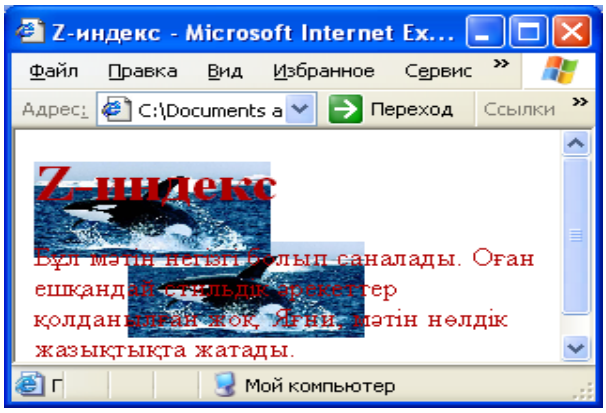
Z-index стильдік қасиеті экранда құжатты әр түрлі жазықтықта бейнелеу мүмкіндігін береді. Оның мәні бүтін сан болады, ол сан – элемент орналасатын жазықтықтың нөмірі (өз ішкі туындыларымен бірге). Негізгі мәтін нөлдік деңгейде болады (z-index:0). Z-index оң мәні элементті негізгі

мәтіннің жоғарғы жағына, ал теріс мәні – төменгі жағына орналастырады. Z-index мәні үлкен объект жоғары жақта (бетінде) орналасады.

```
<HTML>
 <HEAD>
  <TITLE> Z-индекс </TITLE>
 </HEAD>
<BODY bgcolor=white  text=brown">
   <H1>Z-индекс</H1>
 <P>
```

Бұл мәтін негізгі болып саналады. Оған ешқандай стильдік әрекеттер қолданылған жоқ. Яғни, мәтін нөлдік жазықтықта жатады.

```
  <IMG src=12106[1].jpg width=126
      height=60 border=0 alt=Балық
      align=left hspace=10
      style="position:absolute;
      left:0px;top:20px;z-index=-1;">
  <IMG src=12106[1].jpg width=126 height=60
      border=0 alt=Балық align=left
      hspace=10 style="position:absolute;
      left:50px;top:70px;z-index=-2;">
  </BODY>
</HTML>
```



3.31-сурет.Әр түрлі элементтердің орналасуы

Бақылау сұрақтары:

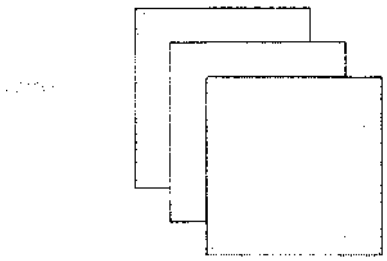
1. «Орналастыру» (Позиционирование) деген не?
2. Элементті орналастыру CSS қандай қасиеті арқылы беріледі? Бұл қасиет қандай мәндерді қабылдай алады?
3. Браузер терезесінде элемент координаталары қандай стильдік нұсқаулармен беріледі?
4. Абсолютті орналастыру орындалатын алгоритмді сипаттаңыз.
5. Салыстырмалы орналастыру орындалатын алгоритмді сипаттаңыз.
6. absolute [relative] схемасы бойынша аралас орналастыру орындалатын

алгоритмді сипаттаңыз.

7. relative [absolute] схемасы бойынша аралас орналастыру орындалатын алгоритмді сипаттаңыз.
8. absolute [absolute] схемасы бойынша аралас орналастыру орындалатын алгоритмді сипаттаңыз.
9. relative [relative] схемасы бойынша аралас орналастыру орындалатын алгоритмді сипаттаңыз.
10. z-index стильдік қасиеті бар элементтерді экранға шығару алгоритмін сипаттаңыз.

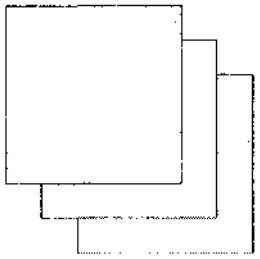
Тапсырмалар:

1. Суретті (100,50) нүктесіне, одан кейін 150,150 нүктесіне орналастырыңыз. Браузер терезесінің көлемін өзгертіп, экранда суреттің қалай жылжытындығын анықтаңыз.
2. Сурет өлшемін (130,50) пикселге, одан кейін (150,100) пикселге өзгертіңіз. Мәтіннің айналасындағы фонның түсін сарыға өзгертіңіз.
3. Пілдер суретін кестемен салыстырмалы (80,40) нүктесіне, одан кейін 150,150 нүктесіне орналастырыңыз. Терезе көлемін өзгертіп, суреттің экранда қалай жылжитынына көңіл аудырыңыз.
4. Маймылдар суретін құжатпен салыстырмалы (50,100) нүктесіне орналастырыңыз. Сурет көлемін (140,800) пикселге өзгертіңіз. Мәтін айналасындағы фонның түсі көк болсын.
5. Суреттердің орнын алмастырыңыз, сәйкесінше көлемдерін өзгертіңіз. Маймылдар суретін құжаттың сол жақ жоғарғы бұрышына қатысты орналастырыңыз.
6. Бір сурет бірнеше рет шығатындай, алдыңғыға қарағанда оның көшірмесі төмен және оңға қарай, оның бетінде орналасатын парақ құрыңыз. (3.32-сурет).



3.32-сурет. 6-тапсырма схемасы

7. Бір сурет бірнеше рет шығатындай, алдыңғыға қарағанда оның көшірмесі төмен және оңға қарай, оның астында орналасатын парақ құрыңыз. (3.33-сурет).



3.33-сурет. 7-тапсырма схемасы

8. Overflow стильдік қасиеті, элемент мазмұны берілген аймаққа сыймаған жағдайда қолданылады. Стандартты мәндердің ішінде scroll ғана дұрыс жұмыс істейді. overflow қасиетінің жұмысын көрсететін парақ құрыңыз (3.34-сурет).

3.34-сурет. 8 -тапсырма схемасы

4 JAVASCRIPT ТІЛІ

4.1 Жалпы мағлұматтар

JavaScript программалау тілін Netscape пен Sun Microsystems фирмасы бірлесіп ұсынған, ол интерактивті HTML-құжаттарды құруға арналған. JavaScript тілін қолданылатын аймақтары:

- Динамикалық, яғни құжат жүктелгеннен кейін мазмұнын өзгертуге болатын парақтарды құру;
- Серверге жөнелтілгенге дейін пайдаланушының форманы дұрыс толтырылғандығын тексеру;
- Сценарийлер көмегімен «жергілікті» (локальный) және басқа да мәселелерді шешу.

JavaScript клиент және сервер жағында да орындалатын қосымшаларды құру мүмкіндігін береді. Екі түрлі қосымшаларды құру кезінде де, стандартты объектілерді анықтаулардан тұратын, *ядро* қолданылады. Пайдаланушы компьютерінде клиенттік қосымшаларды браузер орындайды.

JavaScript тіліндегі программа (сценарий) браузерге ендірілген интерпретатор арқылы өңделеді. Сценарийлерді кез келген браузерде орындалатындай түрде жазуға тырысу керек.

JavaScript тіліндегі жазылған программалар (сценарий) операторлар тізбегінен тұрады. Егер бір жолға бірнеше операторлар жазылса олардың арасына нүктелі үтір (;) қойылады. Егер әрбір оператор жеке жолға жазылса, онда ешнәрсе қоймауға болады. Бір операторды бірнеше жолға жазуға болады.

Құрылымдық программалау принципіне сәйкес, программаны бөліктік құрылым болатындай түрде жазу ұсынылады. Бұл программаны зерттеу мен қателерді іздеуді жеңілдетеді.

JavaScript программаларында түсініктемелерді қолдануға болады. Бір жолға сиятын түсініктеме беру үшін екі қиғаш сызықты қолдануға болады (//). Егер түсініктеме мәтіні бірнеше жолға жазылса, онда оларды /* және /* символдарының ішіне жазу керек. JavaScript тілінде бас әріптер мен кіші әріптер әр түрлі символ болып саналады.

4. 2 Литералдар

Программа әр түрлі операциялар орындата алатын қарапайым мәліметтер *литералдар* деп аталады. Литералдар өзгертілмейді. Бүтін типті литералдар *ондық* (10), *он алтылық* (16) және *сегіздік* (8) жүйелерде берілуі мүмкін. Ондық жүйедегі бүтін типті литералдар таңбамен немесе таңбасыз жазылатын ондық цифрлар тізбегінен тұрады, мысалы, 15, 123, -156, +3567. Он алтылық жүйедегі сандар 0–9 цифрларынан және a, b, c, d, e, f әріптерінен тұрады. Он алтылық сандар оның алдында 0x символын қосып жазу арқылы жазылады, мысалы, 0x25, 0xa1, 0xff. Сегіздік жүйеде тек 0–7 цифрлары жазылады және олар нөлден басталады, мысалы, 03, 0543, 011. Литералдарды жазудың математикадағы нақты сандарды жазудан айырмашылығы, оның бүтін және бөлшек бөлігін бөліп тұратын үтірдің орнына нүкте қойылады,

мысалы, 123.34, -22.56. Сонымен қатар, нақты сандарды жазуда экспоненциалдық форманы қолдануға болады, мысалы, 0,000000273 санын мынадай түрде 2.73x10 $^{-7}$ жазуға болады, ал оны JavaScript тілінде былай 2.73e-7жазады. Бұл жазуда көбейту белгісі мен 10 саны e символымен алмастырылады. Нақты литералдарды жазуда, кем дегенде, бір сан және ондық нүкте немесе экспонента (e немесе E) символы болуы керек. JavaScript тілінде бүтін және нақты мәндерден басқа логикалық мәндер де кездесуі мүмкін. Мұнда тек екі түрлі логикалық мән болады: ақиқат және жалған. Біріншісі – true, екіншісі – false литералы түрінде жазылады. JavaScript тілінің кейбір нұсқаларында true ретінде бір, ал false ретінде нөл қолданылуы мүмкін. Сөз тіркестерінен тұратын тіркестік литерал апостроф (‘) немесе тырнақша (“) белгілеріне алынып жазылады, мысалы, "нәтиже" немесе 'нәтиже'. Бос қатарды көрсететін тіркестік литерал " " немесе ' ' арқылы белгіленеді.

4.3 Айнымалылар

Айнымалылар ақпараттарды сақтау үшін қолданылады. Айнымалылар сценарийлерде идентификаторлардың көмегі арқылы көрсетіледі. *Идентификатор* латын әріптерінен, цифрлардан, астын сызу белгісінен тұрады және ол міндетті түрде латын әрпінен немесе астын сызу белгісінен басталуы керек, мысалы, _my_test, test_l. Айнымалының типі сценарийде сақталған ақпараттың типіне байланысты болып, сол мәліметтің типі өзгерсе, айнымалының типі де өзгереді. Айнымалы var операторы арқылы анықталуы тиіс, мысалы:

```
var test1
```

Мұнда test1 айнымалысының типі әлі анықталмаған, ол айнымалыға белгілі бір мән меншіктеген кезде ғана айқындалады.

var операторын айнымалыны инициализациялағанда да қолдануға болады, мысалы

```
var test2=276
```

Мұнда test2 айнымалысының типі анықталды және ол 276 мәніне тең болды.

Меншіктеу операторынан кейін айнымалының мәні өзгереді. Меншіктеу операторы программаның кез келген жерінде тұра береді және ол айнымалының тек мәнін ғана емес, типін де өзгерте алады. Меншіктеу операторы мынадай түрде беріледі:

```
a=b
```

мұндағы a – біз бір мән бергіміз келетін айнымалы, b – айнымалының жаңа мәнін анықтайтын өрнек.

Сценарийде мынадай айнымалылар жазылған болсын делік:

```
var n=3725
var x=2.75
var p=true
var s="Программа орындалды"
```

n және x айнымалыларының типі number, p айнымалысының типі – логикалық, s айнымалысының типі string. JavaScript тілінде барлық

стандартты және тұтынушының өзі анықтайтын функциялар үшін function типі анықталған. JavaScript объектілері үшін object типі бар. Object типті айнымалылар объектілерді сақтайтындықтан, оларды жай объектілер деп атай береді.
Сценарийлердегі <HEAD> бөлігінде және <BODY> бөлігінде де сипатталған айнымалылар екеуінде де бірдей жұмыс істей береді, бұларды осы құжаттағы кез келген сценарий пайдалана алады. Мұндай айнымалылар *глобальді* (ауқымды) деп аталады, ал функция ішінде анықталған айнымалылар *локальді* , яғни жергілікті болып саналады.

4.4 Өрнектер

Өрнектер операциялар (амалдар) таңбаларымен біріктірілген литералдардан, айнымалылардан және жақшалардан тұрады. Өрнектерді есептеу нәтижесінде сан (бүтін не нақты), сөз тіркесі немесе логикалық типте болатын бір ғана мән шығады. Өрнектерде қолданылатын айнымалылар оған дейін инициалданған болуы керек. Өрнектерді есептеу кезінде анықталмаған немесе инициалданбаған айнымалылар кездессе қате шығады. JavaScript тілінде анықталмаған мәнді белгілеу үшін null литералы қолданылады. Егер айнымалы null мәнін меншіктесе, ол инициалданған болып саналады.

Өрнек операндтардан (мәндер мен айнымалылардан) және операциялар таңбаларынан (+, -, *, /) тұрады. Мысалы, a*b формуласында a және b операндтар, * таңбасы – көбейту операциясы.
Операциялар *унарлық (бірорындық)* немесе *бинарлық (екіорындық)* болып екіге бөлінеді. Өрнек +A түрінде жазылса, ондағы + – унарлық операцияны белгілейді, ал егер ол A + B түрінде берілсе, ондағы + – бинарлық операцияның орындалатынын көрсетеді. +A өрнегін есептеу A операндының мәнін есептеп, оған + операциясын қолдану дегенді білдіреді. Ал A+B өрнегін есептеу төмендегі әрекеттерден тұрады:
1. A және B мәндері есептеледі.
2. Операция 1-ші қадамда табылған операндтар мәндеріне қолданылады.
Есептелген мәннің типіне қарай өрнек арифметикалық, логикалық және тіркестік типтердің біріне жатқызылады. 4.1-кестеде көрсетілген арифметикалық операциялардың орындалу нәтижесінде өрнектер құрылады.

4.1-кесте. Арифметикалық операциялар

Операция	**Аталуы**
+	Қосу
-	Азайту
*	Көбейту
/	Бөлу
%	Бүтін сандарды бөлгеннен кейінгі қалдық
++	Операнд мәнін бірге өсіру
--	Операнд мәнін бірге кеміту

Өрнектегі операторлар арифметикалық операциялардың басымдылықтарына (приоритеттеріне) қарай солдан оңға қарай есептеледі. Керекті жағдайда өрнекке жақша енгізу арқылы операциялар реттілігін өзгертуге болады. JavaScript тілінде теңдіктің оң және сол жағында орналасқан операндтарға арифметикалық амалдар қолданып, нәтижені сол жақтағы операндқа меншіктейтін операторлар анықталған. Операциялардың мұндай қысқарған түрлері 4.2-кестеде көрсетілген.

4.2- кесте. Меншіктеу операторының қысқаша жазылу жолдары

Оператор	Соған сәйкес меншіктеу операторы
X += Y	X = X+Y
X –= Y	X = X-Y
X *= Y	X = X*Y
X /= Y	X = X/Y
X %= Y	X = X%Y

Қатынас операциялары кез келген типтегі операндқа қолданыла береді. Келесі мысалда:

```
var x = 1;
var y;
y = (x += 2) + 1;
```

y айнымалысының мәні 4, ал x айнымалысының мәні – 3. Мынадай тізбекті түрдегі меншіктеулерді де пайдалануға болады :

```
x = y = z = t = өрнек;
```

Мұнда бірнеше айнымалының бәріне бір ғана мән меншіктеледі.

«+ +» және «--» операциялары тек айнымалыларға тіркеледі, оларды өрнектерге қосып жазуға болмайды.

Мынадай командалардың + + x және x + + (-- x және x --) айырмашылығы бұлар басқа командалар құрамына кіріп тұрғанда ғана есепке алынады. Алғашқысында (олар айнымалы алдында) операция айнымалыны пайдаланғанға дейін орындалады, ал соңғысында – пайдаланғаннан кейін орындалады.

```
x = 5;                  Меншіктеулерінен кейін:
y = ++ x;     x және y-тің мәні 6-ға тең.

x = 5;                  Ал мына меншіктеулерден соң:
y = x ++;     x-тің мәні 5-ке, ал y-тің мәні 6-ға тең болады.
```

Операция нәтижесі – салыстыру мәндері дұрыс болса, логикалық true, ал оған кері жағдайда – false мәні болып табылады. Салыстыру операциялары төмендегідей:

- < (кіші);
- <= (кіші немесе тең – артық емес);
- == (тең);
- != (тең емес);

- >= (үлкен немесе тең – кіші емес);
- > (үлкен).

! (логикалық ЕМЕС) операциясы логикалық типтегі операндтарға қолданылады, егер операнд мәні true болса, онда !a өрнегінің мәні – false, ал егер операнд мәні false болса, онда !a = true болады. && (логикалық ЖӘНЕ) пен | | (логикалық НЕМЕСЕ) операцияларының қолданылу ережесі 4.3-кестеде көрсетілген.

4.3- кесте. Логикалық операциялардың орындалу ережелері

A	B	A&&B	A||B
true	true	true	true
true	false	false	true
false	true	false	true
false	false	false	False

A&&B өрнегінің мәні – екі операнд мәндері де ақиқат болғанда ғана ақиқат, қалған жағдайларда жалған болады. A| |B өрнегінің мәні – кем дегенде, бір операнд ақиқат болса, ақиқат, ал басқа жағдайларда – жалған болып саналады.

Символдар тізбегінен тұратын тіркестік мәндер үшін *символдарды конкатенациялау (біріктіру)* операциясы анықталған. Операция «қосу» таңбасымен белгіленеді. Бұл операцияның нәтижесі екі операндтар тізбегінен құралатын жаңа сөз тіркесі болады, мысалы, төмендегі меншіктеу операторының орындалуы нәтижесінде

st = "озат "+"студент"

st айнымалысы "озат студент" мәнін меншіктейді. Тағы бір мысал қарастырайық. Мынадай екі оператор берілсін:

stl = "озат "; st2 = "студент"

Төмендегі операторды орындау нәтижесінде

stl += st2

st1 айнымалысы "озат студент" мәнін меншіктейді.

Операцияның басымдылығы өрнектегі амалдардың орындалу реттілігін анықтайды. 4.4-кестеде орындалу басымдылығы кему реті бойынша орналасқан операциялар тізімі қарастырылған.

4.4-кесте. Операциялардың басымдылық кестесі

Операция аты	Белгіленуі
Инкремент	++
Декремент	--
Терістеу	!
Унарлы минус	-
Көбейту	*
Бөлу, қалдық табу	/, %
Қосу	+
Азайту	-

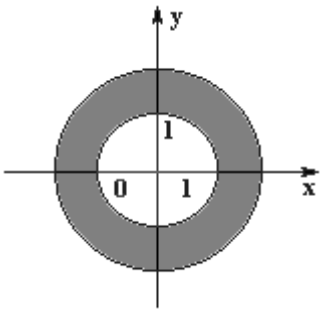
Салыстыру	<, >, <=, >=
Теңдік	==
Теңсіздік	! =
Логикалық ЖӘНЕ	&&
Логикалық НЕМЕСЕ	||
Меншіктеу	=, +=, -=, *=, /=, %=, ! =

Жаттығулар

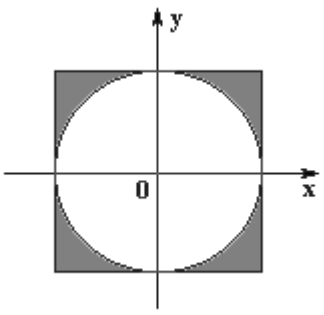
1. Төмендегі тұжырымдар нәтижесінде ақиқат болатын өрнек жазыңыздар:
   - m бүтін айнымалысы k бүтін айнымалысына қалдықсыз бөлінеді;
   - A, B және C нақты айнымалылардың мәні кемімейтін реттілік құрайды;
   - үш x,y,z айнымалыларын жұптасқан түрде қарастырғанда, олардың ішіндегі ең үлкені x айнымалысы болып шыққан;
   - A,B,C логикалық айнымалылары мәндерінің бірде-біреуі ақиқат емес;
   - A, B, C логикалық айнымалылары мәндерінің, кем дегенде, біреуі ақиқат.

2. Берілген x және y мәндері жазықтықтағы нүкте координаталары болып табылады, төмендегі өрнектердің әрқайсысы true мәнін қабылдайтын аймақты штрих сызықпен бояңыздар:
   - (X*Y>0);
   - Y+X<5 && X*X+Y*Y>i;
   - Y<X*X+2 || Y>6.

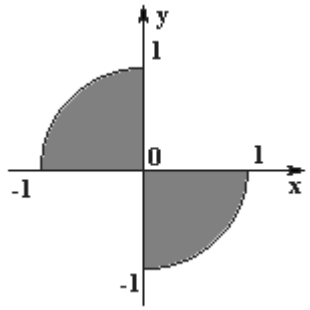
3. Жазықтықтағы координаталары x және y болып келген нүкте үшін, оның штрихталған аймаққа түсуі ақиқат болып келетін формула жазыңыздар (4.1–4.4 сурет.).



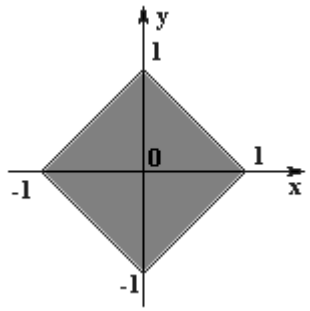
4.1- сурет**.** Нүкте екі шеңбермен қоршалған аймаққа түседі



4.2- сурет. Нүкте квадрат пен шеңбер аралығындағы аймаққа түседі



4.3- сурет. Нүкте көрсетілген екі секторға түседі



4.4- сурет. Нүкте ромб ішіне түседі

4.5 HTML-құжатындағы сценарий

JavaScript тілінде жазылған сценарий тікелей HTML-құжатынының ішінде <script> және </script> тәгтерінің арасына жазылады.

Қолданылатын сценарийлер тілін анықтайтын language, <script> тәгінің параметрлерінің бірі болып табылады. JavaScript тілі үшін бұл параметрдің мәні "JavaScript"-ке тең. Егер VBScript, сценарийлер тілі қолданылса, онда параметрдің мәні "VBScript" тең болуы керек. JavaScript қолданған жағдайда language параметрін жазбауға да болады, бұл тілді браузер үнсіз келісім бойынша таңдайды

Әдетте браузерлер қолдамайтын кез келген HTML тәгтерін ескерусіз қалдырады. Қолдамайтын тәгтердің мазмұнын талдаудағы браузер әрекеті парақтың дұрыс көрінбеуіне әкеледі. Мұндай жағдай болмауы үшін JavaScript тілінің операторларын <! — ... —> тәгтегі түсініктеме ішіне орналастыру ұсынылады. Интерпретатор дұрыс жұмыс істеуі үшін түсініктемені жабу тәгінің --> алдына // символдарын қою керек.

Сонымен HTML-құжатына сценарий орналастыру үшін келесі түрде жазу қажет:

```
<script language="JavaScript">
<!--
JavaScript тілінің операторлары
//-->
```

</script>

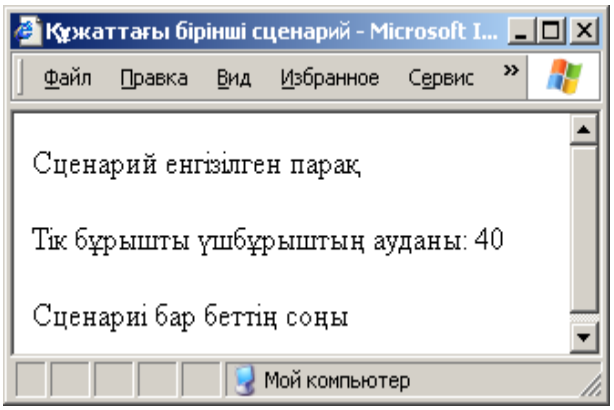
Құжат ішінде бірнеше<script> тәгтері жазылуы мүмкін. Олардың барлығы ретімен JavaScript интерпретаторымен өңделеді.

Үшбұрыш ауданын есептеу

Катеттері берілген тік бұрышты үшбұрыштың ауданын анықтайтын сценарий жазу керек болсын делік. Сценарийді HTML-құжаттың <BODY> бөлігінде жазайық.

4.1- листинг. Құжаттағы бірінші сценарий:

```
<HTML>
<HEAD> <title> Құжаттағы бірінші сценарий /title>
</HEAD>
<BODY>
<P> Сценарий енгізілген парақ </P>
<script>  <!-- //
var a=8; h=10
document.write ("Тік бұрышты үшбұрыштың ауданы: ", a*h/2)
//-->
</script>
<P>Сценарийі бар беттің соңы </P>
</BODY> </HTML>
```



4.5-сурет. Сценарий нәтижесі

Сценарийде екі айнымалы сипатталып инициалданады, содан соң өрнектің мәні құжатқа жазылады. Сценарий нәтижесін экранға шығару үшін document объектісінің write тәсілі қолданылады. Құжатта жазылған жолдар HTML тәгтері мен JavaScript өрнектерінен тұрады.

Браузер JavaScript тілін қолдамайтын болса немесе оны сүйемелдеу алып тасталынған жағдайда, <noscript> тәгінде жазылған сөз тіркесі экранға шығарылады. Бұл тәг <script> ... </script> тәгтерінен кейін жазылады.

Келесі мысалдарда браузер JavaScript тілін толық сүйемелдейді деп есептейміз.

4.6 Функцияларды сипаттау және пайдалану

Программа құрғанда ондағы логикалық тәуелсіз бөліктерді (*программа ішіндегі подпрограммалар*) жеке-жеке бөліп қарауға болады. Программаны

мұндай бөліктерге бөліп қарастыру оның жұмысын түсінуді жеңілдетеді. Программаны ішкі подпрограммаларға бөлу оны түзету ісін жеңілдетеді. Оның үстіне бір жазылған подпрограмманы әр түрлі программаларда қолдануға болады. Көптеген программалау тілдерінде подпрограммалар функциялар, модульдер және т.б. көмегі арқылы жүзеге асырылады.

Функция – JavaScript тілінің негізгі элементі. Функция мынадай түрде сипатталады

function F(v) {S}

мұндағы F – функцияны шақыруға болатын ат тағайындайтын функция идентификаторы; v – үтір арқылы бөлініп жазылатын функция параметрлерінің тізімі; S – нәтиже алу үшін орындалатын іс-әрекеттерден тұратын функция тұлғасы, яғни операторлар тізбегі. Міндетті түрде жазылмайтын return операторы функцияның программаға қайтарылатын мәнін анықтайды.

Бір функцияның сипаттамасы басқа бір функция сипаттамасы ішіне жазылмайды. Функция параметрлері оның ішкі операторларында, яғни тұлғасында қарапайым айнымалылар рөлін атқарады, бірақ бұл параметрлерге бастапқы мәндер функцияны шақырғанда ғана беріледі.

Егер сипаттама мынадай түрде болса:

function F(vl,v2,...,vn) {S}

онда функцияны шақыру мынадай болуы керек:

F(el,e2,...,en)

мұндағы el,e2,...,en – параметрлердің *нақты (нақтық)* мәндерін беретін өрнектер. Функция сипаттамасында көрсетілген vl,v2,...,vn параметрлері функцияны шақыру кезінде el,e2,...,en сияқты нақты параметрлерді бергеннен кейін мән қабылдайтын болғандықтан, *формальді* параметрлер деп аталады. Егер функцияда параметрлер болмаса, функция сипаттамасы мынадай түрде болады:

function F() {S}

Операторда функция атынан соң, жақшаның болуы міндетті, яғни функцияны шақыру мына түрде болуы керек:

F()

Көбінесе барлық анықтаулар мен функциялар құжаттың <HEAD> бөлігінде беріледі. Бұл құжатты браузерге жүктегенде, оны интерпретациялауды және барлық функциялардың компьютер жадында сақталуын қамтамасыз етеді.

Келесі мысалда катеттері берілген тікбұрышты үшбұрыш ауданын есептеудегі care функциясы құжаттың тақырып бөлігінде сипатталған.

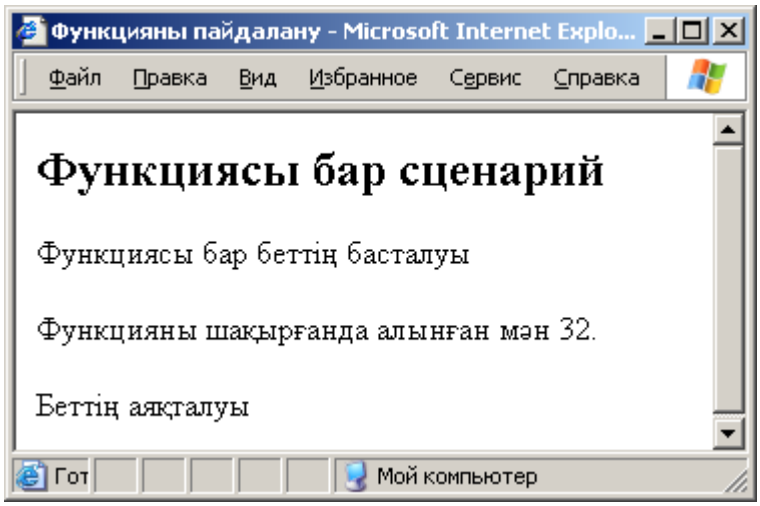
4.6.1 Функциясы бар сценарий

Қабырғасы мен биіктігі берілген үшбұрыштың ауданын анықтайтын сценарий жазу қажет. Үшбұрыштың ауданын есептеу үшін HTML-құжаттың <HEAD> бөлігінде сипатталған care функциясын қолданамыз (4.2 листинг).

4.2 листинг. Функциясы бар сценарий жазу

<HTML> <HEAD> <H2> Функциясы бар сценарий</H2>

```
        <title> Функцияны пайдалану </title>
<script language="JavaScript">
<!-- //
  function care(a,h) {return a*h/2}
//-->
</script>
</HEAD>
<BODY>
<P> Функциясы бар беттің басталуы </P>
<script>
<!--
  var a1=4; h1=16
  var s=care(a1,h1)
  document.write("Функцияны шақырғанда алынған мән ", s,".");
//-->
</script>
<P>Беттің аяқталуы</P>
</BODY> </HTML>
```



4.6-сурет. Функциясы бар сценарий нәтижесі

Функция тұлғасы функцияның қайтарылатын мәнін анықтайтын тек return операторынан тұрады. s=care(a1,h1) меншіктеу операторы орындалғанда құжат тұлғасындағы функцияны шақыру іске асырылады. А және h формальді параметрлеріне a1 и s нақты параметрлерінің мәні меншіктеледі де, функция тұлғасы есептеледі. Табылған мән write тәсілі арқылы экранға шығарылады.

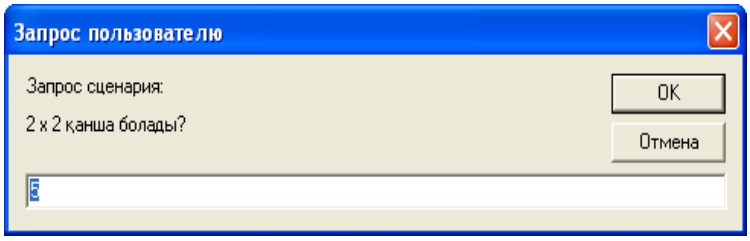
4.6.2 Ақпарат енгізу

Alert функциясы информацияны экранға шығару үшін қолданылады. Ал информация енгізу үшін prompt функциясы қажет: prompt("жазу", "келісім бойынша енгізілетін мән"); мұнда экранға ішінде екі батырмасы бар терезе шығады. Біз жазба мәліметті енгізу жолына жазып, ОК батырмасын басамыз. Сонда бірінші терезе жабылып, келесі терезеге енгізілген мәліметтер шығады. Ол мәнді мысалы, айнымалыға меншіктеуге, артынан

басқа командаларда пайдалануға болады. Егер Cancel батырмасын шертетін болсақ, онда prompt функциясы арнайы null мәнін қайтарады (бұл "null“ сөз тіркесі емес, яғни бос жол ""емес, мәліметтің арнайы мәні).
Мысалы:

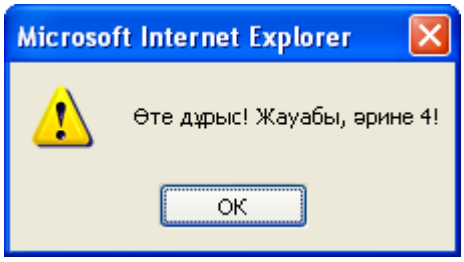
```
<script language="JavaScript">
<!--
var str = prompt("2 х 2 қанша болады?", "5");
if (str == "4") alert (“Өте дұрыс! Жауабы, әрине 4!"); else alert(“Әзілді түсінсең ғана, ол қалжыңға айналады!"); //-->
</script>
```

Осы скрипті іске қосқанда экранға суреттегі терезе шығады.



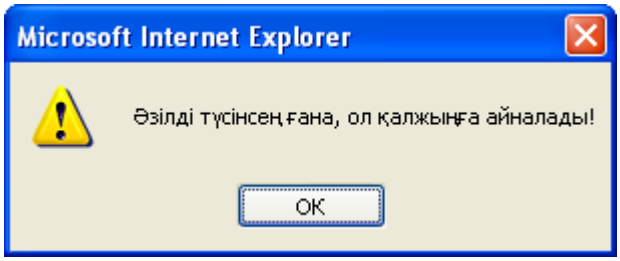
4.7-сурет. Программа нәтижесі

Егер 4 санын енгізіп, ОК батырмасын шертетін болсақ, скрипт жұмысы төмендегі суреттегідей болып жалғасады.



4.8-сурет. Сұхбат терезеге жауаптың шығуы

Егер енгізу өрісінде 5 санын қалдырсақ (не 4-тен басқа кез келген сан енгізсек), экранға төменгі суретте көрсетілген хабарлама шығады



4.9-сурет. Программа нәтижесінің сұхбат терезеге шығуы

4.6.3. Шартты команда

Шартты команданың жалпы жазылу түрі :
Жалпы түрі: if (шарт) 1команда;
else 2 команда;
Мысал ы:

```
if (a > b) c = a;
else c = b;
```
Мұнда шарт тексеріледі. Егер ол ақиқат болса, онда 1 команда, әйтпесе – 2 команда атқарылады.
Мына кодтардан соң:
```
 x = 1;
 if (x == 1) y = 10;
 else    y = 20;
 x += y;
```
айнымалы х-тің мәні 11-ге тең болады.
Мына командалардан соң:
```
x = 1;
if(x != 1) y = 10;
else            y = 20;
x += y;
```
х айнымалысының мәні 2-ге тең болады.
Шартты команданы қысқа түрде else тармағынсыз жазуға да болады:

Жалпы түрі	Мысал
if (шарт) команда1	if (x < 0) x = -x;

Мына кодтардан кейін:
```
x = 1;
y = 10;
if  (x = = 1) y += 10;
x += y;
```
х айнымалысы 21-ге тең болады.
Ал мынадай кодтардан соң:
```
x = 1;
y = 10;
if(x != 1) y += 10;
x += y;
```
х айнымалысы 11-ге тең.

Ауыстырғыш

Жалпы түрі
```
if (1шарт) 1ком;
else if(2шарт) 2ком;
else if(3шарт) 3ком;
 ...
 ...
 ...
else  комN;
```
Мысал:
```
if (Day = = 1) NameDay = "Дүйсенбі";
else if (Day == 2} NameDay = "Сейсенбі";
else if (Day == 3) NameDay = "Сәрсенбі";
```

```
else if (Day == 4) NameDay = "Бейсенбі";
else if (Day == 5) NameDay = "Жұма";
else if (Day == 6) NameDay = "Сенбі";
else if (Day == 7) NameDay = "Жексенбі";
else NameDay = "қате";
```

JavaScript тілінде арнайы switch конструкциясы бар, оны Netscape Navigator және Internet Explorer браузерлерінің 4-нұсқасынан кейінгілері түсінеді. Ол Си және Java тілдеріндегі осы командаға сәйкес келеді.

Жалпы түрі:

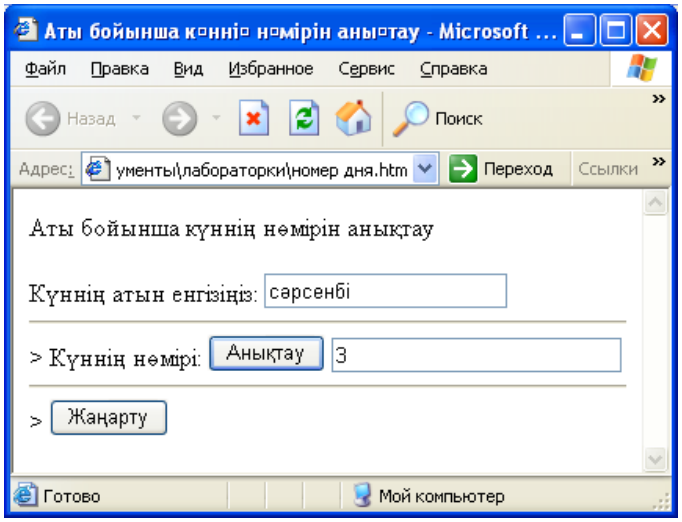
```
switch (өрнек)
{
 case вариант1:
 командалар
        break;
 case вариант2:
 командалар
        break;
 ...
 default:
 командалар
```

Мұнда switch (өрнек) мәні case сөзінен кейінгі қандай мәнге тең болса, сол жол орындалады. Онан кейінгі break командасы басқаруды switch сөзінен кейінгі жолға береді. Егер break командасы жоқ болса, онда варианттарды тексеру ары қарай жалғаса береді. Default нұсқасы (ол болмауы да мүмкін) өрнек мәні case сөзінен кейінгі бірде бір мәнге сәйкес келмеген кезде орындалады.

Мысалы: Күннің аты енгізілген кезде оның нөмірін анықтау мысалын қарастырайық:

```
<HTML>
<HEAD>
<TITLE>Аты бойынша күннің нөмірін анықтау</TITLE>
<script language="JavaScript">
<!-- //
function daynum (obj)
{ var m = obj.numl.value;
var s
switch (m)
{ case 'дүйсенбі': s=l; break;
case 'сейсенбі': s=2; break;
case 'сәрсенбі': s=3; break;
case 'бейсенбі': s=4; break;
case 'жұма': s=5; break;
case 'сенбі': s=6; break;
```

```
case 'жексенбі': s=0; break;
default: s=' күннің атын қате енгіздіңіз'
}
obj.res.value=s
}
//-—>
</script>
</HEAD>
<BODY>
Аты бойынша күннің нөмірін анықтау
<FORM name="form1">
Күннің атын енгізіңіз: <input type="text" size=20 name="numl"><hr>>
Күннің нөмірі:
<input type="button" value=Анықтау onClick="daynum (form1)">
<input type="text" size=25 name="res"><hr>>
<input type="reset" value="Жаңарту">
</FORM>
</BODY>
</HTML>
```



4.10-сурет. Программа нәтижесі

Бақылау сұрақтары:

1. JavaScript тілінің қолданылатын аймақтары.
2. Меншіктеу тәсілдеріне сипаттама беріп, мысалдар келтіріңіз.
3. Шартты командалар қалай жазылады?
4. Ақпаратты енгізу және шығару әрекеттері қалай орындалады?
5. Ауыстырғыштың орындалу ретіне сипаттама беріңіз.
6. switch конструкциясына сипаттама беріңіз.

Тапсырмалар:

1. Күннің нөмірі бойынша оның атын анықтау программасын жазыңыз.
2. Айдың нөмірі бойынша оның қандай кварталға жататынын анықтау программасын жазыңыз. Мысалы 1квартал (1,2,3), 2 кватал (4,5,6) т.с.с

3. Қызметкердің ағымдағы жылдың бірінші кварталына арналған табысы жөнінде ақпарат енгізіледі. Келесілерді анықтаңыздар:

• бірінші кварталдағы жалпы табысты;

• табыс салығының қосындысын (13%);

• қолға алатын кварталдағы табысты.

4.Жазықтықтағы үш нүктенің координаталары берілген. Үшбұрыштың ауданын анықтайтын сценарий құрыңыз.

5. Жазықтықтағы координатасы берілген нүктенің координаталардың бас нүктесіне дейінгі ара қашықтығын анықтау сценарийін жазыңыз.

6. Енгізілген екі айнымалының мәндерін бір-бірімен алмастыратын сценарий жазыңыз.

7. Радиусы белгілі болған жағдайда шардың көлемін анықтау сценарийін жазыңыз.

8. Дөңгелектің радиусы берілген. Шеңбердің ұзындығын және сәйкес шеңбердің ауданын есептеу сценарийін жазыңыз.

9. Жазықтықта координаталары берілген екі нүктенің ара қашықтығын анықтау программасын құрыңыздар.

Логикалық операциялар

Шарт ретінде логикалық өрнектер де жазыла береді, ондайда келесі логикалық операциялар қолданылады:

Белгіленуі	Сипаттамасы	Мысалы
==	Тең	x + 1 == 8
!=	Тең емес	str != "да"
>	Үлкен	x * y > 5
>=	Үлкен немесе тең	d >= 0
<	Кіші	num < 10
<=	Кіші немесе тең	bonus <= 5
&&	Логикалық ЖӘНЕ	1 < x && x < 10
||	Логикалық НЕМЕСЕ	x == 1 || x == 10
!	Логикалық ТЕРІСТЕУ	!(1 < x && x < 10)

Блок

Жүйелік жақшаға алынған командалар тізбегі {команда1; команда2; ...} бір команда ретінде орындалады. Мұндай күрделі команда құрама немесе блок деп аталады. Жақшадан «}» кейін «;» символы қойылмайды. Блок арқылы орындалатын шартты команда мысалдары:

Жалпы түрі

```
if(шарт)
{
  ...
  командалар
  ...
}
else
```

```
 {
  ...
  командалар
  ...
 }
```
Мысал
```
 if(d > 0)
 {
  xl = -(b + Math.sqrt(d))/(2*a);
  x2 = -(b - Math.sqrt(d))/(2*a);
  mes = “екі түбірі бар";
 }
 else if (d == 0)
 {xl = -b / (2 * a);
  mes = “бір түбірі бар";
 }
 else mes = “түбірі жоқ" ;
```
Шағын емтихан программасын жасайық. Емтихан бес сұрақтан тұрады. Бір дұрыс жауапқа бір балл беріледі. Нәтижесінде 2-ден 5-ке дейінгі бір баға қойылады:

- баға = 2, егер дұрыс жауаптар саны 3-тен аз болса;
- әйтпесе баға = дұрыс жауаптар саны.

Мұның программасы төмендегідей болады:
```
<HTML>
<HEAD>
 <TITLE>Логикалық өрнектер бойынша емтихан </TITLE>
</HEAD>
<BODY bgcolor=white text=black>
  <H1>Логикалық өрнектер бойынша емтихан </H1>
  <HR>
  <SCRIPT language=JavaScript>
  <!--
        var bonus =0; // Дұрыс жауаптар саны.
        var num =1; // Сұрақ нөмірі.
        var question; // Сұрақ.
  // 1-сұрақ.
  question = "Сұрақ " + num + ". Дұрыс жауап саны = " + bonus + ".\n Егер
x=5,то 1<x  &&&& x<10 тең  true немесе false?";
  if(prompt(question," true")=="true") bonus++; num++;
  // 2-сұрақ.
  question = "Сұрақ " + num + ".Дұрыс жауаптар саны  = " + bonus + ".\n Егер
x=5, то !(l<x &&&& x<10) тең true немесе false?";
  if(prompt(question, " true")=="false") bonus++;  num++;
  // 3-сұрақ.
```

```
  question = "Сұрақ " + num + ". Дұрыс жауаптар саны   = " + bonus +".\n Егер
x=5, то x != 5 тең
   true немесе false?";
  if(prompt(question," true")=="false") bonus++;  num++;
// 4-сұрақ.
  question = "Сұрақ " + num + ".Дұрыс жауаптар саны = " + bonus + ".\n Егер
x=5, то x++ != 5 тең
   true немесе false?";
  if(prompt(question, " true")=="false") bonus++;  num++;
  // 5-сұрақ.
  question="Сұрақ "+num+".Дұрыс жауаптар саны="+bonus+   ".\n Егер x=5,то
++x !=5 тең true немесе false?";
  if(prompt(question, "true") == " true") bonus++;
  // Нәтижені alert терезесінде көрсету.
  if(bonus < 2) bonus = 2;
  alert("Сіздің бағаңыз: " + bonus);
 //-->
 </SCRIPT>
 Емтихан аяқталды!
</BODY>
</HTML>
```

Ескерту. Браузерлердің NN және IE төртінші нұсқасынан кейінгілері үшін экранға бір «&» символ орнына екі «&&» символын жазу керек.

Арифметикалық және тіркестік өрнектерді шарт ретінде пайдалану

Шартты if командасының шарты ретінде логикалық операциялардан бөлек арифметикалық және тіркестік өрнектерді де жазуға болады. Мұнда егер өрнек мәні 0-ге немесе бос мәнге"" тең болса, ол жалған, ал қалған жағдайда ақиқат болып саналады.

Мысал

```
var x = 20;
if(20 - x) x ++;
var x = 20;
if(2 * x) x ++;
var x = "";
if(x) x += "киса";
var x = "";
if(x+"?") x += "киса";
var x = "";
if(!x) x += "киса";
```

Түсініктеме

```
Шарт жалған, x мәні өзгермейді

Шарт ақиқат, x мәні бірге артады
Шарт жалған, x мәні өзгермейді

Шарт жалған, x мәні «киса» сөзіне тең
Шарт ақиқат, x мәні «киса» сөзіне тең
```

JavaScript тілінде айнымалылар логикалық типтегі мәндерді қабылдай алады. Мұндай тип true және false тәрізді екі түрлі мән қабылдайды.

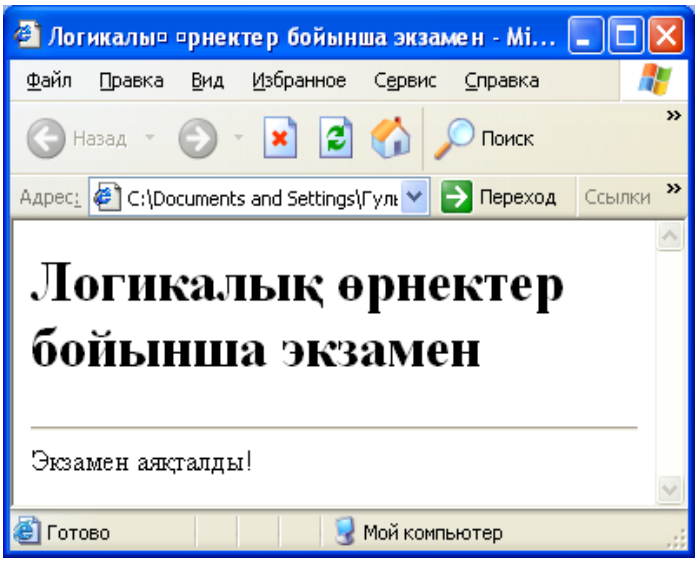
Мысал	Түсініктемелер

`var x = true;`
`var y = 2;`
`if (!x) y ++;`
	Команда орындалған соң, y = 3

`var x = true;`
`var y = 2;`
`if ( !x) y ++;`
	Команда орындалған соң, y = 2

`var x = true;`
`var y = 2;`
`if (x || !x) y ++;`
	Команда орындалған соң, y = 3

`var x = true;`
`var y = 2;`
`if (x && !x) y ++;`
	Команда орындалған соң, y = 2



4.11-сурет

Қиын шарттар

Құрамына арифметикалық және тіркестік (строковые) өрнектер кіретін шарттар алғашқы кездерде қиындықтар туғызады. Оларды түсіну үшін алдымен шарттар құрамындағы арифметикалық және тіркестік өрнектер мәнін есептер алу керек, сонан соң барып олардың мәні логикалық true және false мәндерімен салыстырылады. «Қасқыр» және «крокодил» сияқты тіркестік тұрақтылар және де 25 және 3.14 тәрізді сандық тұрақтылардың барлығы да логикалық true мәніне сәйкес келеді. Сәйкесінше, олардың логикалық терістеу мәндері false болады.

Мысал	Түсініктемелер
`if ("қасқыр")`	`if ( true) орындалады`
`if ("крокодил")`	`if ( true) орындалады`
`if (25)`	`if ( true) орындалады`
`if (3.14)`	`if ( true) орындалады`
`if (!"қасқыр")`	`if (false) орындалады`
`if (!"крокодил")`	`if (false) орындалады`
`If (!25)`	`if (false) орындалады`
`if (!3.14)`	`if (false) орындалады`

Мысал	Түсініктемелер
"қасқыр" && 25	true && true сияқты, мәні – **true**
!"қасқыр" && 25	! true && true сияқты, мәні – **false**
!"қасқыр" || 25	! true || true сияқты, мәні – **true**
!"" || 22 – 22	true || false сияқты, мәні – **true**
! (("" + "мысық") || (22-22))	false || false сияқты, мәні – **false**

Соңғы мысалды түрлендіре отырып түсіндірейік:
!("" + "мысық") || 22 - 22
!"мысық" || 0
! true || false
false || false
false

Балама мән енгізу

Егер екі балама мәннің бірін таңдап алу керек болса, онда prompt функциясын пайдалану керек.

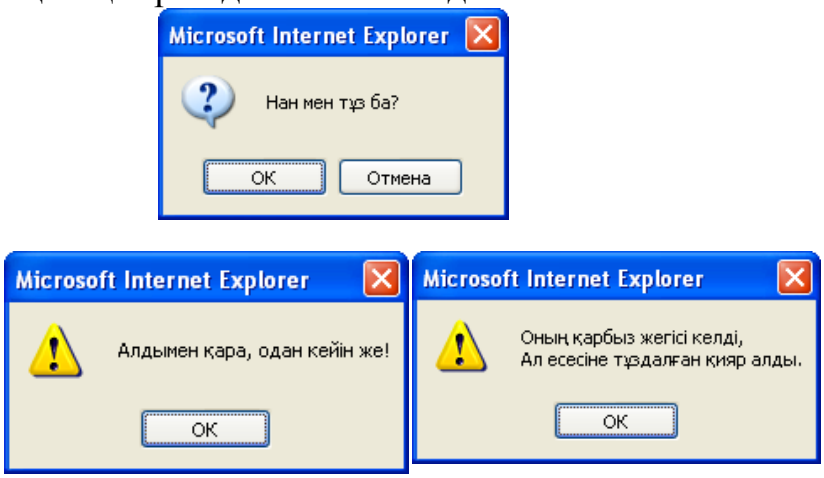
Бірақ бұдан гөрі қарапайым әрі жеңіл тәсіл бар – ол confirm функциясын пайдалану: confirm("жазу");

Бұл функция былай жұмыс істейді. Экранға екі батырмасы бар терезе шығады: Егер ОК батырмасын шертсек, confirm функциясы true мәнін қайтарады, ал егер Отмена (Cancel) батырмасы шертілсе – false мәні қайтарылады. Бұл батырмалардың орнына оларға эквивалентті <Enter> және <Esc> пернелерін де пайдалануға болады.

Мысалы:

(confirm ("Нан мен тұз ба?")) alert ("Алдымен қара, одан кейін же!");
else alert("Оның қарбыз жегісі келді,\n" + "Ал есесіне тұздалған қияр алды");

Осы кодты орындағанда, экранға төмендегі сурет терезе шығады. ОК батырмасы шертілсе, сол жақ терезедегі мәнді шығарады, ал Отмена басылса, оң жақ терезедегі мән шығады.



4.12-сурет. Программа нәтижесі

Бақылау сұрақтары:

1. Шартты команданың жалпы жазылу түрі қандай?
2. Логикалық типтің мәндеріне сипаттама беріңіз.

3. Блок немесе күрделі команда деген не?
4. Қиын шарттар қалай орындалады? Мысалдар келтіріңіздер
5. Балама мән енгізу қалай орындалады? Confirm функциясы қалай жұмыс істейді?

Тапсырмалар:

1. Бес сан енгізілген. Сол сандардың үлкенін анықтау программасын жазыңдар.

2. Қабырғаларының ұзындықтары берілген. Үшбұрыш тұрғызуға болатынын немесе болмайтынын анықтаңдар.

