Ф.7.11-18



Ибрагимов О.М.

WEB-БАҒДАРЛАМАЛАУҒА КІРІСПЕ

оқу құралы

Шымкент, 2012

ҚАЗАҚСТАН РЕСПУБЛИКАСЫ БІЛІМ ЖӘНЕ ҒЫЛЫМ МИНИСТРЛІГІ

М. ӘУЕЗОВ АТЫНДАҒЫ ОҢТҮСТІК ҚАЗАҚСТАН МЕМЛЕКЕТТІК УНИВЕРСИТЕТІ

Ибрагимов О.М.

WEB-БАҒДАРЛАМАЛАУҒА КІРІСПЕ

оқу құралы

Шымкент, 2012

УДК 004 (075.8)
ББк 32.973-018.2 я 7
И 15

Ибрагимов О.М. Web-бағдарламалауға кіріспе. Оқу құралы. –Шымкент: ОҚМУ, 2012. –128 бет.

ISBN 9965-467-41-2

Оқу құралында динамикалық гипермәтіндік құжаттарды құру технологиясы қарастырылған. Онда HTML, JavaScript және PHP бағдарламалау тілдерінің синтаксисі, тэгтері, операторлары мен функциялары қаралған. Сонымен қатар CSS технологиясын және скрипттерден пайдалану әдістері көрсетілген. Оқу құралында осы аталған мәселелер мысалдармен көрнекі түрде сипатталған. Ал оқу құралында берілген жеке тапсырмалар мен тест сұрақтары, оның мазмұнын түсінуге көмектеседі.

Оқу құралы университеттің 5В060200 – «Информатика» мамандығы бойынша оқитын студенттерге арналған.

Пікір жазғандар: Нысанов Е.А. – ф.-м.ғ.д., М.Әуезов атындағы Оңтүстік Қазақстан мемлекеттік университетінің «Информатика» кафедрасының доценті;
Марасулов А.М. – т.ғ.д., Қ.А.Ясауи атындағы Халықаралық Қазақ-Түрік университетінің «Информатика және сандық әдістер» кафедрасының меңгерушісі.
Ибрагимов Р.И. – п.ғ.д., Қазақстан халықтар достығы университетінің «Информатика және математика» кафедрасының меңгерушісі.

М.Әуезов атындағы Оңтүстік Қазақстан Мемлекеттік университетінің оқу-әдістемелік кеңесі баспаға ұсынған, хаттама №__«__» _____ 2012ж.

ISBN 9965-467-41-2

© М.Әуезов атындағы Оңтүстік Қазақстан Мемлекеттік университеті

Мазмұны

Кіріспе..... 5
1 HTML тілі негіздері
1.1 Интернет қызметтері 8
1.2 Web-құжат құру технологиясы 13
1.3 Web-құжат құрылымы 17
1.4 HTML тілінің мәтінді пішімдеу тэгтері 21
1.5 HTML тілінде тізімдерден пайдалану 26
1.6 Web-құжатқа сілтемелер, сурет және объекттер енгізу 28
2 Web-құжатты безендіру және CSS технологиясы
2.1 Web-құжатта кестелерден пайдалану 34
2.2 Web-құжатта қалыптардан пайдалану 38
2.3 Web-құжатта фреймдерден пайдалану 43
2.4 CSS технологиясы 46
2.5 CSS технологиясының қасиеттері 50
2.6 Web-құжаттың интерактивтік интерфейсі 55
3 JavaScript тіліне кіріспе
3.1 JavaScript тілінің қарапайым элементтері 59
3.2 JavaScript тіліндегі амалдар 63
3.3 Шартты операторлар 67
3.4 Цикл операторлары 70
3.5 Web-құжатқа скриптті орналыстыру әдістері 74
3.6 JavaScript-ке мысалдар 76
4 PHP тіліне кіріспе
4.1 PHP тілінің қарапайым элементтері 79
4.2 Деректердің типтері және амалдар 83
4.3 Операторлар 88
4.4 Функциялар 93
4.5 PHP және Арасне локальді серверін орнату 97
4.6 PHP-скрипттерге мысалдар 102
Жеке жұмысқа тапсырмалар 113
Тест сұрақтары 117
Пайдаланылған әдебиеттер 123
Қосымшалар 124

Кіріспе

Интернет – бүкіл әлемдегі миллиондаған шағын компьютерлік желілерді бір – бірімен байланыстырып тұрған орасан зор компьютерлік желі. Желідегі компьютерлерде мәтін, сурет, аудио – бейнематериалдар, т.с.с. өте үлкен ақпарат көлемі сақталады.

Web-дизайн термині - Web (Интернеттің WWW - World Wide Web қызметінің қысқартылған сөзі) және ағылшын тілінің дизайн (design – жобалау, құрастыру, құру) сөздерінің бірігуі арқылы туындап, «Web-құжатты көркем құрастыру» деген мағынаны білдіреді.

Дизайн сөзі күнделікті өмірде - объектті жобалау, яғни жаңа құрылғыны, құрал-саймандарды көркем құрастыру деген көп мағыналы термин ретінде қолданылады. Оның негізгі мақсаты – объектті қолдануына қарай көркемдеу, пішімдеу. Классикалық дизайн көркемөнерінен Web-дизайнның айырмашылығы, мұнда қалам, бояу, мольберт сияқты құрал-саймандардың орнына, компьютерлік құрал-саймандар - бағдарламалау жүйелері, Web-редакторлар, графикалық процессорлар, аниматорлар, цифрлік фотоаппараттар, сканерлер т.с.с. пайдаланылады.

Сонымен, Web-дизайн – IT-технологиялар (компьютер және жаңа ақпараттық технологиялар), арнайы бағдарламалау тілдері (HTML, JavaScript, PHP, Java, Perl, C++) және желілік технологиялар (CSS, CGI, SSI) негізінде құрылған көркемөнердің жаңа түрі болып есептеледі. Кейбір әдебиеттерде компьютерлік дизайн, Web-шебер терминдері де кездеседі.

Web-дизайншы деп, әдетте, Web-сайт дизайнымен айналысушы бағдарламашыны айтады. Бірақта олар дизайнмен қатар, жүйелік администратор және бағдарламашы міндеттерін де атқарады. Кейде сайт модераторлары да Web-дизайншы деп аталады.