3. Жазықтықтағы нүктенің координатасы берілген. Осы нүктенің координаталардың тікбұрышты жүйесінің қай ширегінде жататынын анықтау сценарийін құрыңыз.

4. Тестке қатысушыларға алты есеп берілген. Әрбір есептің нәтижесіне: 0, 1/3, 2/3, немесе 1 деген бал қойылады. Тест тапсырушылардың барлығын нәтиженің төрт категориясы бойынша топтайды. Бірінші категорияға барлық есептерді ең жақсы 1 деген балмен бағаланған қатысушылар кіреді. Екінші категорияға барлық есептер нәтижелері 2/3-тен кем емес және ең болмағанда бір есебінің нәтижесі 1 бал болатын қатысушылар кіреді. Үшінші категорияға барлық есептердің шешімі 2/3-ке бағаланған қатысушылар, ал қалған қатысушылардың барлығы төртінші категорияға жатады.

• Нақты тестке қатысушы үшін сауалнама құрыңыз. Сауалнамада қатысушының тегі, оқитын оқу орны, есепке берілетін балы көрсетілетін болсын. Қатысушыға арналған сауалнаманы өңдеу кезінде жинаған балдарының қосындысы және оның қандай категорияға жататыны анықталуы керек.

• Әрқайсысында категория нөмірі көрсетілген төрт сурет әзірлеңіз. Сауалнанаманы өңдеу кезінде тестке қатысушының категориясы көрінетін сценарий құрыңыз.

5. Сессия емтихандерін тапсыру нәтижесі бойынша келесі ереже бойынша стипендия тағайындалады. Барлық алты емтиханді 5-ке тапсырған студент 50 мың теңге, үш емтиханін 5-ке қалғандарын 4-ке тапсырған студенттер үшін 25 мың теңге, барлық емтихандерді 4-ке тапсырған студенттер үшін 10 мың теңге. Қалған студенттер үшін стипендия тағайындалмайды.

• Студент үшін оның тегі, тобының нөмірі және барлық алты емтиханнен алған нәтижелері көрсетілетін сауалнама құрыңыз. Сауалнаманы өңдеу кезінде, емтихан кезіндегі орта балы мен тағайындалған стипендия мөлшері анықталатын болсын.

• Әрқайсысында стипендия мөлшері көрсетілген төрт сурет дайындаңыз. Сауалнаманы өңдеу кезінде студенттің стипендиясының мөлшері көрсетілетін сценарий құрыңыз.

4.7. Циклдер мен функциялар

Белгілі бір әрекеттердің қайталанып орындалуын цикл операторларының көмегімен ұйымдастыруға болатындығы бізге белгілі.

JavaScript программалау тілінде циклді ұйымдастырудың өзіндік ерекшеліктері бар.

4.7.1 While циклі

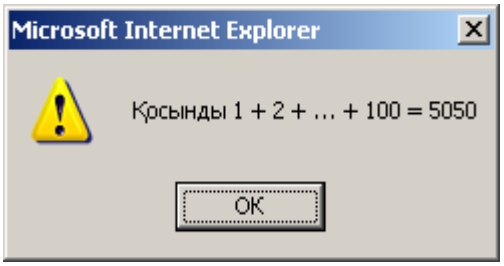
Цикл орындалуы: алдымен шарт тексеріледі. Егер ол ақиқат болса, командалар (цикл денесі) орындалады. Келесі жолы да осы әрекеттер қайталанады, яғни шарт тексеріледі, егер ол ақиқат болса, цикл орындалады, т.с.с. Кезекті тексеру кезінде шарт жалған болған кезде, цикл жұмысы аяқталады. Циклда шарт алдын ала тексерілетін болғандықтан, ол бір де бір рет орындалмауы да мүмкін.

Жалпы жазылу түрі:

```
while (шарт)   команда;
```

Мысал:

```
var i = 1;
var sum = 0;
while(i <= 100)
{
 sum += i;
 i ++;
}
alert("Қосынды 1 + 2 + ... + 100 = " + sum);
```



4.13-сурет. Мысалды

4.7.2 For циклі

Төменде for циклының жалпы жазылу түрі мен алдыңғы мысалдың осы команда арқылы орындалуы көрсетілген.

Жалпы жазылу түрі:

```
 for(цикл басы; шарт; қадамы)
  команда;
```

Мысалдар

```
 var i;
 var sum = 0;
 for(i=1; i<=100; i ++)
  sum += i;
 alert ("Қосынды 1 + 2 +  ... + 100 = " + sum);
```

Бұл мысалдың да нәтижесі алдыңғы мысалдағыдай болады. Цикл жұмысы келесі түрде атқарылады: циклдегі алғашқы команда орындалады (мысалдағы i=1; командасы), ал цикл келесі әрекеттер арқылы орындалады:

- шартты тексеру (мысалдағы i<=100);
- цикл денесін орындау (мысалдағы sum+=i);
- қадамды көрсету командасын орындау, (мысалдағы i++).

Егер шарт бірден жалған болса, while командасындағы сияқты цикл денесі бір де бір рет орындалмауы да мүмкін. Мұндайда қадам беру

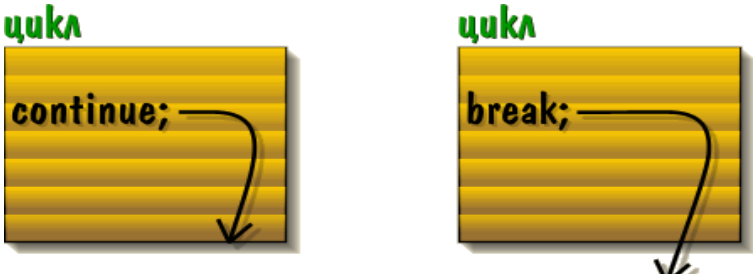
командасы да орындалмайды. Ал циклдің алғашқы командасы әрқашанда кем дегенде бір рет орындалады.
Цикл басындағы for жолының үш командасының - басы, шарты, қадамы кез келгені жазылмай кетуі де мүмкін, бірақ олардың арасында тұратын нүктелі үтір міндетті түрде сақталады. Егер шарт көрсетілмесе, оның мәні ақиқат (true) болып саналады. Мұндайда, цикл шексіз түрге айналып кетеді:
for (;;) команда

Бұл цикл шексіз орындала береді, яғни аяқталмайды. Оны тек цикл ішіндегі break командасы көмегімен ғана аяқтауға болады.

4.7.3 Break және continue командалары

Бұл командалар циклдағы командалардың орындалу реттілігін өзгерту үшін қолданылады.

Continue командасы циклдың онан кейінгі тұрған барлық командаларын аттап өтіп, цикл параметрінің келесі мәніне көшіреді. **Break** командасы жалпы цикл орындалуын аяқтап, одан кейінгі келесі командаларға көшіреді.

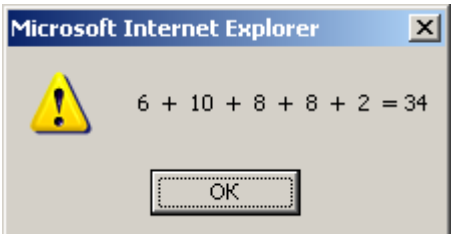


4.14-сурет. Continue және Break командаларының орындалуы

1-мысал (continue). [1,20] аралығынан кездейсоқ түрде алынған 5 жұп санның қосындысын табу керек.

```
var   len     = 5;   // Сандардың қанша екендігі.
var   a       = 1;   // Аралықтың сол жақ шекарасы.
var   b       = 20; // Аралықтың оң жақ шекарасы.
var  sum     = 0;   // Қосқыш.
var  counter = 0;  // сандардың санауышы.
var  number;       // Кездейсоқ сан.
var  str = " ";        // Шығаруға арналған қатар.
while (counter < len)
{  number = Math.floor(a + (b-a+1)*Math.random());
   if (number % 2) continue;
   sum += number;       str += number;
   if (counter  <  len-1)  str  +=  "  +  ";
   else                          str  +=  "  = ";
   counter++;
}
str += sum;  alert(str);
```

Осы мысалдың орындалу нәтижесінің бірі келесі суреттегідей болады. Программаны әр орындаған кезде 1 мен 20 сандарының арасынан жұп сандар кездейсоқ алынатын болғандықтан, нәтиже әрбір программаны қайталап орындаған кезде өзгеріп отырады.



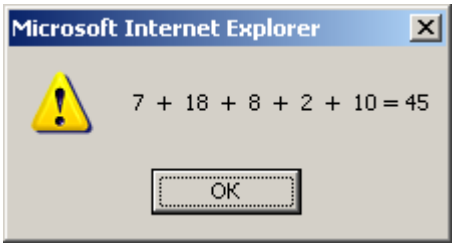
4.15. сурет. 1-мысалдың орындалу нәтижесі

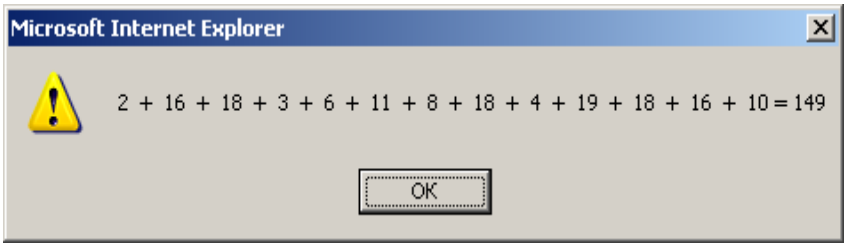
Math.random() стандартты функциясы [0,1] аралығынан кез келген кездейсоқ сан береді. Math.floor(num) стандартты функциясы аргументке тең не одан кіші бүтін сан мәнін береді.

2-мысал (break). Керекті сандар [1,20] аралығынан кездейсоқ түрде алынып отырады. Осы сандардың кезекті келесісі 10-ға тең болғанша, олардың қосындысын табу керек.

```
var a       = 1;      // аралықтың сол жақ шекарасы.
var b       = 20;     // аралықтың оң жақ шекарасы.
var special = 10; // кездейсоқ санның критикалық мәні
var sum     = 0;   // қосқыш.
var number;        // кездейсоқ сан.
var str     = "";      // шығару жолы.
for ( ; ; )              // шексіз цикл.
{  number = Math.floor(a + (b-a+1)*Math.random( ));
   sum += number;   str += number ;
   if (number == special)   break;
   str  +=  "  +  "; }
str += " = " + sum;   alert(str);
```

Бұл мысалда да сандар кездейсоқ алынатын болғандықтан әр түрлі нәтиже береді (4.16-сурет).

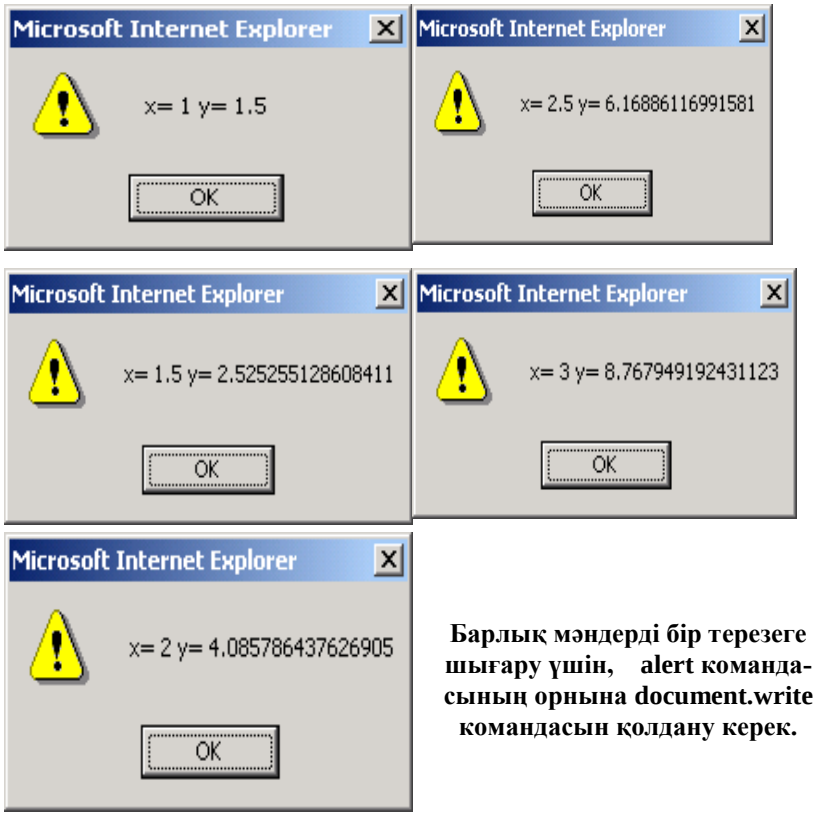




3- мысал. $y = x^2 - \sqrt{x} + 1.5$ функциясы берілген. x аргументі 0-ден 3-ке дейін, 0,5 қадаммен өзгергенде, y функциясының мәндерін табу керек.

```
var  x0 = 0;      // x-тің бастапқы мәні
var  xk = 3;      // x-тің соңғы мәні
var  dx = 0.5;    // x-тің өзгеру қадамы
var  x=x0;        // x-ке алғашқы мәнін меншіктеу
var y;       // y айнымалысын сипаттау
while (x =< xk)
{  y = x*x - Math.sqrt(x) +1.5;
   alert(" x= " + x + " y= " + y);
   x+=dx ;
}
```

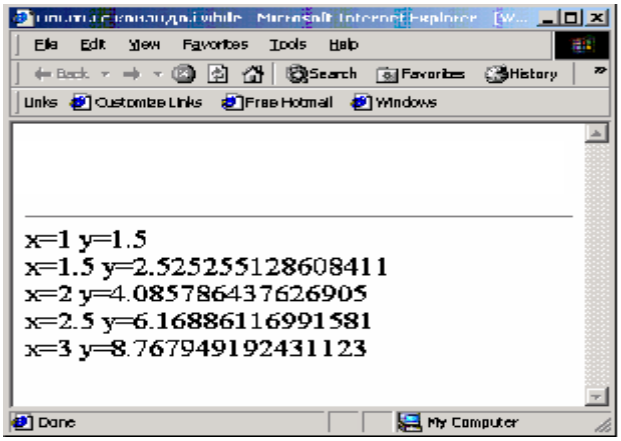
Бұл программа орындалғанда x-пен y-тің әрбір мәндері жеке терезелерге шығады (4.17-сурет).



Барлық мәндерді бір терезеге шығару үшін, alert командасының орнына document.write командасын қолдану керек.

4.17-сурет. 3-мысалдың орындалу нәтижесі

Осы мысалдың document.write командасының көмегімен орындалуының нәтижесі келесі суретте көрсетілген:



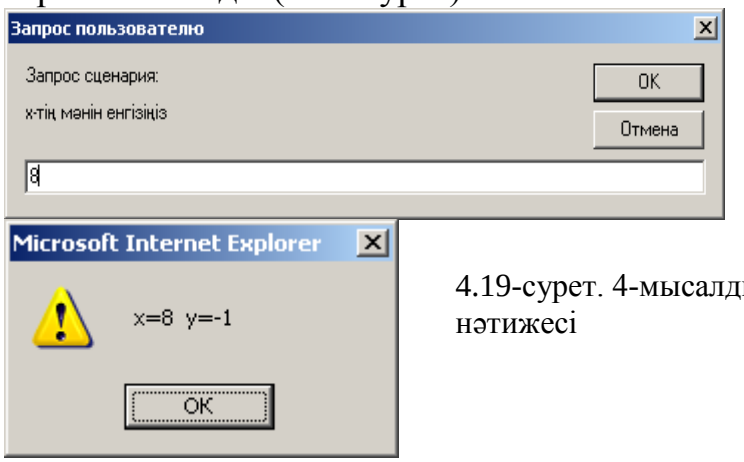
4.18-сурет. document.write командасының көмегімен

4-мысал. $y = \begin{cases} x^2 + x + 1, & \text{егер } x \le 0 \\ x - \sqrt{x+1}, & \text{егер } x > 0 \end{cases}$ функциясы берілген.

х-тің мәнін енгізе отырып, у функциясының мәнін анықтау керек.

```
var x = prompt("х-тің мәнін енгізіңіз", "1");
 y = (x <= 0) ? x*x+x+1:x-Math.sqrt(x+1);
 alert ("x="+x+"  y=" + y);
```

Программаны орындаған кезде, х-тің мәнін енгізуге мүмкіндік беретін терезе ашылады (4.19-сурет).



4.19-сурет. 4-мысалдың орындалу нәтижесі

Енді осы функцияның мәнін, х-тің мәні 0-ден 3-ке дейін, 0,5 қадаммен өзгерген кезде анықтау керек болсын.

```
var  x0 = 0;    // х-тің бастапқы мәні
 var  xk = 3;    // х-тің соңғы мәні
 var  dx = 0.5;  // х-тің өзгеру қадамы
 var  x = x0;    // х-ке алғашқы мәнін меншіктеу
  var y;         // у-ті сипаттау
  while (x =< xk)
  {  y = x*x - Math.sqrt(x) +1.5;
     alert(" x= " + x + " y= " + y);
```

```
    x+=dx ;
  }
```

4.7.4 Циклдерді кері бағытта программалау

Келесі мысалда set жиымы (массиві) элементтері қосындысын табу жолы көрсетілген (элементтерді нөмірлеу 0-ден басталады; set.length – жиым қасиеті – оның ұзындығын білдіреді):

```
var sum = 0;
for(var i=0; i < set.length; i++)
   sum += set[i];
```

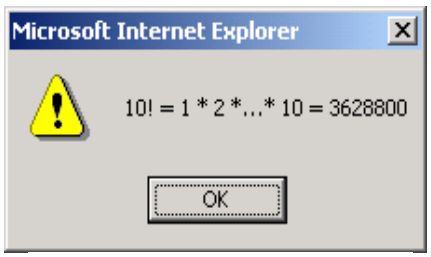
Былай да жазуға болады:

```
var sum = 0;
for(var i = set.length; -- i >= 0; )
   sum += set[i];
```

Екінші тәсіл дұрысырақ, өйткені санды нөлмен салыстыру басқа санмен салыстыруға қарағанда, жылдамырақ орындалады. Басқаша айтсақ, «--i >= 0» шарты «i < set.length» шартына қарағанда тиімдірек болады. Жалпы ереже: санауышы бар циклде санауыш мәні мүмкіндігінше нөлге дейін төмендегені дұрыс болып табылады.

Келесі мысалда 10!= 1*2*. . .*10 мәні есептеледі.

```
var proizvedenie = 1;
for(var i=10; i; --i)
   proizvedenie *= i;
alert("10!=1 * 2 *...* 10 =
"+proizvedenie);
```



4.20-сурет. 10! Есептеу

Келесі нұсқа:

```
var sum = 0;
for(var i = set.length; (i --) >= 0; )
   sum += set[i];
```

қате болып табылады, өйткені ондағы і мәнін бірге кеміту шартты тексеруден кейін жазылған, ол оның алдында тұруы тиіс.

4.7.5 Цикл айнымалысын сипаттау

Цикл айнымалысын JavaScript тілінде келесі түрде жазуға болады:

for(var i = 100; i; i --) ...

немесе

```
var i;
for(i = 100; i; i --) ...
```

Браузер үшін бұл екеуі де бірдей болады. Келесі цикл құрылымында:

for(басы; шарт; өсуі)

команда;

басы командасы цикл басында бір-ақ рет орындалады. Сондықтан і айнымалысы 100 рет var сипаттауышы арқылы тексерілмейді.

Келесі ұсыныстар беріледі:

- і цикл санауышын функция немесе скрипт басында басқа айнымалылар сипатталаған кезде сипаттап кету керек;
- егер бір санауыш бірнеше циклде пайдаланылатын болса, оны жалпы сипаттау бөлімінде көрсету қажет;
- егер функция құрамында цикл біреу ғана болса, онда **for (var i = …,…; …)** – түрінде жазған дұрыс, өйткені і айнымалысы бір-ақ рет кездеседі.

Егер бір digit айнымалысы цикл ішінде пайдаланылып, сыртында кездеспесе, оны циклдың ішінде локальды айнымалы ретінде сипаттаған дұрыс.

Төмендегі while командасының жазылуы:

```
while (...)
{ var digit = ...
  ... }
```

мынадай сипаттаумен бірдей болып саналады:

```
var digit;
...
while (...)
{... digit = ...}
```

Бақылау сұрақтары:

1. While операторының көмегімен цикл қалай ұйымдастырылады?
2. Циклдің орындалу реттілігіне түсініктеме беріңіз.
3. For операторының көмегімен цикл қалай ұйымдастырылады?
4. Break және continue командалары қай кезде қолданылады? Олардың айырмашылықтар қандай?
5. [0,1] аралығынан кездейсоқ санды беру үшін қандай функция қолданылады?
6. Циклді кері бағытта программалау дегенді қалай түсінуге болады?
7. Цикл айнымалысын қалай сипаттауға болады? Оның қандай нұсқалары бар?
8. Функцияны кестелеуде нәтижені жеке терезелерге шығаруды қалай ұйымдастыруға болады?

Тапсырмалар:

1. 1-ден 20-ға дейінгі кездейсоқ тақ сандардың қосындысын есептеу программасын құрыңыз.
2. y=5x+1 функциясының мәнін, x-тің мәні 1 мен 5 аралығында 0,5 қадаммен өзгергенде анықтауды жеке терезеге және бір терезеге шығару программасын құрыңыз.

3. $y = \begin{cases} x^4 + x^2 + 1, & \text{егер } x \le 0 \\ \sqrt{x+1}, & \text{егер } x > 0 \end{cases}$

функциясының мәнін х-тің мәні -2 мен 2 аралығында 0,1 қадаммен өзгергенде анықтауды жеке терезеге және бір терезеге шығаруды ұйымдастырыңыз.

4. 5-тен 15-ке дейінгі кездейсоқ сандардың келесісі 9-ға тең болғанда, олардың қосындысын есептеуді тоқтатып, нәтиже шығаратын программа құрыңыз.
5. 1-ден 5-ке дейінгі кездейсоқ жұп сандардың көбейтіндісін табу программасын құрыңыз.

4.8 JavaScript функциялары

Барлық стандартты математикалық функциялар мен жиі қолданылатын тұрақтылар Math - математикалық объектісінің тәсілдері мен қасиеттері түрінде беріледі.

Math объектісі JavaScript тілінің ішкі объектісі болып табылады. Мысалы, PI тұрақтысын пайдалану үшін Math.PI түрінде жазу керек. JavaScript тұрақтыларының дәлдігі үтірден кейінгі 15-цифрды қамтиды. Math функцияларын да тәсіл ретінде пайдаланып, мысалы, синус функциясы - Math.sin(*argument*), түрінде жазылады, мұндағы *argument* функция аргументі болып табылады.

4.8.1 Стандартты тұрақтылар

1. Натуралдық логарифм негізі – E болып жазылады.

Мысалы:

Alert(Math.E)

2. 10 санының натуралдық логарифмі – LN10.

мысалы:

Alert(Math.LN10)

3. 2 санының натуралдық логарифмі – LN2.

мысалы:

Alert(Math.LN2)

4. ПИ саны – PI

мысал:

Alert(Math.PI)

5.1/2 санының квадраттық түбірі – SQRT1_2

мысал:

Alert(Math.SQRT1_2)

6. 2 санының квадраттық түбірі – SQRT2

мысал:

Alert(Math.SQRT2)

4.8.2 Стандартты функциялар

1. Аргументтің абсолюттік мәні – abs(arg)

мысал:

```
var x = -25;
var y
```

```
y = Math.abs(x);
alert(y);
```
2. sin(arg), cos(arg), tan(arg) тригонометриялық функциялары:
мысал:
```
var x = Math.PI/4;
var y
y = Math.sin(x);
alert(y);
```
3. Кері тригонметриялық функциялар asin(arg), acos(arg),
atan(arg) :
```
var x = 0.5;
var y
y = Math.asin(x);
alert(y);
```
4. Экспонента және натуралдық логарифм: exp(arg), log(arg)
```
var x = 1;
var y
y = Math.exp(x);
alert(y);
```
5. Аргументтен үлкен немесе соған тең сан: ceil(arg)
```
var x = -2.69;
var y
y = Math.ceil(x);
alert(y);
```
6. Аргументтен кіші немесе соған тең сан: floor(arg)
```
var x = -2.69;
var y
y = Math.floor(x);
alert(y);
```
7. Аргументті жақын бүтін санға дейін дөңгелектеу: round(x)
```
var x = -2.69;
var y = Math.round(x);
alert(y);
```
8. Экспонента мен натуралдық логарифм: exp(arg), log(arg)
```
var x = 1;
var y
y = Math.exp(x);
alert(y);
```
9. аргументтің квадраттық түбірі : sqrt(arg)
```
var x = 3;
var y
y = Math.sqrt(x);
alert(y);
```

10. Екі аргументтің үлкенін немесе кішісін анықтау max(arg1, arg2), min(arg1, arg2) :

```
var x = 3;
var y = Math.min(x,10);
alert(y);
```

11. Санның дәрежесін табу pow(arg1, arg2) : arg1-дің arg2 дәрежесі

```
var x = 2;
var y = Math.pow(x,10);
alert(y);
```

12. [0,1] аралығынан кездейсоқ сан алу: random()

```
alert(Math.random());
```

4.8.3 Тұтынушы функциялары

Программалауда бір командалар тобын бірнеше рет қайталауға тура келетін жағдайлар жиі кездеседі. Егер қайталау «бір орында» орындалатын болса – (while, for) циклдар қолданылады. Ал егер кодтарды программаның әр жерінде қайталау қажет болса – оны тұтынушы функциясы ретінде жазу керек болады.

Функцияларды сипаттау және пайдалану. Мысалы, егер берілген бүтін санның цифрларының қанша екенін анықтау керек болса, оны бір рет анықтап алып, одан кейін математикадағы сияқты F(num) деп жазып, ол функцияның ішкі әрекетін қарастырмай, кодтарын жазбай, нәтижесін есептеуге болады. Мысалы, тұтынушы үш сан енгізіп солардың жалпы цифрлары санын анықтауы қажет болсын.

Осы мысалдың программасы:

```
 // Бүтін оң num санындағы цифрлар санын анықтау
 // Енгізу: num (бүтін оң сан).
    // Шығару: num санындағы цифрлар саны.
   function F(num)
{
 var len = num ? 0 : 1;
 while(num)
 {
         num = (num - num % 10) / 10;
         len ++;
 }
 return len;
}
var sum = 0;
var num1, num2, num3;
num1 = prompt("Бірінші санды енгізіңіз", "");
sum += F(numl) ;
num2 = prompt("Екінші санды енгізіңіз", "");
sum += F(num2) ;
```

```
num3 = prompt("Үшінші санды енгізіңіз", "");
sum += F(num3) ;
alert("Енгізілген цифрлардың жалпы саны: " + sum);
```

Браузер бұл скрипті былай орындайды. Бұл жазу:

```
function F(num)
{
  ...
  }
```

орындалмайды, бірақ браузер: «F атты функция сипатталды» деген мәліметті есте сақтайды. Өйткені function түйінді сөзі және оның блогы {...} командалар тізбегі емес, тек декларация (хабарлама, сипаттама) ғана болып саналады.

Функциядан кейін орналасқан командалар әдеттегідей орындалады. Соның ішінде F функциясы кездессе, браузер функция сипаттамасына оралып, оның формальды аргументі num орнына num1, num2, num3 тәрізді нақты параметрлерді қойып орындап шығады. Негізінде параметрлерді ауыстыру кезінде браузер F функциясын орындамай тұрып, мынадай меншіктеулер жасайды:

num = num1; — сонан соң F(numl)орындау;
num = num2; — сонан соң F(num2) орындау;
num = num3; — сонан соң F(num3) орындау.

Жалпы функцияны сипаттау былай атқарылады:

function Функция_аты(үтірмен бөлінген формальды аргументтер тізімі)
{
...
функция денесі
...
return (мәні);
}

return командасы программада қолданылатын функция мәндерін анықтайды, ол бірнешеу болуы да мүмкін, тіпті болмауы да мүмкін. Ол болмаса, функция ешқандай мән бермейді, мұндайда функцияны өрнектерге енгізуге болмайды. Мысалы, мәні жоқ функцияны меншіктеу командасында пайдалануға болмайды.

Функцияны шақыру (пайдалану) былай орындалады:

Функция_аты (үтірмен бөлінген нақты аргументтер тізімі)

Нақты аргументтер ретінде тұрақты, айнымалы, өрнек немесе сан қолданылады. F атауын функцияға дұрыс берілген атау деп айта алмаймыз, мұнда математикадағы сияқты функция бір әріппен белгіленген. Программалауда бір әріппен емес бір сөзбен белгілеу қалыптасқан, ол функция әрекетін білдіретін сөз болуы тиіс. Жоғарыда келтірілген мысалда DlinaChisla немесе LenOfNumber атаулары F атауына қарағанда түсініктірек болар еді. Функция аты шектелмейді, бірақ ол бір сөзден ғана тұруы тиіс (бос

орын болмауы керек). Атау латын әріптері мен цифрлардан тұрады, астын сызу таңбасын қолдануға болады. Орыс, қазақ әріптерін пайдалануға болмайды. Атаудың алғашқы символы әріп немесе астын сызу таңбасы болуы керек. Әріптер регистрі бірдей болып қабылданбайды. Мысалы, «LenOfNumber» және «Lenofnumber» атаулары бірдей емес болып саналады.

«Жоғарыдан төмен қарай» жобалау

Функцияны бірнеше рет пайдаланғанда ғана, оның тиімділігі артады және оны әр түрлі программаларда пайдаланған дұрыс. Программалауда кітапханалық функциялар жиі қолданылады. Мәліметтерді енгізу/шығару, математикалық есептеулер, терезелер жасау, тышқан шертулерін өңдеу – көптеген программаларда кездеседі. Әрине, оларды бір рет дұрыстап программалап алу керек. Кейбір жиі орындалатын функциялар программалау жүйелерінде де қолданылады, мысалы, alert және prompt функциялары, олар браузердің өзінде кодталған. Дегенмен, функцияларды ең жиі әрі кең қолданылатын аймақ – программалар құру технологиясы, мұнда алдымен мәселені шешудің көп сатылы схемасы жобаланады да, әр сатысына жеке функция тағайындалады. Программалау сатының төменгі (түпкі) жағынан жоғары қарай біртіндеп жүргізіледі. Осылай жобалау тәсілі «шашыраңқы (нисходящая) технология» немесе «жоғарыдан төмен қарай» деп аталады. Ол адамның ойлау қабілетіне сәйкес етіп құрастырылған: біз жобаны барлық жағынан бірден қарастыра алмаймыз.

Бақылау сұрақтары:

1. JavaScript тілінің функциялары қалай жазылады?
2. Тұтынушы функциясы қай кезде қолданылады?
3. 10 санының квадрат түбірін жазыңыз.
4. x^5 қалай жазылады?
5. x пен y-тің үлкенін анықтау қалай жазылады?
6. 5 санынынң натурал логарифмі қалай жазылады?
7. x пен y-тің кішісін анықтау қалай жазылады?
8. Жоғарыдан төмен қарай жобалау дегенді қалай түсінесіз?

Тапсырмалар:

1. Төрт сан енгізіп олардың үлкенін анықтауды, екі санның үлкенін анықтау функциясын қолданып программа құрыңыз.
2. Санды дәрежелеу функциясы арқылы кез келген енгізілген санның 5 дәрежесін анықтау программасын құрыңыз.
3. Үш таңбалы санның цифрларының қосындысын анықтау программасын құрыңыз.
4. sinx пен cosx/2 функциясының мәндерінің үлкенін анықтау программасын құрыңыз.
5. Кез келген енгізілген санның квадрат түбірін, квадратын, абсолют шамасын шығаратын программа құрыңыз.

4.9 Объект түсінігі

Объект – бұл мәліметтер мен функциялар жиынынан тұратын бірыңғай конструкция немесе JavaScript терминологиясы бойынша қасиеттер мен тәсілдер жиыны болып табылады.

Функция = тәсіл (метод).

Айнымалы = қасиет (свойства).

Инкапсуляция термині «қара жәшік» ретінде қарастырылатын объектінің ішкі құрылымын жасыру деген сөз. Объектінің қасиеттері белгілі болып саналады, яғни олар - сырттан қол жеткізуге болатын айнымалылар. Бірақ бұл функциялар қалай құрылған, олар қандай алгоритммен жұмыс істейді, ол туралы программалаушыға айтылмайды. Программалаушы немесе объектіні тұтынушы адам объектінің қосымша ішкі функциялары мен айнымалылары бар ма, олар қол жеткізуге болатын қасиеттер мен тәсілдермен қалай байланысқан, ол жағын білмейді.

Объект және объектінің бір данасы (экземпляры)

Мысалы, нақты телевизор – бұл JavaScript терминологиясы бойынша объект емес, ол объектінің бір данасы (экземпляры). Объект болып зауытта нақты өнім шығаруға арналған құжаттамалар жинағы саналады. Конвейерден шығып жатқан барлық телевизорлар бейнелерінің *қасиеттері де* оларды басқаратын *тәсілдер* де бірдей дана (экземпляр) болып табылады. Программалауда да дәл осылай болады. Объект – бұл шаблон, құжаттар жиыны, ал объектінің бір данасы (экземпляры) оның жұмыстық көшірмесі ғана болып табылады.

Объект интерфейсі және объектінің ішкі құрылымы

Rectangle объектісін қарастырайық. Rectangle объектісі тіктөртбұрыштармен жұмыс істеуге мүмкіндік береді. Объектінің бір данасын жасау үшін былай жазу керек:

```
var x = new Rectangle(a,b);    // Мұндағы x Rectangle
 // объектінің бір данасы (экземпляры).
// a мен  b тіктөртбұрыш ені мен биіктігі.
// new сөзі бір дана жасау үшін керек.
```

Объектінің бір данасы жасалған соң, келесі тәсілдер мен қасиеттерді пайдалауға болады:

Қасиеттер: width, height; Тәсілдер square(), perimeter(), radius()

Пайдалану мысалы:

```
var p =x.perimeter();
var r = x.radius();
var m = x.width;
if(x.height > m)
  m = x.height;
```

Мұнда объект *интерфейсі* келтірілген, яғни объектімен қатынасуға қажет ақпарат берілген.

Мұнда square, perimeter, radius функцияларының программалық кодтары келтірілмеген. Басқаша айтқанда, объектінің ішкі құрылымы көрсетілмеген.

Жұмыс кезінде тек интерфейсті пайдаланып, оның ішкі құрылымын қажет етпеуге болады.

Объект интерфейсі – пайдалануға болатын объектінің айнымалылары мен функциялары болып табылады.

Объектінің ішкі құрылымы – программалау тілінде объектінің ішкі айнымалылары мен функцияларын сипаттау.

JavaScript тілінде *Rectangle* объектісінің x данасының қасиеттері мен тәсілдері келесі түрде жазылады:

x.height; x.perimeter();

Жалпы жазылу форматы мынадай болады:

дана_аты.объект_қасиеті_не_тәсілі_

Нүкте – сатылы бөлу таңбасы: ол атасын баласынан (тегін мұрагерінен) бөліп тұрады.

Құрамдас ішкі объектілер және тұтынушы объектісі

JavaScript тілінде ішкі құрамдас объектілер көп. Оларды программалау қажет емес, олар программалау тілі ішінде орнатылған. Бұл – браузердің программалық кодына осы объектілер коды кіреді деген сөз. Программалаушы осы объектілердің интерфейсін білуі тиіс, олардың бір данасын жасай білу керек, сонда ол өз қалауынша ішкі объектілерді пайдалана алады.

JavaScript жаңа объектілерді программалауға және олардың ішкі объектілерін өзгертуге мүмкіндік береді. Енді бірнеше ішкі объектілер жұмысын қарастырайық.

***Date* объектісі**

JavaScript тілінің ішкі объектілерін қарастыруды өте пайдалы болып саналатын Date объектісінен бастайық. Бұл объект күн, ай мерзімімен (датамен) және уақытпен жұмыс істеу үшін керек. Объект данасын жасау үшін (Date объектісінің ғана емес, одан басқасының да) JavaScript тілінде new түйінді сөзі қолданылады:

var now = new Date();

Енді now айнымалысы Date объектісі данасы болып табылады да, ол үстіміздегі мезгіл мен уақытты береді.

Жалпы дана жасау былай орындалады:

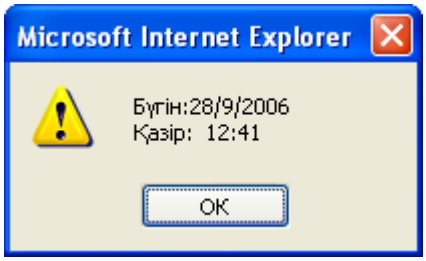
var айнымалы = new Date(параметрлер);

Келесі параметрлерді көрсетуге болады:

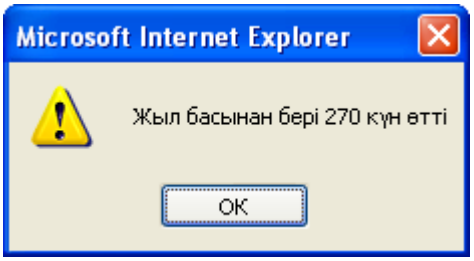
Мысалы:

```
var now = new Date();
var birthday =
  new Date(1954,1,8);

var bell = new Date(2003,
    0,14,12,20,0);
```

Date объектісі данасы құрылғаннан кейін, оның ішкі мәліметтерін көруге болады, оны өзгерту мүмкіндігі де бар. Ол үшін көптеген тәсілдер бар, олардың тізімі тілге арналған кітап қосымшаларында келтіріледі.
Объект тәсілі аты (тәсіл – JavaScript терминологиясында функция) дана атынан нүктемен бөлініп жазылады.
Былай жазуға болады:
var year = bell.getYear();
Date объектісінің бірнеше қарапайым скриптерін қарастырайық.
Ағымдағы мерзім және уақыт
var now = new Date();
alert("Бүгін:"+now.getDate()+"/"+ (now.getMonth()+1)+"/"+
now.getYear()+"\nҚазір: "+now.getHours() +":" + now.getMinutes());



Осы кодтарды жазған кезде скрипті орындау нәтижесінде мерзім мен уақытты көрсететін хабарлама шығады.
Жыл басынан бергі күндер саны
var now = new Date(); // Ағымдағы мерзім мен уақыт.
var begin = new Date(now.getYear(),0,1); // Ағымдағы жыл басы.
// жыл басы ішіндегі миллисекунд саны:
var num = now.getTime()-begin.getTime();
var msPerDay = 24 * 60 * 60 * 1000; // тәулік ішіндегі миллисекунд саны.
num /= msPerDay;
// жыл басынан бергі күннің саны.
// Нәтижесін көрсетеміз:
alert("Жыл басынан бері " + Math.floor(num) + " күн өтті");



Осы кодтарды жазған кезде скрипті орындау нәтижесі оң жақтағы суретте келтірілген

Date ішкі объектісі және оның тәсілдері

1. Тәсіл түрі – getYear(). Жыл нөмірінің соңғы екі цифрын береді.

Мысалы:
var d = new Date(2005,1,2);
var y = d.getYear();
alert(y);

2. Тәсіл түрі – setYear(). Жыл нөмірін тағайындайды.

Мысалы:
var d = new Date();
d.setYear(2004);
alert(d.getYear());

3. Тәсіл түрі – getMonth(). Ай нөмірі мәнін береді.

var d = new Date(2005,1,2);
var m = d.getMonth();

```
alert(m);
```
4. Тәсіл түрі – setMonth(). Айды енгізеді.

Мысалы:

```
var d = new Date();
d.setMonth(4);
alert(d.getMonth());
```
5. Тәсіл түрі – getDate(). Ай күнін береді.

```
var d = new Date(2005,1,2);
var m = d.getDate();
alert(m);
```
6. Тәсіл түрі – setDate(). Ай күнін енгізеді (тағайындайды).

Мысалы:

```
var d = new Date();
d.setDate(2);
alert(d.getDate());
```
7. Тәсіл түрі – getDay(). Апта күнін береді (жексенбі – 0, дүйсенбі – 1, …, сенбі - 6)

```
var d = new Date(2005,1,2);
var m = d.getDay();
alert(m);
```
8. Тәсіл түрі – getHours(). Уақыттың сағатын береді

```
var d = new Date();
var t = d.getHours();
alert(t);
```
9. Тәсіл түрі – setHours().

Сағатты енгізеді (тағайындайды).

Мысал:

```
var d = new Date();
d.setHours(22);
alert(d.getHours());
```
10.Тәсіл түрі – getMinutes(). Уақыт минутын береді.

```
var d = new Date();
var m = d.getMinutes();
alert(m);
```
11.Тәсіл түрі – setMinutes(). Уақыт минутын енгізеді.

```
var d = new Date();
d.setMinutes(32);
var t = d.getMinutes();
alert(t);
```
12.Тәсіл түрі – getTime().

1 янв 0 сағатынан 1970 ж. бергі миллисекунд мөлшерін береді.

Мысал:

```
var d = new Date();
var t = d.getTime();
```

alert(t);

***Array* объектісі**

Бұл объект мәліметтер жиымын (массивін) жасау үшін керек. *Жиым — элементтердің реттелген жиыны*. Жеке элементтің орны оның аты мен индексін (нөмір) көрсету арқылы орындалады. JavaScript тілінде элементтерді нөмірлеу нөлден басталады.

Мысал: апта күндерінің аттары жиымы.

var dayNames = new Array("жексенбі", "дүйсенбі", "сейсенбі", "сәрсенбі", "бейсенбі", "жұма", "сенбі");

Жиымның жеке элементтерін пайдалану үшін былай жазылады:

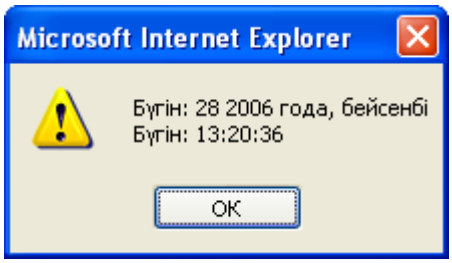
массив__аты [индекс]

Төмендегі скрипт:

var day Names = new Array("жексенбі", "дүйсенбі", "сейсенбі", "сәрсенбі", "бейсенбі", "жұма","сенбі"); alert(dayNames[0]);

жұмысы нәтижесінде alert терезесіне «жексенбі» мәтіні шығады.

Мысалы: ағымдағы дата мен уақытты көрсету.



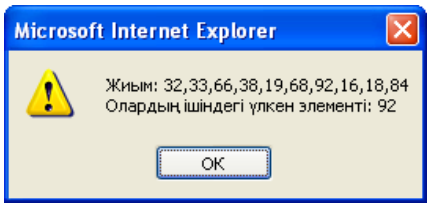

```
 // Ай аттарын көрсету:
 var monthNames = new Array ("қаңтар",
"ақпан",

"наурыз","сәуір","мамыр","маусым","шілде",
"тамыз",
    "қыркүйек","қазан", "қараша",
"желтоқсан");
 // Апта күндері:
 var dayNames = new Array ("жексенбі", "дүйсенбі", "сейсенбі", "сәрсенбі",
"бейсенбі", "жұма","сенбі");
 var today = new Date(); // Ағымдағы мерзім мен уақыт.
 // Нәтижені шығаруға дайындау:
 var str = "Бүгін: "
 // Айдың күні:
 str += today.getDate() + " ";
 // айдың аты: str+=monthNames[today.getMonth()]+" ";
 // жыл:
 str += today.getYear() + " года, ";
 // Апта күні:
 str += dayNames[today.getDay()] + "\n";
 // Уақыт:
 str += "Бүгін: " + today.getHours() + ":" +
                today.getMinutes () +":"+
                today.getSeconds();
 // Нәтижені шығару:
 alert(str);
```

Жиым ұзындығы (оның элементтерінің саны) программа жұмысы кезінде өзгере алады:

```
 var f = new Array(); //Қазір жиым бос,
                      //оның элементі жоқ.
f[0] = 1;      //Жиымда бір элемент бар.
f[l] = 1;            //Жиымда екі элемент бар.
f[2] = f[0] + f[1];  //Жиымда үш элемент бар.
f[5] = 8;      //Жиымда алты элемент бар
                     // f[0]...f[5]
```

Мысал: Жиымның ең үлкен элементін анықтау.

```
// num кездейсоқ сандардан жиым құрайық,
// олардың әрқайсысы [a, b] аралығында жатады
 var num = 10; // кездейсоқ сандардың саны
 var a = 1;    // аралықтың сол жақ шекарасы
 var b = 100;  //аралықтың оң жақ шекарасы
 var set = new Array(); // жиым құрылды
 // жиымды кездейсоқ сандармен толтыру
 for (i=0; i<num; i++)
   set[i] = Math.round(a+(b-a)*Math.random());
 // Ең үлкен элементін анықтау
 var max = a;
 for (i=0; i<num; i++)
   if(set[i] > max) max = set[i];
// Жиым элементтері мен үлкен элементін
көрсетейік:
 alert("Жиым: "+set+"\n Олардың ішіндегі
үлкен элементі: "+max);
```



Қарастырылған мысалда екі цикл бір циклге біріктірілген:

```
var max = a;
for(i = 0; i < num; i ++}
{
 set[i]-Math.floor(a+(b-a+1)*Math.random());
 if(set[i] > max) max = set[i];
}
```

Осыған дейін біз объектілер тәсілдерімен танысқанымен, олардың қасиеттерін қарастырмадық. Объект қасиеттері дегеніміз – JavaScript терминологиясы бойынша объект тұтынушысына арналған интерфейстік айнымалылар болып табылады. Негізінде, мұнда объект туралы емес, тек объектінің нақты данасы жайлы сөз болады. Қасиеттерді пайдалану үшін объект қасиеті аты мен нүкте арқылы бөлінген қасиет аты жазылады, мысалы:

```
var set = new Array("Алпамысов","Азамат",  "Адамұлы");
  alert (set.length);
```

length қасиеті массив элементтерінің санын – оның ұзындығын береді.

JavaScript тілінде объектілермен жұмыс істеу кезінде бірсыпыра «еркіндіктерге» жол беріледі. Мысалы, массив данасын new түйінді сөзінсіз және Array объектісін де көрсетпей жазуға болады:

```
var set = [1, 4, 9, 16, 25, 36];
```

Әрине, браузер, мұндай жазбаны кездестіріп, бәрібір Array объект данасын төмендегі жазба түрінде ашады:

```
var set = new Array(1, 4, 9, 16, 25, 36);
```

Осыған дейін біз string объектісі данасын басқа тәсілдермен құрған болатынбыз. Мысалы,

```
var title = "Ақыл арымас, алтын шірімес"; жазбасы
```

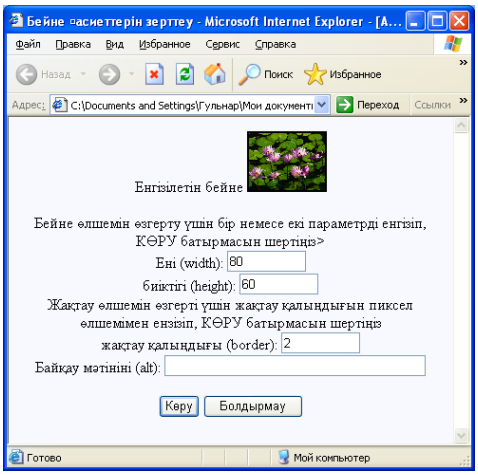
төмендегі жазбамен бірдей болып табылады:

```
var title = new String("Ақыл арымас, алтын шірімес");
```

JavaScript ішкі объектілерімен жұмыс істеу үшін анықтамалық материалдар болғаны дұрыс, олар көбінесе кітап қосымшаларында келтіріледі.

Мысалы: Енгізілген суреттің қасиеттерін (енін, биіктігін, жақтауының қалыңдығын) өзгертіп, экранда бақылауға болатын программа құрайық.

```
<HTML>
<HEAD>
<TITLE> Бейне қасиеттерін зерттеу </TITLE>
<script>
function chpict(obj)
{ var w=obj.wd.value
var h= obj.hg.value
if (w !=0) document.mypict.width=w
if (h !=0) document.mypict.height=h
document.mypict.border=obj.br.value
document.mypict.alt=obj.al.value
}
</script>
</HEAD>
<BODY bgcolor="F8F8FF">
<CENTER>
Енгізілетін бейне
<IMG src=111.jpg name=mypict>
<FORM name="form1">
Бейне өлшемін өзгерту үшін бір немесе екі параметрді енгізіп,  КӨРУ батырмасын шертіңіз><br>
Ені (width): <input type="text" name="wd" size=8><br>
биіктігі (height): <input type="text" name="hg" size=8><br>
```



4.21-сурет

```
Жақтау өлшемін өзгерту үшін жақтау қалыңдығын пиксел өлшемімен енгізіп,
КӨРУ батырмасын шертіңіз<br>
жақтау қалыңдығы (border): <input type="text" name="br"
size=8 value=0><br>
Байқау мәтінін (alt): <input type="text"
name="al" size=40><P>
<input type="button" value= "Көру" onclick="chpict(form1)">
<input type="reset" value="Болдырмау">
</FORM>
</CENTER>
</BODY>
</HTML>
```

4.9.1. Объектілік бағытталған программалау

Объектілік бағытталған программалау – программалық кодтар жасаудың қазіргі тәсілі, ол құрылымдық программалаудан кейін дамыды. Объектілік бағытталған программалау құрылымдық программалауды алмастырған жоқ, ол оны ары қарай логикалық жетілдіру кезеңінен өткізді.

Құрылымдық программалаудың негізі есепті шешудің сатылы бұтақ тәріздес құрылымын жасау және оның программалық кодын жекелеп жазып шығу болып табылады. Құрылымдық программалаушылар бір логикалық бірлікке қатысты процедуралар мен мәліметтерді жеке файлға құрылым (Си-де struct) арқылы жинақтай отырып жасап шығады.

Объектілік бағытталған программалау ортасында тек жаңа мәліметтерді құруға емес, оларды өңдеу функциялармен біріктіруге мүмкіндік беріледі. JavaScript тілінде ол объект деп аталады.

Мысалы:

```
// Rectangle объектісінің құрылымы.
function Rectangle(a, b)
{ // Объект қасиеттерінің сипатталуы.
  this.a = a;   // тіктөртбұрыш ені,
  this.b = b;  // тіктөртбұрыш биіктігі. }
// Rectangle объектісі құжаттамасының соңы.
```

Бұл мысалда Rectangle объектісі C++ тіліндегі құрылым сияқты сипатталған. Бұл объектіде мәлімет бар болғанымен, функцияның өзі жоқ. Сондықтан жаңа тіктөртбұрыш қажет болғанда, оны былай анықтауға болады:

```
var rec = new Rectangle(10,20);
// rec объектісінің данасы құрылды.
```

Кейіннен rec айнымалысы мәнін өрнектерде пайдалануға болады:

```
var perimeter = 2 * (rec.a + rec.b);
// тіктөртбұрыш периметрі.
```

Объект данасын функция аргументі ретінде де пайдалануға болады:

```
function Perimeter(x)
{ return 2 * (x.a + x.b);    }
```

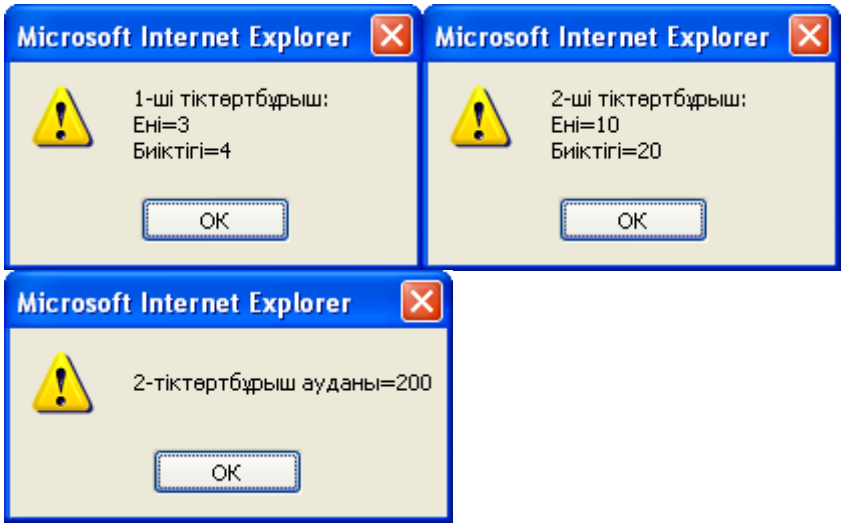
```
var p = Perimeter(rec);
```
Осы объектіге оның мәліметтерін өңдейтін функция (тәсіл) қосайық.
Мысалы:
```
//  Rectangle объектісі.
// Объект құрылымы.
function Rectangle(a,b)
{ // Объект қасиеттерінің сипатталуы
 this.a = a;  // Тіктөртбұрыш  ені
 this.b = b;  // Тіктөртбұрыш биіктігі
  // Объект тәсілдерінің (функцияларының) сипатталуы
  this.perimeter = _perimeter;
  // _perimeter функциясына сілтеме
}
  // perimeter тәсілінің сипатталлуы.
function _perimeter()
{ return 2 * (this.a + this.b); }
//-- Rectangle объектісі құжаттамасының соңы
```
Енді Rectangle объектісі тек мәліметтерден ғана емес, функциядан да тұрады. Мынадай код жазуға болады:
```
// Бірінші тіктөртбұрышты құрайық:
var p1 = new Rectangle(10,20);
// екінші тіктөртбұрышты құрайық:
var p2 = new Rectangle(35,70);
// периметрлер қосындысын табайық:
var sum = p1.perimeter() + p2.perimeter();
```
perimeter тәсілін анықтайтын жолға назар аударыңыздар:
```
This.perimeter = _perimeter;
```
Мұнда perimeter тәсілі анықталған және бұл тәсілді _perimeter атты функция жүзеге асырады. Тәсіл және функция аттары әр түрлі бола береді. Бірақ түсінбеушілік тудырмас үшін функция атын тәсіл атына төменгі сызықша «_» қою арқылы анықтау ұсынылады.

Объект түсінігі – қиын болмағанымен, оны дұрыс сипаттауға дағдылану керек. Rectangle объектісіне тағы бір мысал келтірейік. JavaScript тіліндегі объект *кәдімгі функция* сияқты function *түйінді сөзімен* сипатталады:
```
  // Rectangle объектісі
function Rectangle (a,b)
{ // Қасиеттері.
 this.width = a; // Тіктөртбұрыша ені.
 this.height = b; // Тіктөртбұрыш биіктігі.
 // тәсілдері.
 this.square= _square;//Тіктөртбұрыш ауданы
 this.perimeter = _perimeter;// Тіктөрбұрыш периметрі
 this.radius=_radius;//Сырттай сызылған шеңбер радиусы
}
```

```
// объекті тәсілдерін жүзеге асыру.
 function _square()
{ return this.width * this.height; }
function _perimeter()
 { return 2 * (this.width + this.height);}
function _radius()
{ var temp = this.width * this.width +
          this.height * this.height;
  return Math.sqrt(temp)/2; }
// Rectangle құжаттамасының соңы
```
Сипатталудың келесі түрі:
```
function Rectangle (a,b)
{
…
}
```
объект *конструкторы* деп аталады, оның ішіндегі айнымалылар this сөзі арқылы жазылады:
```
this.width = a; //Тіктөртбұрыш ені.
this.height = b; // Тіктөртбұрыш биіктігі.
```
this түйінді сөзі конструктор арқылы жасалатын объект данасының көрсеткіші болып табылады. Яғни ол айнымалылар мен функцияларға болашақ объект данасының қасиеттері мен тәсілдері ретінде мағына береді. Шартты түрде this түйінді сөзі конструктор сипатталуындағы айнымалыны қасиетке, ал объект тәсіліндегі сілтемені функцияға «айналдырады». Кәдімгі функцияны объект конструкторынан айырудың оңай тәсілі: «егер function {...} ішінде var сөзі орнына this пайдаланылса – ол объект конструкторы болғаны».
```
var r1 = new Rectangle(3,4);
// Тіктөртбұрыштың бірінші данасы.
var r2 = new Rectangle(10,20);
// Тіктөртбұрыштың екінші данасы.
```
Браузер осы екі команда және объект «құжаттамасы» арқылы тіктөртбұрыштың екі данасын (rl және r2) құрады. Әрбір дананың өз айнымалылары (width және height) болады. Оларды былай пайдалануға болады:
```
var x = rl.width; // x мәні 3-ке тең.
var y = r2.height; // y мәні 20-ға тең.
```
Келесі программаны браузерде тексеріп көрейік:
```
<HTML>
 <HEAD>
  <TITLE>Объектіні тексеру</TITLE>
 </HEAD>
 <BODY bgcolor=white text=black link=blue
      alink=red vlink=purple>
```

```
 <H1>Объектіні тексеру</H1> <HR>
 <SCRIPT language=JavaScript>
 <!--
 // -- Rectangle объектісінің құжаттамасы.--
 // Объект конструкторы.
 function Rectangle(a,b}
 {
  // Қасиеттері.
  this.width = a; // Тіктөртбұрыш ені.
  this.height = b; // Тіктөртбұрыш биіктігі.
  // тәсілы.
  this.square =_square;// Тіктөртбұрыш ауданы
  this.perimeter =_perimeter; //Оның периметрі
  this.radius = _radius; //Шеңбер радиусы.
 }
// Объект тәсілін жүзеге асыру
function _square()
{ return this.width * this.height;     }
function _perimeter()
{ return 2*(this.width + this.height); }
 function _radius()
         { var temp = this.width*this.width +
       this.height*this.height;
 return Math.sqrt(temp)/2;            }
  // Rectangle объектісі құжатының соңы.
  // Құрылған объетіміен жұмыс мысалы.
var rl = new Rectangle(3,4);
alert("1-ші тіктөртбұрыш:\nЕні="+
      r1.width+"\nБиіктігі=" + rl.height);
var r2 = new Rectangle(10,20);
alert("2-ші тіктөртбұрыш:\nЕні="+
      r2.width+"\nБиіктігі=" + r2.height);
alert("2-тіктөртбұрыш ауданы="+r2.perimeter());
//-->
 </SCRIPT>
 </BODY>
</HTML>
```

Жоғарыдағы келтірілген скрипт жұмысының нәтижелері:



Бақылау сұрақтары:

1. Объект деген не? Түсініктеме беріңіз.
2. Объектінің бір данасы дегенде қалай түсінуге болады?
3. Ағымдағы мерзім мен уақытты қандай объектінің көмегімен шығаруға болады?
4. Жиым деген не? Жиым объектісінің тәсілдеріне сипаттама беріңіз.
5. Объектілі бағытталған программалау деген не?

Тапсырмалар:

1. Сауалнамада бес қызметкердің әрқайсысы үшін келесі мәліметтер толтырылады: тегі, жалақысы, баласының саны. Жанұядағы кісі басына шаққандағы табысты есептеу сценарийін жазыңыз.

2. Сауалнамада алты қызметкердің әрқайсысына келесі мәліметтер толтырылады: тегі, жұмысқа алынған жылы. Жұмыс стажын және стаждары бірдей қызметкерлердің ең үлкен санын анықтау программасын жазыңыз.

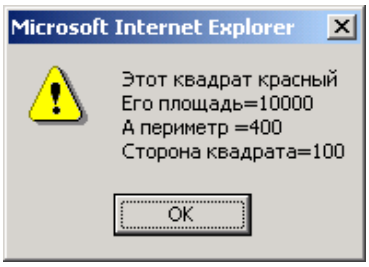
3. Сауалнамада алты қызметкердің әрқайсысына келесі мәліметтер толтырылады: тегі, жалақысы. Әрбір қызметкерге келесі принципке сүйене отырып сыйақы тағайындау шешілді: егер оның жалақысы орта жалақыдан аз болса, онда сыйақы оның жалақысының 50 пайызын, басқа жағдайда жалақысының 30 пайызын құрайды. Қызметкердің қолына алатын жалақысын (жалақы мен сыйақы) анықтау сценарийін жазыңыз. Ең көп сыйақы алған қызметкерлердің санын анықтау керек.

4. Бес түрлі тауарды сатып алу жөнінде мәлімет берілген: бір тауардың бағасы, алынған тауардың саны. Тауарды алуға кеткен қаражатты анықтау сценарийін жазыңыз. Бірдей қаражат жұмсалған тауар бар ма, болса, олардың саны қанша?

4.10. Мұралау

Тіктөртбұрыш объектісінің негізінде жаңа объект Kvadrat құрайық. Тіктөртбұрыштың барлық қасиеттері мен тәсілдері мұра ретінде квадратқа өтеді, айырмасы болуы үшін түс элементін қосайық:

```
// конструктора сипаттамасы (a -қабырғасы, c -түсі)
function Kvadrat(a,c)
{
 this.parent = Rectangle;
 //Аналық объект.
 this.parent(a,a);
 //оның конструкторын шақыру.
 this.color = c;
 // "түсі" қасиетін анықтадық
}
var kv = new
Kvadrat(100,"қызыл"); //Квадрат
var s = kv.square();//ауданы
var p = kv.perimeter();//периметрі
alert("Бұл квадрат " + kv.color + "\nауданы=" + s + "\nA периметрі =" + p +
"\nКвадрат қабырғасы=" + kv.width);
```

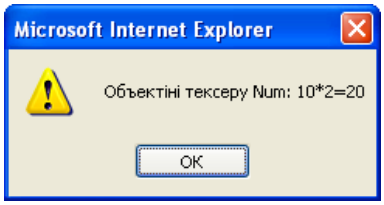


Осы сипатталған механизм объектілік бағытталған программалау ортасында *мұралау* (наследование) деп аталады.

Статикалық және динамикалық мұралау

Мынадай объект құрайық:

```
//-- Объект Num --
function Num(a)
{ this.number = a; this.mul2 = _mul2; }
function _mul2()
{ return this.number * 2; }
//-- Num объектісі құжаттамасының соңы. --
```

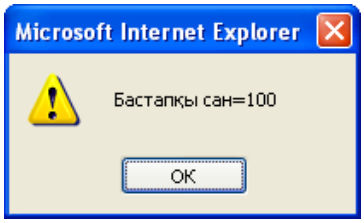


Объект санды сақтап, оны 2-ге көбейте алады.

Енді былай жазамыз:

```
var x = new Num(10);
var y = x.mul2();
alert("Объектіні тексеру
Num: 10*2="+y);
```

Енді Num объектісі үшін мұрагер жасаймыз:




```
// Num объектісі (Num объектісінен
мұраланған)
function Numa(a)
{ this.parent = Num; // Num анасы болып
хабарланған
```

```
this.parent(a); // анасы конструкторы шақырылған
this.put = _put;// мұрагерінің жаңа тәсілі хабарланған }
function _put()
{  alert("Бастапқы сан=" + this.number);}
// Num объектісі құжаттамасының соңы.
```

Мұрагерді тексеру үшін былай жазамыз:

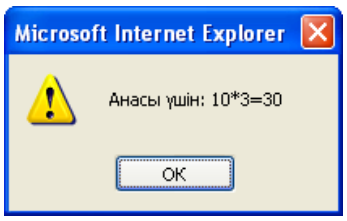
```
 // Numa тексеру
var z = new Numa(100);
z.put();
```

Экранда мынадай жазу пайда болады:

«Бастапқы мән=100»

Num объектісіне жаңа тәсіл қосып былай жазамыз:

```
Num.prototype.mul3 = _mul3;
function _mul3()
{  return this.number*3;}
Тексереміз:
var t = new Num(10);
alert ("Анасы үшін: 10*3=" +t.mul3());
```



Мұрагердің өз тегінің жаңа тәсілін қалай түсінетінін тексерейік, ол үшін былай жазайық:

```
var k = new Numa(10);
alert(k.mul3());
```

Браузер қате жайлы хабарлама береді. Мұндай мұралау *статикалық* деп аталады. Мұрагер жасау кезінде оның ішкі құрамына тегінің барлық қасиеттері мен тәсілдері көшіріледі де, содан кейін туыстық байланыс жойылады. Егер оның ата тегі жаңа тәсілге ие болатын болса, оның мұрагері ол туралы ешнәрсе білмейді. Мұрагер жасау кезінде оның ата тегімен байланысын үзбеуге болады. Ол *динамикалық* мұралау арқылы іске асырылады. Егер объект мұралау тәсілімен жасалса, онда ата тегінің құжаттамасы көшірілмейді, тек оған *сілтеме* жасалады. Егер ата тегінде бір өзгеріс болса, онда ол барлық «динамикалық» түрде құрылған мұрагерлерге қатысты болып саналады.

Динамикалық мұралауы бар объектіге мысал:

```
//Объект Numa (Num-нан динамикалық мұраланған)
function Numa(a)
{  this.parent = Num;
   // Num анасы болып хабарланған
   this.parent(a);
   // анасы конструкторы шақырылған
   this.put = _put;
   // мұрагердің жаңа тәсілі хабарланған
}
Numa.prototype = new Num;
```

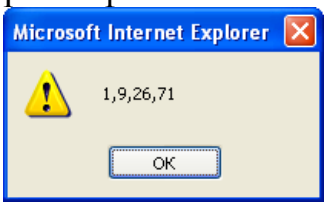
```
  //анасымен динамикалық байланыс берілген
function _put()
{
alert("Бастапқы мән="+this.number);
}
//-- Numa объектісі құжаттамасының соңы.--
```

Array объектісінің sort тәсілі

Жиым элементтерін сұрыптау әрекеттерін орындауға болады. Array объектісінің sort тәсілін қарастырайық. Ол үшін sort(function) немесе sort() тәсілі енгізілген, мұнда function параметрі екі элементті салыстыру ережесін береді, ол болмаған жағдайда сұрыптау лексографикалық тәртіпте жүргізіледі. Мысалы:

```
var set = new
      Array("zebra","ant","dog","cat");  set.sort();
alert(set);
```

Бұл жиым элементтерін алфавит бойынша сұрыптау әрекетін орындайды.



Бұл sort(function) тәсілінде function функциясының екі аргументі болуы тиіс, ол мынадай мәндер қайтарады:

– теріс сан, реттелуі бойынша бірінші аргумент екіншісінен сол жақта орналасқанда;

– 0, аргументтері тең мәнді болғанда;

– оң сан, реттелуі бойынша бірінші аргумент екіншісінен оң жақта орналасқанда. Мысалы:

```
var set = new Array(26,71,9,1);
function Compare(a,b)
{ return a - b; }
set.sort(Compare);
alert(set);
```

Array құрамдас объектісі және оның тәсілдері

1. Тәсіл түрі – concat(array). Бұрынғы жиымға array жиымы қосылған жаңа жиым құрады. Бастапқы жиым өзгермейді.

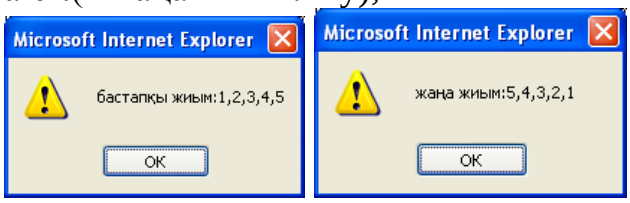
Мысал:

```
var set1 = new Array(1,2);
var set2 = new Array(3,4);
var set = set1.concat(set2);
alert(set1+"\n"+set2+"\n"+set);
```

2. Тәсіл түрі – join (ажыратқыш). Ажырату символы арқылы бөлініп орналасқан массив элементтері жолын береді. Бастапқы массив өзгермейді.

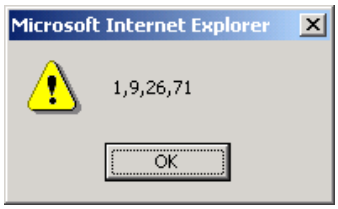
Мысал:
var set= new Array(1,2,3,4,5);
var set2 = set.join(“+”);
alert(set1+”\n”+set2);

3. Тәсіл түрі – reverse(). Жиым элементтерін оның 1-элементі соңғысы болатындай етіп кері бағытта орындарын ауыстырып береді.
var set= new Array(1,2,3,4,5);
alert(“бастапқы жиым:”+set);
var y = set.reverse();
alert(” жаңа жиым:”+y);



4. Тәсіл түрі slice(ind1) емес– slice(ind1,ind2) Бастапқы жиымнан ind1-ден бастап ind2-1 позициясына дейінгі элементтерден тұратын жаңа жиым құрады. Егер 2-индекс жоқ болса, онда жиымның соңына дейінгі элементтер алынады.
var set= new Array(0,1,2,3);
var set1= set.slice(1,3);
var set2= set.slice(1);
alert(”set=“+set+”\nset1=”+
set1+”\nset2=”+set2);

5. Тәсіл түрі sort() емес – sort(function). Жиымды сұрыптау мүмкіндігін береді. function параметрі екі элементті салыстыру ережесін береді, ол жоқ болса, сұрыптау лексографикалық тәртіппен жүргізіледі. Мысалы:
var set = new Array("zebra","ant","dog","cat"); set.sort();
alert(set);
sort(function) тәсілінде function функциясының екі аргументі болуы тиіс, оның қайтаратын мәндері:
теріс сан, реттелуі бойынша бірінші аргумент екіншісінен сол жақта орналасқанда;
— 0, аргументтері тең мәнді болғанда;
— оң сан, реттелуі бойынша бірінші аргумент екіншісінен оң жақта орналасқанда. Мысалы:
var set = new Array(26,71,9,1);
function Compare(a,b)
{ return a-b; }
set.sort(Compare);
alert(set);



6. length тәсілі – массив ұзындығын (оның элементтері саны) анықтайды. Мысалы:

```
var set = new Array(0,1,2,3,4,5,6,7,8,9,10); alert(set.length);
```

Бақылау сұрақтары:

1. Мұралау дегенді қалай түсінуге болады? JavaScript тілінде ол қалай жүзеге асады? Мысалдар келтіріңіз.
2. sort(function), concat(array), join тәсілдеріне сипаттама беріңіз.
3. reverse(), slice(ind1,ind2) тәсілдері қандай қызмет атқарады.
4. sort(function), length тәсілдеріне сипаттама беріңіз.

Тапсырмалар:

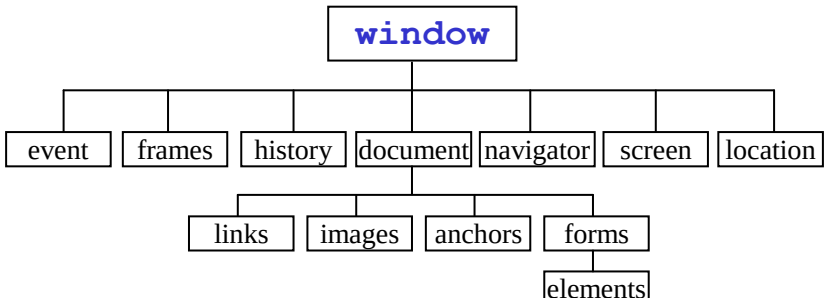
1. Қазақстан облыс орталықтарының алфавит бойынша реттеңіз.
2. 15-ке дейінгі жұп сандарды өсу реті бойынша шығарыңыз.
3. 20 дейінгі сандардан тұратын жиымның ұзындығын анықтау сценарийін құрыңыз.
4. 4,5,6,7 және 7,8,9 жиымдардан slice(ind1,ind2) тәсілінің көмегімен жаңа жиым құру сценарийін жазыңыз.

4.11 Браузер объектілері мен оқиғалары

Құжаттың объектілік моделі

Браузер экранын өзгерту немесе жаңа терезелер жасау үшін браузердің ішкі мүмкіндіктерімен танысып, оның ішкі объектілерін пайдалану керек. Браузер тәгтерді экранда көрсетіп қана қоймай, ол құжаттың иерархиялық моделіне сәйкес объектілер тұрғызады. Браузер жұмыс кезінде модельді тұрақты түрде бақылап отырады. Егер объект қасиеті өзгерсе, экрандағы бейне де өзгереді. JavaScript тілі арқылы объект қасиеттерін өзгертуге болады, сондықтан ол экран динамикасын жүзеге асыра алады.

Кез келген құжаттың объектілік моделі әрқашанда нақты сайттың бейнесінен тәуелсіз түрдегі бір құрылымда болады (4.22-сурет).



4.22-сурет. Объектілер құрылымы

Схемада көрсетілген барлық «объектілер» негізінде бір объектінің даналары болып табылады. Браузер әр нақты құжат үшін сол даналарды жеке-жеке түрде құрастыра алады. Сол даналарды бұдан кейін «объектілер» деп атаймыз.

Window объектісі

Window иерархияның жоғарғы жағында орналасады. Бұл қалған объектілердің одан төмен орналасып, соның қасиеттері түрінде қарастырылатынын білдіреді. Үш тәсілді біз бұрыннан білеміз, олар: alert, prompt және confirm.

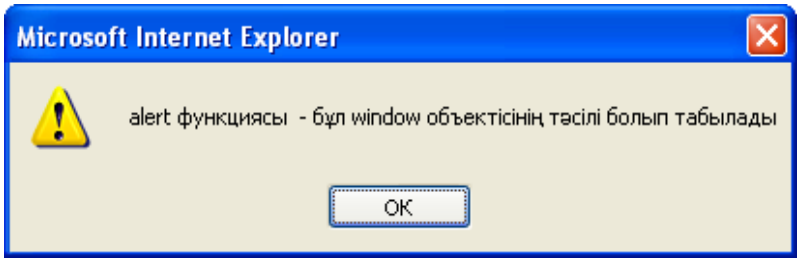
Егер келесі түрде жазылса:

Window.alert ("alert функциясы – бұл window объектісінің тәсілі болып табылады "};

– онда браузер осы хабарламаны шығаратын қосалқы терезе ашады.

Біз бұрын былай жазатынбыз:

alert("alert функциясы – бұл window объектісінің тәсілі болып табылады");



Негізгі объект болып саналатын window объектісінің атын жазбауға да болады. Келесі жолдардың нәтижелері бірдей болады:

window.alert("Ат айналып қазығын табар"); alert("Ат айналып қазығын табар");

Жаңа терезелер ашу

Көп жағдайда пайдаланушыға жұмыс нәтижесін көрсету үшін бұрынғы терезе ашық тұрғанда, жаңа терезе ашуға тура келеді. Жаңа терезе ашу үшін window объектісінің open тәсілін қолданамыз. Оны келесі тәсілдердің бірін қолдану арқылы жазуға болады:

var айнымалы = open();

var айнымалы = open(файл);

var айнымалы = open(файл, терезе_аты);

var айнымалы = open(файл,терезе_аты, терезе_параметрлері);

Мұндағы:

· айнымалы – ашылатын терезеге нұсқауыш;

· файл – сөз тіркесі (строка). Ашылған терезеде көрсетілетін файл аты. Егер бұл параметр берілмесе, бос терезе ашылады;

· терезе_аты — сөз тіркесі. Терезе аты сол терезеге мәлімет шығару үшін керек (window объектісінің name қасиеті мәні);

· терезе_параметрлері – сөз тіркесі. Терезе қасиеттерін сипаттау. Егер параметр берілмесе, үнсіз келісім бойынша тағайындалған қасиеттер қолданылады. Параметрлер арасындағы үтірден соң, бос орын қоймаған дұрыс.

Параметр	Мәні	Сипаттамасы
width	Сан	Пиксель, терезе ені. Минимал мәні – 100
Height	Сан	Пиксель, терезе биіктігі. Минимал мәні – 100
scrollbars	yes, no,1, 0	Айналдыру жолағын береді
resizable	yes, no,1, 0	Терезе көлемін өзгертуді көрсету үшін қажет
menubar	yes, no,1, 0	Меню өрісі бейнелетінін көрсету үшін қажет
location	yes, no,1, 0	Адрес енгізу өрісі бейнелетінін көрсету үшін қажет
status	yes, no,1, 0	Статус жолағы бейнелетінін көрсету үшін қажет
toolbar	yes, no,1, 0	Батырмалар (саймандар) тақтасы бейнелетінін көрсету үшін қажет

Мысал. Пілдердің суреті slon.jpg орналасқан жеке терезе ашу

```
<HTML>
  <HEAD>
      <TITLE>Браузерді тексеру</TITLE>
  </HEAD>
  <BODY bgcolor=white text=black>
      <H2>Браузерді тексеру</H2> <HR>
      <SCRIPT language=JavaScript>
      <!--
       var win = open("slon.jpg","",
          "width = 320,height = 260"+
          "resizable =0,scrollbars =1"+
                  "menubar =0,location = 1" +
                  "status = 0, toolbar = no");
      //-->
      </SCRIPT>
      <P>
Негізгі мәтінге оралу үшін браузердің саймандар тақтасында орналасқан
<EM>Артқа қарай (Назад)</EM>батырмасын басыңыз.
  </BODY>
</HTML>
```

Терезеде көрсетілетін файл ретінде мыналарды көрсете аламыз:

- HTML-файл, мысалы, file1.htm;
- сурет файлы, мысалы, ./pic/fish.gif;
- мәтіндік файл, мысалы, 010401.txt.

Осы программа нәтижесі:

4.23-сурет. Жеке терезенің көрінісі

Бұдан кейін әр түрлі параметр-лерді өзгертіп көруге болады.

Close тәсілі ашық терезені жабады. Мұндағы close()немесе window.close() ағымдағы терезені жабады. Ал win.close() немесе window, win.close() — осы терезеден open тәсілі арқылы ашылған win нұсқауышы бар терезені жабады:

```
var win = window.open(...);
```

Document объектісі

Document объектісі (ол document объектісі қасиеттері болып табылады) <html> ...</html> блоктарында орналасқан тәгтерді модельдейтін объектілер үшін қажет.

Document объектісінің кең тараған тәсілі – write тәсілі. Бұл тәсіл құжат терезесіне мәлімет жазу үшін керек. Write тәсілін қолдану алдында жазба ашу керек (open тәсілі), ал шығарған соң, оны жабу керек (close тәсілі).

Open тәсілі форматы:

айнымалы.document.open();

Мұндағы: айнымалы – бұл мәлімет жазылатын терезеге нұсқауыш.

Write тәсілі форматы:

document.write(жазба); немесе

айнымалы.document.write(жазба);

Мұндағы : айнымалы – бұл шығарылатын мәлімет көрсетілетін терезеге нұсқауыш. Егер айнымалы берілмесе, онда мәлімет ағымдағы терезеге жіберіледі; жазба – шығарылатын мәлімет көрсетілетін сөз тіркесі.

Close тәсілі форматы:

document.close(); немесе

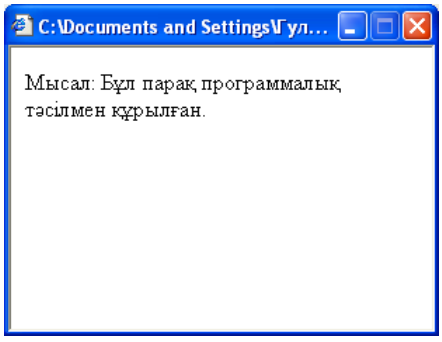
айнымалы.document.close();

Мұндағы айнымалы – жазба шығарылмай жабылатын терезеге нұсқауыш. document.open тәсілін қолдану міндетті емес, өйткені document.write тәсілі жабылған құжатқа мәлімет жазу кезінде оның ішіндегі ескі мәліметті өшіріп, оны жаңа жазба үшін ашады.

Мысал. Келесі суреттегі бейнені құру программасы:

```
var win = open("","","width=300,height=200"); win.document.write("Мысал: Бұл парақ "); win.document.write("программалық тәсілмен құрылған."); win.document.close();
```

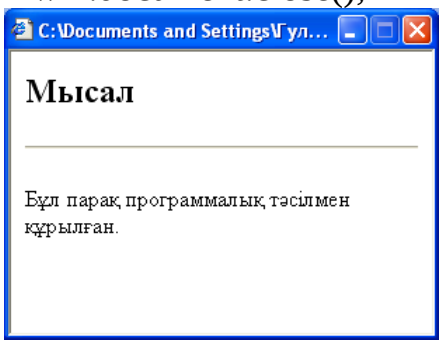
Терезеге мәтін шығару кезінде write функциясы бірнеше рет қолданылғанымен, оның нәтижесі бір тұтас азат жол ретінде шығарылады.



4.24-сурет. Программалық тәсілмен құрылған парақ

Write тәсілі арқылы терезеге жай мәтіннен басқа тәгтермен белгіленген мәтінді де шығаруға болады (4.24- сурет).

```
var win = open("","","width=300,height=200"); win.document.write("<H2>Мысал </H2>"+
 "<HR>" +
 "<P>" +
 "Бұл парақ" +
 " программалық тәсілмен құрылған.");
  win.document.close();
```



4.25-сурет Программа нәтижесі

4.12. Оқиғалар

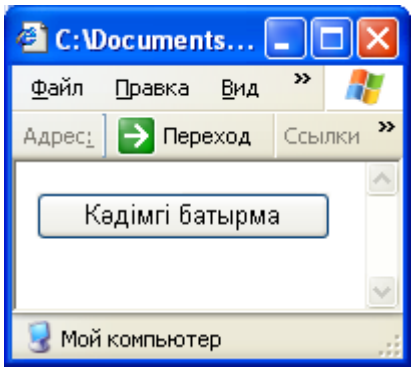
Браузер тұтынушы әрекеттерін қадағалап, соларға сәйкес әрекет ете алады. Егер терезеде тышқанды шерту сілтемеге сәйкес келсе, браузер жаңа құжат жүктейді. “Alt + ←” пернелерін басу терезені бұрынғы қалпына қайтарады.

Егер тұтынушы тышқан курсорын суретке алып барса, онда браузер шағын терезеге сол <img> тәгіне, яғни суретке сәйкес alt-мәтін шығарады. Тұтынушы бағыттауыш пернелерді басса, браузер сәйкесінше терезедегі мәтінді жылжытады.

Бұлардың бәрі тышқан мен пернеліктің браузерге тұтынушының не істеп жатқаны жайлы « хабар беріп» отыратынын білдіреді. Оған қоса, браузер құжаттың жүктелуі жайлы да толық хабардар болып отырады. Егер

сурет жүктелмесе (дискіде жоқ, сервермен байланыс үзілді), браузер оның орнына тіктөртбұрыш қояды. Осындай информациялық хабарламалар *оқиғалар* (события) деп аталады. Кейбір оқиғаларға браузер бірден жауап береді, ал басқаларына жайбарақат қарап тұра беруі де мүмкін. Мұндайда браузерге не істеуі керек екендігі айтылмаған деп ұққан жөн.

HTML-код арқылы браузер салған батырма мынадай болады(4.25 суретті):



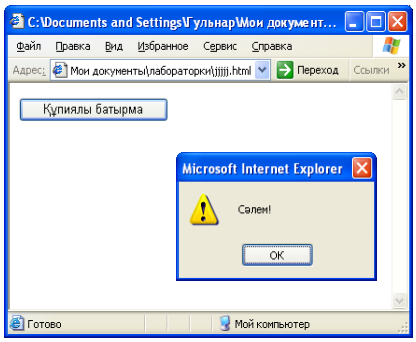

```
<FORM>
  <INPUT type=button
   value="Кәдімгі батырма">
</FORM>
```

Батырма <form>...</form> тәгінің ішкі <input> тәгі арқылы берілген. value атрибуты батырма бетіндегі жазуды береді, ал type атрибуты енгізу өрісінің типін сипаттайды. Браузер батырманы шерткенде, оған жауап бере ме? Әрине, ол экранда батырманы басқанды көрсетеді, бірақ басқа ешнәрсе істемейді.

Ал енді басқа батырмаларды қарастырайық.

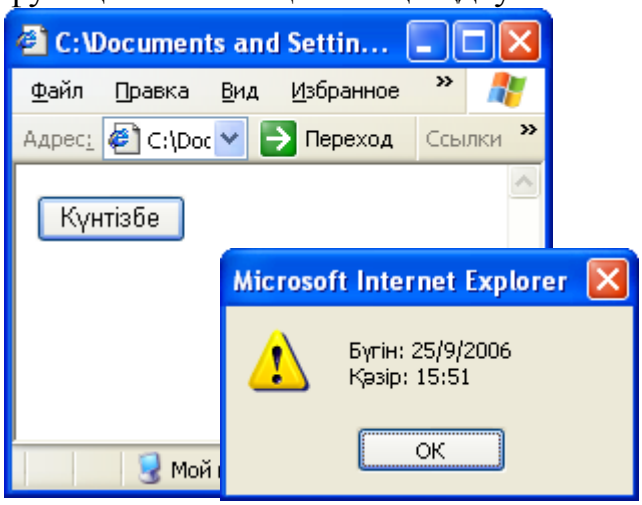
```
<FORM>
   <INPUT type=button value="Құпиялы батырма"
          onclick="alert('Сәлем!')">
</FORM>
```

Экранға «Сәлем!» сөзі бар alert терезесі шығады (4.26-сурет). <input> тәгінің onclick атрибуты бар. Оның мәні ретінде: "alert('Сәлем!')" сөзі жазылған. onclick атрибуты браузерге батырма басылғанда не істеу керек екендігін көрсетіп тұр. Атрибут мәні ретінде ұзын сөз жазудың қажеті жоқ, көбінесе келесі түрде программаланады.




```
<SCRIPT language=JavaScript>
<!--
 function Fun()
 { var now = new Date() ;
  alert("Бүгін: "+now.getDate()+"/"+
  (now.getMonth()+1)+"/"+ now.getYear() +
  "\nҚазір: "+now.getHours()+":"+now.getMinutes());
}
//-->                                              </SCRIPT>
<FORM>
<INPUT type=button
value=Күнтізбе onclick="Fun();">
</FORM>
```

onclick оқиғасы туғанда – батырманы шерткенде жұмыс істейтін Fun() функциясы сол оқиғаның өңдеуіші болып саналады.



4.27-сурет

Осы программаны браузерге жүктеп бірнеше тәжірибе жүргізіңдер.

1. Батырмадағы сөзді өзгертіп «Күнтізбе» орнына басқа мәтін жазыңдар.
2. Fun функциясы кодын басқаға, мысалы, мынаған өзгертіңдер:

```
var sum = 0;
for (var i = 100; i; i --) sum += i;
alert("Қосынды 1 + 2 + ... + 100 = " + sum);
```

3. Өз функцияларыңды жазып шығыңдар. Мысалы, бір сан енгізіп оның квадратын немесе кубын табыңдар.

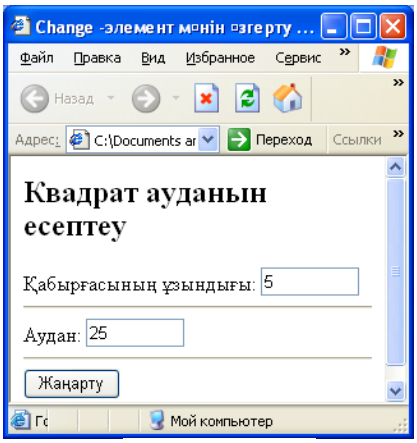
Сонымен, JavaScript HTML-кодтарында оқиғаларды өңдейтін программалар жазуға мүмкіндік береді де, оны интерактивті бет етуге, яғни тұтынушы әрекетіне жауап беретіндей күйге келтіре алады.

Тәгтегі onclick түйінді сөзі - «оқиғалық» атрибуттың аты, алдына «on» сөзі қойылып жазылады. Мұндай атрибут мәні скриптік код болып табылады. Егер оқиға орындалса, онда атрибут аты арқылы кодталған скрипт іске қосылады. Onclick атрибуты «экрандағы белгілі нүктені тышқанмен шерткенде» орындалатын оқиғаны өңдеуді жүзеге асырады .

Мысалы: Қабырғасының ұзындығы берілген жағдайда квадраттың ауданын анықтау программасын жазайық. Ауданның мәні қабырғаның мәні өзгерген сәтте есептелуі тиіс. Екі мәтіндік өріс енгізілген форма құрайық: оның бірі квадраттың қабырғасының мәнін енгізу, ал екіншісі аудан мәнін есептелген мәнін шығару үшін қажет. **Жаңарту** батырмасы форма өрістерін тазалайды. Квадрат ауданы change оқиғасы туындаған мезетте, формадағы numl деген элемент мәні өзгергенде және элемент фокусын жоғалтқанда есептеледі.

```
<HTML>
<HEAD>
<title> Change -элемент мәнін өзгерту оқиғасы</title>
<script>
function srec(obj)
```

{



4.28-сурет

obj.res.value=obj.numl.value* obj.numl.value}
</script>
</HEAD>
<BODY>
<h2>Квадрат ауданын есептеу</h2>
<FORM name="forml">
Қабырғасының ұзындығы: <input type="text" size=7 name="numl"
onChange="srec(forml)">
<hr>
Аудан: <input type="text" size=7 name="res"><hr>
<input type="reset" value=Жаңарту>
</FORM>
</BODY>
</HTML>

Интерактивті құжат мысалы

Экранда бір енгізу өрісі және төмендегідей сипаттамалары бар төрт батырма болсын:

- ағымдағы уақытты береді;
- жаңа терезе ашып, оған сәлемдесу мәтінін шығарады да, терезені жабады;
- есептелетін өрнекті енгізу өрісі;
- енгізу өрісіндегі арифметикалық өрнектің мәнін көрсету өрісі;
- терезе жабу батырмасы.

Енді осы әрекеттерді программалайық.

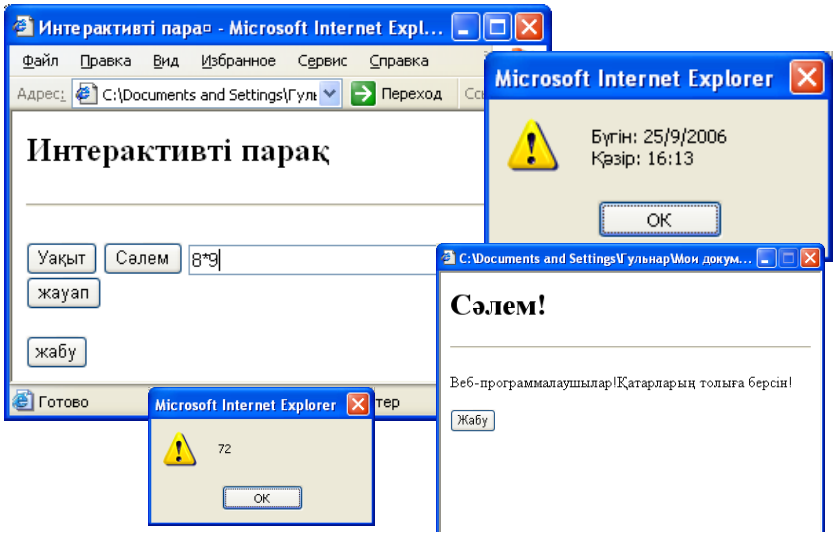
<HTML>
<HEAD><TITLE>Интерактивті парақ</TITLE>
<SCRIPT language=JavaScript>
function CurTime()
{ var now = new Date() ;
alert("Бүгін: "+now.getDate()+"/"+

```
 (now.getMonth()+1)+"/"+ now.getYear() +
 "\nҚәзір: "+now.getHours()+":"+
                    now.getMinutes());      }
 function Hallo()
 { var win=window.open("","",
    "width=400,height=300");
     win.document.open();
     var str = "";
     str += '<H1>Сәлем!</H1>';
     str += '<HR>';
     str += '<P>';
     str+='Веб-программалаушылар! Қатарларың толыға берсін!';
     str += '<P>';
     str += '<FORM>';
     str += '<INPUT type=button value=Жабу';
     str +=' "onclick"="window.close()">';
     str += '</FORM>';
     win.document.write(str);
     win.document.close();     }
</SCRIPT>
</HEAD>
<BODY bgcolor=white text=black>
 <H2>Интерактивті парақ</H2> <HR>
 <FORM>
  <INPUT type=button value=Уақыт
       onclick="CurTime()">
  <INPUT type=button value=Сәлем
       onClick="Hallo()">
  <INPUT type=text value="2*2" size=25
       name=calc>
  <INPUT type=button value=жауап
   onclick="alert(eval(this.form.calc.value))">
  <P>
  <INPUT type=button value=жабу
       onclick="window.close()">
 </FORM>
</BODY>
</HTML>
```

Мұндағы: **CurTime** функциясы **Уақыт** батырмасын шертуді өңдейді. **Hallo** функциясы **Сәлем** батырмасын шертуді өңдейді. **alert** функциясы **Жауап** батырмасын шертуді өңдейді. alert функциясы аргументі болып eval функциясының мәні есептеледі. eval (document,forma,calc.value) Eval функциясы аргументі болып берілген енгізу өрісінің мәні болып табылады. Мұндағы: forma – бұл объект, ол мынадай блок бейнесі

<form name=forma … >...</FORM>;
calc - бұл объект, ол мынадай тәг бейнесі
<INPUT name=calc … >;
value – бұл calc объектісі қасиеті. Window.close (...) функциясы **Жабу** батырмасын өңдейді**.** CurTime және Hallo функциялары сипатталған <script>. . .</script> блогы HTML-коды тақырыбында келтірілген. Себебі браузер функция сипатталуын оларды пайдаланғанға дейін оқуы тиіс, құжат тақырыбы ең алдымен орындалады ғой.

Программа жұмысы нәтижесі:



4.29-сурет. Интерактивті парақ программасының жұмыс

Window объектісі және оның тәсілдері

1. Тәсіл түрі – open(файл, терезе_ параметрі). Жыл мәнін береді, 20-ғасыр үшін соңғы 2 цифрды береді.

Мысал:

```
var okno = window.open(‘.pic/dog03.jpg,””,
”width=250,height=300,left=50,top=100,”+
“scrollbars=1,resizable=no,menubar=no,”+
“location=no,status=no,toolbar=no,”+
“fullscreen=no”);
```

2. Тәсіл түрі – alert(хабарлама). Мәліметі бар ақпараттық терезе шығарады.

Мысал:

```
window.alert(“Палуанның жаманы–\n”+
        “Шалдырған соң өкінер,\n”+
                “Күзетшінің жаманы,\n”+
        “Алдырған соң өкінер\n”+
        “Мақал”);
```

3. Тәсіл түрі – confirm(хабарлама). 2 батырмасы бар сұрақ мәліметі бар терезе шығарады. ОК және Отмена. Функция нәтижесі: - true, егер ОК басылса; - false, егер Отмена басылса.

Document объектісі және оның тәсілдері

Тәсіл	Сипаттамасы
`open()`	Браузер терезесіне жазу жазады. Бұрынғы терезе мәліметі өшіріледі
`close()`	Браузер терезесіндегі жазбаны жабады
`clear()`	Браузер терезесін тазалайды
`write(str)`	Құжаттың str мәні ретіндегі сөз тіркесін - мәтінді және HTML кодын жазады
`writeln(str)`	Каретканы қайтарумен аяқталатын (жаңа жолға көшу) мәтінді және HTML кодын жазады. Жаңа жолға көшу браузер экранында тек <pre> тәгі ішінде орналасса ғана көрінеді.

write және writeln тәсілдерін пайдалану келесі жағдайларда ғана тиімді болып табылады:

- ұзын HTML-кодты қысқа скрипт алмастыра алса;
- құжат браузер ерекшелігі - экран мүмкіндігі және т.б. есепке алынып тұрғызылса;
- құжат толық программалық жолмен жасалса.

Программалық басқару принципі

Браузер гипермәтіндік парақ ішіндегі әрбір тәг үшін объект жасай алады. Ал сол объектімен программалаушы name атрибутында көрсетілетін оның аты арқылы қатынас құра алады. Сонымен, тәгтерді программалық басқару мүмкіндігі пайда болады. Мысалы, HTML-кодта экранға сурет шығаратын мынадай тәг берілген болсын делік: <IMG src=./pic/ris1.gif

border=0 alt="" name=pic>

Осы тәг арқылы pic атты стандартты image объектісінің данасын мынадай эквивалентті конструкциялар арқылы пайдалануға болады: window.document.images["pic"] window.document.imagesрісnемесе ағымдағы терезе ішінде болатын болса: document.images["pic"] document.images.pic. Мысалы, image объектісінің суреті бар файлға нұсқайтын src қасиеті бар екені белгілі. Программада осы қасиетті төмендегі жолды жазып оқуға болады:document.images.pic.src Тек қана оқып қана қоймай, бұл қасиеттің мәнін де өзгерте аламыз:document.images.pic.src = "./pic/ris2.gif"сонда сурет сәйкесінше басқа суретке ауысады.

<Form> блогы

Гипермәтіндік құжаттармен жұмыс істей отырып, мәтіндерді енгізу өрісіне жазып, меню пункттерін таңдап, жалауша арқылы бір жолды таңдап, батырмаларды шертуге болатынын көргенбіз. Мұндайда әрекетке байланысты құжаттың соған жауап беретінін байқаймыз.

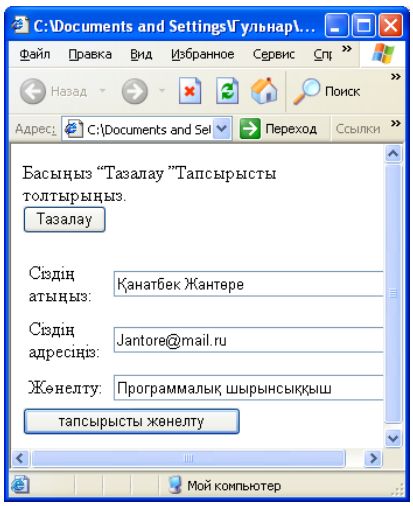
Осы интерфейстік элементтердің бәрі браузер экранына контейнер-команда <form>...</form> арқылы шығарылады, ал батырмалар, енгізу өрісі, жалаушалар және меню осы блок ішіндегі әр түрлі тәгтер арқылы берілетін.

<form>...</form> блогы форма деп аталады. Оның ашылу тәгінде мынадай атрибуттар жазылады.
action - форманы желі бойынша жөнелту адресін береді. Форманы оны өңдейтін программасы (CGI-скрипт) бар серверге жіберуге болады немесе оның мәні ретінде электрондық пошта адресі көрсетілсе, форма e-mail арқылы жөнелтіледі. Мысалы, былай жазуға болады: <FORM action="mailto:myaddres@mail.ru"> Егер ол көрсетілмесе форма сайттағы адресі бойынша жөнелтіледі. enctype – желі арқылы берілетін мәліметтерді кодтауды береді. Мұнда біз формадан алынған мәліметтерді өңдейтін программаларды қарастырмаймыз. Бірақ браузер тұтынушы толтырған форманы сіздің электрондық адресіңізге жібереді. Сол мәліметтерді қарап шығуға немесе басқа программа арқылы оны өңдеуге болады. Көбінесе төмендегідей толтырылған мәндер дұрыс жұмыс істейді:
<FORM action="mailto:myaddres@mail.ru" enctype="text/plain">
method - форманы тасымалдау тәсілін береді. Атрибуттың екі мәні бар: get және post. post тәсілі әмбебап болып саналады (форма URL-ден бөлек жіберіледі). get мәні форманың URL-мен бірге жіберілетінін көрсетеді. e-mail арқылы былай жіберуге болады:
<FORM action="mailto:myaddres@mail.ru" enctype="text/plain" method="post"> name - форма атын береді. Мұны скриптерде форма өрістерін пайдалану үшін береді. Егер форманы ешқайда жібермеу керек болса, онда тек name көрсетіледі: <FORM name=anketa> Енді JavaScript-кодтарында осы форманы былай пайдалануға болады: document.anketa. Егер форманың work атты жолдық өрісі болса, сол жолды айнымалыға меншіктеуге болады:
var str = document.anketa.work.value; Осы өріс мәнін өзгерте аламыз:
document.anketa.work.value = "веб-мастер";
Бұдан кейін бұл өріс мәні өзгереді.

E-mail арқылы форманы жөнелту

Келесі мысалды форманы электрондық пошта арқылы жіберудің үлгісі ретінде қарауға болады (4.30-сурет).

```
<FORM action = mailto:myaddres@mail.ru
    enctype = "text/plain" method="post"> Басыңыз
&#147;Тазалау&#148;Тапсырысты толтырыңыз.<BR>
 <INPUT type=reset value=Тазалау><BR>
<P> <TABLE border=0 cellspacing=0  cellpadding=5>  <TR> <TD>Сіздің
атыңыз:</TD>
<TD> <INPUT type=text size=33 name=name                value="Қанатбек
Жантөре">
</TD>  </TR>  <TR> <TD> Сіздің адресіңіз:</TD>      <TD> <INPUT
type=text size=33 name=email                value="Jantore@mail.ru">      </TD>
</TR>  <TR> <TD>Жөнелту:</TD>      <TD> <INPUT type=text size=33
name=obj                value="Программалық шырынсыққыш">      </TD>
</TR> </TABLE> <INPUT type=submit value="тапсырысты
жөнелту"></FORM>
```



4.30-сурет. Мәліметтерді электрондық пошта арқылы жөнелту

Егер осы кодты HTML-программаға кірістірсе, онда гипермәтіндік бетте төмендегі суреттегідей бейне пайда болады. Егер тазалау, батырмасын басып үнсіз келісім бойынша тағайындалған мәндерді өшіріп, жаңа мәндерді енгізуге болады. Форма толтырылған соң, Тапсырысты жөнелту батырмасын шертіп, оны қажетті адреске myaddres@mail.ru жөнелтуге болады. Әдетте браузер мәліметті өзі жөнелтпейді, оны компьютердің операциялық жүйесіндегі пошта программасы атқарады (мысалы, Outlook).

Батырмалар. <*INPUT*> элементі

Формада әр түрлі енгізу және басқару элементтерін орнатуға болады. Ол үшін <input> тәгі қолданылады. Әрбір <input> тәгінде name атрибуты болады. Оның мәні браузер ашатын объект атауы (типі) болып табылады. Мысалы, бір ғана мәтін енгізу жолы бар форма берілсін делік:

<FORM name=anketa>
<INPUTtype=textvalue=Қаламұш name=obj>
</FORM>

JavaScript программасындағы мынадай конструкция document.anketa.obj.value объектінің <input> тәгі үшін тұрғызылған value қасиетін пайдаланады (өзгертеді), Мұндай жазуды былайша түсіну керек: document – document объектісі (атасы (тегі) - window объектісі); document.anketa – anketa объектісі (атасы - document объектісі); document.anketa.obj – obj объектісі (атасы - anketa объектісі); document.anketa.obj.value – obj объектісінің value қасиеті, ол <input> тәгінің obj атты value атрибутына сәйкес келеді. value қасиетінің мәнін былай меншіктеп оқуға болады: var str = document.anketa.obj.value; Бұл қасиеттің мәнін өзгерте аламыз:document.anketa.obj.value = "қарындаш" Мұндай меншіктеу экрандағы мәнді бірден өзгертеді.Енгізу өрісіндегі «қалам» мәні «қарындаш» мәніне ауысады.

Name атрибуты мәні айнымалы аты тәрізді, сондықтан орыс/қазақ әріптерін пайдаланбау керек және бірінші символ цифр болмауы тиіс.

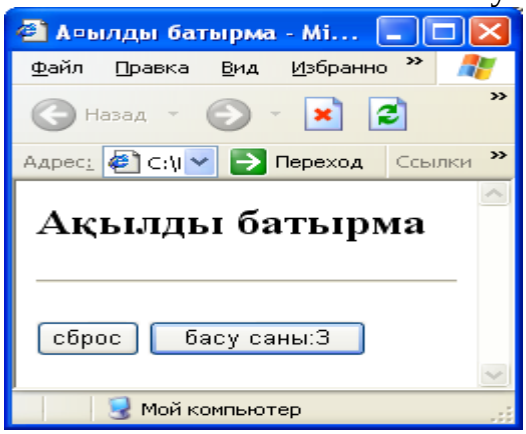
Экрандық батырмалар

<input> тәгіндегі интерфейстік элементтің нақты бейнесі type атрибутымен беріледі. Осы атрибуттың экрандық батырмалар тұрғызатын мәндерін қарастырайық:

reset - шерткен кезде форманы автоматты түрде бастапқы қалыпқа келтіретін батырма жасайды (форманың барлық енгізу өрістері келісім бойынша HTML-кодта орнатылған алғашқы мәндерді қабылдайды). Ал value атрибуты батырмадағы жазуды көрсетеді:<INPUT type=reset value=Тазалау> submit - шерткен кезде форманы желі бойынша жөнелтетін батырма тұрғызады.
value атрибуты батырмадағы жазуды көрсетеді:
<INPUT type=submit value=жөнелту>button – қарапайым батырма тұрғызады:
<INPUT type=button value= "Басу"> программалық басқару Төмендегі тәг үшін браузер тұрғызған объектің <input type=button> (не мынадай тәгтер үшін <INPUT type=submit>,<INPUT type=reset>) value қасиеті болады, ол батырмадағы жазуды анықтайды. Төменде осы қасиетті пайдаланатын мысал келтірілген.