Web-дизайншы Web-құжат құруда гипермәтіндерден, анимациялық суреттерден, фото, видео және аудио файлдардан пайдаланады. Олар коммерциялық және жарнамалық Web-жобалар: мемлекеттік мекеме немесе коммерциялық ұйым Web-сайты, виртуал магазин, электрондық басылым, ақпаратты іздеу жүйелерін құрады. Web-дизайншы өзінің кәсіби міндетін атқаруда декоративті-қолданбалы өнер, көркем фотография, архитектура, скульптура, граффити сияқты көркемөнер салаларында жинақталған тәжірибеден пайдаланыды. Сонымен қатар оның өзіне тән жаңа қасиетері, заңдылықтары бар. Олар Интернеттің техникалық, бағдарламалық және ақпараттық компоненттерінің дамуымен өзгеріп отырады.

Қазіргі таңда Web-құжаттарда гипермәтін мен статикалық суреттермен қатар, қозғалушы объекттер, яғни баннерлер, анимациялар, видеоклиптер қолданылады. Интернетте тұтынушымен интерактивті байланыста болатын динамикалық Web-сайттар, ақпаратты іздеу жүйелері жиі кездеседі. Сонымен қатар, тұтынушының менталитеті мен психологиясын, креативті идеялар мен тақырыптарды есепке ала отыру қажеттігі туындап

отыр. Мұның барлығы Web-дизайншыдан әдебиет, журналистика, кино, телевидения, фото т.с.с. өнер салаларының туындыларынан орынды пайдалануды талап етеді.

WWW (World Wide Web – дүниежүзілік өрмек) - Интернеттегі гипермәтіндік ақпарат іздеу жүйесі, оны Web деп те атай береді. Мәліметтер блоктары (Web-парақтар) мекемелердің немесе жеке тұлғалардың меншігіндегі WWW-сервер немесе Web – сервер деп аталатын жеке компьютерлерде сақталады. Web – құжаттарға енгізілген гипермәтіндік сілтемелер арқылы тұтынушы адам олардың бірінен біріне ауыса береді.

Web–браузерлер HTML тілінде Internet желісіне арнап жазылған Web-сайттар мен Web-құжаттарды парақтап көру, оқу мүмкіндігін береді. Web–браузерлер ішінде ең көп таралғандары Internet Explorer (Microsoft), Opera (Opera Software ASA) және Mozilla FireFox (Mozilla Corporation).

HTML (HyperText Markup Language) – гипермәтінді белгілеу тілі мәтіндерді Web-құжат (HTML-құжат) түрінде бейнелеуге арналған тэгтер жиынын анықтайды. HTML тэгтері құжатты туралауға, пішімдеуге және безендіруге мүмкіндік береді. Интернетке арналған мұндай гипермәтінді Web-құжаттар баспаға шығаруға емес, тек монитор экранында көруге арналған. Сондай-ақ, HTML тілі Web-құжатқа гипермәтін мен қозғалмайтын объекттерді бейнелеу мүмкіндігін береді.

DHTML (Dynamic HTML) тілі HTML-дің мүмкіндіктерін кеңейтуге арналған гипермәтінді белгілеу тілі. Бұл тіл, атынан көрініп тұрғандай, Web-құжатқа қозғалыс енгізеді, яғни мәтіннің түсін өзгертеді, кейбір элементті жасырып, кейбірін экранға шығарады т.с.с. Тілді MicroSoft және NetsCape корпорациялары бір-бірінен тәуелсіз жасап шығарған. Сондықтан DHTML-де жазылған код Internet Explorer мен NetsCape Navigator браузерлерінде әр түрлі жұмыс істейді. Соңғы кезде W^3C консорциум тілдің бірыңғай стандартын шығарған.

Қазіргі таңда тек қана HTML тілін пайдаланып, Интернетке арналған Web-құжат құрастыру жеткіліксіз. HTML тілінің мүмкіндіктерін кеңейтуге арналған қосымша технологиялар дамуда. Мұндай технологияларға төмендегілер жатады:

- құжатты «логикалық» пішімдеуге арналған CSS – каскадты стильдер кестесі;
- JavaScript немесе VBScript тілдерінде жазылған скрипттер;
- Java тілінде жазылған апплеттер;
- Perl, C++ тілдерінде жазылған CGI-скрипттер;
- GIF және Flash-анимациялар;
- AVI, MPEG форматындағы видеоклиптер;
- WAV, MP3, MIDI форматындағы аудиоклиптер.

CSS (Cascade Style Sheets – каскадты стильдер кестесі) технологиясы желідегі бірнеше Web-құжат мәтіндерін бірыңғай пішімдеуге арналған.

Бұл технология Web-сайт парақтарының монитор экранында бірдей бейнеленуін және жылдам жүктелуін қамтамасыз етеді. CSS технологиясын пайдаланып Web-сайттағы барлық құжаттағы мәтіндердің шрифтін, түсін, өлшемін, туралау пішімдерін т.с.с. өзгертуге болады.

Java-апплет деп Java тілінде жазылған бағдарламаны айтады. Бұл бағдарлама компиляцияланып, пайда болған кодты байт-код деп атайды. Байт-код клиенттік компьютерлерде Java виртуал машинасы көмегімен орындалады. Java-апплеттер көлемі бойынша ықшам және компьютер платформасына тәуелсіз. Сондықтан Java-апплет Web-құжатқа енгізілсе, браузер оны автоматты түрде Java виртуал машинасы көмегімен орындайды.

Скрипт (Script) – HTML тілінің мүмкіндіктерін кеңейтуге арналған, арнайы бағдарламалау тілінде жазылған ықшам бағдарлама. Бағдарламалау тілі ретінде JavaScript, VBScript, Java, Perl, C++, т.с.с. тілдері қолданылады. Скрипттер әртүрлі оқиғаларды өңдейді, мысалы, тышқанды шерту, курсордың экран бетіндегі қозғалысы. Оқиғаның жауабы ретінде Web-құжаттың экрандағы түрінің өзгеруі, клиент-компьютерден серверге ақпарат жіберу сияқты іс-әрекеттер орындалады.

Скрипттерді екі топқа бөледі: клиенттік (JavaScript, VBScript); серверлік (CGI-скрипт). Клиенттік скрипттер Web-құжаттың HTML тілінде жазылған кодының ішіне тікелей орналасады. Олар компиляцияны қажет етпейді, яғни компилятордың орнына интерпретатор жеткілікті. Бұл бағдарлама құруды жеңілдетіп, уақытты үнемдейді. Клиенттік скрипттердің негізгі қасиеті, олар клиент компьютерінде, яғни браузерде HTML-кодпен бірге орындалады. Оларды құруға JavaScript немесе VBScript тілдері қолданылады. JavaScript тілі қазіргі заманғы барлық браузерлерде қолданыс тапса, VBScript тілін тек қана Internet Explorer браузері пайдаланады. Бір сөзбен айтқанда, JavaScript тілі Java-ның, ал VBScript тіл Visual Basic-тің скрипттік версиялары.