Браузерге осы мысалды енгізіп, ондағы батырмалар жұмысын тексеруге немесе өзгертуге, яғни алгоритмін бақылауға болады (4.31-сурет).

```
<HTML>
<HEAD>
 <TITLE>Ақылды батырма</TITLE>
  <SCRIPT language=JavaScript>
 <!--
var num=0;
  var name0="басу саны:0";
//-->
</SCRIPT>
 </HEAD> <BODY bgcolor=white text=black>
<H2>Ақылды батырма</H2> <HR>
 <FORM name=count>
 <INPUT type=button value=сброс       on-
click="document.count.key.value=name0;
num=0;">
 <INPUT type=button name=key value="басу саны: 0"
     onclick="document.count.key.value='басу саны:'+(++num);">  </FORM>
```



4.31-сурет. Батырманың басу санын есептеу

Форма өрістерін пайдалануды жеңілдету үшін, this нұсқауын - объектінің ағымдағы данасын көрсетуді және this.form нұсқауын - ағымдағы форма үшін тұрғызылған объектіге нұсқауды пайдаланамыз:

<FORM> <INPUT type=button value=c6poc
onclick="this.form.key.value=name0;
num=0;"> <INPUT type=button name=key value="басу саны on-click="this.value='басу саны:'+(++num);"> </FORM>

Бұл мысалда форма атын көрсету қажет болмады. Форма ішіндегі оның синонимі ретінде this.form конструкциясы қолданылды. Ал екінші <input> тәгінде де оның атын жазбай-ақ, орнына оның синонимі this жазылды.

Енгізу жолы

<input type=... > тәгіндегі type атрибутының text және password мәндері енгізу жолын экранға шығарады: text – қарапайым енгізу қатары; password – құпия сөз енгізу қатары.

Қосымша атрибуттар:

- value атрибуты үнсіз келісім бойынша орнатылған енгізу өрісін береді (осы мән reset батырмасын шерткенде орнатылады: <INPUT type=reset>);
- size атрибуты өріс енін символмен береді. Егер атрибут көрсетілмесе, ол оның (value) мәні ретінде берілген сөздің ұзындығымен анықталады;
- maxlength атрибуты осы өріске енгізілетін сөздің максималды енін символмен береді. Егер ол атрибут көрсетілмесе, ұзындығы шектелмейді;
- readonly атрибуты «тек оқылатын» сөз тіркесін анықтайды. Мұндай жол түзетілмейді.

<INPUT type=text value="Анар" size=10 maxlength=30>

password - «құпия сөз» енгізу жолы. Ол енгізу кезінде көрінбей, жұлдызшалармен бейнеленеді. Құпия сөз буферге де көшірілмейді.

<INPUT type=password value="" size=10 maxlength=10>

Оны программалық басқару Браузердегі <input type=text> (немесе <input type=password>) тәгі үшін тұрғызылған объектінің мынадай қасиеттері болады:

value – енгізу жолының ағымдағы мәні;

defaultValue – value атрибутының соған сәйкес <INPUT> тәгінде көрсетілген мәні.

Формалар (жалаушалар, батырмалар)

Енді интерактивті парақтарда жиі қолданылатын екі интерфейстік элементті қарастырайық, олар: *жалаушалар* мен *радиобатырмалар*.

Жалаушалар Жалаушалар немесе тәуелсіз басылатын батырмалар – басқа жалаушаларға байланысты емес, тәуелсіз жұмыс атқарады. Олардың алдындағы қанат белгісін қоюға және алып тастауға

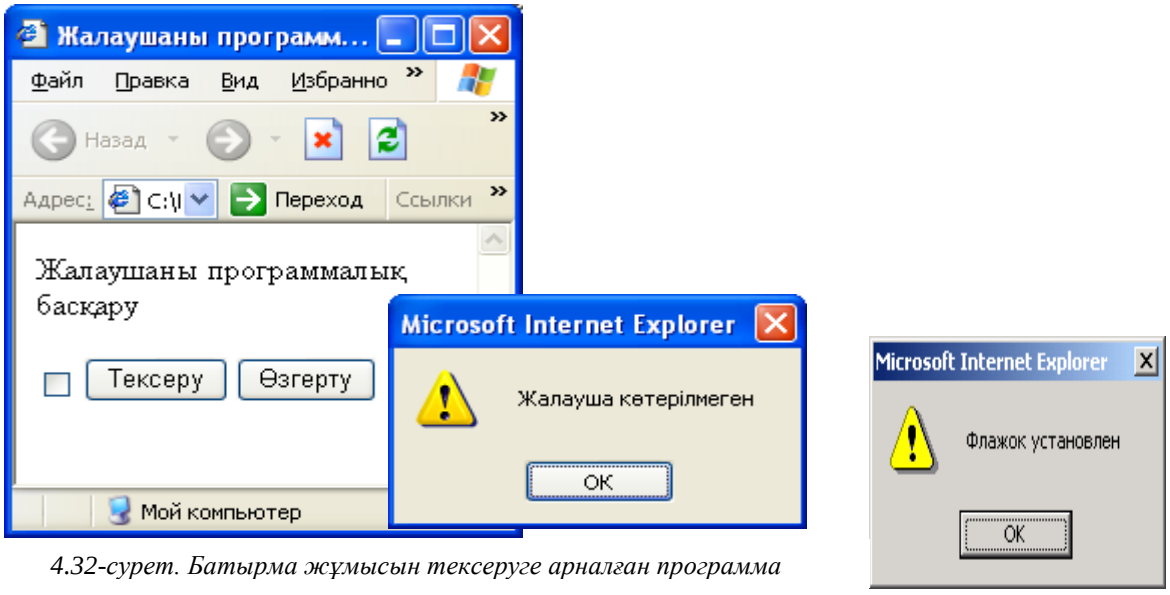
болады. Жалауша құру тәгі: <input type=checkbox>.

<INPUT> тәгінің жалаушаларға арналған қосымша атрибуттары: name - <input> тәгі арқылы жасалған объектіні скриптер ішінде пайдалану үшін

қолданылатын атау; checked - логикалық атрибут, мәні болмайды. Ол тек жалаушаның алдына алдын ала келісім бойынша қанат белгі қойылатынын көрсетеді. Оны программалық басқару браузерде мынадай тәгпен тұрғызылған объектінің <input type=checkbox>checked атты қасиеті болады. Жалауша іске қосылып тұрса, оның мәні – true, әйтпесе – false болады. Осыларды пайдаланатын төмендегі программаны пайдаланып, оның нәтижелерін де қарастырайық.

```
<HTML>
 <HEAD>
  <TITLE>Жалаушаны программалық басқару</TITLE>
  </HEAD>  <BODY bgcolor=white text=black>
  Жалаушаны программалық басқару
  <FORM>
 <INPUT type=checkbox name=flag>
   <INPUT type=button value=Тексеру        onclick="if(this.form.flag.checked)
alert('Жалауша көтерілген');  else
alert ('Жалауша көтерілмеген');">
<INPUT type=button value=Өзгерту         onclick="this.form.flag.checked =
!this.form.flag.checked;">
  </FORM>
 </BODY>
</HTML>
```

Енді мынадай тәжірибелер жасауға болады.

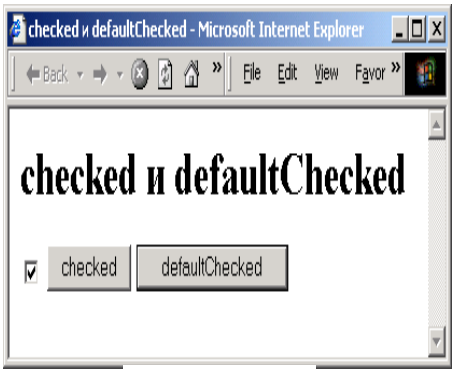


4.32-сурет. Батырма жұмысын тексеруге арналған программа

1.**Тексеру** батырмасын шертсек, «Жалауша көтерілмеген» мәліметі шығады. Өйткені мұнда checked атрибуты іске қосылмаған:
<INPUT type=checkbox name=flag> <input> тәгі объектісінің checked қасиеті мәні – false, оны this.form.flag.checked өрнегімен де беруге болады (if командасында).
2.**Өзгерту** батырмасын шертсек, мына код жұмыс істейді: this.form.flag.checked=!this.form.flag.checked; Checked қасиеті мәнін өзгертеді

(false ® true. Экрандағы жалауша алдында белгі пайда болады, (оны алдын ала қоймасақ та, белгі программа арқылы орнатылды).

3.**Тексеру** батырмасын шертсек, экранға «Жалауша көтерілген» мәліметі шығады. Енді this.form.flag.checked мәні true болады. Сонымен, checked қасиетін тек оқуға емес, оны өзгертіп мәнін қарама қарсыға ауыстыру программалық жолмен де жүргізіледі екен.



4.33-сурет

Мұндағы объектінің checked қасиеті <input> тәгіндегі атрибут мәнінен басқаша, онда ол келісім бойынша іске қосылса, мұнда код бойынша іске қосылады. Егер программада <input> тәгінің checked атрибутының мәнін білгіміз келсе, онда сол объектінің defaultChecked мәнін анықтауымыз керек. Енді браузерге келесі кодты енгізіп, оның нәтижелерін қарап шығайық.

<HTML> <HEAD> <TITLE>checked и defaultChecked</TITLE> </HEAD> <BODY bgcolor=white text=black> <H2>checked и defaultChecked</H2> <FORM> <INPUT type=checkbox checked name=flag> <INPUT type=button value=checked onclick="alert(this.form.flag.checked);"> <INPUT type=button value=defaultChecked on-click="alert(this.form.flag.defaultChecked);"> </FORM> </BODY> </HTML>

Келесі тәжірибелерді жасау керек:

1. checked и defaultChecked батырмаларын біртіндеп басайық Сонда «true» және «true» мәліметтерін аламыз. Бұл жалаушаның да, белгінің де қатар орнатылғанын көрсетеді.
2. Жалауша алдындағы белгіні алып тастап, қайтадан checked және defaultChecked батырмаларын шертеміз. Сонда «false» және «true» мәліметтерін аламыз. Бұл жалаушаның орнатылмағанын, бірақ белгінің алдын ала қойылғанын көрсетеді.
3.<INPUT type=checkbox checked name=flag> тәгінен checked атрибутын алып тастайық, сонан соң checked және defaultChecked батырмаларын қайта басайық. Сонда «false» және «false» мәліметтерін аламыз. Бұл жалаушаның орнатылмағанын және белгінің алдын ала да қойылмағанын көрсетеді.
4. Жалауша алдына белгі қойып, checked и defaultChecked батырмаларын шертеміз. Сонда «true» және «false» мәліметтерін аламыз. Бұл жалаушаның орнатылғанын және белгінің алдын ала да қойылмағанын көрсетеді.

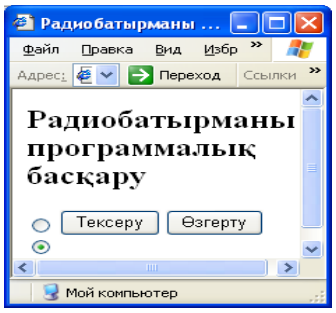
Радиобатырмалар

Радиобатырмалар (селекторлық батырмалар, ауыстырғыш) арқылы бірнеше мүмкіндіктің бірін таңдай аламыз. Батырманың бірін белгілесек, қалғандарындағы белгі автоматты түрде жоғалады. Радиобатырма <input type=radio> тәгімен беріледі.

name - атауы; <input> тәгі арқылы жасалған объектіні скриптер ішінде пайдалану үшін қолданылатын атау. Бір топқа кіретін батырмалар аты бірдей болуы тиіс. Браузер осы атау бойынша ауыстырғыштарды анықтап, батырмаларды белгілейді немесе басқа батырма таңдалса, белгілемейді. Топтағы жеке батырманы анықтау массив элементтерін анықтау сияқты индекс арқылы орындалады. Топтағы батырмалар орналасу кезегі бойынша нөмірленеді, нөмірлеу нөлден басталады.
checked - логикалық атрибут, мәні болмайды. Ол тек батырманың алдына алдын ала келісім бойынша таңдалып алынғанын көрсетеді.
Программалық басқару. Браузердің <INPUT type=radio> тәгі арқылы тұрғызған объектісінің checked қасиеті болады. Ол таңдалып алынса, мәні - true, ал таңдалмаса мәні - false болады.

Келесі программаны қарастырайық.

```
<HTML>
 <HEAD> <TITLE>Радиобатырманы программалық басқару</TITLE>
 </HEAD>
<BODY bgcolor=white text=black>
<H2>Радиобатырманы программалық басқару</H2>
 <FORM>  <INPUT type=radio name=tst>
  <INPUT type=button value=Тексеру
oncliok="if(this.form.tst[0].checked)
alert('1-батырманы белгілеу орындалды') else alert('Бірінші батырманы белгілеу орындалмады');">
<INPUT type=button value=Өзгерту onclick="var x =
this.form.tst[0].checked;this.form.tst[0].checked=this.form.tst[1].checked;this.form
.tst[1].checked=x;">
<BR> <INPUT type=radio checked name = tst>
</FORM> </BODY>
</HTML>
```



3.34-сурет

Келесі әрекеттерді орындап шығайық.
1. *Тексеру* батырмасын шертсек, экранға «1-ші батырманы белгілеу орындалмады» мәліметі шығады. Негізінде батырма іске қосылмаған, өйткені оның коды <INPUT type=radio name=tst>, атрибуттарына checked енгізілмеген. Бұл осы батырма тұрғызған объектінің checked қасиеті іске қосылмағанын көрсетеді, сондықтан оның мәні false. Егер if командасы арқылы this.form.tst[0].checked өрнегін пайдалансақ та, нәтижесі осындай болады.
2. *Өзгерту* батырмасын шертсек, мынадай код орындалады: onclick="var x = this.form.tst[0].checked;this.form.tst[0].checked= this.form.tst[l].checked;this.form, test[1].checked=x;" Бұл код белгіні екінші батырмадан біріншісіне ауыстырады.

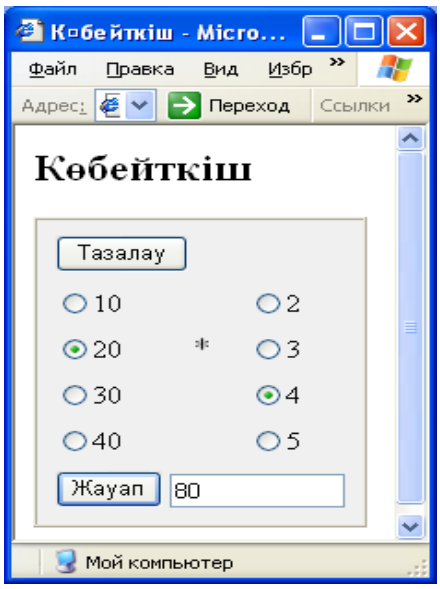
3. Тексеру батырмасын шертсек, «1-ші батырманы белгілеу орындалды» мәліметі шығады. Бұл мынадай өрнектің this.form.tst[0].checked мәні true болатынын білдіреді.

Объектінің checked қасиеті <input> тәгінің осындай атрибутымен бірдей емес. Checked атрибуты батырмаға алдын ала белгі қояды, ал checked қасиеті батырманың дәл сол мезеттегі орнатылған мәнін береді. Жалаушалардағыдай checked атрибутуының defaultChecked қасиеті бар.

Бір батырманы іске қосу арқылы қалғандарының алдындағы checked арқылы қойылған белгі алынатыны программалаушы есінде болуы тиіс (тек бір батырма белгіленеді).

Мысал. Екі бағанаға көбейткіштер мәнін жазып, көбейтіндіні анықтау сценарийін радиобатырмаларды қолдану арқылы жазайық.

```
  <HTML>
<HEAD>
<TITLE>Көбейткіш</TITLE>
</HEAD>
<BODY bgcolor=white text=black>
<H2>Көбейткіш</H2>
 <FORM>
 <TABLE border=1 bgcolor=#F0F0F0          cellspacing=0 cellpadding=5>
 <TR><TD>
 <TABLE border=0 cellspacing=0            cellpadding=5>
 <TR> <TD colspan=3><INPUT type=reset
 value=Тазалау></TD>
</TR>
<TR> <TD>
<INPUT type=radio checked name=op1>10</TD>        <TD> </TD>
<TD><INPUT type=radio checked name=op2>2</TD>       </TR>
<TR> <TD><INPUT type=radio name=op1>20</TD>
<TD><BIG>*</BIG></TD>
 <TD><INPUT type=radio name=op2>3</TD></TR><TR> <TD><INPUT
type=radio name=op1>30</TD>
<TD> </TD> <TD><INPUT type=radio name=op2>4</TD>
</TR><TR> <TD><INPUT type=radio name=op1>40</TD> <TD> </TD>
 <TD><INPUT type=radio name=op2>5</TD></TR><TR> <TD colspan=3>
<INPUT type=button value=Жауап onclick="var op1=new Array(10,20,30,40);
  var op2  = new Array(2,3,4,5); var len  = 4;       var ind1;       var ind2;
 for(var i=len; --i>=0;)       {       if(this.form.op1[i].checked) ind1=i;
if(this.form.op2[i].checked) ind2=i;       }
 this.form.result.value= op1[ind1]*op2[ind2];">       <INPUT type=text size=10
name=result
 value="" readonly>     </TD>
 </TR>   </TABLE> </TD></TR> </TABLE>
</BODY>  </HTML>
```



4.35-сурет. Көбейткіштің экрандағы көрінісі

Бақылау сұрақтары:

1. JavaScript тіліндегі объектілерге сипаттама беріп, мысалдар келітіріңіз.
2. Windows объектісінің атқаратын қызметін сипаттаңыз.
3. Document объектісінің атқаратын қызметі.
4. Объектілерді программалық басқару мүмкіндігін қалай орнатуға болады?
5. JavaScript тілінде батырмалардың қандай түрлері қолданылады.
6. Ауыстырып қосқыш атрибуттарына сипаттама беріңіздер.
7. Радиобатырма атрибуттарына сипаттама беріңіздер

Тапсырмалар:

1. Әрбір он студент үшін сауалнамада келесі мәліметтер келтіріледі: тегі, екі бақылау жұмысының бағасы. Студенттер бірнеше топқа бөлінеді. «Үздіктер» қатарына екі бақылау жұмысының бағасы бес, «жақсы үлгерушілер»– бір бағасы 4, «үлгерушілер», бір бағасы 3, ал «үлгермеушілер» бір бағасы 2 студенттер кіреді. Әрбір топтағы студенттің санын анықтау сценарийін жазу керек.

2. Әрбір алты студент үшін сауалнамада келесі мәліметтер толтырылады: тегі, сессиядағы төрт бағасы. Әр топтағы студенттер санын анықтау сценарийін жазыңыз. Топтар келесі түрде анықталады: барлық сабақтан 5 алғандар «үздіктер», бір сабақтан 2 алғандар «үлгермеушілер», қалған студенттер «үлгерушілер».

3. Тестілеу нәтижесі бойынша сауалнама толтырылады: тегі, әрбір тесті орындау нәтижесі (плюс, егер сәтті тапсырылса, минус, егер тест орындалмаса). Егер тесттің барлығы орындалса жұмыс 5-ке, төртеуі орындалса – 4-ке, үш тест орындалса – 3-ке, басқа жағдайда 2-ге бағаланады. Бағасын есептеу және қорытынды шығару сценарийін жазыңыз. Қорытындыда 5, 4, 3, 2 алған студенттер саны жөнінде мәлімет беріледі.

5 Web-қосымшалар интерфейсі

Common Gateway Interface (CGI - жалпы шлюздік интерфейс) нені білдіретіндігін түсіну үшін тұтас WWW көзқарасымызды талдап көрейік.

Web-ті CERN-де жұмыс істеген физика маманы Тим Бернерс-Ли жоспары 1988 жылы құрылғанымен, тек 1990 жылы табылған. Оның негізгі ойы мультимедиалық мәліметтерді – мәтінді, бейнелер мен дыбыстарды Интернет арқылы алмасу мүмкіндігін жүзеге асыру болды. WWW негізгі үш бөліктен тұрды: HTML, URL және HTTP. HTML – Web-тегі мәліметтерді көрсету үшін қолданылатын форматтау тілі. URL – веб-серверден HTML (немесе басқа) форматындағы мәліметтерді алу үшін қолданылатын адрес. HTTP – бұл веб-серверге түсінікті және клиенттерге серверден құжаттарды сұрауға мүмкіндік беретін тіл.

HTTP хаттамасы

HTTP хаттамасы бойынша жұмыс келесі түрде жүргізіледі: программа-клиент сервермен TCP-байланысты орнатады (порттың стандартты нөмірі - 80) және оған HTTP-сұраныс жібереді. Сервер бұл сұранысты өңдеп, клиентке HTTP-жауап жібереді.

HTTP-сұраныстың құрылымы. HTTP-сұраныс бос қатармен бөлініп жазылған сұраныстың тақырыбымен сұраныс денесінен тұрады. Сұраныс денесі болмауы да мүмкін. Сұраныс тақырыбы сұраныстың негізгі (бірінші) қатарынан және негізгі қатардағыларды анықтай түсетін келесі қатарлардан тұрады. Келесі қатарлардың да болмауы мүмкін. Негізгі қатардағы сұраныс бос орынмен бөлінген үш бөлімнен тұрады:

1. *Әдіс* (басқаша айтқанда, http командасы):
   - GET – құжат сұранысы. Анағұрлым көп қолданылатын әдіс; HTTP/0.9-да, осы әдіс қана болған.
   - HEAD – құжат тақырыбына сұраныс. GET-тен, тек құжат жөнінде ақпарат жазылған тақырып сұранысы шығарылатындығымен ерекшеленеді. Құжаттың өзі шығарылмайды.
   - POST – бұл әдіс мәліметтерді CGI-скрипттеріне жіберу үшін қолданылады. Мәліметтердің өзі келесі қатарларда параметрлер түрінде орналасады.
   - PUT – құжатты серверге орналастыру. Сирек қолданылады. Бұл әдіс қолданылған сұраныс құжаттың өзі жіберілетін бөліктен тұрады.
2. *Ресурс* – бұл клиент алғысы келетін, сервердегі нақты файлдың орналасқан орнының жолы (немесе PUT әдісі үшін орналастыру). Егер ресурс – оқуға арналған жай ғана файл болса, сервер бұл сұраныс бойынша оны жауап бөлігінде шығаруы керек. Егер бұл CGI-скриптке апаратын жол болса, онда сервер скрипті іске қосып, оның орындалу нәтижесін шығарады.
3. *Хаттама нұсқасы* – клиент программасы жұмыс істейтін, http хаттамасының нұсқасы.

Сонымен қарапайым HTTP-сұраныс келесі түрде болады:

GET / HTTP/1.0 - web-сервердің түпкі директориясынан түпкі файл сұралады.

Сұраныстың негізгі қатарынан кейінгі қатардың форматы: *Параметр: мәні* түрінде болады. Осылай сұраныс параметрлері беріледі. Бұл міндетті емес, негізгі қатардан кейінгі қатарлардың барлығы болмауы да мүмкін; бұл кезде сервер олардың мәнін үнсіз келісім немесе алдыңғы сұраныстың нәтижесі (Keep-Alive режимінде жұмыс істеу кезінде) бойынша қабылдайды.

HTTP-сұраныстың анағұрлым көп қолданылатын параметрлерін қарастырайық:

- Connection (байланыстыру)- Keep-Alive және close мәндерін қабылдай алады.
- Keep-Alive ("тірі қалдыру") құжаттты шығарғаннан кейін сервермен байланыс үзілмейтіндігін, тағы да сұраныстар беруді жалғастыруға болатындығын білдіреді. Көптеген браузерлер Keep-Alive режимінде жұмыс істейді, өйткені ол сервермен байланыстырып, html-парақтар мен ондағы суреттерді жазып алу мүмкіндігін береді. Бұл режим қосылғаннан кейін алғашқы қате кездескенге немесе Connection: close режимінің кезекті қосылғанына дейін жұмыс істейді.
- close ("закрыть") – байланыс берілген сұранысқа жауаптан кейін үзіледі.
- User-Agent – мәні браузердің "кодтық белгілеуі" болады, мысалы: *Mozilla/4.0 (compatible; MSIE 5.0; Windows 95; DigExt)*
- Accept – браузер қолдайтын типтер мазмұны берілген браузердің ретіндегідей болады, мысалы, IE5 үшін*: Accept: image/gif, image/x-xbitmap, image/jpeg, image/pjpeg, application/vnd.ms-excel, application/msword, application/vnd.ms-powerpoint,* */*. Бұл параметрдің мәні негізінен жауаптарды құру үшін CGI-скриптерінде қолданылады.
- Referer - URL, осы ресурсқа көшу.
- Host – ресурс сұралатын хостың аты. Серверде бір IP-адреспен бірнеше виртуалды серверлер болған кезде пайдаланылады. Бұл кезде виртуалды серверді осы өрісі бойынша анықтайды.
- Accept-Language – қолдайтын тіл. Бір құжатты әр түрлі тілдер нұсқаларымен беру кезінде маңызды орын алады.

HTTP-жауаптардың форматтары. Жауап форматы сұраныс форматына өте ұқсас. Ол да бос қатармен бөлінген тақырып пен құжат денесінен тұрады. Тақырып негізгі қатар мен парамертлер қатарынан тұрады. Бірақ негізгі қатардың форматының сұраныс қатарындағыдан айырмашылығы бар. Сұраныстың негізгі қатары бос орынмен бөлінген 3 өрістен тұрады:

- Хаттама нұсқасы – сұраныстың сәйкес параметріне ұқсас.
- Қате коды – кодтық белгілеу. 200 коды "барлығы дұрыс" (OK) дегенді білдіреді.
- Қатенің сөзбен талдануы – алдыңғы кодты "талдау". Мысалы, 200 үшін OK, 500 үшін - Internal Server Error.

http-жауаптың анағұрлым көп қолданылатын параметрлері:

- Connection – сұраныстың сәйкес параметрі сияқты. Егер сервер Keep-Alive режімін қолдамаса, онда Connection мәні әрқашан close.
- Content-Type ("мазмұнның типі") – жауап мазмұнының типі.

- Content-Type мәніне байланысты браузер жауапты HTML-парақ, gif немесе jpeg суретті, дискіге сақтауға арналған файл т.с.с сияқты қабылдайды. Content-Type мәні браузер үшін Windows арналған файл типіне ұқсаса болады.

  Құжаттардың типтері:
  - ♦ text/html - HTML (веб-парақ) форматындағы мәтін;
  - ♦ text/plain – жай мәтін ("блокноттағыдай");
  - ♦ image/jpeg – JPEG форматындағы сүрет;
  - ♦ image/gif – GIF форматындағы сурет;
  - ♦ application/octet-stream - "октеттер" (т.е. жай байт) ағыны дискіге жазу үшін.
- Content-Length ("мазмұн ұзындығы") –жауап мазмұнның байтпен берілген ұзындығы.
- Last-Modified ("Соңғы рет жаңартылған") – құжаттың соңғы өзгертілген уақыты.

CGI – бұл сервердегі программалар веб-сервер арқылы клиенттерге мәліметтерді жіберетін ережелер жинағы.

CGI программаларға мәліметтерді клиенттерге жөнелту мүмкіндігін берсе, формалар CGI-программалар үшін осы мүмкіндікті кеңейтеді.

CGI қосымшаларын таратуға келесілер жатады:
- Динамикалық HTML. Тұтас сайттар бір CGI-программасымен басқарылады.
- Пайдаланушы енгізген сөздері бар құжаттарды табатын іздеу механизмдері.
- Пайдаланушылар өз хабарламаларын қоса алатын хабарландыру тақталары мен қонақ кітаптары.
- Тапсырыс қағаздары.
- Сауалнамалар (анкеты).
- Серверге орналастырылған мәліметтер базасынан ақпараттарды шығару.

Осылардың барлығы CGI-ді мәліметтер базасымен байланыстыруға мүмкіндік береді.

CGI спецификациясы

CGI-дің CGI-программасы, веб-сервер және Web клиенті арасында мәліметтерді жіберетін төрт тәсілі бар:
- Айнымалылар қоршауы.
- Командалық қатар.
- Стандартты енгізу құрылғысы.
- Стандартты шығару құрылғысы.

Осы төрт әдістің көмегімен сервер клиент жіберген барлық мәліметтерді CGI-программаға қайта жібереді. Одан кейін CGI-программа өзінің таңғажайып ісін орындап, барлық шығу ақпараттарын қайтадан серверге жібереді.

Айнымалылар қоршауы

Сервер CGI-программаны орындаған кезде, ең алдымен оған жұмысқа арналған бірқатар мәліметтерді айнымалылар қоршауы түрінде жібереді. Спецификацияда ресми түрде он жеті айнымалы анықталған, бірақ HTTP_mechanism деп аталатын механизм көмегімен бұдан көбірек қолданылады. CGI-программа бұл айнымалылардың кез келгенін командалық қатардан іске қосу кезіндегі командалық үрдісор ортасының кез келген айнымалысы сияқты қолдана алады. Мысалы, командалық үрдісор сценарийінде FOO айнымалылар қоршауымен $FOO түрінде қатынасуға болады. 5.1-кестеде мәні null болса да, әрқашанда сервер орнататын айнымалылардың тізімі берілген. Бұл айнымалылардан өзге, клиентке оралған сұраныстың тақырыбында HTTP_FOO түріндегі айнымалылар бар, мұндағы FOO – тақырып аты. Мысалы, көптеген веб-броузерлерде USER_AGENT түріндегі мәліметтер бар. Біздің CGI-программа бұл мәліметтерді HTTP_USER_AGENT айнымалысынан ала алады.

5.1- кесте. CGI айнымалылар қоршауы

Айнымалылар қоршауы	**Сипаттамасы**
CONTENT LENGTH	POST немесе PUT әдісімен берілген мәліметтер ұзындығы, байтпен
CONTENT_TYPE	POST немесе PUT әдісінің көмегімен байланыстырылған MIME мәліметтерінің типі.
GATEWAY_INTERFACE	Сервер қолдайтын CGI спецификациясының нұсқасының нөмірі
PATH_INFO	Клиент жіберген жол жөнінде қосымша ақпарат. Мысалы, http://www.myserver.com/test.cgi/this/is/a/ path?field=green сұранысы үшін PATH_INFO айнымалысының мәні /this/is/a/path болады
PATH_TRANSLATED	PATH_INFO сияқты, бірақ сервер барлық мүмкін болатын трансляцияны орындайды, мысалы «~account» типі атауын орналастыру.
QUERY_STRING	URL-де «?» символынан кейінгі барлық мәліметтер. Бұл сонымен бірге GET формасы бар REQUEST_METOD берілетін мәліметтер.
REMOTE_ADDR	Сұраныс жасап отырған клиенттің IP-адресі.
REMOTE_HOST	Қатынас құруға болатын клиент машинасы түйінінің аты.
REMOTE_IDENT	Егер веб-сервер мен клиент identd түріндегі идентификацияны қолдаса, сұраныс жасайтын тіркеу жазбасының пайдаланушы аты.
REQUEST_METHOD	Клиент қолданатын сұранысқа арналған әдіс. CGI-программалары үшін, бұл әдетте POST немесе GET болады.
SCRIPT_NAME	Клиент көрсеткен орындалатын сценарий жолы. URL сілтемесі кезінде өзіне өзісілтеме ретінде қолданылады және әр түрлі орындардағы сілтемелер орналасқан орнына қатыссыз орындалу үшін қолданылады.
SERVER_NAME	Машинаның аты қатынас құруға мүмкін болмаған жағдайда Веб-сервер орындалатын түйін аты немесе IP-адрес.
SERVER_PORT	Веб-сервер қолданатын порттың нөмірі..
SERVER_PROTOCOL	Сервермен байланысу үшін клиент қолданатын хаттама. Біздің жағдайымызда бұл әркезде дерлік HTTP болады.
SERVER_SOFTWARE	CGI-программаны орындайтын веб-сервердің нұсқасы жөнінде

	мәлімет.

Сервер жіберілген, сервер орнатқан барлық айнымалылар қоршауын, сонымен бірге командалық үрдісор орнатқан барлық мұраланған айнымалыларды шығаратын, Perl-де жазылған CGI сценарийіне мысал келтірейік.

5.1- листинг. Айнымалалар қоршауы мәндерін шығару.

```
print "Content-Type: text/html\n\n
<HTML><HEAD><TITLE></title></head><BODY>\n
<p>Айнымалылар қоршауы:<p>\n";
foreach (keys %ENV) {print "$_: $ENV{$_}<br>\n" }
print "</body></html>";
```

Бұл айнымалылардың барлығы біздің CGI-программада қолданылып, тіпті өзгертілуі де мүмкін. Бірақ, бұл өзгертулер программаны іске қосатын веб-серверге әсер етпейді.

CGI кіріс мәліметтерді стандартты **stdin** енгізу ағынынан немесе айнымалылар арасынан алады да, өз жұмысының нәтижесін стандартты шығару ағыны **stdout** жібереді.

- *Стандартты енгізу ағыны* – программа (скрипт) үнсіз келісім бойынша кіріс ақпараттарды алатын орын. Әдетте бұл пернетақта болады, бірақ оны өзгерте отырып, программа кіріс ақпараттарды файлдан, сокеттен немесе басқа программаның шығыс ағынынан алатындай етуге болады.
- *Айнымалылар қоршауы (Переменные окружения)* (environment variables) – жүйе және сервер үшін анықталған, CGI орындалатын айнымалылар.
- *Стандартты шығару ағыны* (stdout) - программа (скрипт) өз жұмысының нәтижелерін шығаратын орын. Әдетте бұл экран болады, бірақ оны файлға, сокетке, басқа программаның кіріс ағынына, принтерге т.б. өзгертуге болады.

CGI-ді параметрлерсіз шақыру

Ағымдағы мерзімді шығаратын қарапайым скрипт:

```
#!/bin/sh
echo Content-type: text/html
echo
echo "<h2>Today is "
date
echo "</h2>"
```

HTML құжатында бұған сілтеме келесі түрде жазылады:

```
<a href="/cgi-bin/examples/today.cgi">
```

Ескерту: шығарылатын нәтиже – шығарылатын құжаттың тақырып қатарының типін көрсетуді ұмытпау керек. Бұл мысалдағы екінші және үшінші қатар.

```
echo Content-type: text/html
echo
```

мұндағы **Content-type:** - шығарылатын құжаттың типі (text/html, image/gif, image/jpeg және т.б.).

Бос қатар тақырып қатарының аяқталып, ары қарай құжаттың өзі орналасатындығын білдіреді.

CGI скриптіне немесе программасына параметрлерді жіберу

Параметрлерді жіберу негізгі екі әдіспен жүзеге асады: **GET** және **POST**. Бұл әдістердің әрқайсысының кемшіліктері мен артықшылықтары бар.

GET **әдісін қолданғанда параметрлер сұралып отырған** URL-ге қосылады, оны келесі түрде шақыруға болады:

http:// кез келген_хост/cgi-bin/кез келген_скрипт? HTML құжаттарында осы скриптке сілтеме жасауға мүмкіндік беретін параметрлер.

Ал серверде жіберілген параметрлер QUERY_STRING айнымалысына меншіктеледі.

сол скрипт мәтіні:

```
#!/bin/sh
echo Content-type: text/html
echo
echo "<h2>Сіздің жөнелткеніңіз осы:</h2>"
echo "<b>"
set | grep QUERY_STRING
echo "</b><br><hr>"
echo "<b>Environment</b><br><pre>"
set
echo "</pre>"
```

ол құжаттан келесі түрде шақырылды:

```
<a href="/cgi-
bin/examples/link.cgi?some_parameters">
Жұмыс мысалы (осы жерді шертіңіз)
</a>
```

Бірақ GET әдісін барлық ақпарат ашық түрде жіберілетін болғандықтан, құпия ақпараттарды жөнелту кезінде қолдануға болмайды. Web қорғанысын қамтамасыз ету үшін қосымша ақпараттарды алу үшін WWW Security FAQ хабарласуыңызға болады.

POST әдісі скриптке параметрлерді жіберуде құпиялылықты сақтауға мүмкіндік береді. Бірақ ол параметрлерді стандартты енгізу ағынына жібереді де, ол үшін формаларды қолдануға тура келеді. Сервер скриптке жіберу соңында EOF жөнелтпейді. Оның орнына stdin-нен мәліметтердің оқылатын көлемін анықтау үшін, айнымалылардың арасынан CONTENT_LENGTH қолдану керек.

Мысалдар

GET әдісі

Press me. Submit

Құжатта бұл форма келесі түрде жазылған:

```
<FORM ACTION="/cgi-bin/examples/forms.cgi"
METHOD="GET">
```

```
      <INPUT TYPE="checkbox" NAME="button" VALUE="on">
Press me.
      <INPUT TYPE="submit" VALUE="Submit">
      </FORM>
```

POST әдісі

☐ Press me. Submit

Құжатта бұл форма келесі түрде жазылған:

```
      <FORM ACTION="/cgi-bin/examples/forms.cgi"
METHOD="POST">
      <INPUT TYPE="checkbox" NAME="button" VALUE="on">
Press me.
      <INPUT TYPE="submit" VALUE="Submit">
      </FORM>
```

Ал осы орындалған скрипттің әрқайсысы келесі түрде көрінеді:

```
      #!/bin/sh
      echo Content-type: text/html
      echo
      echo "<b>Сіз жөнелттіңіз (POST әдісі
үшін):</b><br>"
      echo "<h2>"
      cat
      echo "<br>"
      set | grep CONTENT_LENGTH
      echo "</h2><hr>"
      echo "<b>Environment</b><br><pre>"
      set
      echo "</pre>"
```

Скриптердің сақталатын орны

CGI стандарты бойынша алдын ала скриптердің сақталатын орны, яғни каталог немесе диск анықталмайды. Әдетте Web-сервер скриптерді сервер программасының катлогындағы */CGI-BIN* каталогынан іздейді.

CGI-скриптер файлдарының типтері

Windows-жүйелеріне арналған HTTP серверлері әдетте CGI-файлдары үшін EXE немесе PL тіркеулерін қолданады. Мысалы егер CGI-программа (скрипт), C программалау тілінде құрылса, онда скрипт-файлдардың типі EXE болады. Егер скрипт Perl тілінің көмегімен құрылса, онда файлдың типі PL болады.

Бірақ бірқатар серверлер скриптер үшін CGI тіркеуін қолдайды. Сервердің қандай тіркеуді қолдайтынын Web-шеберден анықтап алған абзал.

AUTH_TYPE айнымалылар қоршауы

CGI скриптілері *AUTH_TYPE* айнымалылар қоршауын скриптіге қатынас құратын пайдаланушыларды идентификациялау үшін қолданады.

Егер сервер пайдаланушыны идентификациялауға ыңғайланған болса, онда скриптпен қатынас құру үшін әр пайдаланушы өз аты мен құпия сөзін енгізуі керек. Мысалы, айнымалының келесі мәні пайдаланушыдан идентификацияның негізгі деңгейінен өтуін талап ететіндігін білдіреді:

AUTH_TYPE = Basic

CONTENT_LENGTH айнымалысы

Скрипттер *CONTENT_LENGTH* айнымалылар қоршауын байланыстырылған мәліметтерде дәл қанша байт бар екендігін анықтау үшін қолданады. Мысалы, егер сұраныс ұзындығы 1,024 байттан тұратын мәлімет болса, онда айнымалыға келесі мән меншіктеледі:

CONTENT_LENGTH = 1024

CONTENT_TYPE айнымалысы

Скрипттер бұл айнымалылар қоршауын байланыстырылған ақпараттардан тұратын сұраныстар үшін қолданады. Мұндай сұраныстар типіне *POST* HTTP-операциясы жатады. Айнымалының мәні байланыстырылған ақпараттың типін көрсетеді (MIME-тип. ішкі тип). Мысалы, егер сұраныс HTML типіндегі құжатпен байланыстырылған болса, онда айнымалылар қоршауының мәні:

CONTENT_TYPE = text/html

GATEWAY_INTERFACE айнымалысы

Скрипттер бұл айнымалыны Web-сервер-ді қанағаттандыратын CGI спецификасының нұсқасын анықтау үшін қолданады. Спецификаның шығу нөмірінің форматы: CGI/нұсқасы. Мысалы, CGI-дің 1.1 нұсқасы үшін айнымалының мәні:

GATEWAY_INTERFACE = CGI/1.1

PATH_INFO айнымалысы

Скрипттер бұл айнымалыны клиентті қамтамасыз ететін қосымша жолдарды анықтау мақсатында қолданады. Басқаша айтқанда, сервер скрипт үшін виртуалды жолды қолдана алады. Серверлік программа бұл қосымша ақпаратты декодтауы тиіс. Әдетте бұл қосымша ақпарат сұраныс сәтті орындалған жағдайда ресурсты көрсетеді

Жол сервердің түпкі каталогын негізге алып салыстырмалы формада жазылады. Мысалы, егер *c:/cgi-bin/example1.exe/sports.html* жолы берілсе, онда айнымалы мәні:

PATH_INFO = /sports.html

PATH_TRANSLATED айнымалысы

Скрипттер бұл айнымалыны тікелей қолдануға қажетті жолға қатысты ақырғы ақпаратты алу үшін қолданады. Сервер қажетті ақпаратты айнымалыға жолдың қажетті түрлендірулерін орындай отырып аударады.Мысалы, егер *PATH_TRANSLATED* айнымалысының мәні */sports.html,* болса, сервердің түпкі каталогы *C:* болады, онда айнымалының мәні:

PATH_TRANSLATED = c:sports.html

QUERY_STIRNG айнымалысы

Скрипттер бұл айнымалыны пайдаланушыдан скриптке өңдеу үшін жіберілген URL ден кейін сұрау белгісінің оң жағындағы, ақпараттарды мәтіндік формада (аргументтерден тұратын) алу үшін қолданады. Бұл мәтіндік қатар скриптке кіруден тұрады. Ары қарай сервер бұл мәтіндегі әрбір бос орынды " + " таңбасына, ал барлық басылмайтын символдарды " %dd" таңбасымен алмастырады, мұндағы *d* ондық санау жүйесінің базасы болып табылады.

Скрипт осы мәтіндік қатарды талдай алатын кодтан тұруы тиіс. Сервер, бұл ақпаратты скриптке жібере отырып, қайта декодтауды орындамауы қажет. Сонымен қатар пайдаланушы сұраныстың кез келген ақпаратын қамтамасыз еткен жағдайда сервер *QUERY_STRING* айнымалысын орнатуы керек. Мысалы, *http://www.jamsa.com/cgi-bin/grandma.exe?name=margaret+alarcon* URL-і үшін айнымалының мәні келесі шамаға тең болады:

QUERY_STRING = name=margaret+alarcon

REMOTE_ADDR айнымалысы

Скрипттер бұл айнымалыны сұраныс құратын, қашықтағы браузердің IP-адресін алу үшін қолданады. Мысалы, айнымалының мәні келесі түрде болуы мүмкін:

REMOTE_ADDR = 204.212.52.209

Бақылау сұрақтары:

1. CGI сөзі қалай талданады? Нені білдіреді?
2. HTTP хаттамасы бойынша жұмыс қалай жүргізіледі?
3. HTTP хаттамасы бойынша жауап форматы қандай болады?
4. CGI қосымшаларын таратуға жататындар тізімін анықтаңыз.
5. CGI-дің мәліметтерді жіберетін қандай тәсілдері бар?
6. CGI кіріс мәліметтерді қайдан алады?
7. Стандартты енгізу ағыны деген не?
8. Айнымалылар қоршауы деген не?
9. Стандартты шығару ағыны деген не?
10. CGI өз жұмысының нәтижесін қайда жібереді?
11. Серверде жіберілген параметрлер қандай айнымалыға меншіктеледі?
12. Параметрлерді жіберудің қандай тәсілдері бар?
13. Скриптер қайда сақталады?
14. CGI-файлдарының тіркеулері қандай болады?

5.1 Perl тіліне кіріспе

Perl – программалаушы Лари Уоллом (Larry Wall) құрған, көлемді мәтіндер мен файлдарды өңдеуге арналған, интерпретацияланатын тіл болып табылады. Practical Extraction and Report Language (мәліметтерді практикалық ашу және басылымдарды құру) деп талданады. Perl тілінің көмегімен бір немесе бірнеше файлдарды ашу, ақпараттарды өңдеу, нәтижелерін жазу әрекеттерін орындауға болады.

Perl енгізілген мәліметтерді өңдеу әрекетін жүзеге асыру мүмкіндігін береді. Егер пайдаланушы ақпаратты Perl-дің тіркеу формасына енгізсе оны бірнеше тәсілдермен өңдеуге болады:

- Оларды мәтіндік файлға қосу,
- Мәліметтер базасына енгізу,
- Электрондық хатқа енгізу,
- web-параққа қосу,
- жаңа web-парақ құру,
- браузер терезесінде бейнелеу.

Perl тілін іске қосу үшін келесі компоненттер қажет:

1. Мәтіндік файлға сақталған скрипт;
2. Perl интерпретатор. Кез келген Perl скрипт бірінші қатарында осы программаға жолды көрсетеді;
3. Осы скрипт іске қосылатын Web-парақ. Ол, егер скрипт кіріс ақпараттарды талап етпесе, форма немесе жай сілтеме болуы мүмкін;
4. Web-сервер. web-парақ пен скрипт арасындағы өзара байланыс сервер арқылы орындалады. Сондықтан скриптерді жазу және орындау үшін, web-сервермен қатынас құра алатындай болу керек.

Скриптті құру сатылары

Компьютерге қажетті программалық жабдықтарды орнатқан соң, Perl тілін іске қосу үшін келесі әрекеттерді орындау қажет:

1. Скриптті шақыруға арналған форманы құру.
2. Скриптінің өзін құру.
3. Скриптіні жөндеу. Қателерді тексеру.
4. Скриптіні серверге орналастырып, оны орындау құқығын тағайындау.
5. Скриптіні формамен байланыстыру. Мысалы:
   <form method=post action="/cgi-bin/scriptname.pl">
6. Скриптінің формамен бірге дұрыс жұмыс істейтіндігіне көз жеткізу.

Perl скриптінің негізгі бөлімдері

Жалпы жағдайда кез келген Perl скрипт келесі төрт бөлімнен тұрады:

1. **Баптау**. Скриптінің алғашқы бөлімі интерпретаторды іске қосып, скрипт денесінде қолданылатын айнымалыларды орнатады. Интерпретаторды іске қосу үшін программаның сақталған орнын білу керек.
2. **Кіріс мәліметтерді оқу**. Бұл бөлім кіріс ақпараттарды формада өңдеу үшін оқып, сақтайды. Бұл бөлім барлық скрипттерде бірдей.
3. **Кіріс мәліметтерді өңдеу**.Бұл бөлім енгізілген мәліметтерді өңдейді. Ол шешілетін мәселеге қарай қарапайым (5 қатар шамасында) немесе өте күрделі (1000 қатардан артық болуы мүмкін.
4. **Нәтижелерді шығару.** Пайдаланушы әдетте өзінің әрекетіне қандайда бір жауап күтеді. Бұл бөлімді жүзеге асыру өте оңай орындалады.

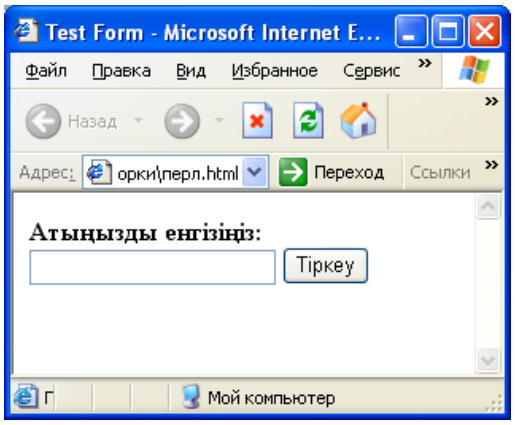
Енді осы айтылған қадамдарды орындауға мысал келтірейік:

1-қадам. Форма құру

Пайдаланушының атын тіркеу мүмкіндігін беретін бір ғана өрістен тұратын қарапайым форма құрайық. Редакторда келесі мәтінді жазайық:

```
<html>
<head>
<title>Test Form</title>
</head>
<body>
<form method=post action"cgi-bin/testform.pl">
<b>Атыңызды енгізіңіз: </b>
<input name="user_name" value="" size=20>
<input type="submit" value="Тіркеу">
</form>
</body>
</html>
```

Файлды дискіге сақтаймыз.



5.1-сурет. Программа нәтижесі

2- қадам. Скрипт құру

Төменде келтірілген скрипт енгізілген мәліметтерді алып, оны файлға сақтап, файлға сілтемеден тұратын хабарламаны көрсетеді. Ол үшін редакторда программа мәтінін теріп, web-сервердің cgi-bin каталогына testform.pl файлына сақтау керек. Программаның бірінші қатарында интерпретатор-программаның сақталған орнын көрсететін жол тұрғанына көзімізіді жеткізуіміз қажет.

```
#!/usr/local/bin/perl # <-- Осыны тексеріңіз
# Read and parse input from the web form
read(STDIN, $buffer, $ENV{'CONTENT_LENGTH'});
@pairs = split(/&/, $buffer);
foreach $pair (@pairs) {
  ($name, $value) = split(/=/, $pair);
  $value =~ tr/+/ /;
  $value =~ s/%([a-fA-F0-9][a-fA-F0-9])/pack("C",
hex($1))/eg;
  $value =~ s/<!--(.|n)*-->//g;
  $input{$name} = $value;
```

```
    }
    # Save the user output in a file
    $targetfile = "/usr/local/www/htdocs/names.html"; #
<-- Осыны тексеріңіз
    open (NAMEFILE, ">>$targetfile");
    print NAMEFILE "<h3>Имя:
",$input{'user_name'},"</h3>n";
    print NAMEFILE "<p><hr><p>n";
    close (NAMEFILE);
    # Send a message back to the user
    print "Content-Type: text/htmlnn";
    print "<h3>Форманы толтырғаныңыз үшін рах-
мет</h3>nБасыңыз ";
    print '<a href="http://server-
name/names.html">сюда</a>'; # <-- Осыны алмастырыңыз
    print ", Өзіңіздің енгізгеніңізді көру үшін.n";
```

3 -қадам. Скриптіні тестілеу

Скриптіні визуалды түрде қате жоқтығына тексеріп, алдын ала cgi-bin каталогына өтіп, командалық жолдан іске қосамыз. Ол келесі түрде орындалады:

```
./testform.pl
/usr/local/bin/perl testform.pl
c:perlperl.exe testform.pl
```

Егер скрипте қате кездесетін болса, келесі түрдегі хабарламалар беріледі:

```
syntax error at testform.pl line 18, near "open"
Execution of testform.pl aborted due to compilation
errors.
```

Мұндай жағдайда көрсетілген қатардағы мәтінді тексеріп шығу қажет. Қателерді түзеткеннен кейін, оны сақтап, келесі түрдегі хабарлама шыққанға дейін тестілеу керек:

```
Content-Type: text/html
<h3>Форманы толтырғаныңыз үшін рахмет!</h3>
Өз енгізгеніңізді көру үшін, <a href="http://server-
ame/names.html"> осы жерді </a>басыңыз
```

4 - қадам. Скриптіні формамен тестілеу.

Егер скрипт өздігінен жұмыс істейтін болса, оны форма арқылы тестілеуге болады:

1. Web-сервердің іске қосылғанына көз жеткізіңіз.
2. Өз браузеріңізді іске қосыңыз.
3. Сіздің форма сақталған URL адресті теріп жазыңыз (адрес қатары http:// түрінде басталуы керек (file:// емес)).
4. Формадағы өріске өз атыңызды теріп, «Тіркеу» батырмасын шертіңіз.
5. **«Форманы толтырғаныңыз үшін рахмет!»** түріндегі хабарлама берілуі тиіс.

Егер сервердің қатесі жөнінде хабарлама берілсе, онда скриптінің дұрыс орналасқандығын, form тәгіндегі action параметрінің мәнінің дұрыстығын тексеріңіз.

6. Дұрыс құрылған парақты көрсеңіз, сілтемеде тышқанды шертіңіз. Осы кезде скриптпен құрылған жаңа парақ ашылуы тиіс.

Енді өзіміз құрған скриптінің төрт қадамын жеке-жеке мұқият талдайық.