PHP (Hypertext Preprocessor – гипермәтінді препроцессор) тілі серверде орындалатын сценарий жазуға арналған тіл. Браузер URL-адресте (Uniform Resource Locator – ықшамдалған ресурстар көрсеткіші) көрсетілген PHP тілінде жазылған Web-құжатты орындау үшін Web-серверге сұраныс жасайды. Web-сервер PHP-процессорын жүктеп, PHP-кодты орындайды ды, нәтижені Web-серверге қайтарады. Ал Web-сервер өз кезегінде нәтижені браузерге терезеде бейнелеу үшін жібереді.

Ұсынылып отырған оқу құралында динамикалық гипермәтіндік құжаттарды құру технологиясы қарастырылған. Онда HTML, JavaScript және PHP бағдарламалау тілдері, CSS технологиясынан пайдалану әдістері берілген. Оқу құралында осы аталған мәселелер мысалдармен көрнекі түрде сипатталған. Оқу құралында берілген жеке тапсырмалар материалды терең түсінуге көмектеседі.

1 HTML тілі негіздері

1.1 Интернет қызметтері

Интернет – бүкіл әлемдегі миллиондаған шағын компьютерлік желілерді бір-бірімен байланыстырып тұрған орасан зор компьютерлік желі, ол хаттама деп аталатын бірыңғай стандартпен жұмыс істейді [1,2].

Интернетті пайдаланудың ең негізгі түрлеріне – Бүкіләлемдік өрмек, электрондық пошта, желілік чат (әңгімелесу), әлеуметтік желілер т.б. жатады. Желіге қойылатын ең негізгі талап – байланыс арналары елеулі өзгерістерге ұшырағанмен, кез келген жағдайда информацияны бір орнынан екінші орынға сенімді түрде жеткізу қажеттілігі болып саналады. Желі тораптарының бірсыпыра бөлігі істен шықса да, қалғандары өз жұмысын атқара алатын қалыпта қалуы тиіс болды. Информацияны тасымалдау сенімділігі мен оның жеткізілу жылдамдығын арттыру мақсатында мәліметтерді тасымалдаудың жаңа түрі – дестелерді коммутациялау технологиясы пайда болды. Оның артықшылығы болып мәліметтерді тасымалдау жылдамдығының жоғарылығы, сенімділігі және икемді жұмыс істеуі саналды.

Пакеттерді коммутациялау принципі тасымалданатын мәліметті бірнеше шағын фрагменттерге бөлуден тұрады. Мәліметтер жеткізілетін пунктке дейінгі аралықтарда орналасқан түйінді компьютерлер арасында пакеттер бір – бірінен тәуелсіз тек өзіне тән байланыс арналары арқылы жылдам жеткізіліп тасымалданатындай етіп ұйымдастырылды. Бір мәліметті құрайтын әрбір пакет баратын пунктіне тек өз маршруты бойынша жеткізілетін болады. Қабылдау пунктінде пакеттер қайтадан біртұтас мәліметке біріктіріліп, өз иесіне, яғни арналған адресатына берілуі тиіс болатын.

Желіні басқару ісінің өте сенімді жұмыс істеуін жүзеге асыру мақсатында оларды бір орталықтан басқармай, барлық түйінді компьютерлер бірдей тең құқықты болып жасалды. Яғни әрбір желі торабындағы компьютер мәліметтерді қабылдау/жөнелту істерін атқарумен қатар басқа тораптардан келіп түскен хабарларда ары қарай жөнелту кезінде оларды басқаша адрестеу ісін де жүргізе алатын еді.

Әр түрлі құрылғылар мен программалар арасында мәлімет алмасу қалай және қандай түрде атқарылатыны жайлы ережелер жиыны хаттама деп аталады. Хаттамалар мынадай топтарға бөлінеді:

- базалық
- қолданбалы

Интернет базалық TCP/IP хаттамасын пайдаланады. Қолданбалы хаттамалар Интернеттің арнайы қызмет баптарының хаттамасынан http, ftp, telnet тұрады.

Гипермәтін – қосымша элементтерді басқару мақсатында ішіне арнаулы код, яғни екпінді элемент (anchor) орналасқан мәтін. Ол мәтін ішіне сурет,

дыбыс енгізу, мәтінді безендіру, пішімдеу ісін орындайтын немесе осы құжаттың басқа бөлігіне гиперсілтемесі бар алғашқы нүкте ретінде қарастырылатын белгіленген сөз. Сөзді ерекшелеп белгілеу дегеніміз – келесі көрсетілетін құжат бөлігі қалай бейнеленетінін анықтайтын айрықша кодты осы сөз ішіне енгізу. Гипермәтінді бейнелеу үшін броузер (browsers) деп аталатын арнайы көрсету программалары қолданылады. Гипермәтін экранда белгіленген қарапайым сөз ретінде тұрады, егер курсорды сол сөзге жеткізіп, тышқанды шертсек, онда ол сілтеп тұрған басқа құжатты оқимыз. Ол құжаттар мәліметтер ішіндегі басқа парақтарды немесе Web жүйесіндегі басқа тораптарда орналасып, бейнежазба, сурет, жазылған дыбыс күйінде болуы мүмкін.

Интернетте көптеген қызмет түрлері бар. Олардың ішіндегі ақпаратты таратуға арналғандары: WWW, FTP, Telnet. т.б.; электрондық мәлімет үшін: E-mail, Usenet, ISQ, IRC, IP-телефония және басқалары. Интернеттің әр қызмет бабы мәлімет таратудың немесе алмасудың әртүрлі функцияларын атқарады, оның үстіне олардың кейбірі ескіріп жойылса, кейбірі жаңадан туындап келеді. Осылардың қазіргі желілерде жұмыс істейтін негізгілерінің қысқаша сипаттамаларын қарастырып өтейік.

FTP. Қашықта орналасқан машинаның каталогтарымен және файлдарымен жұмыс істейтін хаттама (File Transfer Protocol – файлдарды тасымалдау хаттамасы) мен программалар осылай аталады. FTP құралдары сервер каталогтары мен файлдарын көріп, олардың бірінен біріне ауысып, керектілерін көшіріп алуға, жаңартуға мүмкіндік береді.

Telnet. Бұл термин арқылы клиенттің алыстағы сервер-компьютермен қатынасуын қамтамасыз ететін хаттама мен программа белгіленеді. Байланыс орнатылғаннан кейін тұтынушы қашықтағы компьютердің операциялық жүйесіне кіріп, сондағы орнатылған программалармен өз программалары тәрізді жұмыс істей алады.

Usenet жүйесінің бірнеше анықтамалары бар, олар – тармақталған дскуссия клубы, телеконференция, жаңалықтар тобы. Мұның электрондық поштадан айырмашылығы – мәліметті бір адамға ғана емес, бірнеше абоненттер тобына (телеконференциялар) жолдайсыз. Әртүрлі мәселелерді талқылау кезінде конференцияға қатынасушылар тең құқықты болып саналады. Әрбір телеконференция бір тақырыпқа (ғылымда, өнерде, спортта, т.с.с.) арналады және оның өзіндік адресі болады, мұндағы қарастырылатын мәселелер ауқымы әр түрлі бола береді – өте ауқымды тақырыптан (бір сұрақтың барлық жағы) тек бір ғана сұраққа дейін (мысалы, филателия) қамтылады.

ISQ (I seek you – мен сені іздеймін) – желі тұтынушыларына нақты уақыт кезеңінде хат-хабарлармен алмасып, бірден чат (chat- әңгіме) құрып, файл жіберіп, әңгімелесуге мүмкіндік беретін қызмет түрі.

IRC (Internet Relay Chat) – нақты уақыт кезеңінде телеконференция өткізудің бір түрі, тұтынушылардың бір-бірімен әңгіме-дүкен құруға болатын қызметі. IRC-серверлері мен IRC-клиенттері көмегімен

тұтынушылар бір-бірімен «виртуальды» әңгіме өткізуді пернетақтада сөздерді теру арқылы жүргізеді.

IP-телефония. Интернетте телефондағы дауыс арқылы мәлімет алмасуды өте тиімді деп айтуға болмас, бұл тек TCP/IP хаттамалары негізінде электрондық мәлімет алмасудың жеке бір түрі ғана. Адамның дауысы цифрлы файлға түрлендіріледі де, желідегі қарапайым мәліметтер пакеттері тәрізді түрде тасымалданады. Қазақстан Республикасы байланысының ұлттық операторы болып саналатын «Қазақтелеком» ААҚ-ы осы байланыс түрін практикада пайдаланып жүр.

WWW (World Wide Web – дүниежүзілік өрмек) - Интернеттегі гипермәтіндік ақпарат іздеу жүйесі, оны Web деп те атай береді. WWW мен оның программалық жабдықтары – желінің ең қуатты, әрі болашағы бар құралдары. WWW негізінде гипермәтіндік мәліметтерді тасымалдау хаттамасы – HTTP (Hyper Text Transfer Protocol) арқылы жұмыс істейді, ал оның ішкі ақпараттары құжаттарды белгілеудің гипермәтіндік HTML (Hyper Text Markup Language) тілі көмегімен құрастырылады.

Бүкіләлемдік немесе корпоративтік желіге қосылған әрбір компьютердің жеке IP-адресі (Internet Protocol) болады. Бұл адрес нүктелермен бөлінген төрт ондық саннан тұрады. Сандардың өзгеру аралығы 0-ден 255-ке дейін (бір байтқа осынша сан сияды). Мысалы: М.Ауезов атындағы ОҚМУ серверінің адресі 192.168.32.1. Желідегі компьютерлер бірін-бірі осы адрес көмегімен тауып алады.

IP-адресі екіге бөлінеді:

- тұрақты;
- динамикалық.

Желіге қосылған серверлер тұрақты адреске ие болады. Олар әдетте күні-түні жұмыс істейді. Ал, желіге сымтетік арқылы қосылған компьютер динамикалық адрес алады. Бұл компьютер желіге қосылған уақытша адрес алып отырады. Компьютерді осылай адрестеу 256^4=4,3 млрд физикалық адрес бөлу мүмкіндігін береді. Мұндай адрес 32-разрядты екілік сан түрінде (4 байт) сақталады.

Құжаттар үшін адрестеудің екінші түрі URL (Uniform Resource Locator – Ықшамдалған ресурстар көрсеткіші) деп аталады. Мұндай адрестеу желідегі әрбір компьютерге және құжатқа меншіктеледі.

Желі компьютерлерінің IP-адрестерін пайдалану ыңғайсыз. Сондықтан олармен қатар жүретін мағынасы бар домендік адрестер жасалған. Мысалы: сымтетік номерінің жанына оның иесінің аты жазылғандай.

Домендік адрестің жазылу ережесі мынадай:

хаттама://компьютер аты. домен аты [/каталог/ішкікаталог/файл аты]

Бірінші болып қолданылатын хаттама көрсетіледі, мысалы, http. Келесі компьютер аты жазылады да, одан кейін домен аты тұрады, мысалы, microsoft.com. Адрес соңында кейде сервердегі каталог аттары мен файл

аттары жазылады, мысалы /mprses/cpnew.htm. Тік жақша ішіндегі адрес бөлігі жазылмаса да болады. Бірнеше нақты адрестермен мысал келтірейік:

http://www.microsoft.com/mprses/cpnew.htm
http://www.rambler.ru
ftp://mathsoft.com

Домендік адрестің жазылу ережесі төменде көрсетілген:

- сервер жұмыс істеп тұрған хаттама (WWW үшін http хаттамасы, көрсетпесе де болады);
- компьютердің IP-адресі немесе URL-адресі;
- сервер өңдейтін TCP порты (WWW үшін 8080, көрсетпесе де болады);
- сервердегі HTML-құжаттың адресі.

Мысалы: http://www.ukgu.kz – М.Ауезов атындағы ОҚМУ.

Домен аттары екі түрде беріледі:

- қызмет бабына қарай;
- аймақтық.

Қызмет бабына қарай қойылған домендік аттар алғаш рет АҚШ-та қолданылған. Бұл домендік аттар мекеме типін көрсетеді. Олар төменде көрсетілген:

- com – коммерциялық ұйымдар;
- edu – білім мекемелері;
- gov – мемлекеттік ұйымдар;
- mil – әскери ұйымдар;
- org – басқа мекемелер;
- net – желілік ұйымдар;
- int – халықаралық ұйымдар.