Скрипті параметрлерін тағайындау

Жоғарыда айтылғандай, скриптінің бірінші бөлімі бірнеше элементтерді қамтитын баптау бөлімінен тұрады. Бірінші қатар программа-интерпретаторға баратын жолды анықтайды:

```
#!/usr/local/bin/perl          #UNIX үшін
```

немесе

```
Program FilesPerl5perl.exe     # Win32 үшін
```

Сонымен бірге скриптінің басында осы скриптінің не үшін жазылғандығы жөнінде түсініктеме қосуға болады. Түсініктемені программаның кез келген жеріне # символынан бастап жазуға болады.

```
# Бұл түсініктеме
```

немесе

```
open (NAMEFILE, ">$testfile"); #Жазуға арналған файлды ашамыз.
```

Сонымен бірге скриптінің басында барлық тұрақтылар мен айнымалыларды анықтауға да болады (файлдардың орналасқан орнын өзгертуге ыңғайлы болу үшін,барлық жолдарды айнымалылар түрінде көрсеткен дұрыс әсіресе программа 50 қатардан артық болса). Мысалы:

```
$homepage = "http://server_name/home/index.html";
```

Perl тілінде барлық қарапайым айнымалылар$ символынан басталып жазылады. Айнымалылардың басқа көптеген түрлері, мысалы, массив т.с.с. бар. Программаның бірінші және түсініктемеден басқа қатарларының барлығы нүктелі үтірмен «;» аяқталуы тиіс.

Мәліметтерді формадан оқу

Пайдаланушы енгізген мәліметтерді Perl айнымалыларына «оқу» қажет. Пайдаланушы Submit батырмасын басқаннан кейін, браузер формадағы скрипттің аты мен формадан алған мәліметтерді серверге жөнелтеді. Мәліметтер скриптке стандартты кіріске беріледі.

Формада келесі өрістер бар деп есептейік:

5.2-кесте. Өріс сипаттамасы

Формадағы өріс аты	Өріс тәгінде name="xxx" параметрімен анықталған атау	Пайдаланушы мәліметтері
Аты:	user_name	Arman Murat
Мекеме:	co_name	Akbota
Телефон:	Phone	(3272) 55-45-35

Бұл кезде мәліметтер скриптке келесі форматта жіберіледі: user_name= Arman +Murat&co_name= Akbota &phone=(3272)+55-45-35

Perl скрипт бұл қатарды бөліктер бойынша талдап, келешекте өңдеу мақсатында айнымалыларға сақтауы керек.

Бұл әрекетті жүзеге асыратын қатар жеткілікті түрде стандартты болып табылады:

```
read(STDIN, $buffer, $ENV{'CONTENT_LENGTH'});
```

Бұл қатар стандартты кірістен мәліметтерді оқып, оларды $buffer айнымалысына орналастырады. Қатардың ұзындығы скриптке CONTENT_LENGTH айнымалылар қоршауы арқылы беріледі.

Мәліметтер $buffer айнымалысына орналасқаннан кейін, оларды сәйкес мәндерімен жеке айнымалыларға бөлуге болады:

```
@pairs = split(/&/, $buffer);
```

Енді біз келесі жолдық айнымалыларынан тұратын @pairs массивін алдық:

[1] user_name= Arman +Murat
[2] co_name= Akbota
[3] phone= (3272) 55-45-35

Енді осы жолдарды параметр-мәндерге бөлу керек:

```
foreach $pair (@pairs) {
  ($name, $value) = split(/=/, $pair);
  $value =~ tr/+/ /; # қосу таңбасын бос орынмен алмастырамыз
  $value =~ s/%([a-fA-F0-9][a-fA-F0-9])/pack("C", hex($1))/eg;
  $value =~ s/<!--(.|n)*-->//g;
  $input{$name} = $value;
}
```

Осылайша айнымалылар массивін аламыз, олардың индексі форма өрісінің name параметрінің мәні, ал мәні сәйкес өріске енгізілген шама болады.

Мәліметтерді өңдеу

Келесі қадам мәліметтерді өңдеу. Қалай? Оның барлығы мәліметтерді қалай өңдеу керектігіне байланысты болады? Біздің мысалымызда алынған мәліметтер негізінде жаңа HTML құжат құрылады. Өрістің мазмұнының дұрыстығын тексеру әрекеті орындалмай отырғанына көңіл бөлейік. Нақты скриптерде бұл кейбір жағдайларда міндетті түрде қайсыбірінде қалауымыз бойынша орындалады.

Файлға жазу

Мәліметтермен орындалатын әрекеттердің бірі, оларды файлға жазу болып табылады. Perl тілінде файлдарды ашу және жабу үшін қолданылатын функциялар жиыны бар. Жаңа HTML құжатын құруға, мәліметтерді бұрын сақталған HTML құжатына қосуға немесе келешекте өңдеу мақсатында мәтіндік файлға сақтауға болады.

Файлды ашу функциясы келесі түрде жазылады:

```
open (HANDLE, "name"); # Файлды оқу үшін ашу
open (HANDLE, ">name"); # Файлды өзгерту үшін ашы
open (HANDLE,">>name");# Файлдың соңына қосу үшін ашу
```

HANDLE амалдарды орындау кезінде файл идентификаторы ретінде қолданылатын кез келген уақытша атау болып табылады. Файл ашылғаннан кейін, print функциясын қолдана отырып, жазуға болады:

```
print HANDLE "Бұл мәтін файлға дәл осы күйінде жазылады.n";
print HANDLE "Бұл кезде ",$variable," айнымалысы мәтінге ішіне орналасады.";
print HANDLE " $variable айнымалысын мәтін ішіне осылай орналастыруға да болады.n";
print HANDLE "Тырнақша және басқа да арнайы сиволдарды ; "көрсету" қажет.n";
```

Файлмен жұмыс соңында оны міндетті түрде жабу керек:

```
close (HANDLE);
```

Осыдан кейін барлық мәліметтер дискке жазылады.

Файлға жазу алдында web-сервер-дің файл орналасқан директориямен қатынас құра алатындығына және осы файлға өзгеріс енгізу құқығы бар екендігіне көз жеткізу керек.

Close функциясы соңғы файлға жазу функциясына мүмкіндігінше жақынырақ жазылуы керек. Бұл web-сервер көп пайданушыларға арналған ортада орындалатындығымен және скрипті бірмезгілде бірнеше пайдаланушының қосу мүмкіндігі бар екендігімен түсіндіріледі. Файлды өзгерту үшін ашу кезінде файл блоктанады да, скриптің басқа даналары оны аша алмайды, ол сұраныстың орындалуының тежелуіне әкеледі.

Жаңа HTML парағын құру

Бұл мысалда бірқатар ақпаратты файлға жазуды көрсетейік. web-парақ құратын болғандықтан файл кәдімгі мәтінмен бірге HTML тәгтерінен де тұрады. n құрылған файлды келесі жолы ашқанда ыңғайлы болу үшін келесі қатарға көшуді қамтамасыз ету мақсатында қолданылады.

Формадан барлық кіріс мәліметтер $input{field_name} айнымалысында сақталған. Баспа кезінде мұндай айнымалылар тырнақшаға алынып, үтірмен бөлінуі керек.

```
# жазылатын файлдың орнын көрсететін айнымалыны анықтаймыз
$newfile = "c:webserverhtdocsmynewpage.html";
# USERINFO идентификаторын қолданып, файлды ашамыз
open (USERINFO, ">$newfile");
# Мазмұнын жазамыз
print USERINFO "<html>n<head>n";
print USERINFO "<title>Тіркеу жөнінде ақпарат</title>n</head>n";
print USERINFO "n<body>n<h3>Тіркеу мәліметтері:</h3>";
```

```
    print USERINFO "<p><hr></p>n<p>n";
    print USERINFO "Аты: ", $input{'user_name'},"n<br>";
    print USERINFO "Мекеме: ", $in-
put{'co_name'},"n<br>";
    print USERINFO "Телефон: ", $in-
put{'phone'},"n</p>n<p><hr></p>n";
    print USERINFO "<!-- NEW INSERTS GO HERE --
>nn</body>n</html>";
    # Файлды жабамыз
    close (USERINFO);
```

Файл соңына қосу

Бұрыннан сақталған файлдың соңына ақпарат қосу оңай орындалады. Мысалы, жаңа пайдаланушы жөніндегі мәліметті жоғарыда құрылған файлға қосу үшін келесі скрипт қолданылады:

```
    # жазылатын файлдың орнын көрсететін айнымалыны
анықтаймыз
    $targetfile = "c:webserverhtdocsmynewpage.html";
    # NEWINFO идентификаторын қолданып, файлды ашамыз
    open (NEWINFO, ">>$targetfile");
    # жаңа мәліметтерді файлға қосамыз:
    print NEWINFO "nn";
    print NEWINFO "Аты: ", $input{'user_name'},"n<br>";
    print NEWINFO "Мекеме: ", $input{'co_name'},"n<br>";
    print NEWINFO "Телефон: ", $in-
put{'phone'},"n</p>n<p><hr></p>n";
    close (NEWINFO);
```

Файлдың ортасына мәлімет қосу

Жаңа мәліметтерді файлдың ортасына қосу анағұрлым күрделі болып саналады. Бірінші мысалда файлға біз келесі түрдегідей түсініктеме сызығын қосқанымызға назар аударыңыз:

<!-- NEW INSERTS GO HERE -->

Осы түсініктеме жаңа мәліметтерді қосатын орынды білдіретін болсын. Бұл мысалда уақытша файлды құруды қолданамыз.:

```
    # жазылатын файлдың орнын көрсететін айнымалыны
анықтаймыз
    $origfile = "/pathname/originalfile.htm";
    # уақытша файлдың орнын көрсететін айнымалыны
анықтаймыз
    $newfile = "/pathname/newfile.htm";
    open(INFILE, "<$origfile");
    open(OUTFILE, ">$newfile");
    while ($line = <INFILE>) {
      printf OUTFILE $line;
      if ($line =~ /<!-- NEW INSERTS GO HERE -->/i) {
        # Жаңа мәліметтерді файлға қосамыз:
```

```
        print OUTFILE "nn";
        print OUTFILE "Аты: ", $in-
put{'user_name'},"n<br>";
        print OUTFILE "Мекеме: ", $in-
put{'co_name'},"n<br>";
        print OUTFILE "Телефон: ", $in-
put{'phone'},"n</p>n<p><hr></p>n";
      }
    }
    # Файлдарды жабамыз
    close(INFILE);
    close(OUTFILE);
    # Бастапқы файлды өшіріп, жаңа файлды бастапқының
атына ауыстырамыз
    unlink($origfile);
    rename($newfile, $origfile);
```

Мәліметтерді e-mail арқылы жөнелту

Кейбір жағдайда формаға енгізілген мәліметтерді электрондық адрес бойынша жөнелту қажеттігі туындайды.

Ол үшін командалық қатар интерфейсі бар поштаны жөнелту программасы қажет болады. UNIX-те бұл sendmail немесе mail болады. Осы мысалда мәліметтер sendmail программасының көмегімен жіберіледі. Файлға жазудың орнына біз арнайы ашылған каналға (pipe) жазуды қолданамыз:

```
 # e-mail адресі
 $sendto = "webmaster@akbota.kz";
 # каналды ашамыз
 open (MAIL, "| /usr/bin/sendmail $sendto")
 # Каналға арнайы форматпен жазамыз
 print MAIL "From: Web-серверn";
 print MAIL "To: $sendton";
 print MAIL "Subject: Жаңа мәліметтерді енгізу";
 print MAIL "Біреу форманы жаңа мәліметтер енгізу үшін
қолданды ";
 print MAIL "Оның енгізген мәліметі:";
 print MAIL "Аты: ", $input{'user_name'},"n";
 print MAIL "Мекеме: ", $input{'co_name'},"n";
 print MAIL "Телефон: ", $input{'phone'},"n ";
 # каналды жаба отырып, хатты жібереміз
 close (MAIL);
```

web-парақты құруды талдау

Perl скриптің соңғы маңызды бөлімі нәтижені қайтадан пайдаланушыға жіберу болып табылады. Бұл сол print арқылы, бірақ файл немесе канал идентификаторынсыз жүзеге асады. Стандартты шығуға жазылған барлық мәлімет браузер терезесіндегі ағымдағы құжатты қалыптастырады. Мысалы:

```
    print "Content-Type: text/htmlnn";
```

```
    print "<html>n<head>n<title>Рахмет</title>n</head>";
    print "<body>n<h1>Форманы толтырғаныңызға
рахмет</h1>";
    print "Біз сіздің атыңызды, жұмыс орныңызды және те-
лефоныңызды алдық,";
    print " Олар төменде көрсетілген:<br>n";
    print "Аты: ", $input{'user_name'},"n<br>";
    print "Мекеме: ", $input{'co_name'},"n<br>";
    print "Телефон: ", $in-
put{'phone'},"n</p>n<p><hr></p>n";
    print "</body>n</html>";
```

Бірінші қатарға назар аударайық. Бұл қатар қайтарылатын мәліметтердің типі жөніндегі ақпараттан тұрады. Қатарды екі рет аудару міндетті түрде болу керек. Бұл парақ Submit батырмасы басылғаннан кейін бірден пайдаланушыға қайтарылады.

Бақылау сұрақтары:

1. Perl тілін кім құрған? Ол қандай мағынаны білдіреді?
2. Perl тілі қандай мүмкіндіктер береді?
3. Perl тілін іске қосу үшін қандай компоненттер қажет?
4. Perl-дің тіркеу формасына енгізілген ақпаратты қандай тәсілдермен өңдеуге болады?
5. Perl тілін қалай іске қосуға болады?
6. Perl скрипт қандай бөлімдерден тұрады? Олардың әрқайсысына сипаттама беріңіз.

Тапсырмалар:

1. Жоғарыда келтірілген әрбір қадамды орындап, тексеріп шығыңыз.
2. Өзіңіз туралы мәліметтерді электрондық пошта арқылы серверге жөнелтіңіз.
3. Өзіңіз туралы мәліметтерді досыңыздың электрондық поштасына жөнелту программасын құрыңыз.
4. Өзіңіз туралы жазылған мәліметтер жазылған файлдың басына досыңыз туралы ақпараты қосуды ұйымдастырыңыз.
5. Файлдың соңына және ортасына досыңыз туралы мәліметтерді қосыңыз.
6. Соңғы толтырылған мәліметтерді серверге жөнелтіп, сервердің жауабын алуды ұйымдастырыңыз.

5.2 PHP ТІЛІ НЕГІЗДЕРІ

PHP – серверде орындалатын программалау тілі. JavaScript тіліне қарағанда PHP тілі қолданушының программалық жабдықтамасынан тәуелсіз, сондықтан ол әрқашан орындалады.

Программа немесе *скрипт* деп аталатын нұсқаулар тізбегі PHP тілінің интерпретаторы арқылы орындалады. Программа кодының HTML-кодқа енгізілу мүмкіндігі бар. PHP тілі Интернетте қолданылатын басқа тілдерден, мысалы Perl, осы мүмкіндік арқылы ерекшеленеді. PHP-коды серверде ол Web-браузерге берілмей жатып өңделеді. Нәтижесінде Web-браузер қарапайым HTML-код немесе басқа бір шешім қабылдайтын деңгейде болады.

PHP тілінің қысқаша даму тарихы

PHP (ол мына сөз акронимі: "PHP: Hypertext Preprocessor/Гипермәтіндік Преүрдісор ", алғашқыда - Personal Home Page сөзінен шыққан) – HTML ортасына енгізілген сценарийлерді жазу тілі.

Бұл тілдің синтаксисінің басым бөлігі C, Java және Perl тілдерінен алынған (әрине өзінің ерекшелігіне байланысты жаңа мүмкіндіктер қосылған). Тілдің мақсаты – Web-программалаушыларға жеңіл әрі жылдам өзгертуге болатын динамикалық түрдегі ықшам html-парақтар құру мүмкіндіктерін беру. PHP – HTML ортасына енгізілген, бірақ серверде орындалатын сценарийлер жазу тілі.

PHP тілінің бір ерекшелігі – мәліметтер базаларымен бірігуінің жоғары деңгейде болуы. Қазіргі кезде бұл тіл бірсыпыра мәліметтер базаларын сүйемелдей алады, олар: Oracle, Adabas D, Sybase, FilePro, mSQL, Velocis, MySQL, Informix, Solid, dBase, ODBC, Unix dbm, PostgreSQL

PHP тілін 1994 жыл аяғында Расмус Ледорф (Ras-mus Lerdorf) ұсынған болатын. Оның алғашқы нұсқалары сайтты және ондағы резюмені қанша адам көрге-нін қадағалау мақсатында жасалған болатын. Бұл тіл-дің бастапқы бір нұсқасы 1995 ж. шығып, ол Personal Home Page Tools деген атпен белгілі болады. Мұнда қонақтарға арналған мәлімет, санауыш және т.с.с. қосымшалар болған еді.

Қазіргі кезде 1996 ж. өзінде PHP тілі әлемдегі 15,000 веб-сайтта қолданылғаны белгілі. Ал 1997 ж. бұл көрсеткіш 50,000-нан асты. 1997 ж. PHP тіліне бірсыпыра жаңалықтар қосылды. Содан бастап Расмустың жеке жобасы бірсыпыра программалаушылардың қолдауымен жақсы ұйымдасқан жұмысшы топқа айналды. Бұл тілдің синтаксистік анализаторын Зев Сураски (Zeev Suraski) мен Анди Гутманс (Andi Gutmans) қайтадан жазып шықты, осы нұсқадағы анализатор PHP 3 тілінің негізі болып қалыптасты.

PHP тілі 1998 ж. ортасында дүние жүзіндегі 150,000 сайтта қолданылды. Болашақта олардың саны Интернеттегі Netscape's flagship Enterprise server ортасындағы сайттардан да асып түсті.

PHP тілі барлық ірі-ірі операциялық жүйелерге енді, ол Linux ортасында, Unix (HP-UX, Solaris и OpenBSD) нұсқаларында, Microsoft

Windows, Mac OS X, RISC OS, т.б. орталарда жұмыс істей бастады. PHP тілі көптеген web-серверлерде сүйемелденетін болды. Олар Apache, Microsoft Internet Information Server, Personal Web Server, Nets-cape, iPlanet-серверлер, т.с.с.

Осы серверлердің көпшілігінде PHP тілінің арнайы модульдері бар. PHP тілі суреттерді бейнелеп, PDF-файлдарды, тіпті Flash клиптерін де көрсете алатын болды.

Қазір PHP – жылдам дамып келе жатқан программалау ортасы, ол Интернеттегі көптеген серверлерде жұмыс істейді. PHP-де жасалған файлдар серверлердің барлығында да сақталып, өңделе береді. PHP тілінің артықшылығы оны HTML-парақтың кез келген жеріне қосып, түрлендіруге болады. Оған қоса тілдің синтаксисі, құрылуы, ережелері де онша қиын емес.

PHP тілінің ерекшеліктері

PHP – HTML-кодқа қосылатын, бірақ серверде орындалатын скриптер тілі. PHP тілі скриптерінің құжатқа енгізілуін көрсететін бір мысал қарастырайық.

1 мысал. PHP кодтарын құжатқа енгізу

```
<html>
 <head>
   <title> Listing l </title>
 </head>
<body>
 <?php echo "Hello! This is my first programm!"; ?>
</body>
</html>
```

Блокнотта осы программалық кодты теріп PHP форматында (мысалы, index.php) C:\Apache\Apache2\htdocs бумасына немесе қолданылатын сервер-ге байланысты басқа орында сақтаймыз. Web-браузерді іске қосамыз және адрес өрісінде http://localhost/ теремiз.

Нәтижесінде “Hello, world” сөз тіркесі экранға шығады. Енді жоғарыдағы бастапқы HTML-код жолдарын қарастырып көрейік.

Ашылу тәгі – <html> бұл файлда HTML-кодтары қолданылатынын көрсетеді. <title> Listing 1 </title> құжат атын береді, оны өзгерту өз қолымызда, яғни қалаған атты таңдай аламыз. Бұл тәгтер <head> ... </head> тәгтері арасында тұруы тиіс, олар тақырыпты құрайды.

<body> контейнері құжаттың негізгі бөлігін қамтиды, ол экранда көрінеді. PHP-скрипті де осы бөлімде орналасқан. Мынадай жол:

```
echo "Hello! This is my first programm!";
```

браузер терезесіне қостырнақша ішіндегі мәтінді шығарады. Скрипт экранға мынадай жол шығарады:

```
Hello! This is my first program!
(Сәлем! Бұл менің бірінші программам!)
```

Экранға шығарылатын қазақша сөздерді орыс әріптерімен немесе ағылшын тілінде теремiз, өйткені қазақ әріптері шығарылмайды.

PHP тілінің әрбір операторы нүктелі үтірмен (;) бөлініп жазылады. Егер ол жазылмаса, браузер қате кеткенін білдіріп, жол нөмірін көрсетеді.

PHP артықшылықтары.

- Perl типтес тілде жазылған PHP скриптерінің CGI скриптерден негізгі айырмашылығы мынада – CGI программасында енгізілетін HTML кодтары жазылады, ал PHP тілін пайдаланғанда – программа дайындалған HTML параққа ашылып-жабылатын тәгтердің көмегімен енгізілледі (1 мысалды қара – `<?php` және `?>`).
- PHP тілінің JavaScript тілінен айырмашылығы – PHP скрипті серверде орындалып, клиентке жұмыстың нәтижесі беріледі, ал JavaScript коды толығымен клиенттің машинасында орналасады және тек сол жерде ғана орындалады.
- Internet Information Server жүйесіне қызығушылары PHP-ді Active Server Pages (ASP) ортасымен ұқсастықтары бар деп санайды, ал Java тілін қолдаушылар PHP-ді Java Server Pages (JSP) ортасына ұқсас деп санайды. Негізінде, аталған үш тілдің барлығы да Web-сервердегі HTML парақтардың ішіне код енгізуге мүмкіндік береді.
- PHP-де көптеген мәліметтер базасымен жұмыс істеу мүмкіндігі мәліметтер базасын пайдалану арқылы жазылған Web-қосымшалардың жазылуын жеңілдетеді және ол өте қарапайым түрде болады.

PHP кемшіліктері.

- PHP тілінің бастапқы идеологиясы бойынша оның кішігірім скрип-тердің жазылуына бағытталуы оның негізгі кемшілігі болып саналады. Сондай-ақ бұл кемшілік PHP 4 және одан бұрынғы нұсқаларында қайталанды.
- PHP жартылай компиляцияланатын тіл болғандықтан, ол толық компиляцияланатын С тілімен жылдамдығы бойынша салыстырыла алмайды. Сонымен қатар көптеген кішігірім парақтардан құралған жобалар жазғанда, оларды компьютер жедел жадына жүктеу мен С тілінде жазылған CGI программасы шақыру кезінде біршама артық шығындардың пайда болуына әкеп соқтырады.

5.3 Тіл синтаксисі мен грамматикасы

1 HTML ортасынан PHP тіліне көшу

HTML тілі ортасынан шығып, PHP кодтары режиміне көшу тәсілдері:

```
1. <? echo("қарапайым әдіс");?>
2. <?php echo("құжаттармен жұмыс істеу кезінде");?>
3. <script language="php">
       echo ("FrontPage сияқты кейбір редакторлар осындай
                 нұсқаларды пайдаланады");
   </script>;
4. <% echo("ASP-тәгтері бар парақтарды құру кезінде өзара сәйкестік
```

үшін "); %>

Бұлардың тек екеуін (<?php...?> және <script language="php"> ... </script>) кез келген уақытта қолдануға болады, қалғандары сервер ерекшеліктеріне қарай кейде орындалмайды.

Көптеген программалау тілдері тәрізді PHP тілі де программаның басын және соңын белгілеп отыратын таңбаларды қолданады, яғни PHP кодының басы мен соңын белгілейді. Олардың мұнда, PHP тілінде төрт нұсқасы қолданылады:

1. Бірінші нұсқа, стандартты тәгтермен қоршау тәсілі:
   - бастапқы тәг – <?php
   - соңғы тәг – ?>

Бұл нұсқа жиі қолданылады, кез келген серверде оның параметрлерінен тәуелсіз осы нұсқа жұмыс істей береді. Біздің мысалдарда осы нұсқа қолданылады. Қалған тәсілдер үшін сервер басқарушыларының PHP-ге арналған арнайы параметрлерін енгізу қажет.

2. Екінші нұсқа, қысқартылған тәгтерді қолдану:
   - бастапқы тәг – <?
   - Соңғы тәг – ?>
3. Үшінші нұсқа тәгтері asp ортасы стилінде берілген:
   - бастапқы тәг – <%
   - соңғы тәг– %>

Бұл екеуі серверде арнайы параметрлер арқылы рұқсат беруге байланысты жұмыс істейді, егер керекті параметрлер орнатылмаса, программа іске қосылмай қалады.

1. Төртінші нұсқа:
   - бастапқы тәг – <script language="php">
   - соңғы тәг – </script>

Бұл нұсқа да әмбебап тәсілдерге жатады, кез келген серверлерде іске қосылады .

PHP файлдарының кеңейтілуі, яғни типі php немесе phtml болуы тиіс, соңғысы кейде іске қосылмауы да мүмкін. Кейбір серверлер ерекшеліктеріне байланысты басқа типтер де қолданыла береді, оны тек провайдер арқылы білуге болады. Біздің мысалдарда *.php кеңейтілуі қолданылады.

Әрбір оператор С тіліндегідей түрде (;) символымен аяқталады.

Жабылатын тәг те (?>) сол сияқты операторлардың аяқталғанын білдіреді, сондықтан келесі нұсқалар бірдей болып саналады:

```
<?php echo "Бұл мәтіндік жол";?>
<?php echo "Бұл да мәтіндік жол"?>
```

2 Түсініктемелер беру

PHP тілі C, C++ тілдеріндегі және де Unix ортасындағы комментарийлерді (түсініктемелерді) қолдана береді. PHP-де түсініктемелердің үш түрі бар:

- жол соңында екі слэштен кейін түсініктеме жазылады (// түсініктеме), мысалы:

```
<?php
  echo "Бұл мәтін"; // C++ тіліндегідей түсініктеме
 ?>
```

- хэш символы арқылы жазылатын түсініктеме (# – тор перне арқылы), мұндай түсініктемелерді циклдер мен жиымдар (массивтер) ішіне пайдалануға болады.
- бірнеше жолдан тұратын түсініктеме – слэш пен жұлдызшаның қатарласып келуімен басталып (/*), солардың кері реттілікпен орналасуы (*/) арқылы аяқталады.

```
/* Бұл көп жолдық комментарий (C тіліндегідей)
   Бұл оның тағы бір жолы */
```

Соңғы нұсқаны жиі пайдалану қажет емес, өйткені ол кейбір жағдайларда программаның орындалуы кезінде қате беруі мүмкін.

Мысалдар. Енді біз PHP тілінде алғашқы программа жазайық. Әрине, программа жазу ортасында қалыптасқан салт бойынша, ол "Сәлем, әлем!" деген мәтінді шығаратын нұсқау болып табылады. Келесі кодты теріп, оны **example1.php** файлына жазып қояйық:

```
<?php
      print "Hello, world!";
  ?>
```

PHP тілін іске қосып, браузерде осы файлды адресі бойынша шақырғанда, мысалы төмендегідей түрде:

```
http://localhost/example1.php
```

браузер экранына ағылшын тіліндегі `Сәлем, әлем` сөзі қос тырнақшасыз шығады.

Экранға мәлімет шығаратын `print` операторының басқа да бір нұсқасы бар, ол: `echo();`

Мұнда жақша қою міндетті емес, бірақ `print` және `echo` функциялар болып табылады, сондықтан анықтамалықтарда жақша қою керек деп айтылады. Сонымен, нәтижені екі оператор көмегімен шығаруға болады:

- `echo` – "Hello, world" тіркесін шығару үшін қолданғанбыз;

```
echo  "Hello, world";
```

- `print` – бұл оператор Perl тілінде қолданылады.

```
print "Hello, world";
```

Қандай оператор қолдануды өзіміз таңдаймыз.

Енді осы мысалды HTML-тәгтерін қоса отырып шығарып көрейік.

5.1- мысал. HTML-тәгтерін PHP көмегімен шығару

```
<?php
echo "<HTML>";
echo "<HEAD>";
echo "<TITLE> 4.1 мысал </TITLE>";
```

```
echo "</HEAD>";
echo "<BODY>";
echo "Hello, world!";
echo "</BODY>";
echo "</HTML>";
?>
```

Мұның да нәтижесі де **Hello, world!** сөзі болып табылады.

Ал егер осы кодтардың мәтінін экранға шығарғымыз келсе, онда “ < ” және “ > “ таңбалары орнына сәйкесінше < және >таңбаларын теруіміз керек. Сонда программа коды төмендегідей болады:

```
   echo "<HTML&gt";
   echo "<HEAD&gt";
   echo "<TITLE>4.1 мысал </TITLE&gt";
   echo "</HEAD&gt";
   echo "<BODY>";
   echo "Hello, world";
   echo "</BODY&gt";
   echo "</HTML&gt";
```

Осының нәтижесінде келесідей бастапқы кодты аламыз:

```
   <HTML><HEAD><TITLE> 4.1 мысал </TITLE></HEAD><BODY> Hel-
lo, world</BODY></HTML>
```

Көріп тұрғанымыздай, барлық код бір қатардың бойында орналасады. Әрбір тәгті бөлек қатарларда шығару үшін келесі қатарға көшу символын қосамыз (
). Unix жүйесі үшін \n символы қолданылады. Windows операциялық жүйесінде келесі қатарға көшу символы екі символ \r\n комбинациясынан тұрады.

5.2- мысал. HTML-тәгтерін жеке-жеке жолдарға шығару

```
<?php
echo "<HTML><br>";
echo "<HEAD>\r\n";
echo "<TITLE>4.1 мысал </TITLE><br>";
echo "</HEAD><br>";
echo "<BODY> \r\n";
echo "Hello, world \r\n";
echo "</BODY><br>";
echo "</HTML&gt";
?>
```

Енді әрбір тәг өз қатарында шығады.

Бұл мысалдан HTML тілі мен PHP кодтары араласып қолданыла беретіні көрініп тұр, тек қос тырнақша таңбасын қолдану тәсіліне назар аудару қажет.

3 Айнымалылар

Айнымалылар – әр түрлі мәліметтер типін уақытша сақтау үшін пайдаланылатын шамалар. Программалар жазып, айнымалыларды пайдалану кезінде қолданылатын арнайы символдарға жататындар:

$ – доллар белгісі – айнымалы аты осы таңбадан бастайды,

" – қос тырнақша – сөз тіркесі түріндегі мәліметтерді жазуға арналған,

' – апостроф белгісі – бұл да сөз тіркесі түріндегі мәліметтерді жазуға арналған, бірақ айнымалы мен константа мәндерін емес, тек солардың аттарын шығарады.

Әрбір айнымалының программада латын әріптерінен, цифрлардан және сызу белгілерінен тұратын бірегей атауы (идентификатор) болуы керек. PHP-дегі барлық айнымалылар атауы $ белгісінен басталады.

Айнымалылардың дұрыс атаулары: $x, $strName, $y1, $_name.

Сонымен, PHP тіліндегі кез келген айнымалының аты ($) доллар белгісімен басталып, одан кейін тек латын әріптерінен мен цифрлар орналасады, бірақ доллар белгісінен кейін тек қана әріп тұруы тиіс (айнымалы атында ұлттық әріптер қолданылмайды). Мысалы:

```
$alfa          // Бұл дұрыс жазылған айнымалы
$alfa_5_5      // Бұл да дұрыс жазылған
$54beta_gamma //Бұл айнымалының аты қате, цифрдан басталған
$345           // Бұл айнымалының аты да қате
$alfa-bir      // Бұл да айнымалының қате жазылған түрі
```

Сондай-ақ, бас әріп пен кіші әріп екі түрлі болып саналады, сондықтан **$Alfa_bir** мен **$alfa_bir** екеуі бірдей емес. Айнымалылар ретінде әр түрлі типтегі шамалар қолданыла береді, мысалы, сөз тіркестері, жеке символдар, сандар, жиымдар, логикалық мәндер, т.с.с.

Арифметикалық амалдар (операциялар) таңбалары:
+, -, *, /, % (модуль бойынша бөлу, бөлуден шыққан қалдықты табу),
салыстыру операторлары (= =, !=, >, <, >=, <=),
логикалық операциялар (||, &&, !),

5.3- мысал. Айнымалыларға мәндерді меншіктеу

```
<?php
$first='Text';   //'Text' мәнін $first-ке меншіктеу
$second=$first;  // $first мәнін $second-ке меншіктеу
$first='New text'; // $first мәні – 'New text'
echo "first мәні $first болады <br>";
// $first айнымалысы мәні шығарылады
echo "second мәні $second болады <br>";
// $second мәнін шығарамыз
?>
```

Нәтижесі:

```
first мәні  New text болады
 second мәні Text болады
```

4 Тұрақтылармен жұмыс істеу

Script орындау барысында тұрақты шамаларды, яғни мәні өзгермейтін шамаларды сақтау үшін **константалар** немесе **тұрақтылар** пайдаланылады. Константаларға математикалық *тұрақтылар*, құпия сөздер (парольдар), файл адрестері және т.б. жатады. *Тұрақтының айнымалыдан* негізгі айырмашылығы – оған мән бір-ақ рет беріледі және оның жарияланған мәнін өзгертілмейді. Бұған қоса тұрақтылар аты алдына доллар таңбасы жазылмайды және оларға мәнді қарапайым меншіктеу арқылы беруге болмайды. Тұрақтыларды анықтау үшін **define()** арнайы функциясы қолданылады. Оның синтаксисі мынадай:

```
define("тұрақты_аты","тұрақты_мәні",
      ["регистрден_тәуелсіздігі"])
```

Келісім бойынша тұрақтылар аттары регистрге тәуелді, бірақ әрбір тұрақты үшін қосымша аргумент ретінде **регистрден_тәуелсіздігі** мәнін **true** мәніне өзгертіп, бұл қасиетті жоюға болады. Келісім бойынша әрқашанда тұрақты аты жоғарғы регистрде (бас әріптермен) жазылады.

Тұрақтының мәні оның $ таңбасысыз жазылатын атын көрсету арқылы пайдаланылады. Бұған қоса тұрақты мәнін алу үшін аргументі тұрақты атына сәйкес **constant**() функциясын пайдалануға болады. Келесі жолға көшу
 тәгімен орындалады.

5.4 -мысал. Айнымалыларға тұрақты мәндерді меншіктеу

```
<?php // PASSWORD тұрақтысын анықтаймыз
define("PASSWORD", "qwerty");
define("PI", "3.14",True); // регистрден тәуелсіз
                    // PI константасын анықтаймыз
echo ('PASSWORD ');
// PASSWORD сөзін шығарады
echo (constant("PASSWORD")."<BR>");
// qwerty мәнін шығарады
echo (password."<BR>");
 /* password сөзі және ескерту шығады, өйткені
регистрден тәуелсіз тұрақты енгізілген болатын */
echo (pi);
// PI регистрден тәуелсіз болғандықтан 3.14 шығады
?>
```

Осының нәтижесі:

PASSWORD qwerty
Notice: Use of undefined constant password - assumed 'password' in **c:\Inetpub\wwwroot\php\primer2-4.php** on line **6**
password
3.14

PHP құрамында орнатылған тұрақтылар бар:

□ __file__ (екі астын сызу символына дейін және одан кейін) – программа атынан тұрады;

□ __ line __ (екі астын сызу символына дейін және одан кейін) – сол мезетте интерпретатор өңдеп жатқан қатар нөмірін береді;
□ php_os – операциялық жүйенің нұсқасы мен атынан тұрады;
□ php_version – PHP нұсқасынан тұрады.

Мысалы, төмендегі программаны орындасақ:

```
<?php
echo __FILE__ . "<BR>";
echo __LINE__ . "<BR>";
echo PHP_OS . "<BR>";
echo PHP_VERSION . "<BR>";
?>
```

Оның нәтижесі мынадай болады:

```
Z:\home\localhost\www\vercia.php
3
WINNT
5.3.1
```

Алғашқы жол сервердегі `localhost\www` бумасындағы `vercia.php` файлының орындалғандығын, мәлімет беріп тұрған жол нөмірі 3 екенін және операциялық жүйе мен PHP нұсқалары нөмірлерін беріп тұр.

5 Айнымалылар мен сандар типтері

PHP тілінде сегіз қарапайым айнымалылар типі бар. Солардың ішінде төрт скалярлық тип:

- *integer* – бүтін (мысалы 1000),
- *boolean* – логикалық (true, false; 1, 0),
- *float* – нақты (жылжымалы нүктелі сан, соңғы нұсқаларда
  - *double* – екі еселенген нақты сан (3,14),
- *string* – сөз тіркесі, яғни тіркестік тип ("Алма"),

екі аралас (құрамдас) тип:

- *array* – жиым,
- *object* – объектілі айнымалы(объектіге бағытталған программалауда),

және екі арнайы тип бар:

- *resource* - ресурс
- *NULL*.

Мұнда программалаушы айнымалылар типін енгізбейді. Олар пайдаланылатын мәніне қарай (контекст бойынша) орындалу сатысында анықталады.

Айнымалыны қолдану үшін меншіктеу арқылы оған кез келген бір мән беру керек, сондықтан айнымалыны алдын ала хабарлау қажет етілмейді, яғни айнымалыға мән меншіктелген сәттен бастап оның типі анықталады.

Айнымалыға мән меншіктеу келесі түрде атқарылады:

```
$variable = "мән";      $alfa=5.56;
```

немесе

```
$variable="өрнек";      $alfa=5.56*$x/1.5;
$number=7;     // integer
```

```
$number2 = 7.8;// float
$string="Жол"; // $string айнымалысына Жол мәні берілген
$string2='Жол'; // $string2 айнымалысы мәні де Жол
$boolean=true; // $boolean айнымалысы мәні – true
```

PHP кез келген уақытта өзінде сақтаулы мәліметтерге сәйкес айнымалы типін өзгерте алады.

```
$var="Жол"; // string типі
$var=7; // енді айнымалы integer типінде
```

gettype (<Айнымалы_аты>) функциясы айнымалы типін қайтарады (5.7. листинг).

5.5- мысал. Айнымалы типін шығару

```
<?php
echo "<HTML><HEAD>\n";
echo "<TITLE> Айнымалылар типі </TITLE>\n";
echo "</HEAD> <BODY>\n";
$var=7;
echo gettype($var); //  integer
echo "</BODY></HTML>\n";
?>
```

Мұның жауабы тек `integer` деген сөз болады.

Сонымен қатар, нақты айнымалылар типін тексеретін функциялар бар:

- isint (<айнымалы>) – егер айнымалы бүтін сан болса `true` мәнін қайтарады;
- is_integer (<айнымалы>) – егер айнымалы бүтін сан болса `true` мәнін қайтарады;
- isdouble (<айнымалы>) – егер айнымалы нақты сан болса true мәнін қайтарады;
- is_float (<айнымалы>) – егер айнымалы нақты сан болса true мәнін қайтарады;
- is_string(<айнымалы>) – егер айнымалы сөз тіркесі болса true мәнін қайтарады;
- is_array (<айнымалы>) – егер айнымалы жиым болса true мәнін қайтарады;
- is_object (<айнымалы >) – егер айнымалы объект болса true мәнін қайтарады;
- is_bool (<айнымалы >) – егер айнымалы мәні логикалық болса true мәнін қайтарады;
- isset (<айнымалы >) функциясы көмегімен айнымалының бар екендігін тексеруге болады. Егер айнымалы анықталса true мәнін қайтарады.

Мысал үшін біздің бастапқы программамызды бүкіл әлемге емес, тек бізге ғана сәлем беретіндей етіп өзгертейік.

5.6 мысал. Айнымалының бар екендігін тексеру

```
<HTML>
<HEAD>
```

```
<TITLE> 4.6 мысал </TITLE>
</HEAD>
<BODY>
<?php
if (isset($name)) {
   echo "Hello, $name";
}
else {
   echo "Өз атыңызды енгізіңіз: <BR>\n";
   echo "<FORM>\n";
   echo "<INPUT type=\"text\" name=\"name\">\n";
   echo "<INPUT type=\"submit\" value=\"OK\">\n";
   echo "</FORM>\n";
}
?>
</BODY>
</HTML>
```

Программаны бірінші рет іске қосқанда атын енгізу үшін шақыру пайда болады. Өзіміздің атымызды енгіземіз (мысалы, Бақыт) және ОК шертеміз. Нәтижеде “Hello,Бақыт” сәлем жолы шығады.

Айнымалыны unset () функциясы арқылы жоюға болады.

unset (<айнымалы>);

Бұл функция айнымалы үлкен көлемдегі мәліметтерді өңдеп және енді қажет болмағанда қолданылады. Айнымалыны жою компьютер жадын босатуға мүмкіндік береді.

6 Бүтін сандар типі

Бұлар бүтін сандар жиымындағы мәндерді береді, мысалы:

integer Z = {..., -2, -1, 0, 1, 2, ...}

Бүтін сандар ондық , он алтылық және сегіздік жүйелерде алдындағы міндетті түрде көрсетіле бермейтін таңбасын жазу арқылы бейнеленеді (-7; +456; 4589).

Егер сегіздік санды пайдалансақ, бірінші символ 0 (нөл) болуы керек,он алтылық сандарда алғашқы символдар 0x болуы тиіс. Мысалы:

```
$a = 1234;  # ондық сан
$a = -123;  # теріс сан
$a = 0123;  # сегіздік сан (ондық жүйеде 83)
$a = 0x1A;  # он алтылық сан (ондық жүйеде 26)
```

Мұнда **integer** типіндегі сандар шамасы компьютер платформасына тәуелді болып келеді, оның ең үлкен (максимал) мәні 2 миллиард шамасында болады (32-биттік таңбалы бүтін сан). PHP тілінде таңбасыз **integer** типі қолданылмайды.

Егер берілген сан **integer** шекарасы диапазонынан асып кететін болса, ол **float** типіне айналып кетеді. Сондай-ақ операцияны орындау барысында да оның нәтижесі **integer** арнасынан асып кетсе, **float** типі қолданылады. Санның типін анықтау үшін var_dump() функциясы пайдаланылады.

PHP тілінде бүтін санды бүтін санға бөлу кезінде нақты сан шығады, мысалы, 1/2 = 0.5 болады.

5.7- мысал. Айнымалы типтерін тексеру

```
<?php
$large_number =  2147483647;
var_dump($large_number); echo "<BR>";
// жауабы: int(2147483647)
$large_number =  2147483648;
var_dump($large_number); echo "<BR>";
// жауабы: float(2147483648)
// келесі он алтылық сандарға да сәйкес келеді:
var_dump( 0x80000000 ); echo "<BR>";
// жауабы: float(2147483648)
$million = 1000000;
$large_number =  50000 * $million;
var_dump($large_number);
// жауабы: float(50000000000)
?>
```

Нәтижесі:

```
int(2147483647)
float(2147483648)
float(2147483648)
float(50000000000)
```

Тағы бір мысал келтірейік:

```
<?php
$a = 1234;
$b = -1.2e3;
$c = 0123;  // сегіздік сан ондық жүйеде 83
$d = 0x1A;  //  он алтылық сан ондық жүйеде 26
echo "a = $a <br>";
echo "b = $b <br>";
echo "c = $c <br>";
echo "d = $d <br>";
?>
```

Нәтижесі:

```
a = 1234
b = -1200
c = 83
d = 26
```

7 Нақты сандар типі

Жылжымалы нүктелі нақты сандар ("float", "double" немесе "real") келесі синтаксистерді пайдалану арқылы бейнеленеді:

```
$a = 1.234; $b = 1.2e3; $c = 7E-10;
```

`Float` типті сандар компьютер платформасына байланысты болып келеді, бірақ оның максимал шамасы ~1.8e308, жалпы алғанда, 14 таңбалы ондық санға (яғни 64 биттік формат) сәйкес келеді.

5.8 - мысал. Сандарды экранға шығару

```
<?php
$a = 1234;
$b = -1.2e3;
$c = 0123;  // сегіздік сан – ондық жүйеде 83
$d = 0x1A;  // он алтылық сан – ондық жүйеде 26
echo "a = $a <br>";
echo "b = $b <br>";
echo "c = $c <br>";
echo "d = $d <br>";
?>
```

Нәтижесі:

a = 1234
b = -1200
c = 83
d = 26

8 Сөз тіркестері – тіркестік типтегі шамалар

Сөз тіркесі – бұл символдар тізбегі. PHP тілінде символ дегеніміз – бір байт көлеміндегі мәлімет, яғни ол әр түрлі 256 символдардан тұра алады. Бұл PHP тілінің Unicode жүйесін сүйемелдемейтінін білдіреді. PHP тілінде *сөз тіркесінің* ұзындығына шек қойылмайды, сондықтан олардың ұзындығы жайлы ойламай жұмыс істей беруге болады.

PHP тілінде *сөз тіркесі* 3 түрлі тәсілмен:

- *жалқы тырнақша* (апостроф) көмегімен;
- *қос тырнақша* көмегімен;
- *heredoc-синтаксисі* арқылы анықталады.

Сөз тіркесін анықтаудың ең қарапайым тәсілі – бұл оны ' *апостроф* немесе " *қостырнақша* ішіне алып жазу. Бұл екеуі арасындағы негізгі айырмашылық – қостырнақша ішіндегі айнымалы идентификаторының мәні шығарылады да, апостроф ішіндегі айнымалы идентификаторының аты шығарылады. Мысалы, мына екі жол нәтижесі бірдей:

```
$food = "et";
$food = 'et';
```

Ал келесі екі жол арасында айырмашылықтар бар:

```
$tagam1 = "Менің жақсы көретінім – $food";
$tagam2 = 'Менің жақсы көретінім – $food';
```

Алғашқы жол мынадай мнәтиже береді:

```
Менің жақсы көретінім – et
```

Ал келесі жол нәтижесі:

```
Менің жақсы көретінім – $food
```

PHP тілінде кері слэш таңбасынан басталатын JavaScript тіліндегідей басқару комбинациялары да қолданылады. Олар:

Символдар тіркесі	Мағынасы
\n	**Жаңа жолға көшу**
\r	**Курсорды жол басына көшіру**
\t	**Горизонталь табуляция**
\\	**Кері қиғаш сызық белгісі**
\$	**Доллар белгісі**
\"	қос тырнақша белгісі

Қостырнақша ішінде бұлар мағынасына сәйкес басқару символдары болады да, ал апостроф ішінде тек "\" және "\\" тізбегі ғана өңделеді.

Сөз тіркесін анықтаудың ең қарапайым тәсілі – бұл оны ' *жалқы тырнақша* ішіне алу. Ал сол сөз тіркесі ішінде жалқы тырнақшаны пайдалану үшін, С тіліндегі тәрізді, оның алдына кері қиғаш сызықша (слэш) таңбасын «\» қою керек, яғни оны экрандау қажет. Егер кері сызықша жалқы тырнақшаның алдында немесе сөз тіркесінің соңында тұруы керек болса, онда ол қайталанып жазылуы тиіс «\\».

Егер жалқы тырнақша ішіндегі сөз тіркесінде кез келген басқа таңбалар алдында кері слэш «\» кездессе, онда ол қарапайым символ сияқты анықталады және өзгеріссіз шығарылады. Сондықтан кері слэш таңбасын тек ол сөз тіркесінің соңында, яғни жабылатын тырнақша алдында тұрғанда ғана экрандау керек.

PHP тілінде кері слэш таңбасынан басталатын басқару комбинациялары апострофтар ішінде кездескенімен өңделмейді.

5.9- мысал. Басқару тізбектерін пайдалану

```
<?php
echo 'Жаңа жолға көшу белгісі – <br&gt<br>';
// Келесі жол: ' таңбасын шығару үшін оның алдына \ қою
керек сөзі
echo "' \' таңбасын шығару үшін оның алдына '\\' қою
керек";
// Келесі жол: C:\*.*? тіркесін өшіргіңіз келе ме?  сөзі
echo '<br> C:\*.*? тіркесін өшіргіңіз келе ме?';
// Келесі жол: \n таңбасы жаңа жолға көшірмейді сөзі
echo '<br> \n таңбасы жаңа жолға көшірмейді';
// Келесі жол: $expand және $either мәндері қойылмайды
сөзі
  echo '<br> $expand және $either '.
                'мәндері қойылмайды '
?>
```

Нәтижесі:

```
Жаңа жолға көшу белгісі. – <br>
' \' таңбасын шығару үшін оның алдына '\' қою керек
C:\*.*? тіркесін өшіргіңіз келе ме?
\n таңбасы жаңа жолға көшірмейді
```

```
$expand және $either мәндері қойылмайды
```

Heredoc-синтаксисін қолдану

Сөз тіркесін анықтаудың басқа бір тәсілі – бұл heredoc-синтаксисін пайдалану. Мұнда сөз тіркесі <<< символдарынан басталып, осыдан кейін идентификатор тұрады.

Сөз тіркесі аяқталарда да осы идентификатор қайта жазылады. Жабылатын идентификатор осы жолдағы бірінші сөз болуы қажет. Идентификатор PHP тіліндегі басқа атаулар сияқты жазылуы тиіс: тек әріптерден басталып, сандар мен әріптер (астын сызу таңбасынан да) тізбегінен тұрады.

Heredoc-мәтін тура қос тырнақша ішіндегі сөз тіркесі сияқты болады, бірақ қостырнақшаны жазу қажет емес. Сол себепті тырнақшаны экрандау да керек емес. Бірақ мұнда да жоғарыда айтылған басқару тізбектерін пайдалануға болады. Heredoc тізбегі ішінде тұрған айнымалы мәні де шығарылады.

5.10- мысал. Heredoc-синтаксисін пайдалану

```
<?php
$str = <<<BUL
PHP тілінде heredoc-синтаксисті пайдаланып
бірнеше жолға сөз тіркесін жазу тәсілі
осындай болады
BUL;
 // Мұндағы идентификатор-BUL.Ал ендігі идентификатор-OL
$name = "Бекзат";
$name2 = "Азат";
echo <<<OL
Бала аты – "$name", ал екінші бала аты – "$name2".
OL;
// Экран: Бала аты – "Бекзат", ал екінші бала аты – "Азат".
?>
```

Сөз тіркестерін түрлендіру

Егер сөз тіркесі ретінде сандық мән көрсетілсе, онда нәтиже мен оның типі келесі ережелер бойынша анықталады:

Сөз тіркесіндегі сан құрамында **.** (нүкте), **e** немесе **E** символдары болса, онда ол екі еселенген дәлдіктегі сан болып саналады. Мұндай символдар болмаса, ол бүтін сан болып табылады.

Басқаша жағдайларда сан мәні сөз тіркесінің бастапқы бөлігі бойынша анықталады. Егер сөз тіркесі сандық мәліметтен басталса, онда олар санның мәні ретінде пайдаланылады. Кері жағдайда санның мәні нөлге тең болады. «E», «e» символдары сан құрамындағы экспонентаны білдіреді, яғни сан 10-ның белгілі бір дәрежесіне көбейтілетінін көрсетеді.

5.11- мысал. Сандардың типін шығару

```
<?php
echo "Сөз тіркестерін түрлендіру <br>";
echo "---------------------------------------- <br>";
```

```
   $f = 1 + "10.5"; //$f  екі еселенген дәлдікте (11.5)
   echo "\$f = $f; типі ".gettype($f).'<br>';
   $f = 1 + "-1.3e3"; //$f екі еселенген дәлдікте (-1299)
   echo "\$f = $f; типі ".gettype($f)."<br>"; //gettype()
типті береді
   $f = 1 + "bob-1.3e3"; //$f бүтін (1)
   echo "\$f = $f; типі ".gettype($f)."<br>";
   $f = 1 + "bob3"; //$f  бүтін (1)
   echo "\$f = $f; типі ".gettype($f)."<br>";
   $f = 1 + "10 chislo"; //$f бүтін (11)
   echo "\$f = $f; типі ".gettype($f)."<br>";
   $f = "10.0 chislo " + 1; //$f екі еселенген дәлдікте (11)
   echo "\$f = $f; типі ".gettype($f)."<br>";
   $f = "10.0 chislo" + 1.0; //$f екі еселенген дәлдікте
(11)
   echo "\$f = $f; типі ".gettype($f)."<br>";
   ?>
```

Осы скрипттің нәтижесі келесідей түрде болады:

```
Сөз тіркестерін түрлендіру
-----------------------------------
$f = 11.5; типі double
$f = -1299; типі double
$f = 1; типі integer
$f = 1; типі integer
$f = 11; типі integer
$f = 11; типі double
$f = 11; типі double
```

Сөз тіркестеріне тағы бір мысал келтірейік.