Бүкіләлемдік желінің дамуына байланысты домендер мемлекет атына сәйкес қойыла бастады. Аймақтық домен атары толығымен 1-қосымшада көрсетілген. Мысалы:

- ru - Ресей
- us - АҚШ
- kz – Қазақстан
- ua - Украина

Бірнеше нақты адрестермен мысал келтірейік.
Шет елдік іздеу серверлері мен каталогтары:

- www.google.com
- www.yahoo.com
- www.altavista.com

- www.msn.com
- www.ask.com

Ресейдің іздеу серверлері мен каталогтары:

- www.google.ru
- www.yandex.ru
- www.rambler.ru
- www.aport.ru
- www.mail.ru

Қазақстанның іздеу серверлері мен каталогтары:

- www.google.kz
- www.yandex.kz
- www.qazaq.kz/izdew.php
- www.catalog.site.kz
- www.resurs.kz
- www.catalog.tabu.kz
- www.zero.kz
- www.poisk.kz
- www.catalog.desk.kz
- www.kazakh.ru
- www.catalog.awstudio.kz

Тегін пошталық серверлер:

- www.ok.kz
- www.nur.kz
- www.yandex.ru
- www.mail.ru
- www.rambler.ru
- www.gmail.com

Әлеуметтік желілер:

- Facebook
- Twitter
- MySpace
- YouTube
- Вконтакте
- Одноклассники.ru
- Мой Мир@mail.ru
- В кругу друзей
- Мой Круг

E-mail. Бұл электрондық поштаның ағылшынша аталуы, ол - желі қызметтерінің ішіндегі электрондық мәлімет алмасудың ең негізгісі. Электрондық пошта көмегімен барлық құрылықтарда тұратын адамдар бір-біріне электрондық хаттар мен файлдар жібере алады.

Электрондық пошта жұмысы кәдімгі тұрмыстағы поштаға ұқсас болғандықтан, оның жұмыс істеу принциптерін меңгеру тұтынуылар үшін аса көп қиындықтар туғыза қоймайды. Оның негізгі айырмашылығы – физикалық заттар емес, ақпараттық мәліметтер, бейнелер жөнелтіледі.

Электрондық пошта географиялық адрестеуді қолданбайды. Электрондық поштаның адресі төмендегідей түрде болады:

пұтынушы_аты@домен_аты.

Мысалы: info@mail.ru, us-ibr@rambler.ru, us-ibr@nur.kz

1.2 Web-құжат құру технологиясы

Интернеттің WWW (World Wide Web) жүйесінің негізгі қызметі қажетті ақпаратты іздеу, жинау және экранға шығаруды ұйымдастыру. Оның экранда көрсететін мәліметтері Web-беттер түрінде дайындалып сақталған, электрондық құжаттар. Электрондық құжаттың кәдімгі құжаттан айырмашылығы, оның жазылу форматында. WWW қызметіндегі барлық гипермәтінді құжаттардың жалпы бір қасиеті, олардың барлығы HTML тілінде жазылған. HTML (HyperText Markup Language) –гипермәтінді белгілеу тілі мәтіндерді Web-құжат (HTML-құжат) түрінде бейнелеуге арналған нұсқаулар жиынын анықтайды. Сонымен қатар Web-құжатқа суреттерді, кестелерді түрлі түспен бейнелеу мүмкіндігін береді. Web-құжаттар Интернеттің WWW қызметімен бірге дами отырып, оның соңғы технологияларымен үнемі толықтырылып отырылады. Төменде тілдің Интернет консорциум ұсынған HTML 4.0 версиясын қарастырамыз [3,4].

HTML тілінде жазылған гипермәтіндік құжаттар браузер терезесінде ықшам түрде көрсетілгенімен, HTML тілі мәтіндерді пішімдеп көрсету тіліне жатпайды. Ол гипермәтіндердің атқаратын қызметін анықтап, оларды тұтынушыға бейнелеп жеткізетін тіл болып есептеледі. Web-құжаттар тұтынушы мониторының мүмкіндіктеріне қарап, браузермен әртүрлі бейнеленуі мүмкін. Сонымен, гипермәтіндерді осылай байланыстыра отырып белгілейтін мүмкіндікті беретін HTML тілі. Оның дұрыс нәтиже алуды қамтамасыз ететін өз заңдылықтары мен ережелері бар.

Web–браузерлер HTML тілінде Internet желісіне арнап жазылған Web-сайттарды парақтап көру, оқу мүмкіндігін береді. Web–браузерлер ішінде ең көп таралғандары Internet Explorer (Microsoft), Opera (Opera Software ASA) және Mozilla FireFox (Mozilla Corporation) (1-3 суреттер).

Браузерлердің негізгі атқаратын қызметтері:

- құжат сақталған Web-сервермен байланыс орнату, жинақталған құжаттың барлық бөліктерін жүктеу.
- HTML тілінің тэгтерін нұсқау ретінде орындау (интерпретациялау), Web-құжаттарды браузер жұмыс істейтін компьютердің мүмкіндіктеріне сәйкес бейнелеу және форматтау.
- Web-құжаттарды іздеу жолын автоматтандыруды қамтамасыз ету және бұрында жұмыс істеген Web-парақтарымен қатынас құруды жеңілдету.
- Басқа интернет қызметтерімен жұмыс істеуге арналған ішкі және жеке құралдармен қатынас құруды ұйымдастыру.

Web-құжат файлдары .html немесе .htm кеңеймесі бар мәтіндік файлдар. Оларды құруда қарапайым мәтіндік редакторлардан пайдалануға болады. Мысалы: MS Windows ортасында Notepad (Блокнот) редакторы.

Бірақта Web-құжаттарды арнайы HTML-редакторларда құру ыңғайлы. HTML-редакторлар екі топқа бөлінеді:

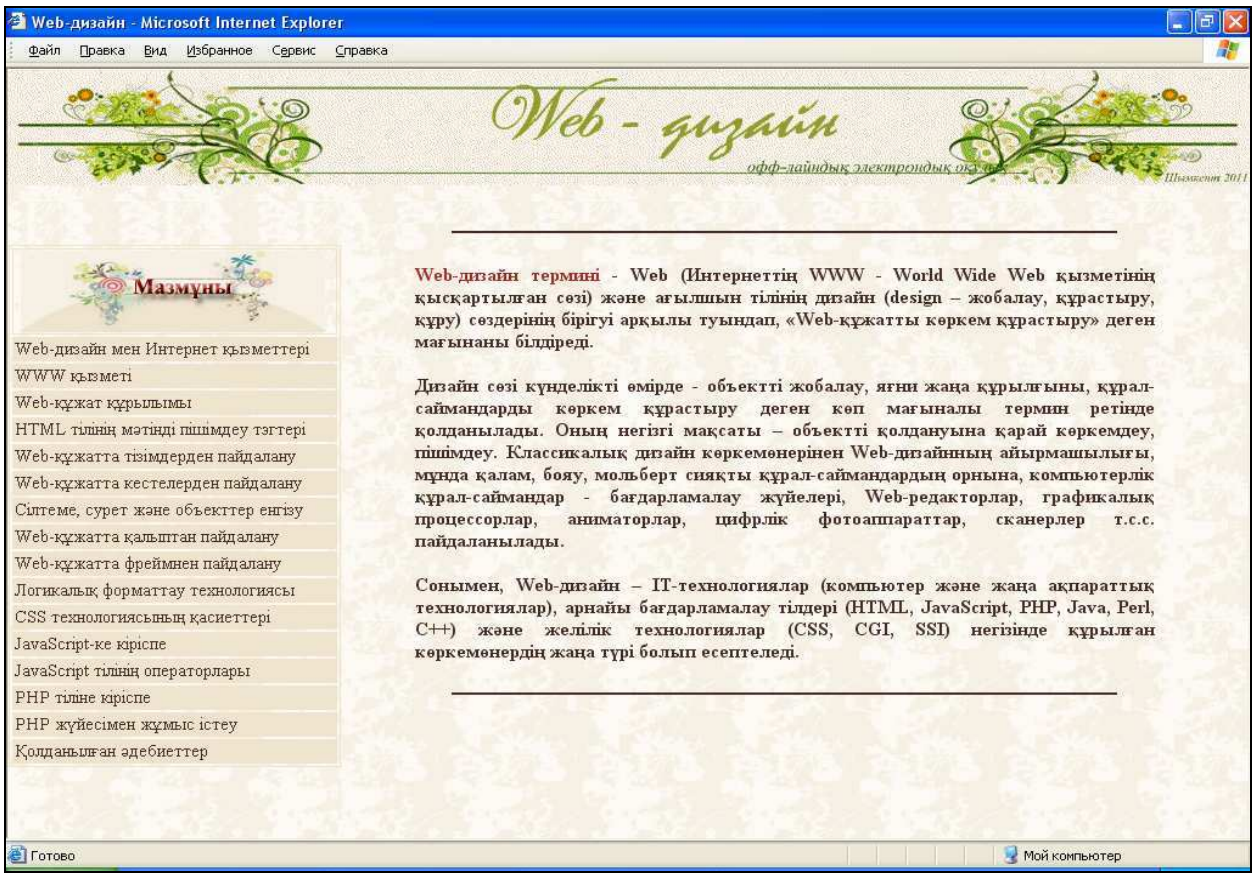
- ASCII-редакторлар (ASCII-editors, Source editors);
- WYSIWYG-редакторлар (WYSIWYG-editors).

ASCII-редакторлар қарапайым мәтіндік редакторлар. Тек олар HTML-құжаттарды құруға арналған қосымша мүмкіндіктерге ие. Оларға Webber, Macromedia HomeSite, DarekWare HTML Author, WebEdit редакторлары жатады.

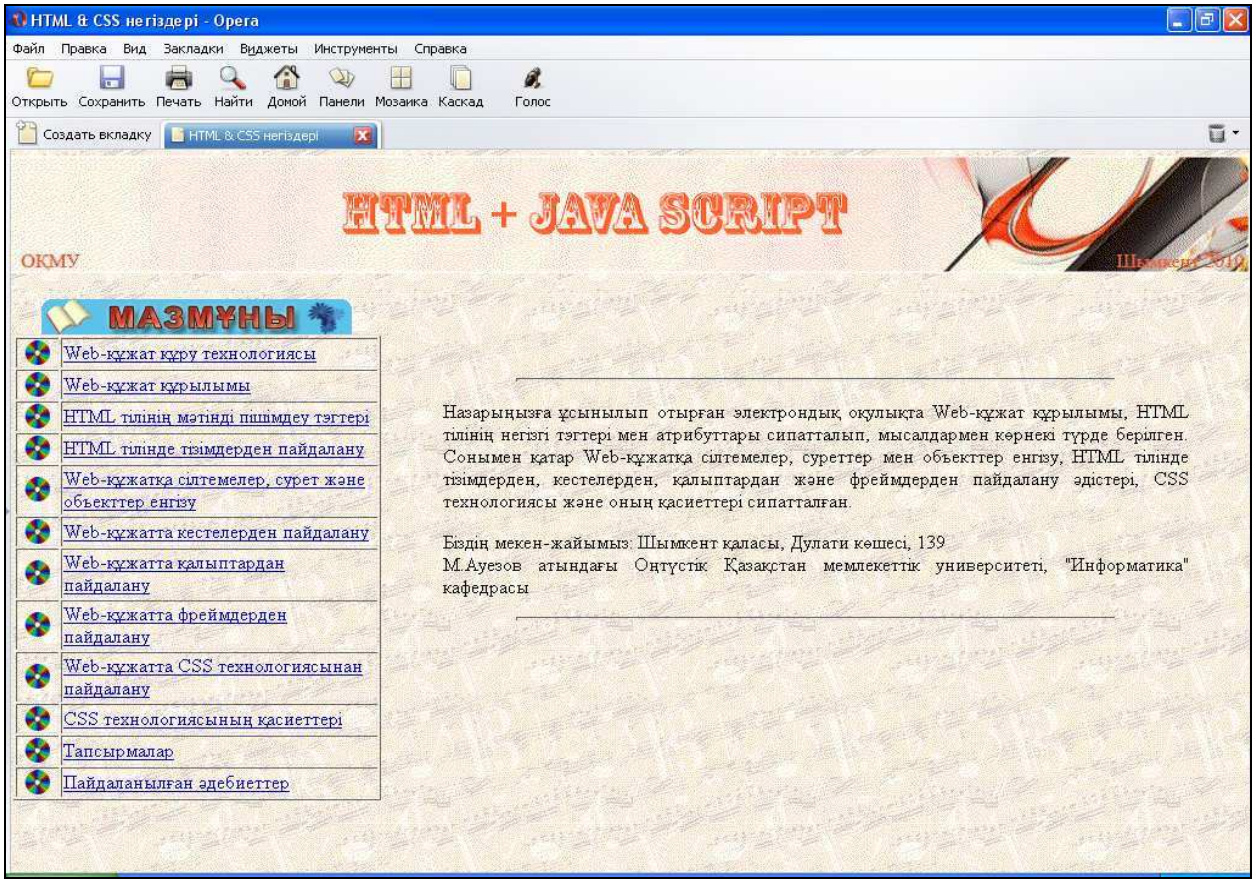
WYSIWYG-редакторлар HTML-құжаттарды дайындалу барысында терезеде көру, құжатқа алдын-ала дайындалған шаблондарды қосу мүмкіндігін береді. Редакторлар қосымша терезе – браузерлермен жабдықталған. Оларға Microsoft FrontPage, Netscape Navigator Gold редакторлары жатады.