5.12- мысал. Құрамында функциялар бар сөз тіркестері

```
<?php
echo "Сөз тіркестері мысалы <br>";
echo "-------------------------------- <br>";
echo "Менің атым Бақыт <br>";      // C++ стиліндегі комментарий
echo "Менің тегім Бөрібаев <br>";  /* C стиліндегі комментарий,
                              ол  бірнеше жолдан тұратын комментарий */
echo "Hello, Бақыт!<br>";
echo "Компьютерде орнатылған PHP нұсқасы:" .  phpversion();
                                    // PHP нұсқасын беру функциясы
echo("<br> Бүгін : ".date("d.m.Y H:i:s"));
$imia='Азат';
$godRogd=1990;
echo "<br>$imia  $godRogd  жылы туылған";
//Массивтермен (жиымдармен) жұмыс істеу
$man=array('Айым',1991);
echo "<br>{$man[0]} біздің топтан {$man[1]} жылы туылған <br><br>";
$str = <<<T15
```

Бұл жерде бірнеше жолдан тұратын мәтін болуы мүмкін.
Оның қалай жазылғанына назар аудару қажет.
Мұның басында 3 "<" белгісінен кейін 'T15' идентификаторы тұруы керек және мәтіннің соңынан кейінгі жолдың басында нүктелі үтірмен аяқталатын тағы да T15 орналасады.

```
T15;
echo $str;
// Ары қарай PHP  нұсқасы туралы жүйелік ақпарат шығады
echo ("PHP орнатылған нұсқасы туралы жүйелік ақпарат :");
phpinfo();
?>
```

Нәтижесі:

```
Сөз тіркестері мысалы
-------------------------
Менің атым Бақыт
Менің тегім Бөрібаев
Hello, Бақыт!
Компьютерде орнатылған PHP нұсқасы:5.3.1
Бүгін : 19.06.2011 14:47:42
Азат 1990 жылы туылған
Айым біздің топтан 1991 жылы туылған

Бұл жерде бірнеше жолдан тұратын мәтін болуы мүмкін. Оның
қалай жазылғанына назар аудару қажет. Мұның басында 3 "<"
белгісінен кейін 'T15' идентификаторы тұруы керек және
мәтіннің соңынан кейінгі жолдың басында нүктелі үтірмен
аяқталатын тағы да T15 орналасады.
PHP орнатылған нұсқасы туралы жүйелік ақпарат :
```

Мұның төменгі жағында PHP тілінің нұсқасы, қай жылы шыққаны, архитектурасы, т.б. жайлы ақпарат берілген көлемді суреттік мәлімет шығады.

5.13-мысал. Құрамында айнымалылар бар сөз тіркесі

```
<?php
echo "Сөз тіркестерімен мысал <br>";
echo "---------------------------------------- <br>";
$str = " сөз тіркесі";
echo $str, "<br>";
// Оған   . – нүкте арқылы сөз тіркесінің қосылуы
$str = " Бұл қосымша мәтінмен" . $str ;
echo $str, "<br>";
$str .= " және сөз тіркесінің соңын тасымалдау.<br>";
echo $str;
$num=9;
$str = "Сан: $num <br><br>";
echo $str;
$str = "Сөз тіркесі: ";
$str.= 'Бұл мәтін.<br>';
echo $str ;
```

```
   $first=$str[0];   //сөз тіркесінің алғашқы символын алу
"С"
   echo "Сөз тіркесінің бірінші символы: $first <br><br>";
   $str = 'Бұл да мәтін.';
   echo "Сөз тіркесі: $str";
   $last=$str[strlen($str)-1]; //соңғы символды алу"."
   echo "<br>сөз тіркесінің соңғы символы: $last";
   ?>
```

Программа нәтижесі:

```
    Сөз тіркестерімен мысал
----------------------------------------
сөз тіркесі
Бұл қосымша мәтінмен сөз тіркесі
 Бұл қосымша мәтінмен сөз тіркесі және сөз тіркесінің соңын тасымалдау.
Сан: 9

Сөз тіркесі: Бұл мәтін.
Сөз тіркесінің бірінші символы: С

 Сөз тіркесі : Бұл да мәтін.
Сөз тіркесінің соңғы символы: .
```

9 Array типі және жиымдарды пайдалану

Жиымдармен жұмыс істеуге арналған бірсыпыра мүмкіндіктер бар. Жиым индексі тік (квадрат) жақшаның ішіне жазылады. Мысалы, құрамында төмендегідей сөз тіркестері бар жиымнан мысал келтірейік: **компьютер, интернет, модем, монитор**. Индекс нөлден басталады. Мысалы:

$m[0]= “компьютер”;

$m[1]= “интернет”;

$m[2]= “модем”;

$m[3]= “монитор”;

Бұл жерде жиым элементтері саны 4-ке тең, бірақта соңғы толтырылған индекс мәні 3-ке тең болады. Ең соңғы элемент $m[4]= “ ”;

Жиымның барлық элементтерін экранға шығаратын программа құрастырайық:

$i=0; while ($i<count($m)) {echo $m[$i].”
”; $i++}

Мұндағы count($m) функциясы жиым элементтерінің санын білдіреді. PHP тілінде жиыммен жұмыс істейтін көптеген функциялар бар.

Жиымды анықтау үшін ***array*()** конструкциясы беріледі немесе оның элементтеріне тұрақты мән беру арқылы да анықтауға болады. Мысалы:

array ([key] => value, [key1] => value1, ...)

Array() конструкциясы параметр ретінде үтірмен бөлініп, қос-қостан жұпталған **кілт => мән** тізбегін қабылдайды. => символдары мәнмен оның кілті (индексі) арасында сәйкестік орнатады. Кілт, яғни жиым индексі бүтін санмен де, сөз тіркесімен де беріле береді, ал оның мәні PHP тіліндегі кез

келген типтегі шама бола алады. Жиымның *сан түрінде берілген кілті индекс* деп аталады. PHP тілінде жиымды индекстеу нөлден басталады. Жиым элементінің мәнін жиым атынан соң тік жақшаға алынған оның *кілтін* (индексін) көрсету арқылы алуға болады. Егер жиым кілті ретінде стандартты түрде жазылған бүтін сан көрсетілсе, онда ол сандық индекс болып табылады, басқаша жағдайларда – сөз тіркесі болады. Сондықтан **$a["1"]** мен **$a[1]** тіркестері және **$a["-1"]** мен **$a[-1]** тіркестері бірдей болып саналады.

1 мысал. PHP тіліндегі жиымдар

```
<?php
$books = array ("php" => "PHP users guide",
                12 => true);
echo $books["php"]; // "PHP users guide" шығады
echo $books[12];    // бір шығады
?>
```

Іші бос тік жақшаны пайдаланғанда, ең үлкен мәнді кілт соңғы қайта индекстеу кезінен бергі жиым кілттері арасынан ізделеді. *Жиымды қайта индекстеу array_values*() функциясы арқылы жүргізіледі.

5.14- мысал. Жиымды қайта индекстеу

```
<?php
$arr = array ("a","b","c"); /* мәндері "a", "b" және "c"
        болатын жиым құрамыз. Кілттері көрсетілмегендіктен олар
        сәйкесінше 0, 1, 2 болады */
print_r($arr); //жиымды шығарамыз (кілттерін де, мәндерін де)
unset($arr[0]); unset($arr[1]); unset($arr[2]);
                // оның барлық мәндерін өшіреміз
echo "<br>";
print_r($arr); // жиымды шығарамыз (кілттері мен мәндерін)
$arr[] = "aa"; /* жиымға жаңа элемент қосамыз. Оны индексі (кілті) 0
                  емес 3 болады */
print_r($arr);
echo "<br>";
$arr = array_values($arr); // жиымды қайта индекстейміз
$arr[] = "bb"; // бұл элементтің кілті 1 болады
print_r($arr);
?>
```

Осы скрипт жұмысының нәтижесі мынадай болады:

```
Array ([0] => a [1] => b[2] => c )
Array ()
Array ([3] => aa)
Array ([0] => aa[1] => bb)
```

Жиымдарға тағы бір мысал

```
<?
// шақырушының қолтаңбасы константа болсын делік
define("SIGN","Құрметпен, Есет");
```

```
// адамдар мен әрекеттер жиымын тағайындаймыз
$names = array("Азат",  "Бекзат",  "Айым");
$events=array("f" => " ашық есік күні",
                  "o" => "көрме ашылуы",
                  "p" => "бітірушілер балы ");
// шақыру мәтінін құрамыз
$str = "Құрметті, $names[0]";
$str .= "<br> Сізді". $events["f"]. " салтанатына
шақырамыз";
$str .= "<br>" . SIGN;
echo $str;    // мәтінді экранға шығару
?>
```

Осы скрипттің жұмысының нәтижесі мынадай болады:

```
Құрметті, Азат
   Сізді ашық есік күні салтанатына шақырамыз
Құрметпен, Есет
```

Жиымның нақты элементін өзгерту үшін оның кілтін көрсетіп, жаңа мән беру керек. Элементтің *кілтін* өзгертуге болмайды, тек элементті өшіріп тастауға (*кілті*/*мәні* екеуін де) немесе жиымға жаңа элемент қосуға болады.

Жиым *элементін өшіру* үшін ***unset***() функциясын пайдалану қажет.

Мысалы:

```
<?php
$books = array ("php" => "PHP users guide", 12 => true);
$books[] = "Book about Perl";
   //  кілті (индексі) 13 болатын элемент қосу
   // ол мынадай жолмен  $books[13] = "Book about Perl";  бірдей
$books["lisp"] = 123456;  /* Бұл жиымға кілті "lisp", мәні 123456 болатын
                             жаңа элемент қосады  */
unset($books[12]);    // Бұл жиымнан кілті 12-ге тең элементті өшіреді
unset ($books);       // жиымды толық өшіреді
?>
```

10 Null типі

NULL арнайы мәні *айнымалының* мәні жоқ екенін білдіреді.

Айнымалы мәні төмендегі жағдайларда *NULL*:

- оған *NULL тұрақтысы* меншіктелген болса, (`$var = NULL`);
- оған әлі ешқандай мән берілмесе;
- ол *unset()* функциясы арқылы өшірілсе, болып саналады.

NULL типінің бір-ақ мәні – регистрден тәуелсіз NULL түйінді сөзі болып табылады.

11. Тілдің resource (ресурстар) және object (объектілер) типтері

Ресурс – бұл сыртқы ресурсқа сілтемесі бар *арнайы айнымалы* (мысалы, мәліметтер базасымен байланысу). *Ресурстар* арнайы функциялар арқылы жасалады және пайдаланылады (мысалы, mysql_connect(), pdf_new() т.с.с.).

Объектілер – объектіге бағытталған программалаудан келген мәліметтер типі. Бұл принцип бойынша, класс – белгілі бір қасиеті мен онымен жұмыс

істейтін тәсілдері бар *объектілер жиыны*, ал *объект* осы сәйкестік бойынша – класс экземпляры, яғни бір данасы. Мысалы, программалаушылар – программа жазатын адамдар класы (тобы), оған қоса, олардың да басқа адамдар сияқты, аты-жөні бар. Енді бір нақты программалаушы Азат Омаровты алатын болсақ, ол программалаушылар класының *объектісі* болып табылады, басқа программалаушылар тәрізді қасиеттері – аты-жөні бар, программа жазады, т.с.с.

PHP тілінде *объект* тәсілдеріне қол жеткізу үшін -> *операторы* пайдаланылады. *Объектіні* инициалдау үшін new өрнегі қолданылады, ол с*обьект экземплярын бір айнымалыға* меншіктейді.

5.15 мысал. PHP тіліндегі объектілер

```
<?php
// адамдарын класын құрамыз
class Person
{
   // PHP  тіліне үйретететін тәсіл
  function know_php()
  {  echo "Енді мен PHP тілін білемін!"; }
}
 $bob = new Person;  // адам класындағы объект құру
 $bob -> know_php(); // оны PHP тіліне үйрету
?>
```

Нәтижесі:

```
Енді мен PHP тілін білемін!
```

Сұрақтар

1. PHP сөзінің толық мағынасы қандай?
2. PHP тілінің қысқаша даму тарихы. Оны кім және қай жылы шығарды?
3. PHP тілінің қандай артықшылықтары бар?
4. HTML тілі ортасынан шығып, PHP кодтары режиміне көшу тәсілдері.
5. Тілдің түсініктемелер беру жолдары және оларды пайдалану.
6. Айнымалыны жариялау, меншіктеу тәсілдері.
7. Логикалық тип және оны пайдалану.
8. PHP ортасындағы сандар типтері, оларды қолдану.
9. Сөз тіркестерін анықтау тәсілдері.
10. Тіркестердегі қостырнақша мен апострофтарды пайдалану өзгешеліктері.
11. PHP ортасында heredoc-синтаксисін қолдану ерекшеліктері.
12. Тұрақтылармен жұмыс істеу қалай атқарылады.
13. Айнымалылармен жұмыс істейтін функциялар.
14. Жиымдармен жұмыс істеу жолдары.
15. Тілдің resource (ресурстар) және object (объектілер) типтері

Тапсырмалар

1. Әр түрлі 5 айнымалысы бар 1-1.php файлын құрып, оларға типтері әр түрлі болып келген мән меншіктеңдер. Типті анықтайтын gettype() функциясы арқылы айнымалыларды зерттеңдер.

2. Мәндері берілген сандық типтегі 2 айнымалысы бар 1-2.php файлын құрып, олармен бірнеше арифметикалық амалдар орындаңдар да, нәтижелерін экранға шығарыңдар.
3. Сөз тіркестерінен тұратын 2 айнымалысы бар 1-3.php файлын ашып, айнымалыларға мән беріңдер, оларды конкатенация арқылы біріктіріп, нәтижесін табыңдар.
4. 10 кездейсоқ сандардан тұратын жиым құрып, оны инициалдайтын 1-4.php файлын ашыңдар. Кейбір жиым элементтерін өшіріп, жиымға жаңа элементтер енгізіңдер. Жиымның алғашқы нұсқасын және нәтижесін экранға шығарыңдар.

5.4 PHP тілі операторлары

Операторлар *айнымалылармен, константалармен және өрнектермен әр түрлі амалдар орындайды*. Өрнек деп кез келген мәні бар шамаларды атайды. *Айнымалылар* мен *константалар* – өрнектердің негізгі және қарапайым формалары. Көптеген операциялар және соларға сәйкес операторлар бар. Солардың бірсыпырасын қарастырып шығайық.

Арифметикалық операторлар

Операторлар мәліметтермен белгілі бір әрекеттер жасауға мүмкіндік береді. Мысалы, меншіктеу операторлары мәліметтерді айнымалыларда сақтау үшін қызмет етеді, математикалық операторлар арифметикалық есептеулер жасау үшін, конкатенация операторлары екі қатарды біріктіру үшін қолданылады. PHP-дегі операторларды нақтырақ қарастырайық.

Математикалық операторлар

- + – қосу. $z=$x+$y;
- - – азайту. $z=$x-$y;
- * – көбейту. $z=$x*$y;
- / – бөлу. $z=$x/$y;
- % – қалдық табу. $z=$x%$y;
- ++ – инкремент операторы. Айнымалы мәнін 1-ге үлкейтеді.

  $z++; // $z=$z+1;
- -- – декремент операторы. Айнымалы мәнін 1-ге кемітеді .

  $z--; // $z=$z-1; операторымен бірдей

Инкремент және декремент операторлары постфиксті және префиксті формада қолданылуы мүмкін

```
sz++; $z--;  // Постфиксті формасы
++$z; --$z;  // Префиксті формасы
```

Бұлардың айырмашылығы – постфиксті формада айнымалы мәні операцияға дейін қайтарылады, ал префиксті формада алдымен операция орындалады, содан кейін барып мәні қайтарылады. Мұны мысалда көрсетейік (5.16-мысал).

5.16-мысал. Постфиксті және префиксті формалардағы операторлар

```
<HTML>
<HEAD>
<TITLE> 5.16- мысал </TITLE>
</HEAD>
<BODY>
<?php
$X=5;
$Z=$X++; // $Z=5, $X=6
echo "<B>Постфиксті форма (\$Z=\$X++;):</B><BR> ";
echo "\$Z = $Z <BR>\$X = $X <BR><BR>";
$X=5;
$Z=++$X; // $Z=6, $X=6
echo "<B> Префиксті форма (\$Z=++\$X;):</B><BR> \$Z = $Z
<BR>\$X = $X";
?>
</BODY>
</HTML>
```

Соңында мынадай нәтиже аламыз:

```
Постфиксті форма   ($Z=$X++;):
$Z = 5
$X = 6
Префиксті форма ($Z=++$X;):
$Z = 6
$X = 6
```

Меншіктеу операторлары

□ = – айнымалыға мән меншіктейді. $z = 5;

□ += – айнымалы мәнін көрсетілген шамаға үлкейтеді.
$z+=5; // $z=$z+5; операторымен бірдей

□ -= – айнымалы мәнін көрсетілген шамаға кемітеді.
$z-=5; / / $z=$z-5; операторымен бірдей

□ *= – айнымалы мәнін көрсетілген шамаға көбейтеді.
$z*=5; // $z=$z*5; операторымен бірдей

□ /= – айнымалы мәнін көрсетілген шамаға бөледі.
$z/=5; // $z=$z/5; операторымен бірдей

□ %= – айнымалы мәнін көрсетілген шамаға бөледі және қалдығын шығарады.
$z%=5; // $z=$z%5; операторымен бірдей

Екілік операторлар

□ ~ – екілік инверсия. $z=~$x;

□ & – екілік ЖӘНЕ. $z = $x & $y;

□ | – екілік НЕМЕСЕ. $z = $x | $y;

□ ^ – екілік арифметикалық НЕМЕСЕ. $z = $x ^ $y;

□ << – солға жылжыту – кіші разрядтарды нөлмен толтыратын бір немесе одан көп разрядтарды солға жылжыту;

□ >> –оңға жылжыту – үлкен разрядтарды өзінің құрамындағылармен толтыратын бір немесе одан көп разрядтарды оңға жылжыту.

$z = $x >> $y;

Шартты if операторы

Шартты if операторының құрылымын келесідей түрде көрсетуге болады:

if (өрнек) орындау_блогы;

Мұндағы өрнек кез келген PHP өрнегі (яғни, мәні барлардың барлығы) болып табылады. Скрипті өңдеу барысында өрнек логикалық типке түрлендіріледі. Егер түрлендіру нәтижесінде өрнек мәні ақиқат (true) болса, онда *орындау блогы* атқарылады. Кері жағдайда орындау блогы атқарылмайды. Егер орындау блогы бірнеше командалардан тұрса, онда ол жүйелі жақшаның {} ішіне алынуы тиіс.

5.17 - мысал. if шартты операторы

```
<?$names = array("Азат","Бекзат","Айым");
if ($names[0]=="Азат")  {
echo "Сәлем, Азат!";
$num = 1;
$account = 2000;}
if ($num) echo "<br> Азат тізімде бірінші!";
$bax = 30;
if ($account > 100*$bax+3)
echo "Бұл жол тізімде болмайды, өйткені шарт
орындалмады";
?>
```

Осы скрипт жұмысының нәтижесі:

```
Сәлем, Азат!
Азат тізімде бірінші!
```

Кеңейтілген, яғни *толық if операторы* құрылымы *else* операторы көмегімен келесі түрде жазылады:

```
if (өрнек) 1_орындау_блогы;
   else 2_орындау_блогы;
```

Егер шарт орындалса (яғни **өрнек=true**), онда **1_орындау_блогы**, кері жағдайда – **2_орындау_блогы** атқарылады. Мұндағы *else* операторының міндетті түрде болуы қажет емес екендігі түсінікті шығар.

Жоғарыда көрсетілген мысалды өзгертіп, шарт орындалмаған жағдайда да әрекеттердің орындалуын қарастырайық.

5.18- мысал. if .. else шартты операторы

```
<?php
$names = array("Азат","Бекзат","Айым");
if ($names[0]=="Азат") {
    echo "Сәлем, Азат!<br>";
```

```
        $num = 1;
        $account = 2000;}
      else {
        echo "Сәлем, $names[0] . Біз Азатты күтіп едік :( "; }
    if ($num) echo "Азат тізімде бірінші!<br>";
     else echo " Азат тізімде бірінші емес?!";
    $bax = 30;
    if ($account > 100*$bax+3)
        echo "Бұл жол тізімде болмайды, өйткені шарт
орындалмады ";
     else echo "Бірақ мына жол шығады!";
    ?>
```

Осы скрипт жұмысының нәтижесі мынадай болады:

```
Сәлем, Азат!
Азат тізімде бірінші!
Бірақ мына жол шығады!
```

Elseif операторы

Шартты операторлардың тағы да бір кеңейтілген түрі – *elseif* операторын пайдалану. *elseif* – бұл *else* және *if* сөздерінің бірігуінен тұрады. Бұл да *else* сияқты *if* шарты орындалмаған жағдайда, бірнеше әрекеттерді тармақты түрде орындау кезінде қолданылады. Мұның *else* операторынан айырмашылығы – *elseif* шарты дұрыс болғанда ғана альтернативтік әрекеттер орындалады. *else* және *elseif* операторлары арқылы кеңейтілген *if* операторы құрылымын былай етіп көрсетуге болады:

```
if (1-өрнек) 1-орындау_блогы;
    elseif (2-өрнек) 2-орындау_блогы;
    ... else N-орындау_блогы;
```

Бір *if* блогында бірнеше *elseif* операторлары бола береді. Егер алғашқы *if* шарты мен алдыңғы *elseif* шарттары *False* болып, қарастырылып отырған *elseif* шарты *True* болса, онда осы *elseif* операторы орындалады. Мысалы:

5.19- мысал. elseif шартты операторы

```
<?
$names = array("Азат","Бекзат","Айым");
if ($names[0]=="Азат") {
  // егер жиымдағы бірінші ат Азат болса
  echo "Сәлем, Азат!";}
 elseif ($names[0] == "Бекзат"){
  // егер жиымдағы бірінші ат Бекзат болса
  echo "Сәлем, Бекзат!";}
 elseif ($names[0] == "Айым"){
  // егер жиымдағы бірінші ат Азат, Бекзат емес, Айым
болса
  echo "Сәлем, Айым!";}
 else {
```

```
    // егер жиымдағы бірінші ат Азат,Бекзат,Айым болмаса
  echo "Сәлем, $names[0]. Ал сен кімсің?";}
  ?>
```

Осы скрипттің жұмысының нәтижесі мынадай болады::

```
Сәлем, Азат!
```

Switch операторы

switch – бұл шарттарды тексеру мен соған байланысты әр түрлі әрекеттерді тармақты түрде орындаудың тағы да бір тәсілі. Бұл оператор аты қазақ тіліне «ауыстырғыш» (переключатель) деп аударылады. Оның мағынасы да осыған сәйкес келеді. Айнымалының қандай мәнге ие болғанына байланысты әр түрлі әрекеттер блогы орындалады. Бұл операторлардың атқаратын жұмысы *if...elseif...else* операторлары тізбегіне немесе *if* операторлары тобына ұқсас болып келеді. *switch* құрылымын мынадай түрде жазуға болады:

```
switch (өрнек немесе айнымалы){
  case 1-мән: 1-орындау_блогы break;
  case 2-мән: 2-орындау_блогы break;
  ...
  default: келісім_бойынша_орындау_блогы
}
```

if операторынан айырмашылығы – бұл жерде өрнектің мәні логикалық типке келтірілмейді, ол тек case түйінді сөзінен (мәні1, мәні2 және т.б.) кейінгі мәндермен салыстырылады. Егер өрнектің мәні қандай да бір нұсқамен сәйкес келсе, онда *сол жолдың орындау блогы* атқарылады да, қос нүктеден кейінгі әрекеттер жолдың соңына дейін немесе *break* операторына (ол бар болса) дейін орындалады. Егер өрнектің мәні бірде-бір нұсқамен сәйкес келмесе, онда *келісім бойынша орындау блогы*, яғни default түйінді сөзінен кейінгі әрекеттер атқарылады. *switch* операторында өрнек бір рет қана есептеледі, ал *elseif* операторында ол әр тексеру сайын есептелінетін еді, сондықтан өрнек күрделі болғанда, *switch* операторы жылдам жұмыс істейді.

Алдыңғы мысалды осы *switch* операторын қолданып қайтадан шығарайық.

5.20- мысал. switch шартты операторы

```
<?
$names = array("Азат","Бекзат","Айым");
switch ($names[0]){
  case "Азат": echo "Сәлем, Азат!"; break;
  case "Бекзат": echo "Сәлем, Бекзат!"; break;
  case "Айым": echo "Сәлем, Айым!"; break;
  default: echo "Сәлем, $names[0].Ал сіздің атыңыз
кім?";
}
?>
```

Осы скрипттің де жұмысы нәтижесі мынадай болады:

Сәлем, Азат!

While операторы

While операторының құрылымы:

while (өрнек) { орындау_блогы }

немесе

while (шарт): орындау_блогы endwhile;

түрінде болады. Бұл – қарапайым цикл. Ол өрнек мәні *false* болғанша, орындау_блогын қайталап отырады (if операторындағы сияқты өрнек логикалық типке келтіріледі). Мұнда өрнек мәні цикл ішінде өзгертіледі де, ол цикл басында қайта есептелініп отырады. Енді осы оператор көмегімен 10-ға дейінгі жұп сандарды анықтау алгоритмін келтірейік.

5.21-мысал. while цикл операторы

```
<?
//бұл программа жұп цифрларды анықтайды
$i = 1;
while ($i <= 10) {
 if ($i % 2 == 0) print $i." "; //жұп цифрды шығару
  $i++;  // $i-ді бірге арттыру
}
?>
```

Нәтижесі: **2 4 6 8 10**

Do .. while операторы

Do while операторы while операторына ұқсас, оның бір ғана өзгешелігі – өрнектің ақиқаттылығын циклдің басында емес аяғында тексереді. Осының арқасында орындау_блогы кем дегенде бір рет орындалады. Операторлың жазылуы:

do {орындау_блогы} while (өрнек);

Енді бір мысал келтірейік.

5.22 мысал. Do .. while цикл операторы

```
<?
  /* цикл шарты орындалмағанына қарамастан, бұл
программа
      12 санын шығарады */
$i = 12;
do {
  if ($i % 2 == 0) print $i; //егер сан жұп болса,шығару
  $i++; // санды бірге арттыру
} while ($i<10)
?>
```

Нәтижесі: **12**

For операторы

For операторы PHP тіліндегі күрделі циклдер тобына жатады. Ол C тіліндегі циклге ұқсас болып келеді. Жазылуы:

for (1-өрнек; 2-өрнек; 3-өрнек) { орындау_блогы }

немесе

for (1-өрнек; 2-өрнек; 3-өрнек): орындау_блогы endfor;

Мысалы, 10-ға дейінгі барлық жұп сандарды экранға шығару үшін *for* циклі келесідей түрде жазылады:

```
<?php
for ($i=0; $i<10; $i++) {
    if ($i % 2 = = 0) print $i." ";
    // жұп сандарды шығару
}
```

Егер оператор жақшасындағы екінші өрнекті (**$i<10** шарты) алып тастасақ, онда циклден шығу үшін *break* операторын қолдану керек болады:

```
<?php
for ($i=0; ; $i++){
    if ($i>=10) break; /* егер $i 10-нан артық немесе тең
                          болса, циклді аяқтаймыз */
    if ($i % 2 = = 0) print $i. " ";
     // егер сан жұп болса, оны шығару }
?>
```

Жақша ішіндегі барлық үш өрнекті де алып тастауға да болады. Мұндай жағдайда $i санауышының алғашқы мәні де берілмейді және ол циклдің аяғында да өзгертілмейді. Сондықтан осы әрекеттерді жеке команда түрінде – орындау блогында немесе цикл алдында жазу қажет болады:

```
<?php
$i=0; // санауышқа бастапқы мән береміз
for ( ; ; ) {
  if ($i>=10) break;
  // егер $i 10-нан артық немесе тең болса,циклді аяқтау
  if ($i % 2 == 0) print $i;
  // егер сан жұп болса, оны шығару
  $i++; // санауышты бірге арттыру
}
?>
```

for жақшасы ішіндегі үшінші өрнекте бірнеше қарапайым командаларды үтір арқылы жазуға да рұқсат етілген. Мысалы, егер біз барлық сандарды экранға шығарғымыз келсе, онда программаны қарапайым нұсқада жазуға болады:

```
<?php
for ($i=0; $i<10; print $i, $i++);
/* Егер орындау_блогында команда болмаса немесе тек бір ко-
манда ғана болса, жүйелі жақшаларды қоймаса да болады */
?>
```

Foreach операторы

foreach операторы тағы да бір керекті цикл конструкциясы болып саналады. Ол тек PHP4 тілінен бастап жиымдармен жұмыс істеу мақсатында енгізілді. Жазылуы:

foreach ($array as $value) {орындау_блогы}

немесе

foreach ($array as $key => $value) {орындау_блогы}

Бірінші нұсқада цикл **$array** айнымалысы арқылы берілген жиымның барлық элементтерін қамтиды. Циклдің әр қадамында жиымның ағымдағы элементінің мәні $value айнымалысына жазылады да, ішкі санауыш бірге өседі (келесі қадамда жиымның келесі элементі өңделеді). Орындау_блогы ішінде жиым элементінің ағымдағы мәнін **$value** айнымалысы көмегімен алуға болады. Цикл **$array** жиымында қанша элемент болса, сонша рет орындалады.

Жазудың екінші нұсқасында жоғарыда айтылғандарға қосымша циклдің әрбір қадамында жиымның ағымдағы элементінің кілті (индексі) **$key** айнымалысының мәні ретінде жазылып отырылады, оны орындау_блогында пайдалануға болады.

foreach орындалуы басталғанда, жиымның ішкі көрсеткіші автоматты түрде оның бірінші элементіне орналасады.

5.23 -мысал. foreach цикл операторы

```
<?php
$names = array("Азат","Бекзат","Айым");
foreach ($names as $val) {
    echo "Сәлем, $val <br>";
    // бәріне сәлем жолдайды
}
foreach ($names as $k => $val) {
    // сәлемдесіп, тізімдегі нөмірлерін (кілт-
терін)шығару
    echo "Сәлем, $val! Сенің тізімдегі нөмірің $k <br>";
}
?>
```

Нәтижесі:

Сәлем, Азат

Сәлем, Бекзат

Сәлем, Айым

Сәлем, Азат! Сенің тізімдегі нөмірің 0

Сәлем, Бекзат! Сенің тізімдегі нөмірің 1

Сәлем, Айым! Сенің тізімдегі нөмірің 2

Break операторы

break операторы бірден атқарылып жатқан *for*, *foreach*, *while*, *do..while* сияқты циклдердің немесе *switch* операторының орындалуын аяқтайды.

break операторының сандық аргументі де болуы мүмкін, ол пайдаланылғанда, сыртқы неше басқару операторларының жұмысы аяқталатыны көрсетіледі.

5.24- мысал. break операторы

```
<?php
$i=1;
while ($i) {
    $n = rand(1,10);
    // 1 ден 10-ға дейінгі кездейсоқ сандар аламыз
    echo "$i:$n ";
    // қадам нөмірі мен кездейсоқ санды  шығарамыз
    if ($n= =5) break;
    /* Егер 5 саны шыққан болса, циклді доғарамыз.
       Мұнда бұл жолдан төмен орналасқан операторлар
       орындалмай қалады */
    echo "Цикл жұмыс істеп тұр <br>";
    $i++;
}
echo "<br> Цикл қадамдары саны $i ";
?>
```

Осы скриптің жұмысы нәтижесі мынадай болады:

```
1:8 Цикл жұмыс істеп тұр
2:9 Цикл жұмыс істеп тұр
3:5
Цикл қадамдары саны 3
```

Егер *break* операторынан кейін сан көрсетілсе, онда осы санға сәйкес сыртқы цикл операторларының жұмысы аяқталады. Жоғарыда келтірілген мысалда бұл мүмкіндікті пайдалана алмаймыз, өйткені онда сыртқы циклдер жоқ. Біз ол скрипті аздап өзгертейік.

5.25- мысал. break операторы

```
<?php
$i=1;
while ($i) {
  $n = rand(1,10); //1 ... 10 арасындағы кездейсоқ сандар
алу
  switch ($n) {
    case 5: echo "<font color=blue> switch-тен шығу
                (n=$n)</font>";  break 1;
   // switch жұмысын аяқтау
    case 10: echo "<font color=red> switch пен
                  while-дан шығу (n=$n)</font>"; break 2;
   // switch пен while жұмыстарын аяқтау
    default: echo "switch істеп тұр (n=$n), ";
}
echo " while істеп тұр – $i қадам <br>";
```

```
$i++;
}
echo "<br> Цикл қадамдары саны $i ";
?>
```
Нәтижесі:

switch істеп тұр (n=2), while істеп тұр – 1 қадам
switch істеп тұр (n=6), while істеп тұр – 2 қадам
switch істеп тұр (n=4), while істеп тұр – 3 қадам
switch істеп тұр (n=4), while істеп тұр – 4 қадам
switch істеп тұр (n=6), while істеп тұр – 5 қадам
switch пен while-дан шығу (n=10)
Цикл қадамдары саны 6

Continue операторы

Кейде цикл жұмысын толығымен аяқтамай, тек оның жаңа қадамын (итерациясын) қайта бастау керек болып жатады. *Continue* операторы кез келген циклдің орындау_блогындағы өзінен кейінгі нұсқауларды аттап өтіп, жаңа цикл қадамын орындауды жалғастырады. *continue* операторын да сандық аргументпен пайдаланып, алдыңғы басқарушы конструкциялардың қаншасы жұмысын аяқтау керек екендігін көрсетуге болады.

Жоғарыда келтірілген мысалдағы *break* сөзін *continue* сөзіне ауыстыралық. Осымен қатар цикл қадамын төртке тең етіп алайық.

5.26- мысал. continue операторы

```
<?php
$i=1;
while ($i<4) {
   $n = rand(1,10);
        // 1 мен 10 аралығындағы кездейсоқ сан алу
   echo "$i:$n ";
        // итерация нөмірі мен кездейсоқ санды шығару
   if ($n==5) {
        echo "Жаңа итерация ";
        continue;
       /* Егер 5 саны шықса,жаңа итерация бастаймыз, цикл
          қадамы $i арттырылмайды */
    }
   echo "Цикл істеп тұр <br>";
   $i++;
}
echo "<br> Цикл итерациялары саны $i ";
?>
```

Осы скрипттің жұмысы нәтижесі мынадай болады:

```
1:2 Цикл істеп тұр
2:4 Цикл істеп тұр
3:5 Жаңа итерация 3:2 Цикл істеп тұр
Цикл итерациялары саны 4
```

continue операторының орындалуынан кейін циклдің жұмысы аяқталмайды. Мысалда циклдің санауышы 5 болғанда, ол өзгертілмейді, өйткені ол *continue* операторынан кейін тұр. Сондықтан continue операторының орнына өрнектің ақиқаттылығын тексеруді жазуға болады:

```
<?php
$i=1;
while ($i<4) {
    $n = rand(1,10);
       // 1 мен 10 аралығындағы кездейсоқ сан алу
    if ($n!==5) {
        echo "$i:$n <br>";
       // итерация нөмірі мен кездейсоқ санды шығару
    $i++;
    }
}
?>
```

PHP тілінде *continue* операторын пайдаланудың бір ерекшелігі – ол switch конструкциясында *break* сияқты жұмыс істейді. Егер *switch* цикл ішінде болса және жаңа қадам (итерация) бастау қажет болса, онда *continue* 2 деп жазу қажет.

Сұрақтар:

1. PHP тіліндегі математикалық операторлар.
2. Инкремент және декремент операторларының постфиксті және префиксті түрлері.
3. Меншіктеу операторларының қандай түрлері бар?
4. Екілік операторлар не істейді?
5. Шартты if операторының түрлері.
6. *Elseif* операторын пайдалану.
7. Switch операторы
8. While операторының құрылымы.
9. *Do while* операторы жұмысы.
10. *For* операторының жазылу түрлері және оларды пайдалану.
11. *Foreach* операторы жұмысы ерекшеліктері.
12. *Break, continue* операторларын пайдалану мысалдары.

Тапсырмалар:

1. Шартты операторды пайдаланып, $age айнымалысы мәні 0-30 арасында болса, «Арманың алдында!», одан үлкен болса, «Есейіп қалыпсың!» деген сөз шығаратын программа жазыңдар.
2. Алдыңғы шартты көбейтіп, егер айнымалы мәні
   а) 0-13 арасында болса, «Бала»,
   ә) 14-37 арасында болса, «Жігіт»,
   б) 38-49 арасында болса, «Жігіт ағасы»,
   в) 50-65 арасында болса, «Егде адам»,
   ә) 66-100 арасында болса, «Қарт адам» деген мәлімет шығаратын программа құрыңдар.

3. Циклді пайдаланып, 1 мен 100 арасындағы жұп сандардан тұратын a_{25} жиымын құрыңдар. Экранға сол жиым элементтерінің 4-ке бөлінбейтіндерін шығарыңдар.
4. Индекс кілттері: Name, Address, Phone, Mail сөздеріне сәйкес жиым құрып, оларды толтырыңдар. Foreach циклі арқылы сол жиым элементтерін «элемент: мәні» түрінде экранға форматталған күйде шығарыңдар.

ҚОЛДАНЫЛҒАН ӘДЕБИЕТТЕР

1. *Дузбаева Р.М.* Основы создания HTML-документов. Уч. пособ. для студентов. –Алматы, КБТУ 2003. –82 с.
2. *Нидерст Дж*. Web-мастеринг для профессионалов. –СПб.:Питер, 2001. –576 с.
3. Информатика и образование, №8, 2000. *Е. В. Давыдова*
4. *Симонович С., Евсеев Г., Алексеев А*. Специальная информатика: Уч. пособ. - М.: АСТ-ПРЕСС: Инфорком-Пресс, 1998.-480с.
5. *А. Гончаров*. HTML в примерах. – СПб: "Питер", 1997.
6. *Лэмонт Вуд.* Web-графика. Справочник. – СПб: "Питер", 1998.
7. *С. Бейн, Д. Грей*. Как сделать красиво в Интернете. Перевод с англ. – СПб: "Символ-Плюс", 1998.
8. *А.О. Коцюбинский, С.В. Грошев.* Современный самоучитель работы в сети Интернет. М.: "Триумф", 1997.
9. Internet. Шаг за шагом. (на CD-ROM). – СПб: "Питер Мультимедиа", 1997.
10. Энциклопедия пользователя Internet. (на CD-ROM). – М.: "Демос", 2000.
11. *Шапошников И.В.* Самоучитель HTML 4. –СПб.: БХВ-Петербург, 2001. –288 с.
12. *Пайк М.* Internet в подлиннике. СПб.: BHV-Санкт-Петербург, 1996.
13. *Савельева Н.В.* Основы программирования на PHP: курс лекций для ст-тов. -М.: Интернет-университет информационных технологий, 2005. -264 с.
14. *Мазуркевич А.* PHP: настольная книга программиста.–М.: Новое знание, 2004. -479 с.
15. *Дуванов А.А.* Web-констуирование DHTML. –СПб.: БХВ-Петербург, 2003. –512 с.
16. *Джамса Крис.* Эффективный самоучитель по креативному Web-дизайну. HTML, XHTML, CSS, Javascript, PHP, ASP, ActiveX. Пер. с англ./Крис Джамса, Конрад Кинг, Энди Андерсен. –М.: ООО «ДиаСофтЮП», 2005. – 672 с.
17. *Джон Коггзолл.* PHP5. Полное руководство. -М.: Диалектика. 2006 г.
18. *Люк Веллинг, Лаура Томсон.* Разработка Web-приложений с помощью PHP и MySQL. -М.: Вильямс. 2007.
19. AJAX и PHP. Разработка динамических веб-приложений // Кристиан Дари, Богдан Бринзаре, Филип Черчез-Тоза, Михай Бусика. -М.: Символ-Плюс. 2006.
20. *Елена Бенкен.* PHP, MySQL, XML. Программирование для Интернета. - СПб.: БХВ-Петербург. 2007.
21. *Джордж Шлосснейгл.* Профессиональное программирование на PHP. - М.: Вильямс. 2006.
22. PHP тіліне байланысты материалдары бар сайттар:
    http://www.php.net/manual/ru/
    http://www.phpclub.ru/
    http://www.phpworld.ru/
    http://php.su/
    http://www.phpforum.ru/

1-қосымша

HTML тілінің түсініктеме сөздігі

Web-құжат (**HTML-құжат**) – гипермәтінді белгілеу тілінде **<HTML>** және **</HTML>** тәгтері арасында орналасады**.**

Тәг (tag) – HTML тілінің кодтары, яғни мәтіндерді форматтайтын командалары. Кодтар – бұрыштық жақшаларға < > алынып жазылған ағылшын тілінің толық немесе қысқартылған сөз тіркестері.

Web-сайт, Web-сервер – бір-бірімен логикалық түрде байланысып, HTML тілінде жазылған құжаттар тізбегі.

Құжатты белгілеуге арналған HTML тәгтері қызметтері

Негізгі тәгтер

<HEAD> ... **</HEAD>** – бұлардың ортасында құжат туралы мәлімет орналасады және **TITLE** тәгі тұруы мүмкін. **<TITLE>** ... **</TITLE> –** бұлар броузер-программа маңдайшасында жазылатын терезе тақырыбын қоршап тұрады.

<BODY> ... **</BODY>** – бұл екеуінің аралығында құжат "тұлғасы" (мәтін, графика, т.с.с.) жайғасады.

<BODY> тәгінің параметрлері (атрибуттары):

BGCOLOR – фон түсі (<BODY BGCOLOR ="#FFFFFF">);

BACKGROUND – фон орнына "тұсқағаз" (обои) түрінде сурет беру (<BODY *BACKGROUND* ="BERKUT.JPG">);

TEXT – мәтін түсі (<BODY TEXT ="#00FF00">);

LINK – гипермәтіндік сілтеме (байланыс) түсі;

VLINK – алдында қаралған сілтеме түсі;

ALINK – осы сәттегі екпінді сілтеме (курсор меңзеп тұрған) түсі.

Мәтіндерді форматтау тәгтері

<P> ... **</P>** – азат жолты белгілейтін тәгтер. Жаңа азат жол алдыңғысынан бос жолмен бөлініп тұрады.

**
 –** мәтінді келесі жолға көшіретін жалқы тәг. Графиканы, суретті мәтіннен бөліп тұру үшін де қолданылуы мүмкін.

<HR> – мәтіндерді бір-бірінен логикалық түрде бөліп тұратын көлденең сызық жүргізетін жалқы тәг.

<PRE> ... **</PRE>** – бұлардың ортасында алдын ала форматталған мәтін орналасады. Ондай мәтін экранға "курьер" (Courier New) қарпімен шығарылады.

Мәтін логикалық бөліктерін (тақырыптарын) ерекшелеу тәгтері

<H1> ... **</H1>** – бірінші деңгей тақырыбы (ең ірісі).

<H2> ... **</H2>** – екінші деңгей тақырыбы.

.....

<H6> ... **</H6>** – алтыншы деңгей тақырыбы (ең кішісі).

Туралау параметрлері

<P> және **<H*>** тәгтерінде қолданылады, олардың мәндері:

ALIGN=LEFT – сол жақ шетке жақындатып туралау;
ALIGN=RIGHT – оң жақ шетке жақындатып туралау;
ALIGN=CENTER – ортаға жылжытып туралау.

Туралау тәгтері

<LEFT> ... </LEFT> – сол жақ шетке туралау;
<RIGHT> ... </RIGHT> – оң жақ шетке туралау;
<CENTER> ... </CENTER> – ортаға жылжытып туралау.

Мәтін бөліктерін ерекшелеу тәгтері

<B> ... </B> – мәтінді (сөздерді, әріптерді) қалыңдатылған қаріппен **ерекшелейді.**
<I> ... </I> – мәтінді қисайтылған курсив қаріппен (Italic) ерекшелейді.
<U> ... </U> – аралығындағы мәтіннің астын сызады.
<BLINK> ... </BLINK> – аралығындағы мәтін жыпылықтап назар аудартып тұрады.
<FONT SIZE=+n> ... </FONT>
<FONT SIZE=–n> ... </FONT> – қаріп көлемін өзгертетін тәгтер.
<FONT COLOR="#OOOOFF> ... </FONT> қаріп түсін өзгертетін тәгтер.

Сырғымалы жолдар тәгтері

<**MARQUEE**>...мәтін...</**MARQUEE**> – сырғымалы жол жасайды. Оның параметрлері:
DIRECTION=left – мәтін оңнан солға қарай сырғи отырып жылжиды;
DIRECTION=right – мәтін солдан оңға қарай сырғиды;
BEHAVIOR= scrol – мәтін стандартты түрде оңнан солға қарай сырғиды;
BEHAVIOR=slide – мәтін стандартты түрде оңнан солға қарай бір-ақ рет сырғиды да тоқтайды;
BEHAVIOR= alternate – мәтін оңнан солға қарай, содан соң кері қарай, т.с.с. шексіз сырғи береді;
LOOP=n – сырғу саны LOOP атрибутының мәні *n*-мен шектеледі;
WIDHT=n – сырғымалы аймақтың ені *n*-мен шектеледі, ол пиксельмен (*n*) немесе терезе енінің пайызымен де (*n%*) көрсетіле береді;
SCROLLAMOUNT=n – жылжудың бір қадамында мәтіннің қанша пиксельге жылжитынын анықтайтын бүтін сан;
SCROLLDELAY=n – екі сырғудың арасындағы интервалды миллисекундпен көрсететін бүтін сан;
BGCOLOR=”#FFFFFF” – сырғымалы жол аумағын басқа түске бояу, түс ағылшынша атымен немесе он алтылық кодымен көрсетіледі;
HEIGHT=n – сырғымалы жол биіктігін пиксельмен (*n*) көрсету;
<FONT SIZE=n><MARQUEE>...мәтін...*</MARQUEE></FONT>* – сырғымалы жол мәтінінің қаріп мөлшерін, типін, түсін әдеттегідей өзгертуге болады.

Тізім жасайтын тәгтер

<OL> ... </OL> – нөмірленген тізімнің басын, аяғын белгілейді;
<UL> ... </UL> – белгіленген тізімнің басын, аяғын белгілейді;
<LI> – тізім элементтерін бастайды;

<DL>...</DL> түсініктемелердің **(**глоссарий) басын, аяғын белгілейді;
<DT> – сол жақ шетке жақын орналасқан глоссарий терминін сипаттайды;
<**DD**> – сол жақ шеттен оңға ығысып орналасқан глоссарий терминін сипаттайды.

Мәтіндік емес мәліметтерді орналастыру тәгтері

<IMG SRC ="file.gif '> немесе **<IMG SRC ="file.jpg">** – графикалық бейнелерді енгізу командасы.
<IMG SRC ="file.wav"> – дыбыстық бөлікті енгізу командасы.
<IMG SRC ="file.avi"> – бейнелік бөлікті енгізу командасы.

Графикалық бейнелерді енгізу тәгтерінің параметрлері

WIDHT=n – суреттің пиксельмен берілген ені.
HEIGHT=n – суреттің пиксельмен берілген биіктігі.
ALIGN – суретті туралау (ALIGN=LEFT – сол жақ шетке туралау, ALIGN=RIGHT – оң жақ шетке туралау, ALIGN=TOP – жоғарғы жаққа мәтін жолын туралау, ALIGN=BOTTOM – төменгі жаққа мәтін жолын туралау, ALIGN=MIDDLE не CENTER – мәтін жолын ортаға туралау).
HSPACE=n – графикалық бейнеден көлденең (горизонталь) бағытта қалдырылатын бос кеңістік (пиксель)
VSPACE=n – графикалық бейнеден тік (вертикаль) бағытта қалдырылатын бос кеңістік (пиксель)
ALT=”мәтін” – сурет шығарылмағанда көрсетілетін балама мәтін.

Басқа HTML-құжаттармен және Интернет мәліметтерімен гипербайланыс (гиперсілтеме) жасау командалары

<A HREF="fail.htm"> ... **</A>** немесе
<A HREF="http://www. kbtu.edu.kz"> ... **</A> –** гиперсілтеме жасау.
<ADDRESS><A HREF=mailto: bb@kbtu.edu.kz> bb@kbtu.edu.kz
</ADDRESS> – электрондық пошта адресімен гипербайланыс жасау

Кестелер

<TABLE> ... **</TABLE> –** HTML-құжатқа кесте енгізу тәгі, оның параметрлері:
BGCOLOR=”GREEN” – фон түсі, ағылшынша сөзбен немесе он алтылық кодпен беріледі;
BORDER=n – жақтау (бордюр) сызығы ені, пиксельмен беріледі;
WIDHT=n – кесте ені, пиксельмен немесе пайызбен беріледі.

Кесте жасау тәгтері

<CAPTION> ... **</CAPTION>** – кесте тақырыбы, оның параметрі VALIGN=TOP – кесте алдындағы тақырып және VALIGN=BOTTOM – кесте соңындағы тақырып.
<TR> ... **</TR>** – кесте жолы, яғни бір қатары. Мынадай параметрлері болуы мүмкін: BGCOLOR – жолдың ішкі фоны түсі; ALIGN=LEFT, RIGHT, CENTER – жол мәтіндерін көлденең туралау; VALIGN=TOP, BOTTOM, MIDDLE – жол мәтіндерін тік бағытта –вертикаль туралау.
<TD> ... </TD> – кесте бағанасы. Оның параметрлері BGCOLOR – бағананың ішкі фоны түсі; ALIGN=LEFT, RIGHT, CENTER – бағана ішіндегі

мәтінді көлденең туралау; VALIGN=TOP, BOTTOM, MIDDLE – бағана ішіндегі мәтінді тік бағытта туралау; COLSPAN – бірнеше бағананы біріктіру, ROWSPAN – бірнеше жолды біріктіру.
<TH> ... </TH> – бағана тақырыптары. Оның параметрлері BGCOLOR – осы бағананың ішкі фон түсі; ALIGN=LEFT, RIGHT, CENTER – көлденең туралау; VALIGN=TOP, BOTTOM, MIDDLE – вертикаль туралау; COLSPAN, ROWSPAN – бірнеше бағананы және жолдарды біріктіру; WIDHT – тақырып ені.

Фреймдер жасау тәгтері

Фреймдер немесе кадрлар терезені бірнеше бөліктерге бөліп, олардың әрқайсысында жеке web-құжаттар орналастырады.
<FRAMESET> ... </FRAMESET> – фрейм жасау тәгтері. Оның параметрлері:
COLS – экранды бірнеше бағаналық (вертикаль) фреймдерге бөледі.
ROWS – экранды бірнеше қатар түріндегі (горизонталь) фрейм жолдарына бөледі.
BORDCOLOR – фреймнің жақтау сызығы түсі
BORDER – жақтау сызығы (бордюр) ені
FRAMEBORDER – жақтау шекарасы сызығы (FRAMEBORDER=YES – шекара сызығы бар, FRAMEBORDER=NO – шекара сызығы жоқ, *FRAMESPACING=n* – шекара сызығы ені)
<FRAME> – фреймді сипаттау тәгі (<FRAME SRC='file.htm">). Оның параметрлері:
SCROLING – жылжыту жолағын реттеу параметрі:
SCROLING=YES – жылжыту жолағы көрсетіледі
SCROLING=NO – жылжыту жолағы көрсетілмейді
SCROLING=AUTO – жылжыту жолағы қажет болғанда ғана көрсетіледі
MARGINWIDHT және *MARGINHEIGHT* – фрейм ішіндегі суреттерді ығыстырып орналастыруға қажет параметрлер
NORESIZE – фрейм көлемінің өзгертілмейтінін көрсететін параметр.
<A HREF="file.htm" TARGET="frame2"> file.htm</A> – фреймдер арасында байланыс жасау
TARGET – фреймдер арасында байланыс жасау атрибуты, оның бірнеше мәндері бар:
_BLANK – сілтеме көрсетіп тұрған парақты жаңа бос терезеге жүктеу;
_SELF – сілтеме көрсетіп тұрған парақты сол сілтеме тұрған терезеге жүктеу;
_PARENT – сілтеме көрсетіп тұрған парақты фреймдерді ашып тұрған негізгі терезеге жүктеу;
_TOP – сілтеме көрсетіп тұрған парақты фреймдер құрылымы біріктірілген толық терезеге жүктеу.
Фреймдерді көрсету қабілеті жоқ броузерлер үшін:
<FRAMESET>
...Мұнда фреймдер орналасады...
</FRAMESET>

```
<NOFRAMES>
   <BODY>
      ...Мұнда фреймсіз мәтін орналасады...
   </BODY>
 </NOFRAMES>
```

Фреймдерді көре алмайтын тұтынушылар үшін арнайы тәг ішіне түсініктеме мәтін беріледі.