Web–браузерлер HTML тілінде Internet желісіне арнап жазылған Web-сайттарды парақтап көру, оқу мүмкіндігін береді. Web–браузерлер ішінде ең көп таралғандары Internet Explorer (Microsoft), Opera (Opera Software ASA) және Mozilla FireFox (Mozilla Corporation) (1-3 суреттер).

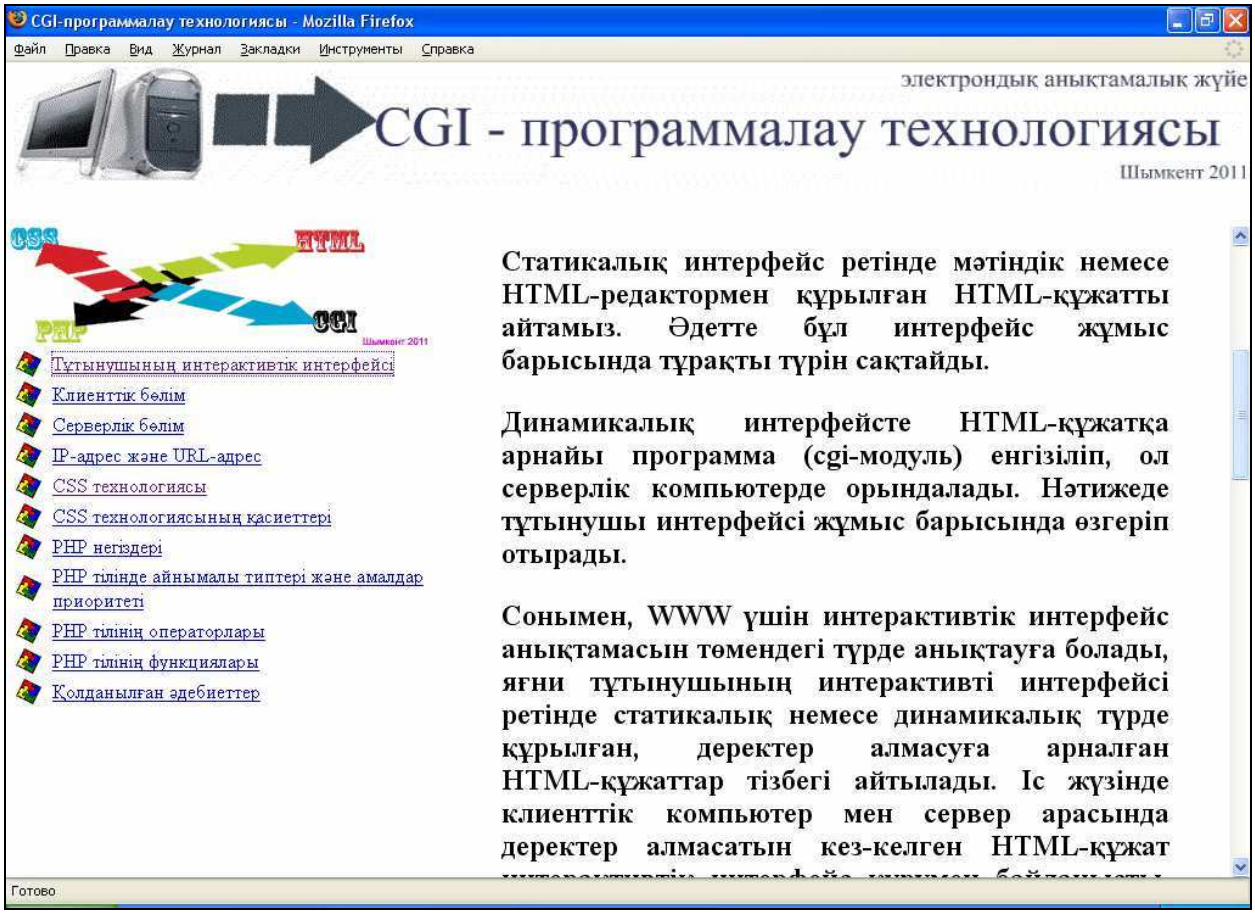
HTML тілінің мәтіндерді белгілеу нұсқаулары тэг (tag) деп аталады. Тэг символдар тізбегінен тұрады. Барлық тэг < (кіші) символынан басталып, > (үлкен) символымен аяқталады. Осы символдар ішіне *қызметші сөз* – тэг орналасады. Әрбір тэг өзіне тән бір арнаулы қызмет атқарады. Оларды жазуда бас әріпті де, кіші әріптерді де қатар қолдануға болады, әріптер бірдей қабылданады. Браузер мәтінді монитор терезеында көрсеткенде тэгтер бейнеленбейді, тек олардың құжат мәтініне тигізетін әсері ғана білініп тұрады.



Сурет 1 - Microsoft Internet Explorer браузері



Сурет 2 - Opera браузері



Сурет 3 - Mozilla FireFox браузері

HTML тілінің әрбір тэгі әдетте құжаттың белгілі бір бөлігіне ғана әсер етеді. Осыған орай екі тэг қатар қолданылады: бірі – тэгті *ашады*, екіншісі – *жабады*. Ашатын тэг белгілі бір нұсқауды бастайды, ал жабатын тэг – оны аяқтайды. Жабу тэгінің алдына / (қиғаш сызық) символы жазылуы тиіс. Кейбір тэгтер мағынасына орай жабылмауы да мүмкін.