2-қосымша

Құжаттардағы түстер және өлшем бірліктері

Web-құжаттарын безендіру кезінде атрибуттар мәндері ретінде әр түрлі объектілердің көлемдері мөлшерлерін көрсету керек және де объектілер элементтерінің түстерін анықтау қажет болады. HTML ортасында түстер мен өлшем бірліктерін анықтаудың стандартты ережелері бар. Түстерден бастайық.

Түстерді екі түрлі тәсілмен беруге болады: код арқылы және ағылшын тіліндегі түс атауын жазу жолымен (black, yellow, т.с.с.). Көбінесе түс мәні он алтылық RGB-кодтарымен беріледі. Кез келген түсті негізгі үш түс арқылы – *қызыл, жасыл, көк* түстерді белгілі бір мөлшерде араластыра отырып алуға болатынын текемет басатын, кілем тоқитын әжелеріміз ертеден-ақ білген болатын. Осы үш түсті пропорция бойынша араластыра отырып, компьютер экранында 16 миллионнан аса табиғи түстерді бейнелеуге болады екен. Мұндай көп түс түрлері негізгі түс мәндерінің әрқайсысының мөлшерін 0-ден 255-ке дейінгі ондық сандар аралығында (немесе 00-ден FF-ке дейінгі он алтылық сандар аралығында) өзгерту жолымен алынады. Сонымен түстер # таңбасынан басталатын алты разрядты он алтылық сандармен өрнектеледі, оның алғашқы екі разряды – қызыл, келесі екі разряды – жасыл, соңғы екі разряды – көк түс мөлшерлерін бейнелейді. Мысалы, color="#FF0000" қызыл түсті көрсетеді, яғни олар “#RRGGBB” түрінде беріледі, мұндағы RR – қызыл түстің он алтылық мәні, GG – жасыл түстің, ал BB – көк түстің он алтылық мәні. Web-палитраға кіретін 216 негізгі түстер алдын ала анықталған алты кодтың өз деңгейлеріне байланысты комбинациялық араласуынан шығады: 00, 33, 66, 99, CC, FF. Төмендегі *кестеде* түстер палитрасын пайдалану кезінде қолданылатын негізгі 20 түстің **RGB** форматындағы кодтары берілген,

Негізгі түстердің RGB форматында жазылу кодтары

Түс	*16-лық коды*	*Түс*	*16-лық коды*
Қара – Black	#000000	Жасыл – Green	#008000
Күміс түсті – Silver	#C0C0C0	Ақ жасыл – Lime	#00FF00
Сұр – Gray	#808080	Олив түсті – Olive	#808000
Ақ – White	#FFFFFF	Сары – Yellow	#FFFF00
Қызылқоңыр – Maroon	#800000	Қара көк – Navy	#000080
Қызыл – Red	#FF0000	Көкшіл – Blue	#0000FF
Күлгін – Purple	#800080	Көк жасыл – Teal	#008080
Қызғыш – Fuchsia	#FF00FF	Толқын түсті – Aqua	#00FFFF
Қоңыр– Brown	#996633	Қызғыл тсары – Orange	#FF8000
Көкжасыл – Azure	#00FFFF	Қызыл көк– Violet	#8000FF

Енді ұзындықты өлшеу бірліктерін қарастырайық HTML тілінің ережелеріне сәйкес web-құжат объектілерінің көлемі екі тәсілмен көрсетіледі. Ұзындық нақты пиксель (экранның ең кіші нүктесі) мәнімен немесе негізгі аналық объект (терезенің) көлемінің пайызымен анықталады. Егер бір кесте жасап, оның енін 50% деп жазсақ, онда ол броузер терезесінің 50 пайызын алып тұрады. Ал егер кесте бағанасының ені пайызбен берілсе, ол осы бағана орналасқан кестенің жалпы енінің неше пайызын құрайтынын көрсетеді. Жалпы көрсетілген ұзындық терезе көлемі өзгертілген сәтте ішкі көлем сәйкестіктерін толық сақтап отыруы тиіс. Объект мөлшерін пиксельмен нақты көрсету кезінде алыстағы тұтынушы компьютерінің сипаттамаларының (экрандағы пиксельдер саны) әр түрлі болатынын есепке алып отыру қажет.

Белгілі бір элементтің көлемін пиксельмен беру кезінде оның мүмкін болатын ең кіші мәнін алған дұрыс. Егер бір объектінің ені отыз пиксель болатын болса, оны былайша жазып көрсету керек:

width="30"

Ал оның ені негізгі объекті енінің отыз пайызын құрайтын болса, онда былайша жазу қажет:

width="30%"

Бұл екі тәсілден бөлек тағы бір тәсіл қолданылуы мүмкін. Біз алдыңғы бір көрсетілген пиксельдер мәнін бірнеше есе үлкейтіп пайдалана аламыз. Мысалы, үш жолдан тұратын кесте тұрғыздық делік. Кестенің биіктігі әйтеуір бір тәсілмен тағайындалған болсын. Енді әрбір жолдың биіктігі отыз пиксельге еселеніп (30, 60, 90, ...) берілуі қажет болса, онда әрбір жолды құратын тәгтер параметрлері ретінде мынадай мән жазуымыз керек:

height="3*"

Ұзындық мөлшерінің еселеніп өзгеру белгісі “жұлдызша” (*) символымен көрсетіледі. Жаңа мән есептелуі кезінде жұлдызшаның сол жағындағы сан онға (жиырмаға, отызға,...) көбейтіледі. Броузер осындай объектілерді барынша үлкейтіп бейнелеуге тырысады. Егер кестенің жалпы биіктігі 180 пиксель болатын болса, онда әрбір жол биіктігі 60 пиксель болып тағайындалады. Кесте биіктігі 200 пиксель болса да, осы мәндер алынып, 20 пиксель пайдаланылмай қалады. Егер жолдар биіктігі бірдей болуы керек болса, онда height="*" болып жазылады. Егер атрибутта екі мән көрсетілетін кездерде, олардың біреуі “жұлдызша” таңбасымен жазылса, онда ол объектінің алғашқысынан қалған бөлігін алып тұрады деген ұғымға сәйкес келеді. Мысалы, фреймдердегі: cols=”30,*” деп жазылған атрибут екі бағана түріндегі фрейм ашылып, оның алғашқысының ені 30 пиксель, ал екіншісі терезенің қалған бөлігін алып тұратынын көрсетеді. Жалпы алдын ала келісім бойынша да, мән көрсетілмеген жағдайда осы тәсіл қабылданған.

Тест сұрақтарына мысалдар

1. Құжат фрагменттерін белгілеу үшін және оның экранда бейнелеуін басқаруға арналған HTML тілінің арнайы элементтері, яғни компоненттері:
   А. тәгтер
   Б. блоктар
   В. тізімдер
   Г. сілтемелер
   Д. кестелер
2. Мына тұжырымдардың қайсысы дұрс:
   А. Тәгтер жазу кезінде бас әріп пен кіші әріп бірдей болып саналады
   Б. HTML тілінің барлық тәгтері &quot&gt&quot символынан басталады
   В. Блокнот арқылы гипермәтіндік құжатты жасауға болмайды
   Г. HTML құжатына коментарийлер енгізуге болмайды
   Д. HTML мәтіндік элементтері ішіне <,&gt, “, @ символдары жазылады
3. HTML элементтері мынадай екі топқа бөлінеді:
   А. Блоктық және мәтіндік
   Б. Мәтіндік және сандық
   В. Символдық және цифрлық
   Г. Жай және күрделі
   Д. Қарапайым және құрамалы
4. <BODY> тәгінің мәтін түсін анықтайтын атрибуты:
   А. TEXT
   Б. BGCOLOR
   В. COLOR
   Г. LINK
   Д. VLINK
5. <BODY>тәгінің курсор көрсетіп тұрған гиперсілтеме сөз түсін беретін атрибуты...
   А. ALINK
   Б. VLINK
   В. TEXT
   Г. COLOR
   Д. LINK
6. HTML тілі мәтін бөліктерін форматтаудың қандай стильдерін жүзеге асыра алады?
   А. логикалық және физикалық
   Б. логикалық
   В. физикалық
   Г. мәтіндік
   Д. математикалық
7. HTML тіліндегі <HR>тәгінің сызықтың қалыңдығын анықтайтын атрибуты:
   А. SIZE
   Б. ALIGN

В. WIDTH
Г. NOSHADE
Д. COLOR

8. HTML тіліндегі <HR>тәгінің сызықтың рельефтілігін анықтайтын атрибуты:
А. NOSHADE
Б. ALIGN
В. WIDTH
Г. SIZE
Д. COLOR

9. HTML тілінде <STRONG> </STRONG>тәгтерінің арасында мәтін қандай қаріппен жазылады:
А. Қарайтылған
Б. Курсив
В. Программа кодтарын жазуға арналған
Г. Асты сызылған
Д. Бел ортасынан сызылған

10. <IMG> тәгінің ALIGN атрибуты қандай қызмет атқарады:
А. Мәтінге байланысты суреттің орналасуын көрсетеді
Б. Суреттің URL адресін береді
В. Суреттің биіктігін береді
Г. Сурет көрсетілмейтін браузерде оның аты мәтін ретінде көрсетіледі
Д. Суреттің сол және оң жақ шеттеріндегі бос аймақ мөлшерін пиксельмен береді

11. <IMG>тәгінің VSPACE атрибуты қандай қызмет атқарады:
А. Суреттің жоғарғы және төменгі жақтарындағы бос аймақ мөлшерін
Б. тағайындайды
В. Суреттің URL адресін береді
Г. Суреттің биіктігін береді
Д. Сурет көрсетілмейтін браузерде оның аты мәтін ретінде көрсетіледі
Е. Суреттің сол және оң жақ шеттеріндегі бос аймақ мөлшерін пиксельмен береді

12. Гиперсілтеме жасағанда <A HREF = …>тәгінде HREF трибутының атқаратын қызметі:
А. Гиперсілтемеде шақырылатын ресурстың URL адресін көрсетеді
Б. Суреттің URL адресін көрсетеді
В. Суретттің енін өзгертеді
Г. Суреттің биіктігін өзгертеді
Д. Мәтіннің түсін өзгертеді

13. <UL>… <UL>тәгтерінің көмегімен...
А. Маркерленген (белгіленген) тізімді жасай аламыз
Б. Нөмірленген тізімдер жасауға болады
В. Жаңа абзацқа көшеміз
Г. Горизонталь сызықты жүргізе аламыз

Д. Бүкіл мәтінді бір жолға жаза аламыз

14. <UL>... </UL>ортасында орналасқан әрбір тізім элементтері қандай тәгтен соң жазылады?
А. <LI>
Б. <UL>
В. <HL>
Г. <DL>
Д. <DD>

15. HTML тілінде <TABLE>тәгінің кестенің жақтау сызығы мен мәтін аралығындағы бос кеңістік көлемін беретін атрибуты:
А. CELLPADDING
Б. CELLSPACING
В. WIDTH
Г. HEIGHT
Д. VSPACE

16. HTML тілінде <TABLE>тәгінің кестенің жоғарғы және төменгі жақтарында қалдырылған бос кеңістік көлемін өзгерту мүмкіндігін беретін атрибуты:
А. VSPACE
Б. CELLSPACING
В. CELLPADDING
Г. WIDTH
Д. HEIGHT

17. HTML тілінде <TABLE>тәгі арқылы кесте жасауда WIDTH атрибуты қандай қызмет атқарады?
А. Кестенің жалпы енін өзгерту мүмкіндігін береді
Б. Кестенің жақтау сызықтарының қалындығын анықтайды
В. Кестенің жалпы биіктігін өзгертуге мүмкіндігін береді
Г. Кестенің жақтау сызығы мен мәтін арасындағы бос кеңістік көлемін өзгертуге мүмкіндік береді
Д. Кесте ұяшығының жақтау сызықтары аралығында бос кеңістікті өзгертуге мүмкіндігін береді

18. HTML тілінде <TABLE>тәгі арқылы кесте жасауда HEIGHT атрибуты қандай қызмет атқарады?
А. Кестенің жалпы биіктігін өзгерту мүмкіндігін береді
Б. Кестенің жақтау сызықтарының қалындығын анықтайды
В. Кестенің жалпы енін өзгертуге мүмкіндігін береді
Г. Кестенің жақтау сызығы мен мәтін арасындағы бос кеңістік көлемін өзгертуге мүмкіндік береді
Д. Кесте ұяшығының жақтау сызықтары аралығында бос кеңістікті өзгертуге мүмкіндігін береді

19. HTML тілінде <TABLE>тәгі арқылы кесте жасауда CELLSPACING атрибуты қандай қызмет атқарады?

А. Кесте ұяшығының жақтау сызықтары аралығында бос кеңістікті өзгерту мүмкіндігін береді
Б. Кестенің жақтау сызықтарының қалындығын анықтайды
В. Кестенің жалпы енін өзгертуге мүмкіндігін береді
Г. Кестенің жалпы биіктігін өзгертуге мүмкіндігін береді
Д. Кестенің жақтау сызығы мен мәтін арасындағы бос кеңістік көлемін өзгертуге мүмкіндік береді

20. HTML тілінде кестелерде ROWSPAN атрибуты қандай қызмет атқарады?
А. Бір бағананың екі жолында қатар тұрған екі ұяны біріктіру үшін қолданылады
Б. Бір жолдың бағанасында қатар тұрған екі ұяны біріктіру үшін қолданылады
В. Мәтіннің фон түсін анықтайды
Г. Кестені жақтай сызығы мен мәтін арасындағы бос кеңістікті көрсетеді
Д. Екі сырғудың арасындағы интервалынды миллисекундамен көрсететін бүтін сан

21. HTML сөзінің толық мағынасы ...
А. HyperText Markup Language
Б. Historial Tehnical Markup Language
В. HyperTehnical Markup Language
Г. Historial Tutorial Markup Language
Д. Дұрыс жауап жоқ

22. Гипермәтіндік құжат құрамында мынадай символдық ақпарат болады:
А. мәліметтік мәтіндер мен тәгтер
Б. командалар мен арнайы белгілер
В. дескрипторлар мен операторлар
Г. функциялар мен операциялар
Д. процедуралар мен контейнерлер

23. HTML-құжаттарды сақтайтын файлдардың типтері келесідей болады:
А. .htm немесе .html
Б. .doc немесе .rtf
В. .gif немесе .jpg
Г. .txt немесе .text
Д. .xml немесе .exl

24. HTML тіліндегі тәгтер арқылы мынадай әрекеттерді орындауға болады:
А. Қалған төрт жауап дұрыс
Б. Құжат тақырыбын ерекшелеуге, гиперсілтемелер жасауға болады
В. Символдардың түсін, мөлшерін және сызылымын өзгертуге болады
Г. Фреймдер, формалар құруға болады
Д. Құжат ішіне суреттер орналастырылады

25. <BODY>тәгінің ALINK атрибутының атқаратын қызметі:
А. Курсор көрсетіп тұрған гиперсілтеме сөз түсін береді

Б. Мәтін түсін анықтайды
В. Әлі қаралмаған гиерсілтеме сөз түсін береді
Г. Алдында қаралған гиперсілтеме сөз түсін береді
Д. Фон түсін анықтайды

26. <BODY>тәгінің TEXT атрибутының атқаратын қызметі:
А. Құжат мәтінінің жалпы түсін анықтайды
Б. Курсор көрсететін гиперсілтеме сөз түсін береді
В. Әлі қаралмаған гиерсілтеме сөз түсін береді
Г. Алдында қаралған гиперсілтеме сөз түсін береді
Д. Құжаттың фон түсін анықтайды

27. <Frameset>тәгінің қандай атрибуты терезені горизонталь – көлденең бағыт бойынша (жолдарға) бөледі ?
А. ROWS
Б. COLS
В. WIDTH
Г. HEIGHT
Д. LOOP

28. <FRAME> тәгінде SRC атрибутының атқаратын қызметі:
А. Фреймде шығарылатын құжаттың URL-адресін анықтайды
Б. Суреттің URL- адресін көрсетеді
В. Текст түсін өзгертеді
Г. Терезені горизонталь бағыт бойынша бөледі
Д. Терезені тігінен бөледі

29. <FRAME>тәгінің MARGINWIDTH атрибутының атқаратын қызметі:
А. Фреймнің оң және сол жақтарындағы пиксельмен берілген бос кеңістік мөлшерін анықтайды
Б. Фрейм өлшемін береді
В. Фрейм көлемі тұрақты болып, оның өзгермейтінін көрсетеді
Г. Фреймнің жоғары және төмен жақтарындағы пиксельмен берілген бос кеңістіктің мөлшерін анықтайды
Д. Фрейм бойынша жылжыту жолақтары бар не жоқ болуын көрсетеді

30. Фрейм көлемі тұрақты болып, оның өзгермейтінін көрсететін атрибут:
А. NORESIZE
Б. MARGINHEIGHT
В. MARGINWIDTH
Г. SCROLLING
Д. BORDERCOLOR

31. Фреймнің жоғары және төмен жақтарындағы пиксельмен берілген бос кеңістіктің мөлшерін анықтайтын атрибут:
А. MARGINHEIGHT
Б. MARGINWIDTH
В. SCROLLING
Г. NORESIZE
Д. BORDERCOLOR

32. Фреймдерді пайдалану кезіндегі <A TARGET=frame1> деген сілтеме нені білдіреді?
А. Осындай аты бар фреймге гипермәтіндік сілтеме жасау
Б. Фреймге frame1 деген ат берілді
В. Берілген фреймді frame1 фреймімен ауыстыру
Г. Frame1 фреймін жою
Д. Frame1 фрейміне жасаған сілтеменің түсін өзгерту

33. HTML. Фреймдерге сілтеме жасайтын <A TARGET=…> TARGET атрибутының _blank тұрақты мәні қандай қызмет атқарады?
А. Көрсетілген файлды ат қойылмаған жаңа терезеге жүктейді
Б. Көрсетілген файлды сілтеме жасалған фреймге жүктейді
В. Көрсетілген файлды кадрлар құрылымы біріктірілген толық терезеге жүктейді
Г. Көрсетілген файлды фреймдерді ашатын түпкі кадрға жүктейді; егер ондай кадр анықталмаған болса, онда оның әсері алдыңғы _self қызметімен бірдей болады
Д. Көрсетілген файлды келесі жасалған фреймге жүктейді

34. Фреймдерге сілтеме сілтеме жасайтын <A TARGET=…>TARGET атрибутының _self тұрақты мәні қандай қызмет атқарады?
А. Көрсетілген файлды сілтеме жасалған фреймге жүктейді
Б. Көрсетілген файлды кадрлар құрылымы біріктірілген толық терезеге жүктейді
В. Көрсетілген файлды фреймдерді ашатын түпкі кадрға жүктейді; егер ондай кадр анықталмаған болса, онда оның әсері алдыңғы _self қызметімен бірдей болады
Г. Көрсетілген файлды ат қойылмаған жаңа терезеге жүктейді
Д. Көрсетілген файлды келесі жасалған фреймге жүктейді

35. Фреймдерге сілтеме жасайтын <A TARGET=…>тәгінің TARGET атрибутының _parent тұрақты мәні қандай қызмет атқарады?
А. Көрсетілген файлды фреймдерді ашатын түпкі кадрға жүктейді; егер ондай кадр анықталмаған болса, онда оның әсері алдыңғы _self қызметімен бірдей болады
Б. Көрсетілген файлды сілтеме жасалған фреймге жүктейді
В. Көрсетілген файлды кадрлар құрылымы біріктірілген толық терезеге жүктейді
Г. Көрсетілген файлды ат қойылмаған жаңа терезеге жүктейді; егер ол анықталмаса, онда ол негізге фреймге әсер етеді
Д. Көрсетілген файлды келесі жасалған фреймге жүктейді

36. <FORM> тәгінің METHOD атрибутының атқаратын қызметі қандай?
А. Форманы өңдейтін программаға қалайша ақпарат жөнелтілетінін көрсетеді
Б. Форма мәліметтерін қабылдап алып, оны өңдейтін URL анықтайды
В. Бірнеше жолдан тұрған мәтіндік ақпарат енгізуге мүмкіндік береді

Г. Ақпарат атын анықтайды
Д. Таңдай операциялары үшін терезе биіктігін анықтайды

37. HTML формаларын құрудағы <TEXTAREA> тәгінің атқаратын қызметі қандай?
А. Кез келген тұтынушы бірнеше жолдан тұратын мәтіндік ақпарат енгізе алатын өрісті анықтайды
Б. Форма мәліметтерін қабылдап алып, оны өңдейтін URL-ды анықтайды.
В. Мәлімет енгізудің кейбір өзге түрлерін: бір жол мәтін енгізу, жалаушаларды көтеріп қою және мәліметтерді жөнелту сияқты әрекеттерді орындауды қамтамасыз етеді
Г. Тұтынушыға жылжымалы жолағы бар терезедегі немесе суырылып шығатын меню ішіндегі бір жолды таңдау мүмкіндігін береді
Д. Кез келген тұтынушы бірнеше жалаушадан тұратын форма жасағанда, соның алғашқысын таңдау мүмкіндігін береді

38. HTML формаларын құрудағы <FORM>тәгінің ACTION атрибутының атқаратын қызметі қандай?
А. Форма мәліметтерін қабылдап алып, оны өңдейтін URL-ды анықтайды. Егер бұл атрибут анықталмаса, онда мәліметтер форма орналасқан web-парақ адресіне жіберіледі
Б. Мәлімет енгізудің кейбір өзге түрлерін: бір жол мәтін енгізу, жалаушалар қою және мәліметтерді жөнелту сияқты әрекеттерді орындауды қамтамасыз етеді
В. Кез келген тұтынушы бірнеше жолдан тұратын мәтіндік ақпарат енгізе алатын өрісті анықтайды
Г. Тұтынушыға жылжымалы жолағы бар терезедегі немесе суырылып шығатын меню ішіндегі бір жолды таңдау мүмкіндігін береді
Д. Кез келген тұтынушы бірнеше жалаушадан тұратын форма жасап, соның бірін таңдау мүмкіндігін береді

39. HTML формаларын құрудағы <INPUT>тәгінің атқаратын қызметі қандай?
А. Мәлімет енгізудің кейбір түрлерін: бір жол мәтін енгізу, жалаушалар қою (check boxes), ажыратып қосқышты таңдау және мәліметтер жөнелту сияқты әрекеттерді орындауды қамтамасыз етеді
Б. Форма мәліметтерін қабылдап алып, оны өңдейтін URL-ды анықтайды. Егер бұл атрибут анықталмаса, онда мәліметтер форма орналасқан web-парақ адресіне жіберіледі
В. Кез келген тұтынушы бірнеше жолдан тұратын мәтіндік ақпарат енгізе алатын өрісті анықтайды
Г. Тұтынушыға жылжымалы жолағы бар терезедегі немесе суырылып шығатын меню ішіндегі бір жолды таңдау мүмкіндігін береді
Д. Кез келген тұтынушы бірнеше жалаушадан тұратын форма жасағанда, соның алғашқысына мәлімет енгізу мүмкіндігін береді

40. HTML тілі – қандай тіл
А. гипермәтінді белгілеу тілі
Б. тек дайын браузерлермен ғана жұмыс істеу тілі
В. сілтеме жасау тілі
Г. программа құрастыру тілі
Д. ақпаратты қорғау тілі
41. “Сырғымалы жолдардың” қайталану санын анықтайтын атрибут...
А. LOOP
Б. SCROLLAMOUNT
В. SCROLLDELAY
Г. WIDTH
Д. SCROL
42. Сырғымалы жолдардағы жылжудың бір қадамында мәтіннің қанша пикселге жылжитынын анықтайтын атрибут...
А. SCROLLAMOUNT
Б. LOOP
В. SCROLLDELAY
Г. WIDTH
Д. SCROLL
43. Сырғымалы жолдың екі қадамы арасындағы интервалды миллисекундпен көрсететін атрибут...
А. SCROLLDELAY
Б. SCROLLAMOUNT
В. LOOP
Г. WIDTH
Д. SCROLL
44. HTML тіліндегі “сырғымалы жолдың” енін анықтайтын атрибут, оның мәнін пиксельмен (бүтін сан) де, пайызбен де (%) көрсетуге болады:
А. WIDTH
Б. SCROLLAMOUNT
В. LOOP
Г. SCROLLDELAY
Д. SCROLL
45. Сырғымалы жолдың жылжу бағытын көрсететін атрибут...
А. DIRECTION
Б. SCROLLAMOUNT
В. LOOP
Г. SCROLLDELAY
Д. SCROLL
46. GIF және JPEG сөздері нені көрсетеді?
А. графикалық форматтар типтерін көрсетеді
Б. мәтін типтерін көрсетеді
В. хабарлама түрін көрсетеді
Г. жіберілген қателіктерді көрсетеді

Д. суреттік файлдар көлемін көрсетеді

47. <OL TYPE=I ...</OL>тізбегі элементтерді қалай нөмірлейді?
А. үлкен рим цифрларымен
Б. араб цифрларымен
В. кіші рим цифрларымен
Г. латынның бас әріптерімен
Д. латынның кіші әріптерімен

48. <OL ... > тізбегі элементтерді 15-тен бастап нөмірлеу үшін нүктелер орнына қандай атрибут беріледі?
А. START=15
Б. BEGIN=15
В. HEAD=15
Г. TITLE=15
Д. STARTAMOUNT=15

49. HTML тілінде мынадай сан #FFFFFF қандай түсті білдіреді?
А. ақ
Б. сары
В. жасыл
Г. ақшыл
Д. көгілдір

50. HTML тілінде мынадай сан #00FF00 қандай түсті білдіреді?
А. жасыл
Б. сары
В. қара
Г. көгілдір
Д. ақ

51. HTML тілінде мынадай сан #0000FF қандай түсті білдіреді?
А. көк
Б. сары
В. қара
Г. қызыл
Д. ақ

52. HTML тілінде мынадай сан #000000 қандай түсті білдіреді?
А. қара
Б. жасыл
В. сары
Г. көгілдір
Д. ақ

53. Браузер дегеніміз не?
А. Гипермәтінді экранға шығарып бейнелейтін арнайы көрсету программалары
Б. Тәгтер жиынтығы
В. Қосымша элементтерді басқару мақсатында ішіне арнаулы код орналасқан мәтін

Г. Құжатты функционалды түрде белгілейтін тіл
Д. HTML тілінде жазылған арнаулы файлдар

54. Гипермәтіндерді түрлендіре алатын арнайы программалау тілінде жазылған программалық кодтар былай аталады:
А. скриптер
Б. жазбалар
В. тәгтер
Г. айнымалылар
Д. функциялар

55. CSS-те жеке бір тәг үшін берілген стильді қай атрибут арқылы жазамыз?
А. STYLE
Б. CSS
В. SHEETS
Г. LINK
Д. CASCAD

56. Стильдерді (CSS) беру тәсілдерін араластыра пайдаланғанда қайсысы басым болады?
А. Жеке тәг үшін жазылған стиль
Б. Жеке HTML-файлға арналған стиль
В. Басқа CSS файлында анықталған стиль
Г. Бірнеше HTML-файлдарға арналған стиль
Д. Бірнеше тәгтерге арналған стиль

57. Символдар мөлшерін беретін 1 пункт неше миллиметрге тең?
А. 0,375
Б. 1/12
В. 12
Г. 1/72
Д. 2,54

58. CSS-те sans-serif қандай қаріп түрін көрсетеді?
А. жұмыр қаріп түрін (шрифт без засечек – рубленый)
Б. шығыңқы қаріп түрі (шрифт с засечками – серифный)
В. символдарының ені бірдей қаріп түрі (моноширинный шрифт)
Г. альбомдық қаріп түрі (альбомный шрифт)
Д. кітаптық қаріп түрі (книжный шрифт)

59. CSS-те serif қандай қаріп түрін көрсетеді?
А. шығыңқы қаріп түрі (шрифт с засечками – серифный)
Б. жұмыр қаріп түрін (шрифт без засечек – рубленый)
В. символдарының ені бірдей қаріп түрі (моноширинный шрифт)
Г. альбомдық қаріп түрі (альбомный шрифт)
Д. кітаптық қаріп түрі (книжный шрифт)

60. CSS-те стильдердің *font-family* сипаттамасы арқылы:
А. қаріп типін өзгертеміз
Б. қаріп түсін өзгертеміз

В. фонның түсін өзгертеміз
Г. сілтеме жасай аламыз
Д. тізімдер құрамыз

61. CSS-те text-align: right; тіркесі нені білдіреді?
А. мәтінді оң жақ шетке туралайды
Б. мәтінді кез келген орынға қояды
В. мәтінді екі шетке қарай туралайды
Г. мәтінді сол жақ шетке туралайды
Д. ешнәрсе жасамайды

62. CSS-те font-family: &quotArial Cyr&quot,Geneva, sans-serif; тіркесі қандай әрекет орындауды талап етеді немесе қай қаріп түрін пайдалануды көрсетеді?
А. Мәтін бірінші кезекте Arial Cyr қарпімен берілуі тиіс, егер ол компьютерде болмаса, сәйкесінше Geneva, sans-serif қаріп түрлерін қолдану керек.
Б. Алғашқы абзац Arial Cyr қарпімен, келесісі – Geneva, ал соңғысы – sans-serif қарпімен берілуі тиіс.
В. Тарау тақырыптары Arial Cyr қарпімен, параграф тақырыптары – Geneva қарпімен, ал негізгі мәтін sans-serif қарпімен берілуі тиіс.
Г. Программа көрсетілген қаріп түрлерінің кез келгенін пайдалана алады.
Д. Қаріп түрінің тек біреуі көрсетілуі тиіс болатын.

63. CSS-те стильдік нұсқауларда пайдаланылатын өлшем бірліктерінің берілу ретіне қарай дұрыс жазылуын көрсету керек:
Пика, Пункт, Пиксель, Миллиметр, Сантиметр, Дюйм:
А. pc, pt, px, mm, cm, in
Б. pic, pt, pix, mm, cm, duim
В. pa, pn, pc, mm, sm, dm
Г. pk, pu, px, mm, sm, “
Д. pica, punkt, mm, cm, dum

64. CSS-те 1 типографиялық пункт қанша дюймге тең?
А. 1/72
Б. 1/12
В. 1/7
Г. 1/24
Д. 1/2

65. CSS-те жұмыр қаріптер (рубленый шрифт) көбінесе нені жазу үшін пайдаланылады?
А. тақырыптарды
Б. майда ескертпелерді
В. индекстер мен дәрежелерді
Г. негізгі мәтінді
Д. программалар мәтіндерін

66. CSS-те шығыңқы қаріптер (шрифт с засечками) көбінесе нені жазу үшін пайдаланылады?

А. негізгі мәтінді
Б. майда ескертпелерді
В. индекстер мен дәрежелерді
Г. программалар мәтіндерін
Д. тақырыптарды

67. CSS-те ендері бірдей қаріптер (моноширинный шрифт) көбінесе нені жазу үшін пайдаланылады?
А. программалар мәтіндерін
Б. майда ескертпелерді
В. индекстер мен дәрежелерді
Г. негізгі мәтінді
Д. тақырыптарды

68. CSS-те стильдердегі border-style сипаттамасы нені анықтайды:
А. элементті қоршаған жақтау сызықтардың стилін
Б. элементтің өзара орналасу стилін
В. құжат ішіндегі барлық сызықтар стилін
Г. құжат ішіндегі барлық мәтін стилін
Д. құжат ішіндегі барлық тақырыптар стилін

69. CSS-те стильдердегі border-width сипаттамасы нені анықтайды:
А. жақтау сызықтарының қалыңдығын
Б. элементті қоршап тұратын түзу сызықтар типін
В. жақтау сызықтарының енін
Г. жақтау сызықтарының биіктігін
Д. жолдағы мәтіндер енін

70. CSS-те стильдердегі margin сипаттамасы нені анықтайды:
А. элементтердегі блок шеттерінде бос қалатын өріс енін
Б. элементтер алдында қалатын бос өріс енін
В. элементтер артында қалатын бос өріс енін
Г. элементтердің астыңғы жағындағы бос өріс енін
Д. элементтердің үстіңгі жақтағы бос өріс енін

71. CSS-те гарнитура дегеніміз:
А. Бір қаріп символдарының сызылымдары жиыны
Б. Кестені жақтау сызығының түсін өзгертуге мүмкіндік беретін атрибут
В. CCS файлдарын сақтайтын арнайы бума
Г. Мультимедиалық файлдарды оқуға мүмкіндік беретін программа
Д. Кестелерді құруға мүмкіндік беретін атрибут

72. CSS-тегі қаріп стильдері:
А. Font-family, font-size
Б. Size, color
В. Courier, background
Г. Font-size, color
Д. Font-size, background

73. CSS-тегі түс стильдері:
А. Color, background-color

Б. Font-family, font-size
В. Size, color
Г. Font-size, color
Д. Font-size, background

74. CSS-тегі мәтін стильдері:
А. Letter-spacing, line-height, text-align
Б. Font-family, font-size
В. Size, color, line-height
Г. Color, background-color
Д. Font-size, color

75. CSS-тегі жақтаулар мен өрістер стильдері:
А. Border-style, border-color, border-width, margin, padding
Б. Font-family, font-size
В. Size, border-style, color
Г. Font-size, color, border-width, margin, padding
Д. Font-size, background

76. CSS-те Font-size стилі бізге қандай мүмкіндік береді?
А. Қаріптің абсолюттік немесе салыстырмалы мөлшерін береді
Б. Қаріп түрін өзгертуге мүмкіндік береді
В. Фонның түсін өзгертуге мүмкіндік береді
Г. Сілтеме жасауға мүмкіндік береді
Д. Тізімдерді құруға мүмкіндік береді

77. CSS-те әріптер арасындағы қашықтықты анықтау үшін қандай стиль қолданылады?
А. Letter-spacing
Б. Font-family
В. Background
Г. Line-height
Д. Border-style

78. CSS-те жолдар аралығын (интервалын) тағайындау үшін қандай стиль қолданылады?
А. Line-height
Б. Font-family
В. Background
Г. Letter-spacing
Д. Border-style

79. CSS-те мәтінді көлденең туралау тәсілін көрсететін стиль:
А. Text-align
Б. Background
В. Letter-spacing
Г. Line-height
Д. Border-style

80. CSS-те элементті қоршап тұрған жақтау сызықтардың типін анықтайтын стиль:

А. Border-style
Б. Font-family
В. Background
Г. Letter-spacing
Д. Line-height

81. CSS-те Letter-spacing стилі бізге қандай мүмкіндік береді?
А. Әріптер арасындағы аралықты анықтау
Б. Жолдар аралығын тағайындау
В. Мәтінді көлденең туралау тәсілін көрсетету
Г. Элементті қоршап тұрған жақтау сызықтардың түрін анықтау
Д. Қаріп түрін өзгерту

82. CSS-те Border-style стилі бізге қандай мүмкіндік береді?
А. Элементті қоршап тұрған жақтау сызықтардың түрін анықтау
Б. Жолдар аралығын тағайындау
В. Мәтінді көлденең туралау тәсілін көрсетету
Г. Әріптер арасындағы аралықты қосымша анықтау
Д. Қаріп түрін өзгерту

83. CSS-те *line-height* стилі бізге қандай мүмкіндік береді?
А. Жолдар аралығын (интервал) тағайындау
Б. Мәтінді көлденең туралау тәсілін көрсетету
В. Элементті қоршап тұрған жақтау сызықтардың түрін анықтау
Г. Әріптер арасындағы аралықты қосымша анықтау
Д. Қаріп түрін өзгерту

84. CSS-те *Text-align* стилі бізге қандай мүмкіндік береді?
А. Мәтінді көлденең туралау тәсілін көрсетету
Б. Жолдар аралығын тағайындау
В. Элементті қоршап тұрған жақтау сызықтардың түрін анықтау
Г. Әріптер арасындағы аралықты қосымша анықтау
Д. Қаріп түрін өзгерту

85. CSS-те *Border-style: none* тіркесі нені білдіреді?
А. Элементті қоршаған жақтау сызығы жоқ
Б. Элементті қоршаған жақтау сызығының түсі қара
В. Элементті қоршаған жақтау сызығының қалыңдығы жуан
Г. Элементті жақтау сызығының қалыңдығы орташа
Д. Элементті жақтау сызығының қалыңдығы жіңішке

86. CSS-те *Border-style: solid;* тіркесі нені білдіреді:
А. Элементті жақтау сызығының кәдімгі түрі (обычная)
Б. Элементті жақтау сызығының түсі қара
В. Элементті жақтау сызығының қалыңдығы жуан
Г. Элементті жақтау сызығының қалыңдығы орташа
Д. Элементті жақтау сызығының қалыңдығы жіңішке

87. CSS. Жақтау сызығының қалыңдығын беретін стиль:
А. Border-width
Б. Font-family

В. Background
Г. Line-height
Д. Border-style

88. CSS-те *Border-width* стилі бізге қандай мүмкіндік береді?
А. Элементті қоршап тұрған жақтау сызықтардың түрін анықтау
Б. Жолдар аралығын тағайындау
В. Мәтінді көлденең туралау тәсілін көрсетету
Г. Жақтау сызығының қалындығын өзгерту
Д. Қаріп түрін өзгерту

89. CSS-те *Border-width: thick* ; жазуы нені білдіреді:
А. Элементті жақтау сызығының қалындығы жуан
Б. Элементті жақтау сызығының түсі қара
В. Элементті жақтау сызығының жалпы түрі (обычная)
Г. Элементті жақтау сызығының қалындығы орташа
Д. Элементті жақтау сызығының қалындығы жіңішке

90. CSS-те *margin* стилі бізге қандай мүмкіндік береді?
А. Элементтегі блок шеттерінде бос қалатын өріс енін анықтау
Б. Жолдар аралығын тағайындау
В. Мәтінді көлденең туралау тәсілін көрсетету
Г. Элементті қоршап тұрған жақтау сызықтардың түрін анықтау
Д. Әріптер арасындағы аралықты қосымша анықтау

91. CSS-те *padding* стилі бізге қандай мүмкіндік береді?
А. Элемент пен жақтау арасындағы қашықтықты анықтау
Б. Жолдар аралығын тағайындау
В. Элементті қоршап тұрған жақтау сызықтардың түрін анықтау
Г. Әріптер арасындағы аралықты қосымша анықтау
Д. Қаріп түрін өзгерту

92. CSS-те <DIV> тәгі құжаттың жеке бөліктерін ерекшелеп, . . .
А. оларды жаңа жолға көшіреді
Б. олар бұрынғы жолда қала береді
В. оларды бояуға мүмкіндік береді
Г. олардың өлшемін өзгертуге мүмкіндік береді
Д. оларды сырғымалы жолдар түрінде қолдануға мүмкіндіктер береді

93. CSS-те <SPAN> тәгтері құжаттың жеке бөліктерін ерекшелеп, . . .
А. Олар бұрынғы жолда қала береді
Б. Оларды жаңа жолға көшіреді
В. Оларды бояуға мүмкіндік береді
Г. Олардың өлшемін өзгертуге мүмкіндік береді
Д. Оларды сығырмалы жолдар түрінде қолдануға мүмкіндіктер береді

94. ?CSS. Экрандағы элементтің қай деңгейде орналасуы керек екендігін анықтайтын стильдік қасиет:
А. Z-index
Б. Font-family
В. Background

Г. Line-height
Д. Border-*style*

95. CSS-те салыстырмалы түрде орналастыру дегеніміз:
А. Координата басы ретінде элементтің позицияланбаған кездегі орналасатын нүктесі қабылданады
Б. Элементтің бастапқы координатасы тікелей сыртқы элемент аймағының жоғарғы сол жақ бұрышы болады.
В. Элемент мәтіннің не басқа элементтің үстіне орналасады
Г. Элемент фон ретінде қолданылады
Д. Элементке сілтеме жасалынады

96. CSS-те салыстырмалы орналастыру нұсқауы:
А. Position: relative
Б. Position: absolute
В. Position {abs}
Г. Position {rel}
Д. Position [absolute]

97. CSS-те абсолютті орналастыру нұсқауы:
А. Position: absolute
Б. Position: relative
В. Position {abs}
Г. Position {rel}
Д. Position [absolute]

98. CSS-те Absolute [relative] жазуы нені білдіреді?
А. Абсолюттік орналастыру ішіндегі салыстырмалы орналастыру
Б. Салыстырмалы орналастыру ішіндегі абсолютті орналастыру
В. Абсолюттік орналастыру ішіндегі абсолют орналастыру
Г. Салыстырмалы орналасу ішіндегі салыстырмалы орналастыру
Д. Осылай жазуға болмайды

99. CSS-те Absolute [absolute] жазуы нені білдіреді?
А. Абсолюттік орналастыру ішіндегі абсолют орналастыру
Б. Абсолюттік орналастыру ішіндегі салыстырмалы орналастыру
В. Салыстырмалы орналастыру ішіндегі абсолютті орналасытру
Г. Салыстырмалы орналасу ішіндегі салыстырмалы орналастыру
Д. Бұлай жазуға болмайды

100. CSS-те Relative [relative] жазуы нені білдіреді?
А. Салыстырмалы орналасу ішіндегі салыстырмалы орналастыру
Б. Абсолюттік орналастыру ішіндегі салыстырмалы орналастыру
В. Салыстырмалы орналастыру ішіндегі абсолютті орналасытру
Г. Абсолюттік орналастыру ішіндегі абсолют орналастыру
Д. Бұлай жазуға болмайды

101. CSS-те Relative [absolute] жазуы нені білдіреді?
А. Абсолюттік орналастыру ішіндегі абсолют орналастыру
Б. Абсолюттік орналастыру ішіндегі салыстырмалы орналастыру
В. Салыстырмалы орналастыру ішіндегі абсолютті орналасытру

Г. Салыстырмалы орналасу ішіндегі салыстырмалы орналастыру
Д. Бұлай жазуға болмайды

102. JavaScript. <script>…</script>тәгтері не үшін қажет:
А. Скриптер енгізу үшін
Б. Коментарий енгізу үшін
В. Формалар жасау үшін
Г. Сығырмалы жолдарды жасау үшін
Д. HTML-да осындай скриптер қолданылмайды

103. JavaScript тілінде alert('It\'s Ok!'); жолы қандай нәтиже береді?
А. It's Ok
Б. It is OK
В. 'It's OK'
Г. "It's OK"
Д. i\s OK

104. JavaScript тілінде шығарылатын жазбаны жаңа жолға көшіру үшін мынадай таңбалар қолданылады:
А. \n
Б. \'
В. n.
Г. &quot+&quot
Д. &quot*&quot

113. JavaScript тілінде айнымалыларды сипаттау үшін қандай түйінді сөз қолданылады?
А. var
Б. alert
В. prompt
Г. concat
Д. slice

105. JavaScript. Төмендегі программалық кодтың нәтижесі қандай?

```
<script language=&amp;quotJavaScript&amp;quot&amp;gt
<!--
var x =&amp;quot60&amp;quot;
var y=20;
alert(x+y);
//--&amp;gt
</script>
```

А. &quot60&quot+20
Б. 6020
В. 60
Г. 20
Д. 80

106. JavaScript тілінде x % 3 деген жазу нені білдіреді?
А. x-ті 3-ке бөлгендегі қалдықты табу
Б. x-тің 3 пайызын табу
В. x-ті 3-ке бөлу
Г. x-тің өсу қадамы 3-ке тең
Д. x-тің кему қадамы 3-ке тең
107. Java Script тілінде коментарий енгізу үшін қандай символдар қолданылады?
А. /* … */
Б. // … //
В. (* *)
Г. { ... }
Д. [...]
108. JavaScript тілінде var x=2; var y= x + +; тіркесі берілген. Осылардың нәтижесінде x және y мәндері нешеге тең болады?
А. y = 2, x = 3
Б. y = 3, x = 3
В. y = 2, x = 2
Г. y = 1, x = 2
Д. y = 3, x = 2
109. JavaScript.Төмендегілердің нәтижесінде x және y мәндері нешеге тең болады?

```
var x=2;
var y= + + x;
```

А. y = 3, x = 3
Б. y = 2, x = 2
В. y = 2, x = 3
Г. y = 1, x = 2
Д. y = 3, x = 2
110. Төмендегілердің нәтижесінде x және y мәндері нешеге тең болады?

```
var x=2;
var y= x - - ;
```

А. y = 2, x = 1
Б. y = 3, x = 3
В. y = 2, x = 2
Г. y = 1, x = 2
Д. y = 3, x = 2
111. JavaScript.Төмендегілердің нәтижесінде x және y мәндері нешеге тең болады?

```
var x=2;
var y= - -x;
```

А. y = 1, x = 1
Б. y = 3, x = 3
В. y = 2, x = 2

Г. y = 2, x = 3
Д. y = 3, x = 2

112. JavaScript. Мына жазу нені білдіреді: x + = 11;
x-тің өсу қадамы 11-ге тең;
А. x = x + 11;
Б. x = x + 1;
В. x = x + 1 + 2 + 3 +…+11;
Г. x = 11 + 10 + 9 +…+x;

113. JavaScript. Мынадай фрагмент берілген:
var x=1;
var y;
y = (x + = 2) +1;
Сонда x пен y айнымалылардың мәндері неге тең?
А. x = 3, y = 4
Б. x = 2, y = 3
В. x = 3, y = 3
Г. x = 4, y = 4
Д. x = 4, y = 3

114. JavaScript тілінде программаға мәлімет енгізетін функция:
А. prompt()
Б. alert()
В. concat()
Г. slice()
Д. var()

115. JavaScript тілінде prompt() функциясының неше аргументі бар?
А. 2
Б. 1
В. 3
Г. 4
Д. 5

116. Төменде көрсетілген таңбалардың қайсысы JavaScript тілінде логикалық ЖӘНЕ операциясын белгілейді?
А. &&
Б. ||
В. ??
Г. //
Д. !

117. Төменде көрсетілген таңбалардың қайсысы JavaScript тілінде логикалық НЕМЕСЕ операциясын белгілейді?
А. ||
Б. ??
В. //
Г. &&
Д. !

118. Төменде көрсетілген белгілердің қайсысы JavaScript тілінде логикалық ТЕРІСТЕУ операциясын белгілейді?
А. !
Б. ||
В. ??
Г. //
Д. &&

JavaScript тілінде жүйелі жақшаға {} алынған командалар тізбегі қалай аталады?
А. блок
Б. тізбег
В. функция
Г. тәсіл
Д. шарт

119. JavaScript тілінде экранға екі батырмасы (ОК және ОТМЕНА) бар терезе шығаратын функция:
А. confirm()
Б. alert()
В. prompt()
Г. slice()
Д. var()

120. JavaScript тілінде confirm() функциясы не үшін қолданылады?
А. ОК және ОТМЕНА батырмалары бар терезе шығарылады
Б. Бір ғана ОК батырмасы бар терезе шығарады
В. Ол ақпаратты енгізу үшін қолданылатын функция
Г. Көлемді мәтінді экранға шығаратын функция
Д. Айнымалыларды сипаттау үшін қолданылады

121. JavaScript тіліндегі prompt() функциясы қандай қызмет атқарады?
А. Ол ақпаратты енгізу үшін қолданылатын функция
Б. Бір ғана ОК батырмасы бар терезе шығарады
В. ОК және ОТМЕНА батырмалары бар терезе шығарылады
Г. Көлемді мәтінді экранға шығаратын функция
Д. Айнымалыларды сипаттау үшін қолданылады

122. JavaScript тіліндегі for циклінің жалпы жазылу түрі:
А. for (цикл басы; шарт; қадамы) команда;
Б. for (шарт) команда;
В. for (цикл басы; шарт) команда;
Г. for (шарт; қадамы) команда;
Д. for (шарт) команда;

123. JavaScript тіліндегі while циклінің жалпы жазылу түрі:
А. while (шарт) команда;
Б. while (цикл басы; шарт; қадамы)
В. while (шарт)
Г. while (цикл басы; шарт)

Д. while (шарт; қадамы)

124. JavaScript тіліндегі экранға шығарылатын мәндердің барлығын бір терезеде орналастыру үшін alert() функциясының орнына қандай команда жазу керек?

А. document.write()
Б. alert.write()
В. prompt()
Г. concat()
Д. slice()

125. PI константасы үшін JavaScript тілінде қандай қасиет қолданылады?

А. Math.PI
Б. Math.E
В. Math.LN10
Г. Math.SQRT314_10
Д. Math(14)

126. e = 71828 константасы үшін JavaScript тілінде қандай қасиет қолданылады?

А. Math.E
Б. Math.PI
В. Math.exp
Г. Math.SQRT2_71
Д. Math(71828)

127. 10 санының натуралдық логарифмі үшін JavaScript тілінде қандай қасиет қолданылады?

А. Math.LN10
Б. Math.PI
В. Math.E
Г. Math.SQRT314_10
Д. Math.sin(14)

128. Аргументтің квадрат түбірі үшін JavaScript тілінде қандай тәсіл қолданылады?

А. Math.sqrt()
Б. Math.sqr()
В. Math.E()
Г. Math.Ln()
Д. Math.sin()

129. x санының синусын табу үшін JavaScript тілінде қандай тәсіл қолданылады?

А. Math.sin(x)
Б. Math.asin()
В. Math.sinus()
Г. Math(x)
Д. Math.SQRT(x)

130. ex табу үшін JavaScript тілінде қандай тәсіл қолданылады?

А. Math. exp(x)
Б. Math.e(x)
В. Math.E(x)
Г. Math.ex()
Д. Math.exp

131. Аргументке ең жақын одан үлкен немесе соған тең бүтін санды табу үшін JavaScript тілінде қандай тәсіл қолданылады?
А. Math.ceil(arg)
Б. Math&gt=(arg)
В. Math.floor(arg)
Г. Math.exp(arg)
Д. Math.sin(arg)

132. Аргументке ең жақын одан кіші немесе соған тең санды табу үшін JavaScript тілінде қандай тәсіл қолданылады?
А. Math.floor(arg)
Б. Math<=(arg)
В. Math.ceil(arg)
Г. Math.exp(arg)
Д. Math.sin(arg)

133. Аргументті оған ең жақын бүтін санға дейін дөңгелектеу үшін JavaScript тілінде қандай тәсіл қолданылады?
А. Math.round(arg)
Б. Math.floor()
В. Math.ceil(arg)
Г. Math.exp(arg)
Д. Math.cel(arg)

134. x^n мәнін табу үшін JavaScript тілінде қандай тәсіл қолданылады?
А. Math.pow(x,n)
Б. Math.round(x,n)
В. Math.floor(x,n)
Г. Math.exp(x,n)
Д. Math.ceil(x,n)

135. JavaScript терминологиясында тәсілдер мен қасиеттер жиыны – бұл :
А. Объект
Б. Функция
В. Аргумент
Г. Инкапсуляция
Д. Экземпляр

136. JavaScript. Date объектісі не үшін қолданылады?
А. Күн-ай мерзімімен және уақытпен жұмыс істеу үшін
Б. Квадрат жасау үшін
В. Тіктөртбұрыш жасау үшін
Г. Жиыммен жұмыс істеу үшін
Д. Мәліметтерді өңдеу үшін

137. JavaScript. Array объектісі не үшін қолданылады?
А. Жиыммен жұмыс істеу үшін
Б. Квадрат жасау үшін
В. Тіктөртбұрыш жасау үшін
Г. Күн-ай мерзімімен және уақытпен жұмыс істеу үшін
Д. Мәліметтерді өңдеу үшін
138. JavaScript. Мынадай программалық код қандай нәтиже береді?
```
var d = new Date(2005,1,2);
var y = d.getYear();
alert(y);
```
А. 2005
Б. 2005/1/2
В. 05/1/2
Г. жыл = = 2005
Д. 2006
139. JavaScript. Мынадай программалық код бізге қандай нәтиже береді?
```
var d = new Date();
d.setMonth(4);
alert(d.getMonth());
```
А. 4
Б. Апрель
В. Сәуір
Г. Май
Д. 5
140. JavaScript. setMonth() тәсілі не береді?
А. Ай нөмірін тағайындайды
Б. Ай нөмірін береді
В. Апта күнін береді
Г. Ай атын береді
Д. Апта күнін тағайындайды
141. JavaScript. Sort, concat, reverse, join, slice бұл қай объектінің тәсілдері?
А. Array
Б. Date
В. Rectangle
Г. Kvadrat
Д. String
142. JavaScript. Sort тәсілінің атқаратын қызметі:
А. Жиым элементтері өсу немесе лексографикалық тәртіпте орналастырады
Б. Ол бастапқы жиымды өзгертпей, жиым элементтерін &quotразделитель&quot арқылы бөліп жібереді
В. Жиым элементтерін оның 1-элементі соңғысы болатындай етіп кері бағытта орын ауыстырып береді
Г. Бұрынғы жиымға array жиымы қосылған жаңа жиым жасап жібереді

Д. Бастапқы жиымнан бір позициядан бастап екінші позицияға дейінгі элементтерден тұратын жаңа жиым жасап жібереді

143. JavaScript тіліндегі reverse тәсілінің атқаратын қызметі:

А. Жиым элементтерін оның 1-элементі соңғысы болатындай етіп кері бағытта орын ауыстырып береді

Б. Ол бастапқы жиымды өзгертпей, жиым элементтерін &quotразделитель&quot арқылы бөліп жібереді

В. Бұрынғы жиымға array жиымы қосылған жаңа жиым жасап жібереді

Г. Жиым элементтері өсу немесе лексографикалық тәртіпте орналастырады

Д. астапқы жиымнан бір позициядан бастап екінші позицияға дейінгі элементтерден тұратын жаңа жиым жасап жібереді

144. JavaScript тіліндегі slice (ind1, ind2) тәсілінің атқаратын қызметі:

А. Бастапқы жиымдағы ind1 позициясынан бастап ind2-1 позициясына дейінгі элементтерден тұратын жаңа жиым жасап жібереді

Б. Ол бастапқы жиымды өзгертпей, жиым элементтерін &quotажыратқыш&quot арқылы бөліп жібереді

В. Жиым элементтерін оның 1-элементі соңғысы болатындай етіп кері бағытта орын ауыстырып береді

Г. Бұрынғы жиымға array жиымы қосылған жаңа жиым жасап жібереді

Д. Жиым элементтері өсу немесе лексографикалық тәртіпте орналастырады

145. JavaScript тіліндегі join (ажыратқыш) тәсілінің атқаратын қызметі:

А. Ол бастапқы жиымды өзгертпей, жиым элементтерін &quotажыратқыш&quot арқылы бөліп орналастырады

Б. Жиым элементтерін оның 1-элементі соңғысы болатындай етіп кері бағытта орын ауыстырып береді

В. Бұрынғы жиымға array жиымы қосылған жаңа жиым жасап жібереді

Г. Жиым элементтері өсу немесе лексографикалық тәртіпте орналастырады

Д. Бастапқы жиымдағы бірінші позициядан бастап екінші позицияға дейінгі элементтерден тұратын жаңа жиым жасап жібереді

146. JavaScript тіліндегі concat (array) тәсілінің атқаратын қызметі:

А. Бұрынғы жиымға array жиымы қосылған жаңа жиым жасап жібереді

Б. Ол бастапқы жиымды өзгертпей, жиым элементтерін &quotразделитель&quot арқылы бөліп жібереді

В. Жиым элементтерін оның 1-элементі соңғысы болатындай етіп кері бағытта орын ауыстырып береді

Г. Жиым элементтері өсу немесе лексографикалық тәртіпте орналастырады

Д. Бастапқы жиымнан бір позициядан бастап екінші позицияға дейінгі элементтерден тұратын жаңа жиым жасап жібереді

147. JavaScript тіліндегі length тәсілінің атқаратын қызметі:

А. Жиым ұзындығын береді
Б. Ол бастапқы жиымды өзгертпей, жиым элементтерін &quotразделитель&quot арқылы бөліп жібереді
В. Жиым элементтерін оның 1-элементі соңғысы болатындай етіп кері бағытта орын ауыстырып береді
Г. Жиым элементтері өсу немесе лексографикалық тәртіпте орналастырады
Д. Бастапқы жиымнан бір позициядан бастап екінші позицияға дейінгі элементтерден тұратын жаңа жиым жасап жібереді

148. JavaScript. Бастапқы жиымды өзгертпей, жиым элементтерін &quotажыратқыш&quot (разделитель) арқылы бөліп жіберетін тәсіл:
А. join
Б. slice
В. reverse
Г. concat
Д. sort

149. JavaScript. Жиым элементтерін оның 1-элементі соңғысы болатындай етіп кері бағытта орын ауыстырып беретін тәсіл:
А. reverse
Б. slice
В. join
Г. concat
Д. sort

150. JavaScript. Жиым элементтері өсу немесе лексографикалық тәртіпте орналастыратын тәсіл:
А. Sort
Б. slice
В. join
Г. reverse
Д. concat

151. JavaScript. Бастапқы жиымның көрсетілген алғашқы позициясынан бастап екінші көрсетілген позициясына дейінгі элементтерден тұратын жаңа жиым жасап беретін тәсіл:
А. slice
Б. join
В. reverse
Г. concat
Д. sort

152. JavaScript. Бұрынғы жиымға *array* жиымы қосылған жаңа жиым жасап жіберетін тәсіл:
А. concat
Б. slice
В. join

Г. reverse
Д. sort

153. JavaScript. Жиым ұзындығын беретін тәсіл:
А. length
Б. slice
В. reverse
Г. concat
Д. sort

154. JavaScript.Бұрынғы терезе ашық тұрғанда жаңа терезе ашу үшін қандай тәсіл қолданылады?
А. Open()
Б. Close()
В. Clear()
Г. Write(str)
Д. Writeln(str)

155. JavaScript. Браузер терезесіндегі жазбаны жабу үшін қандай тәсіл қолданылады?
А. Close()
Б. Open()
В. Clear()
Г. Write(str)
Д. Writeln(str)

156. JavaScript. Төмендегілердің қайсысы дұрыс тұжырым?
А. Alert функциясы ақпаратты экранға шығару үшін, ал ақпаратты енгізу үшін prompt функциясы керек.
Б. Prompt функциясы ақпаратты экранға шығару үшін, ал ақпаратты енгізу үшін alert функциясы керек.
В. confirm функциясы ақпаратты экранға шығару үшін, ал ақпаратты енгізу үшін prompt және alert функциялары керек.
Г. Alert функциясы ақпаратты экранға шығару, енгізу үшін қолданылады.
Д. Prompt функциясы ақпаратты тек экранға шығару үшін қолданылады.

157. JavaScript. Идентификатор дегеніміз не?
А. Әріптен басталып жазылатын латын әріптер мен араб цифрларының тізбегі.
Б. Цифрдан басталып жазылатын сөйлем.
В. Ақпаратты шығарушы терезе
Г. Құрамына арифметикалық және тіркестік (строковые) өрнектер кіретін тізбек
Д. Кез келген тұтынушы бірнеше жолдан тұратын мәтіндік ақпарат енгізе алатын өрісті анықтайды

158. JavaScript. x%=9 нені білдіреді?
А. x = x%9;
Б. x = x/9;

В. x = 9%x;
Г. x%x=9;
Д. 9= x%;

159. JavaScript. 9 % 3 нәтижесі неге тең?
А. 0
Б. 3
В. 9
Г. 1
Д. 3,33

160. JavaScript. Шартты команданың жалпы жазылу түрі :
А. if (шарт) команда1; else команда2;
Б. if (шарт): команда1; else команда2;
В. if (шарт) команда1 else команда2;
Г. if (шарт); команда1;else команда2;
Д. if (шарт); команда1 else команда2;

161. JavaScript. 3 санына 1-ді қосу үшін қалай жазу керек?
А. *x=3+1;
Б. x=”3”+”1”;
В. x=3+”1”;
Г. x=”3”+1;
Д. x=’3’+’1’;

162. JavaScript-те:

```
var x = 1;
var y;
if (x == 1) y = 10;
else  y = 20;
x += y;
```

осы кодтардан соң:
А. x 11-ге тең, y 10-ға тең
Б. x және y 11-ге тең
В. x 1-ге тең y 20-ға тең
Г. x және y 10-ға тең
Д. x 1-ге тең y 11-ге тең

163. JavaScript:

```
x = 1;
if (x != 1)  y = 10;
else y = 22;
x += y;
```

осы кодтардан соң:
А. x 23-ке тең y 22-ге тең
Б. x және y 23-ге тең
В. x 1-ге тең y 10-ға тең
Г. x және y 10-ға тең
Д. x 11-ге тең y 10-ға тең

164. JavaScript:

```
x = 1;
y = 10;
if  (x = = 1) y += 10;
x += y
```

осы кодтардан соң:

А. х 21-ге тең, у 20-ға тең
Б. х және у 21-ге тең
В. х 23-ке тең, у 22-ге тең
Г. х және у 10-ға тең
Д. х 11-ге тең, у 20-ға тең

165. JavaScript:

```
x = 1;
y = 10;
if(x != 1) y += 10;
x += y;
```

осы кодтардан соң:

А. х 11-ге тең, у 10-ға тең
Б. х және у 21-ге тең
В. х 11-ге тең, у 20-ға тең
Г. х 15-ке тең, у 22-ге тең
Д. х және у 10-ға тең

166. JavaScript: switch(өрнек) не үшін қажет?

А. өрнек мәні case сөзінен кейінгі қайсы мәнге тең болса, сол жол орындалады
Б. өрнек мәні case сөзінен кейінгі бірде бір мәнге сәйкес келмеген кезде орындалады (ол болмауы да мүмкін)
В. өрнек мәні case сөзінен кейінгі екі мәнге сәйкес келген кезде орындалады (ол міндетті түрде керек)
Г. өрнек мәні case сөзінен кейінгі барлық мәнге сәйкес келген кезде орындалады (ол міндетті түрде керек)
Д. өрнек мәні switch сөзінен кейінгі екі мәнге сәйкес келмеген кезде орындалады

167. JavaScript: continue операторы не үшін қажет?

А. циклдың онан кейінгі барлық командаларын аттап өтіп, цикл параметрінің келесі мәніне көшіреді.
Б. Циклдағы алғашқы команданы орындайды
В. Циклдағы соңғы команданы орындайды
Г. Кезекті тексеру кезінде шарт жалған болған кезде, циклдің жұмысын аяқтайды.
Д. жалпы цикл орындалуын аяқтап, одан кейінгі келесі командаларға көшіреді.

168. JavaScript:

```
var x = 20;
```

if (20 - x) x ++;
командалары орындалған соң:
А. х мәні өзгермейді
Б. х мәні 21-ге тең
В. х мәні бірге тең
Г. х мәні 2-ге артады
Д. х мәні 22-ге тең

169. JavaScript:
var x = 20;
if(2 * x) ++x;
осы кодтардан соң:
А. х мәні 21-ге тең
Б. х мәні 22-ге тең
В. х мәні бірге тең
Г. х мәні 0-ге тең
Д. х мәні өзгермейді

170. JavaScript:
var x = &quot "
if(x) x += &quotкітап"
осы командалар орындалған соң:
А. х мәні &quot &quot тең
Б. х мәні &quotкітап&quot сөзіне тең
В. х мәні 1-ге артады
Г. х мәні 0-ге тең
Д. х мәні өзгермейді

171. JavaScript:
var x = &quot"
if(x+&quot?&quot) x += &quotкітап"
командалар орындалған соң:
А. х мәні &quotкітап&quot сөзіне тең
Б. 1 х мәні өзгермейді.
В. х мәні 0-ге тең
Г. х мәні 1-ге артады
Д. х мәні &quot &quot тең

172. JavaScript:
var x = true;
var y = 2; if (!x) y ++;
команда орындалған соң:
А. у мәні 2-ге тең
Б. у мәні 3-ке тең
В. у мәні 1-ге артады
Г. у мәні &quotFalse&quot сөзіне тең
Д. у мәні өзгермейді

173. JavaScript:

var x = true;
var y = 2; if (x || !x) y ++;
командалар орындалған соң:

А. у мәні 3-ке тең
Б. у мәні 0-ге тең
В. у мәні 5-ке тең
Г. у мәні 1-ге артады
Д. у мәні &quotFalse&quot сөзіне тең

174. JavaScript:

```
var x = true;
var y = 2; if (x && !x) y ++;
```

команда орындалған соң:

А. у мәні 2-ге тең
Б. у мәні &quotFalse&quot сөзіне тең
В. у мәні 3-ке тең
Г. у мәні &quottrue&quot сөзіне тең
Д. у мәні 1-ге тең

175. JavaScript:

```
x = &quotБағa=&quot+5;
```

x нәтижесі:

А. &quotБағa=5&quot тіркесі
Б. &quotБағa=+5&quot тіркесі
В. &quotБағa+=5&quot тіркесі
Г. &quotБағa=&quot+5 бүтін саны
Д. +5 бүтін саны

176. JavaScript: Нәтижесі `false` болатын команданы көрсет:

А. !&quotқасқыр&quot && 25
Б. &quotқасқыр&quot && 25
В. !&quot&quot || 22 - 22
Г. !&quotқасқыр&quot || 25
Д. !&quotқасқыр`&quot`

177. JavaScript: Math.floor(num) стандартты функциясы:

А. аргументке тең не одан кіші бүтін сан мәнін береді.
Б. Циклдағы алғашқы команданы орындайды
В. Кезекті тексеру кезінде шарт жалған болған кезде, циклдің жұмысын аяқтайды.
Г. стандартты функциясы [0,1] аралығынан кез келген кездейсоқ сан береді.
Д. жалпы цикл орындалуын аяқтап, одан кейінгі келесі командаларға көшіреді.

178. JavaScript: x^{10} – санның дәрежесін табу:

А. Math.pow(x,10);
Б. pow(x,10);

В. math.pow(x,10);
Г. Math.sin(x);
Д. Math.abs(x);

179. JavaScript: Қай нұсқада аргументтің синусын есептеу дұрыс көрсетілген?

А.
```
var x = Math.PI/4;
var y
y = Math.sin(x);
alert(y);
```
Б.
```
var x = math.PI/4;
var y
y = math.sin(x);
alert(y);
```
В.
```
var x = Math.pi/4;
var y
y = Math.sin(x);
alert(y);
```
Г.
```
var x = math.pI/4;
var y
y = Math.sin(x);
alert(y);
```
Д.
```
var x = Math.PI/4;
var y
y = math.sin(x);
alert(y);
```

180. JavaScript: Аргументтің – х квадраттық түбірі :

А. Math.sqrt(x);
Б. Math.round(x);
В. Math.ceil()
Г. Math.sin(x);
Д. Math.abs(x);

181. JavaScript: Екі аргументтің – х,у үлкенін анықтау коды:

А.
```
var x = 3;
var y = Math.max(x,10);
alert(y);
```
Б.
```
var x = 3;
var y=max(x,y)
```
В.
```
var x = 3;
y=max(x,y)
```
Г.
```
var x = 3;
var y = Math.max(y,10);
alert(y);
```
Д.
```
var x = 3;
```

```
var y = Math.min(x,10);
alert(y);
```

182. JavaScript. Объект дегеніміз:
   А. бұл мәліметтер мен функциялар жиынынан тұратын бірыңғай конструкция немесе, JavaScript терминологиясында, қасиеттер мен тәсілдер жиыны.
   Б. &quotқара жәшік&quot ретінде қарастырылатын объектінің ішкі құрылымын жасыру деген сөз.
   В. программалау тілінде объектінің ішкі айнымалылары мен функцияларын сипаттау.
   Г. элементтердің реттелген жиыны.
   Д. бұлар пайдалануға болатын объектінің айнымалылары мен функциялары.
183. JavaScript. Инкапсуляция дегеніміз:
   А. &quotқара жәшік&quot ретінде қарастырылатын объектінің ішкі құрылымын жасыру деген сөз.
   Б. бұл мәліметтер мен функциялар жиынынан тұратын бірыңғай конструкция немесе, JavaScript терминологиясында, қасиеттер мен тәсілдер жиыны.
   В. программалау тілінде объектінің ішкі айнымалылары мен функцияларын сипаттау.
   Г. элементтердің реттелген жиыны.
   Д. бұлар пайдалануға болатын объектінің айнымалылары мен функциялары.
184. JavaScript. Объект интерфейсі дегеніміз:
   А. бұлар пайдалануға болатын объектінің айнымалылары мен функциялары.
   Б. бұл мәліметтер мен функциялар жиынынан тұратын бірыңғай конструкция немесе, JavaScript терминологиясында, қасиеттер мен тәсілдер жиыны.
   В. &quotқара жәшік&quot ретінде қарастырылатын объектінің ішкі құрылымын жасыру деген сөз.
   Г. программалау тілінде объектінің ішкі айнымалылары мен функцияларын сипаттау.
   Д. элементтердің реттелген жиыны.
185. JavaScript. Тәсілдің қай түрі жыл нөмірін береді?
   А. getYear()
   Б. getDate()
   В. setYear()
   Г. setDate()
   Д. getDay()
186. JavaScript. Тәсілдің қай түрі жыл нөмірін тағайындайды?
   А. setYear()
   Б. getDate()

В. getYear()
Г. setDate()
Д. getDay()

187. JavaScript. Тәсілдің қай түрі айды енгізеді?
А. setMonth()
Б. getMonth()
В. setYear()
Г. getYear()
Д. setDate()

188. JavaScript. Тәсілдің қай түрі ай нөмірі мәнін береді?
А. getMonth()
Б. setYear()
В. setMonth()
Г. setDate()
Д. getDay()

189. JavaScript. Тәсілдің қай түрі апта күнін береді?
А. getDay()
Б. getDate()
В. setYear()
Г. getYear()
Д. setDate()

190. JavaScript. Тәсілдің қай түрі уақыт минутын енгізеді?
А. setMinutes()
Б. setYear()
В. getYear()
Г. getMinutes()
Д. getDay()

191. JavaScript. concat(array) тәсілі не үшін керек?
А. Бұрынғы массивке array массиві қосылған жаңа массив жасап береді
Б. Бөлініп орналасқан массив элементтері жолын береді.
В. Массив элементтерін оның 1-элементі соңғысы болатындай етіп кері бағытта орын ауыстырып береді.
Г. Массив ұзындығын береді.
Д. Массивті сұрыптайды.

192. JavaScript. Объект жасау кезіндегі мұралаудың түрлері:
А. Статикалық және динамикалық
Б. Статикалық және физикалық
В. Логикалық және физикалық
Г. Статикалық және логикалық
Д. Динамикалық және физикалық

193. JavaScript. Динамикалық мұралау :
А. Мұрагер жасау кезінде оның ата тегімен байланысын үзбеу.
Б. Мұрагер жасалу кезінде оның ішкі құрамына тегінің барлық қасиеттері мен тәсілдері көшіріледе, содан кейін туыстық байланыс жойылады.

В. Мұрагер жасалу кезінде алдымен туыстық байланыс жойылады да, содан кейін оның ішкі құрамына тегінің барлық қасиеттері мен тәсілдері көшіріледі.
Г. Мұрагер жасалу кезінде алдымен туыстық байланыс жойылады да, содан кейін оның ішкі құрамына тегінің барлық қасиеттері мен тәсілдері көшірілмейді.
Д. Мұрагер жасау кезінде оның ата тегімен байланысын үзбеу және қасиеттерін қайталамау.

194. JavaScript. reverse()тәсілі не үшін керек :
А. Массив элементтерін оның 1-элементі соңғысы болатындай етіп кері бағытта орын ауыстырып береді.
Б. Бөлініп орналасқан массив элементтері жолын береді.
В. Бұрынғы массивке array массиві қосылған жаңа массив жасап береді
Г. Массив ұзындығын береді.
Д. Массивті сұрыптайды.

195. JavaScript. join(ажыратқыш) тәсілі не үшін қажет?
А. Ажыратқыш арқылы бөлініп орналасқан массив элементтері жолын береді.
Б. Бұрынғы массивке array массиві қосылған жаңа массив жасап береді
В. Массив элементтерін оның 1-элементі соңғысы болатындай етіп кері бағытта орын ауыстырып береді.
Г. Массив ұзындығын береді.
Д. Массивті сұрыптайды.

196. JavaScript. sort(function) не sort() тәсілдері нені береді?
А. Массив элементтерін сұрыптайды.
Б. Бөлініп орналасқан массив элементтері жолын береді.
В. Бұрынғы массивке array массиві қосылған жаңа массив жасап береді
Г. Массив элементтерін оның 1-элементі соңғысы болатындай етіп кері бағытта орын ауыстырып береді.
Д. Массив ұзындығын береді.

197. JavaScript. length тәсілі не үшін керек:
А. Массив ұзындығын береді.
Б. Бөлініп орналасқан массив элементтері жолын береді.
В. Бұрынғы массивке array массиві қосылған жаңа массив жасап береді
Г. Массив элементтерін оның 1-элементі соңғысы болатындай етіп кері бағытта орын ауыстырып береді.
Д. Массивті сұрыптайды.

198. JavaScript. Ашылатын терезеге нұсқауыш:
А. айнымалы
Б. сөз тіркесі
В. терезе_қасиеттері
Г. файл
Д. терезе_аты

199. JavaScript. Терезе ашу кезінде resizable параметрі не істейді?
А. Терезе көлемін өзгертуді көрсету үшін қажет
Б. Меню өрісі бейнелетінін көрсету үшін қажет
В. Статус жолағы бейнелетінін көрсету үшін қажет
Г. Батырмалар (саймандар) тақтасы бейнелетінін көрсету үшін қажет
Д. Адрес енгізу өрісі бейнелетінін көрсету үшін қажет
200. JavaScript. Терезе ашу кезінде menubar параметрі не істейді?
А. Меню өрісі бейнелетінін көрсету үшін қажет
Б. Терезе көлемін өзгертуді көрсету үшін қажет
В. Статус жолағы бейнелетінін көрсету үшін қажет
Г. Батырмалар (саймандар) тақтасы бейнелетінін көрсету үшін қажет
Д. Адрес енгізу өрісі бейнелетінін көрсету үшін қажет
201. JavaScript. Терезенің toolbar параметрі не істейді?
А. Батырмалар (саймандар) тақтасы бейнелетінін көрсету үшін қажет
Б. Терезе көлемін өзгертуді көрсету үшін қажет
В. Меню өрісі бейнелетінін көрсету үшін қажет
Г. Статус жолағы бейнелетінін көрсету үшін қажет
Д. Адрес енгізу өрісі бейнелетінін көрсету үшін қажет
202. JavaScript. Терезе ашу кезінде scrollbars параметрі не істейді?
А. Айналдыру жолағының шығатынын/шықпайтынын көрсетеді
Б. Терезе көлемін өзгертуді көрсету үшін қажет
В. Меню өрісі бейнелетінін көрсету үшін қажет
Г. Статус жолағы бейнелетінін көрсету үшін қажет
Д. Адрес енгізу өрісі бейнелетінін көрсету үшін қажет
203. JavaScript. Формалардағы қай атрибут батырма бетіндегі жазуды береді?
А. value
Б. type
В. onclick
Г. input
Д. select
204. JavaScript. Тәсілдің қай түрі мәліметі бар информациялық терезе шығарады?
А. window.alert(хабарлама)
Б. window.open(файл, имя_окна, параметры_окна)
В. window.close (хабарлама)
Г. window.prompt(хабарлама)
Д. window.confirm(хабарлама)
205. JavaScript. document.write(str) тәсілінің сипаттамасы :
А. Құжаттың str мәні ретіндегі сөз тіркесін – мәтінді және HTML кодын жазады
Б. Браузер терезесіндегі жазбаны жабады
В. Браузер терезесіне жазу жазады. Бұрынғы терезе мәліметі өшіріледі
Г. Браузера терезесін тазалайды

Д. Каретканы қайтарумен аяқталатын (жаңа жолға көшу) мәтінді және HTML кодын жазады.

206. HTML-дан шығып, PHP режиміне өту тәсілі:
А. <?php echo("..."); ?>
Б. < echo("..."); >
В. <? echo("..."); >
Г. < echo("..."); ?>
Д. <! echo "..." !>;

207. HTML-дан шығып, PHP режиміне өту тәсілі:
А. <script language = "php" >
Б. <script language: "php" >
В. <language = "php" >
Г. <?script language = "php" ?>
Д. <script language: ?"php" ?>

208. PHP тілінде жол соңында комментарий жазу қай символдан басталады:
А. //
Б. */
В. /*
Г. (*
Д. {

209. PHP тілінде айнымалы аты қай символдан басталады?
А. $
Б. #
В. “
Г. *
Д. !

210. PHP тілінде $ символы не үшін қолданылады?
А. айнымалы аты осы символдан басталады
Б. осы символдардың ортасында сөз тіркесі типіндегі (строковый тип) мәліметтер жазылады
В. осы символдардың ортасында көпжолдық комментарийлер жазылады
Г. қалдықпен бөлу үшін қолданылады
Д. PHP тілінде әр-бір жол осы символдан бастап жазылады

211. Тек PHP тілінде жиымдармен жұмыс істеуге арналған цикл операторы:
А. foreach
Б. do... while
В. while
Г. for
Д. switch

212. PHP тілінде include операторы не үшін қолданылады?

А. осы оператор кездескен сайын, көрсетілген файлдың құрамындағы код пайдаланылады
Б. екі батырма (ОК және ОТМЕНА) бар терезе шығарады
В. ОК батырмасы бар терезені шығарады
Г. шартты операторларды біріктіру үшін қолданылады
Д. нәтижені экранға шығару операторы

213. PHP тілінде require операторы не үшін қолданылады?
А. осы оператор кездескен сайын, көрсетілген файлдың құрамындағы код пайдаланылады
Б. екі батырма (ОК және ОТМЕНА) бар терезе шығарады
В. ОК батырмасы бар терезені шығарады
Г. шартты операторларды біріктіру үшін қолданылады
Д. нәтижені экранға шығару операторы

214. PHP тілінде foreach операторы не үшін қолданылады?
А. жиымдармен жұмыс істеуге арналған цикл операторы
Б. ОК батырмасы бар терезені шығарады
В. осы оператор кездескен сайын, көрсетілген файлдың құрамындағы кодты пайдалануға болады
Г. шартты операторларды біріктіру үшін қолданылады
Д. нәтижені экранға шығару операторы

215. Сервер дегеніміз – бұл ...
А. әр түрлі процестерді басқаратын программалар жиыны
Б. клиенттермен жұмыс істейтін программа
В. сұраныстармен жұмыс істеуге арналған программа
Г. әрбір компьютер үшін арнайы жазылған программа
Д. мәліметтерді жөнелту үшін арнайы жазылған программа

216. PHP тілінде төмендегі программаның нәтижесі қандай:

```
<?php
$first=' Text ';
$second=$first;
$first = ' New text ';
echo "Переменная с именем first равна $first <br>";
echo 'Переменная с именем second равна $second';
?>
```

А. Переменная с именем first равна New text
Переменная с именем second равна $second
Б. Переменная с именем first равна $first
Переменная с именем second равна $second
В. Переменная с именем first равна Text
Переменная с именем second равна $first
Г. Переменная с именем second равна $first

Д. Переменная с именем first равна Text
Переменная с именем second равна Text

217. PHP тілінде қате жазылған айнымалы атауы:
А. $345
Б. $alfa
В. $alfa_romeo5_5
Г. $_alfa
Д. $alfa7889

218. PHP тілін кім алғаш рет шығарды?
А. Расмус Ледорф
Б. Зев Сураски
В. Никлаус Вирт
Г. Чарльз Бэббидж
Д. Билл Гейтс

219. Қай жылы PHP тілі алғаш рет шығарылды?
А. 1994
Б. 1993
В. 1992
Г. 1995
Д. 1996

220. PHP тіліндегі әрбір оператор қай символмен ақталады:
А. ;
Б. :
В. >
Г. ?>
Д. ”

221. PHP тіліндегі программада кездескен сайын оның құрамындағы файлдың кодын пайдалануға мүмкіндік беретін оператор:
А. include
Б. switch
В. foreach
Г. require
Д. while

222. PHP тілінде бірнеше жолдық комментарий жазу тәсілі:
А. /* ... */
Б. (*...*)
В. // ... //
Г. # ... #
Д. { ... }

223. PHP тілінде = = = тіркесі нені білдіреді:
А. айнымалылардың мәндері және типтері бірдей;
Б. айнымалылардың мәндері тең емес;
В. айнымалылар эквивалентті емес ;
Г. айнымалылардың мәндері тең
Д. айнымалылардың типтері әр түрлі.
224. PHP тіліндегі echo “ ” функциясы:
А. қостырнақша арасындағы мәтінді экранға шығарады
Б. мұндай функция жоқ
В. тырнақшаның орнына жақша қойсақ, соның ішіндегі мәтінді экранға шығарады
Г. экранға echo “ ” сөзін шығарады
Д. тырнақшаның орнына апостроф қойсақ, ол функцияға айналады
225. PHP кодының бастапқы және соңғы символдары:
А. бастапқы — <?php, соңғы — ? >
Б. бастапқы — <php, соңғы — >
В. бастапқы — <, соңғы — >
Г. бастапқы — ?php, соңғы — ?
Д. бастапқы — ?, соңғы — ?
226. PHP программалық файлдарының кеңейтілуі:
А. *.php
Б. *.hphp
В. *.phpl
Г. *.ptxt
Д. *.htm
227. PHP тіліндегі \n басқару тізбегі нені білдіреді?
А. жаңа жолға көшу
Б. каретканың қайтарылуы
В. нұсқауыш операторы
Г. көлденең табуляция
Д. тырнақша таңбасы
228. PHP тіліндегі \t басқару тізбегі нені білдіреді?
А. көлденең табуляция
Б. каретканың қайтарылуы
В. нұсқауыш операторы
Г. жаңа жолға көшу
Д. тырнақша таңбасы
229. PHP тіліндегі \” басқару тізбегі нені білдіреді?
А. тырнақша белгісі
Б. каретканың қайтарылуы
В. доллар белгісі
Г. көлденең табуляция
Д. жаңа жол

230. PHP тілінде heredoc синтаксисі қолданылған кезде қандай символдар жазылады?
А. <<< және идентификатор
Б. << және идентификатор
В. < және идентификатор
Г. <<& және идентификатор
Д. &< және идентификатор
231. PHP.Төмендегілердің қайсысы for операторы үшін дұрыс жазылған?
А. for (өрнек1; өрнек2; өрнек3) { орындалу_блогы };
Б. for (өрнек1; өрнек2; өрнек3) < орындалу_блогы >
В. for (өрнек1; өрнек2; өрнек3) /орындалу_блогы /
Г. for (өрнек1; өрнек2; өрнек3) (орындалу_блогы);
Д. for (өрнек1; өрнек2; өрнек3) [орындалу_блогы];
232. PHP.Төмендегілердің қайсысы foreach операторы үшін дұрыс жазылған?
А. foreach ($array as $value) { орындалу_блогы }
Б. foreach (array as value) { орындалу_блогы }
В. foreach ($value as $array) { орындалу_блогы }
Г. foreach ($array as $value):: { орындалу_блогы }
Д. foreach ($array as $value) begin орындалу_блогы end;
233. PHP.Төмендегілердің қайсысы foreach операторы үшін дұрыс жазылған?
А. foreach ($array as $key => $value) {орындалу_блогы}
Б. foreach (array as key => value) {орындалу_блогы}
В. foreach ($value as $key=> $array) {орындалу_блогы}
Г. foreach ($array as $key => $value):: {орындалу_блогы}
Д. foreach ($array as $key <>value) begin орындалу_блогы end;

234. PHP тілінде foreach ($array as $value) {орындалу_блогы} циклі қанша рет қайталанады?
А. $array жиымында неше элемент бар болса, сонша рет
Б. 0 рет
В. 1 рет
Г. $array жиымында неше элемент $value айнымалысына тең болса, сонша рет
Д. индекске қатысты болады
235. PHP. Төменде көрсетілген функциялардың қайсысы санның факториалын дұрыс есептейді?
А. function fact($n)
{$fact=1;
for ($i=1; $i<=$n; $i++)
$fact*=$i;
return $fact;}
Б. function fact(n)

{if (n==0) return 1;
else return fact = n * fact(n-1);}

В. function fact($n)
$fact=1;
while ($n>=1;)
[$fact*=$n; $n--;]
return $fact;

Г. function fact($n)
$fact=1;
while ($n>=1;)
[{$fact*=$n; $n--; }
return $fact;]

Д. ешқайсысы дұрыс есептемейді

236. PHP. Төменде көрсетілген функциялардың қайсысы санның факториалын дұрыс есептейді?

А. function fact($n)
{if ($n==0) return 1;
else return $fact = $n * fact($n-1);}

Б. function fact($n)
{$fact=0;
for ($i=1; $i<=$n; $i++)
$fact*=$i;
return $fact;}

В. function fact($n)
{$fact=1;
while ($n<1;)
{$fact*=$n; $n--; }
return $fact;}

Г. function fact($n)
$fact=1;
while ($n>=1;)
[$fact*=$n; $n--;]
return $fact;

Д. ешқайсысы дұрыс есептемейді

237. PHP. Нақты (бүтін емес) санды көрсетіңіз:

А. $a=7E-10;
Б. $a=0123;
В. $a=0X1A;
Г. $a=1234;
Д. дұрыс жауабы жоқ.

238. PHP. array_values() функциясының атқаратын қызметі:

А. жиымды қайтадан индекстейді;
Б. жиымды жиымға қосады;
В. жиымды программаға енгізеді;

Г. жиымды құрастырады;
Д. жиымды жояды.

239. PHP. $a=($b=5)+7; өрнегінің нәтижесі:
А. a=12; b=5;
Б. a=7; b=5;
В. a=12; b=12;
Г. a=7; b=7;
Д. a=5; b=7.

240. PHP программасындағы сан бүтін болып саналады, егер:
А. оның құрамында нүкте, e, E символдары жоқ болса
Б. оның құрамында нүкте, e, E символдары бар болса
В. оның құрамында әріптер болмаса
Г. оның құрамында нүкте болмаса
Д. оның құрамында “0” символы жоқ болса

241. PHP тілінде төмендегідей түрде берілген айнымалының сандық мәні:
$f = 1 + "bob-1.3e3"
А. бүтін сан болады
Б. 0-ге тең болады
В. сан болмайды
Г. кездейсоқ сан болады
Д. нақты сан болады

242. PHP тіліндегі жиым (массив) қай конструкция арқылы анықталады:
А. array ([key] => value, [key1] => value1, ...)
Б. new array ([key] => value, [key1] => value1, ...)
В. new array ([key] => value, [key1] => value1, ...) ();
Г. $array ([key] => value, [key1] => value1, ...)
Д. var array ([key] => value, [key1] => value1, ...)

243. PHP тілінде жиымдармен жұмыс істегенде оның кілті (индексі) ретінде нені қолдануға болады:
А. бүтін сандарды және сөз тіркесін
Б. объектілер мен субъектілерді
В. бүтін және нақты сандарды
Г. операторлар мен айнымалыларды
Д. түйіндік сөздерді

244. PHP тілінде жиым элементтерін жою үшін қандай функция қолданылады?
А. unset();
Б. print_r();
В. delete();
Г. remove();
Д. undo();

245. PHP тілінде NULL мәні нені білдіреді?
А. айнымалының мәні жоқ екенін
Б. айнымалының кездейсоқ санға тең екенін

В. айнымалы ның «_» тең екенін
Г. айнымалы 0-ге тең екенін
Д. айнымалы 1-ге тең екенін

246. PHP тілінде ресурс (resource) дегеніміз – бұл:
А. сыртқы ресурсқа сілтеме түріндегі айнымалы
Б. құрамында сыртқы ресурсы бар айнымалы
В. ресурстармен жұмыс істейтін функция
Г. типке сәйкес келмейді
Д. кез келген типті білдіреді

247. PHP тіліндегі switch құрылымының дұрыс жазылуы:
А. switch (өрнек не айнымалы) {
case 1-мән: 1_әрекеттер_блогы break;
case 2-мән: 2_әрекеттер_блогы break;
...
default: келісім_бойынша_ әрекеттер_блогы }
Б. switch (өрнек не айнымалы) {
case 1-мән: 1-әрекеттер блогы;
case 2-мән: 2-әрекеттер блогы;
...
default: келісім_бойынша_ әрекеттер_блогы }
В. switch (өрнек не айнымалы) {
case: 1-әрекеттер блогы; break;
case: 2-әрекеттер блогы; break;
...
default: келісім_бойынша_ әрекеттер_блогы }

Г. switch {
case 1-мән: 1_әрекеттер_блогы break;
case 2-мән: 2_әрекеттер_блогы break;
...
default: келісім_бойынша_ әрекеттер_блогы }
Д. switch (өрнек не айнымалы) {
case 1-мән: break;
case 2-мән: break;
...
default: n-мән }

248. PHP тілінде while циклінің дұрыс жазылуы:
А. while (өрнек) {орындалу_ блогы};
Б. while (өрнек) do {орындалу_ блогы};
В. while (өрнек); {орындалу_ блогы};
Г. while өрнек {орындалу_ блогы};
Д. while (өрнек) do орындалу_ блогы;

249. PHP тілінде do..while циклінің дұрыс жазылуы:
А. do {орындалу_блогы} while (өрнек);
Б. do {орындалу_блогы}; while (өрнек);
В. do {орындалу_блогы}; while өрнек;
Г. do (өрнек) {орындалу_блогы} while;
Д. do while (өрнек) {орындалу_блогы};

250. PHP.Төмендегілердің қайсысы for операторы үшін дұрыс жазылған?
А. for (өрнек1; өрнек2; өрнек3): орындалу_блогы endfor;
Б. for (өрнек1; өрнек2; өрнек3) орындалу_блогы endfor;
В. for (өрнек1; өрнек2; өрнек3); орындалу_блогы endfor;
Г. for (выражение1; выражение2; выражение3): орындалу_блогы end;
Д. for (выражение1; выражение2; выражение3) begin орындалу_блогы endfor;

Жауап кілттері

Сұрақ нөмірі	Дұрыс жауабы	Сұрақ нөмірі	Дұрыс жауабы
1.	А	126.	А
2.	А	127.	А
3.	А	128.	А
4.	А	129.	А
5.	А	130.	А
6.	А	131.	А
7.	А	132.	А
8.	А	133.	А
9.	А	134.	А
10.	А	135.	А
11.	А	136.	А
12.	А	137.	А
13.	А	138.	А
14.	А	139.	А
15.	А	140.	А
16.	А	141.	А
17.	А	142.	А
18.	А	143.	А
19.	А	144.	А
20.	А	145.	А
21.	А	146.	А
22.	А	147.	А
23.	А	148.	А
24.	А	149.	А

25.	A	150.	A
26.	A	151.	A
27.	A	152.	A
28.	A	153.	A
29.	A	154.	A
30.	A	155.	A
31.	A	156.	A
32.	A	157.	A
33.	A	158.	A
34.	A	159.	A
35.	A	160.	A
36.	A	161.	A
37.	A	162.	A
38.	A	163.	A
39.	A	164.	A
40.	A	165.	A
41.	A	166.	A
42.	A	167.	A
43.	A	168.	A
44.	A	169.	A
45.	A	170.	A
46.	A	171.	A
47.	A	172.	A
48.	A	173.	A
49.	A	174.	A
50.	A	175.	A
51.	A	176.	A
52.	A	177.	A
53.	A	178.	A
54.	A	179.	A
55.	A	180.	A
56.	A	181.	A
57.	A	182.	A
58.	A	183.	A
59.	A	184.	A
60.	A	185.	A
61.	A	186.	A
62.	A	187.	A
63.	A	188.	A
64.	A	189.	A
65.	A	190.	A
66.	A	191.	A
67.	A	192.	A

68.	A	193.	A
69.	A	194.	A
70.	A	195.	A
71.	A	196.	A
72.	A	197.	A
73.	A	198.	A
74.	A	199.	A
75.	A	200.	A
76.	A	201.	A
77.	A	202.	A
78.	A	203.	A
79.	A	204.	A
80.	A	205.	A
81.	A	206.	A
82.	A	207.	A
83.	A	208.	A
84.	A	209.	A
85.	A	210.	A
86.	A	211.	A
87.	A	212.	A
88.	A	213.	A
89.	A	214.	A
90.	A	215.	A
91.	A	216.	A
92.	A	217.	A
93.	A	218.	A
94.	A	219.	A
95.	A	220.	A
96.	A	221.	A
97.	A	222.	A
98.	A	223.	A
99.	A	224.	A
100.	A	225.	A
101.	A	226.	A
102.	A	227.	A
103.	A	228.	A
104.	A	229.	A
105.	A	230.	A
106.	A	231.	A
107.	A	232.	A
108.	A	233.	A
109.	A	234.	A
110.	A	235.	A

111.	A	236.	
112.	A	237.	
113.	A	238.	
114.	A	239.	
115.	A	240.	
116.	A	241.	
117.	A	242.	
118.	A	243.	
119.	A	244.	
120.	A	245.	
121.	A	246.	
122.	A	247.	
123.	A	248.	
124.	A	249.	
125.	A	250.	

Түсіндірме сөздер

IP адрес – желідегі компьютерді анықтайтын адрес (адрес 32 екеулік разрядтан құралған және TCP/IP желінің бойында қайталанбауы тиіс). Әдетте IP адрес сегіз екеулік разряды бар төрт октетке бөлінеді (бір байт); әрбір октет ондық санға түрленіп, нүктемен бөлінеді, мысалы 102.54.94.97.

HTTP гипермәтінді хаттама – гипермәтіннің блоктарын алмастыру процедурасын сипаттайтын интернет желісінің хаттамасы.

HTML тілі – гипермәтін технологиясын қолданатын құралдық бағдарламалық қамтамасыз ету.

FTP (File Transfer Protocol) - файлдарды беру хаттамасы. Желі бойынша файлдармен алмасуға мүмкіндік береді.

IP (Internet Protocol) – Internet хаттамасы, TCP/IP стегінің желілік хаттамасы, ол адрестік және бағыттық ақпаратпен қамтамасыз етеді.

Telnet – қашықтан қатынау. Абонент Internet желісінің кез-келген ЭЕМ-де өз компьютеріндегідей жұмыс істеуге мүмкіндік береді.

TCP (Transmission Control Protocol) – берілістерді басқару хаттамасы.

URL (Uniform Resource Locator) – ресурстардың әмбебап көрсеткішінің адресі.

Архитектура – желідегі құрамдастардың өзара байланысын, орындайтын қызметтерін, құрылымын, моделін анықтайтын концепция. Архитектура желінің логикалық, физикалық және бағдарламалық құрылымын, қызмет етуін, элементтердің әсерлесуінің сипаты мен топологиясын қамтиды.

Адаптер (adapter) – объектілерді біріктіру мақсатындағы енгізу және шығу параметрлерін келісімдеу программасы немесе құрылғысы.

Адрес (adress) – деректердің шығу немесе бару пунктін кодталған белгісі.

Аналогтық сигнал (analog signal) – уақыт бойынша шамасы өзгеретін сигнал. Аналогтық сигнал деректерді уақыт бойынша үзбей өзгерту арқылы береді.

Аналогтық-дискреттік түрлену (analog-to-digital conversion) – аналогтық сигналдың дискретті сигналға айналу үрдісі.

Архитектура – желідегі құрамдастардың өзара байланысын, орындайтын қызметтерін, құрылымын, моделін анықтайтын концепция. Архитектура желінің логикалық, физикалық және бағдарламалық құрылымын, қызмет етуін, элементтердің әсерлесуінің сипаты мен топологиясын қамтиды.

Арна (link) – деректерді беру ортасы немесе жолы.

Ауқымды есептеуіш жүйе, АЕЖ (Wide Area Network, WAN) – алыстан әсер ету байланысының құралдарын қолданатын компьютерлер желісі.

Аттардың домендік жүйесі (DNS – Domain Name System) – пайдаланушыға түсінікті, IP адрестері мен аттарын салыстыруға қажетті

белгілердің жүйесі, Internet желісінде қолданылады. DNS жүйесі кейде DNS қызметі деп те аталады.

Ақпараттық желі (information network) – деректерді өңдеу, сақтау және беруге арналған желі.

Әкімшілік жүйе (manegment system) – желіні немесе оның бөлімін басқаруды қамтамасыз ететін жүйе.

Бөлінген сервер (dedicated server) – тек сервер түрінде қызмет көрсететін желілік сервер.

Біррангілі архитектура (peer-to-peer architecture) –қорларды ұсынатын және қолдана алатын әрбір абоненттік жүйесі бар ақпараттық желінің концепциясы.

Виртуалды желі – сипаттамалары негізінен оның бағдарламалық қамтамасыз етілуімен анықталатын желі.

Құпия түрде қосылу – аты мен құпия сөзді көрсетпей, компьютердің есептік жазбалары бойынша компьютердің қорына қашықтан қатынауға рұқсат беретін функция.

Гиперорта – қабылдаушы жақтан қабылдауды растауды қажет етпейтін, бір-бірімен ассоциацивті байланысқан, блоктар түріндегі ақпарат түрлерін ұсыну технологиясы.

Гипермәтін – блоктардың бір-бірімен ассоциацивті түрде байланысқан мәтіні.

Доменнің бас бақылаушысы (Primary Domain Controller, PDC) – есептік жазбалардың деректер базасының негізгі көшірмелерін сақтау үшін PDC режіміндегі Windows NT Server орнатылған компьютер.

Дискретті сигнал (discrete signal) – соңғы, әдетте көп емес мәндердің саны бар сигнал. әдетте барлық жағдайларда дискретті сигналдың екі немесе үш мәндері бар. Кейде оны сандық сигнал деп атайды.

Интернет – ауқымды желіге біріктірілген компьютер жиынтығы.

Кабель (cable) – геметивті қапшыққа біріктірілген изоляцияланған жүргізушілердің бірі немесе тобы.

Кадрлардың коммутаторы – ортаның жылдамдығымен бірге жұмыс істейтін ортаға қатынау деңгейінің көппорттық көпірлері, ол клиенттік және серверлік жүйелерді байланыстыруда бөлінетін қатынауы бар орта үшін концентраторлармен салыстырғанда жоғары өтімділікке кепілдік береді. ЛВС-ті сегментациялағанда кадрлардың коммутаторлары баға/өнімділіктің ең дұрыс шамаларын қамтамасыз етеді және дәстүрлі көпірлер мен маршрутизаторларға қарағанда көрсетеді.

Клиент (client) – басқа бірнеше компьютерлерден қызметтер немесе ресурстарды сұрататын желідегі компьютер.

Клиент – сервер (client-server) – есептеу моделі, мұнда кейбір компьютерлер қызметтерді сұратады (клиент), ал кейбірі мұндай сұранымдарға қызметпен жауап береді (сервер).

Коаксиалды кабель (coaxial cable) – ішкі және сыртқы бір-бірінен изоляцияланған құралған кабельдер. Коаксиалды кабель бір немесе бірнеше

орталы мыс сымдары болады, ол диэлектрлі изоляциямен қапталған, ол өз кезегінде орталық сымдарды сыртқы электромагниттік әсерлерден қорғау үшін металл торшасымен немесе түтікпен жабылған.

Коммутатор (switch) – деректердің берілуінің мүмкін бағыттарын таңдайтын программа немесе құрылғы.

Көпір (bridge) – бір желінің жұмыс станцияларының басқа сжұмыс станцияларына қатынауға мүмкіндік беретін құрал. Көпірлер желіні кішкене сегменттерге бөлуге қолданылады. OSI моделінің арналық деңгейінде байланысуды жүзеге асырады. Көпір түрлі физикалық және арналық деңгейлерді түрлендіреді. Түйіндердің ұзындығы немесе санын көбейту үшін қолданылады.

Концентратор немесе hub (concentrator or hub) – жұлдыз топологиялы желілерде барлық компьютерлер қосылатын желінің байланыстырушы құрамдасы. Концентратор шырмалған жұпты қолданғанда компьютерлердің бір бірімен байланысын қамтамасыз етеді, сондай-ақ FDDI желілерінде компьютерлерді орталық түйіндерге қосу үшін қолданылады.

Құпия сөз (Пароль) (password) – пайдаланушының немесе бағдарламаның кез келген қорды пайдалануына құқығын растайтын белгі.

Топ (group) – бір атпен және қорларға қатынау құқығымен анықталатын пайдаланушылардың жиынтығы.

Жергілікті желі (Local-Area Network) – жүйелері бір-бірінен жақын арақашықтықта орналасқан желі.

Магистраль (backbone) – трансиверлер кабельдері компьютерлерге, қайталауыштарға және көпірлерге таралатын негізгі кабель.

Маршрутизатор (router) – екі желіні қосатын, кейде әртүрлі деңгейлі MAC (арналық деңгей, ортаға қатынауды бақылау) біріктіретін бағытталған құрылғы – хаттама.

Маршрутизация (routing) – деректер блогы адресатқа жететін жолды қатынастық желіде анықтау үрдісі.

Желінің маскасы (network mask) – бір IP–желі/бағыныңқы желіде орналасқан IP адрестердің диапазонын анықтайтын 32 биттік сан.

Модем (modem) – МОДулятор-ДЕМодулятор ден қысқартылған. Компьютерге деректерді әдеттегі телефондық тізбек бойынша беруге мүмкіндік беретін байланыс құрылғысы. Сандық сигналдарды беріліс кезінде аналогты түрлерге түрлендіреді. Ал қабылдау кезінде аналогты сигналдарды сандық түрлерге түрлендіреді.

Желінің мониторы (network monitor) – желілік трафикті бақылайтын бағдарламалық-аппараттық құрылғы кадрлар деңгейлерін дестелер типтері мен қателер туралы ақпаратты жинайды.

Қайталауыш немесе репитер (repeater) – кабельдің бір кесіндісінен екіншісіне мазмұнын өзгертпей, сигналдарын күшейте өткізетін құрылғы. Қайталауыштар ЛВС ұзындығын көбейтеді.

Құқық, мүмкіндік (token) – жүйеге кадрларын беруге рұқсат беретін символ немесе символдар тобы.

Өткізу жолы (bandwidth) – берілген диапазондағы максималды және минималды жиіліктер арасындағы айырмашылық; тасушы жұмыс істеуіне болатын диапазондардың жиілігі.

Пайдаланушы (user) – қандай да бір қорларды немесе мүмкіндіктерді қолданатын заңды немесе жеке тұлға.

Порт (port) – құрылғы немесе бағдарламаға қатынау нүктесі. Логикалық және физикалық порттар белгілі.

Провайдер (provider) – Интернет желісіне қосылуды қамтамасыз ететін (немесе ақылы еңбек негізінде басқа қызметтерді атқаратын) ұйым.

Хаттама – желі бойынша берілуге арналған жіберуші жұмыс станциясынан шығатын басқарылатын ақпарат пен деректері бар дестелерді жинау және алушы жұмыс станциясына жеткенде дестелерді босату тәртібін ұсынатын ережелер жиынтығы.

Таратқыш (hub) – жұлдыз топологиялы кабельдік жүйе немесе ЛВС-тің орталығы. Бұл файл-серверлер немесе концентраторлар болуы мүмкін. Оларды желілік бағдарламалық қамтамасыз ету бар және олар желі ішінде қатынастарды басқарады, сондай-ақ шлюз есебінде басқа компьютерлерде жұмыс істей алады.

Шифрлеу (encryption) – рұқсатсыз қатынаудан қорғау үшін ақпаратты түрлендіру.

Шлюз (gateway) – түрлі архитектуралы желілерді біріктіретін құрылғы.

МАЗМҰНЫ

Б Бөрібаев., Г.А. Мадьярова

Web-технологиялар

БАҚЫТ БӨРІБАЕВ – Қазақ политехникалық институтының автоматика және есептеу техникасы факультетін бітірген. Көп жылдардан бері жоғарғы оқу орындарында ұстаздық қызмет атқарып келеді, техника ғылымдарының кандидаты, профессор. Ол информатика және компьютерлік технологиялар саласынан мемлекеттік тілде шыққан алғашқы оқу құралдары мен сөздіктердің авторы. Қазақстан республикасының үздік білім қызметкері. 200-ден аса ғылыми еңбектері жарыққа шыққан, оның ішінде оннан аса оқулықтары, елуге тарта оқу құралдары мен әдістемелік нұсқаулары бар. Ғылыми еңбектері мен оқу-әдістемелік құралдарының басым бөлігі қазақ тілінде жарияланған.

МАДЬЯРОВА ГҮЛНӘР АТЫХАНҚЫЗЫ

Абай атындағы мемлекеттік университетінің физика және математика факультетін бітірген.

Педагогика ғылымдарының кандидаты, Алматы экономика және статистика академиясының, Информатика кафедрасының меңгерушісі, 2010 жылдың «Жоғарғы оқу орнының үздік оқытушысы мемлекеттік грантының иегері». 100- ден аса ғылыми еңбектердің авторы, оның ішінде төрт электрондық оқулық, бес оқулық, оннан аса оқу құралдары мен әдістемелік нұсқаулары бар.