Тэгтердің мәтінге тигізетін әсерлерін түрлендіруде олардың *атрибуттарынан* пайдаланады. Әдетте атрибуттар ашылу тэгтерінде кездесіп, қосымша қызметші сөздерден тұрады. Көбінесе атрибуттар " (қос тырнақша) символдары ішіне жазылған мәнімен беріледі. Атрибут мәні оның қызметші сөзінен = (теңдік) символы арқылы бөлініп жазылады.

Web-құжатта *түсініктеме* жазу үшін <!-- comment --> тэгінен пайдаланамыз. Түсініктемеде жазылған мәтін немесе тэг өңделмейді, яғни браузер терезесіне шықпайды. Түсініктемені құжаттың кез келген жеріне жазуға болады.

1.3 Web-құжат құрылымы

Web-құжатты құрудан алдын, оның қай тілде жазылғанын браузер үшін анықтау қажет. Ол үшін құжаттың басына

<!doctype html public"-//W3C//Dtd HTML 4.0//EN">

тэгін орналастырамыз. Бұл қатар құжаттың HTML 4.0 тілінде жазылғанын білдіреді.

Web -құжат құрылымы төмендегі негізгі тэгтерден тұрады:

<html>
<head>
<title> *тақырып* </title>
</head>
<body>
негізгі бөлім
</body>
</html>

Web-құжат <html> тэгімен басталып, сәйкес </html> тэгімен аяқталады. Осы екеуінің арасына құжаттың *тақырыбы* және *негізгі бөлімі* орналасады.

Құжаттың тақырыптық бөлімінде төмендегі тэгтер кездеседі:

- <head>
- <title>
- <base>
- <meta>
- <style>
- <link>

Құжаттың тақырыптық бөлімі <head> және </head> тэгтерінің ортасына орналасады ды, жалпы құжат жайлы мәлімет береді. Бұл бөлім браузер терезесінде бейнеленбейді. Әдетте бұл бөлімде <title> және </title> тэгтерімен шектелетін құжаттың ресми тақырыбы жазылады. Браузерлер бұл атауды браузердің терезе тақырыбында жазылатын файл аты есебінде пайдаланады.

<base> – тэгі Web-құжаттың ішкі бумалары мен файлдары үшін негізгі URL-адресін жазуға арналған. <base> тэгінің жабылу тэгі жоқ. Тэг төмендегі атрибуттарға ие:

- href – атрибуты құжаттың негізгі URL-адресін анықтайды. URL-адрес толық көрсетілуі тиіс;
- target – атрибуты сілтемеде көрсетілген құжаттың бейнелену фрейм терезесін анықтайды. Егер атрибут жазылмаса негізгі терезе түсініледі.

Мысалы: <base href ="http://domen.ru" target=="frame">

<meta> – тэгі Web-құжат жайындағы ақпарат беруге арналған. Бұл ақпаратты іздеу серверлері құжатты индекстеуде пайдаланады. <meta> тэгінің жабылу тэгі жоқ. Тэг төмендегі атрибуттарға ие:

- http-equiv="refresh" – атрибуты браузерге content атрибутында берілген уақыт аралығында көрсетілген құжатқа өту керектігін көрсетеді; content=”number” – берілген уақыт; url="url-адрес" – http-equiv атрибуты үшін адресі;
- name="description/keywords" – атрибуты браузерге content атрибутында құжатты сипаттау немесе түйінді сөздер тізімі берілгенін анықтайды; content="text/мәндер тізімі" – атрибут мәні description болғанда құжаттың сипатталуы, ал атрибут мәні keywords болғанда түйінді сөздер тізімі беріледі.

Мысалы: <meta http-equiv="Content-Type" content="text/html">.

<style> тэгі құжатқа стильдер кестесін (CSS – Cascade Style Sheet) енгізуге арналған. Стильдер кестесінде абзац, мәтін, тақырып сияқты түсініктердің жалпы пішімдеу стильдері анықталады. Тэг </style> тэгімен жабылады. Тэг төмендегі атрибуттарға ие:

- type – міндетті атрибут. Ол стильдер кестесінің MIME-типін анықтайды. Әдетте атрибуттың мәні ретінде "text/css" жазылады;
- title – атрибуты пайдаланылатын стильдер кестесінің атын анықтайды.

Мысалы: <style type="text/css" title="style1"> стильдер </style>

<link> тегі құжатты желідегі басқа Web-құжаттармен байланыстырады. Тэгтің жабылу тэгі жоқ. Тэг төмендегі атрибуттарға ие:

- href – атрибуты байланысу файлының url-адресін анықтайды;
- type – атрибуты байланысу файлының MIME-типін анықтайды;
- rel – атрибуты url-адрестегі файлмен байланысу типін көрсетеді. rel атрибуты төмендегі мәндердің бірін қабылдайды:
  - stylesheet – стильдер кестесімен байланысуды көрсетеді. Браузер href атрибутында көрсетілген стильдер кестесін жүктеп, құжатта пайдаланады;
  - home – сайттың негізгі Web-құжатын анықтайды;

- toc, contents – құжаттың негізгі мазмұны сақталатын файлды анықтайды;
- index – индекстік іздеу серверлеріне кілттік сөздер тізімін сақтаушы файлды көрсетеді;
- glossary – құжаттқа тиісті терминдер тізімі сақталатын файлды анықтайды;
- copyright – сайттың құжат авторлары жайында деректерді сақтаушы файлын көрсетеді;
- up, parent – иерархиялық ағашта өзінен жоғары тұрған құжатты анықтайды;
- child – иерархиялық ағашта өзінен төмен тұрған құжатты анықтайды;
- next – құжаттар тізбегіндегі келесі құжатты көрсетеді;
- previous – құжаттар тізбегіндегі өзінен алдыңғы құжатты көрсетеді;
- last, end – құжаттар тізбегіндегі соңғы құжатты көрсетеді;
- first – құжаттар тізбегіндегі бастапқы құжатты көрсетеді;
- help – түсініктеме сақталатын файлды анықтайды.

Мысалы: <link rel="home" href="index.html">
<link rel=="stylesheen" type=="text/css" href=="files/us-ibr.css">

Құжаттың негізгі бөлімі <body> және </body> тэгтерінің ортасында жазылады. Негізгі бөлімде құжат мәтіні, кестелер, суреттер, сілтемелер сәйкес тэгтермен өрнектеліп жазылады. Сонымен қарапайым құжат көрінісі мынадай түрде болады:

```
<html>
<head>
<title> Web-бағдарламалау </title>
</head>
<body>
        HTML негіздері
</body>
</html>
```

Жоғарыда көрсетілген <html>, <head>, <body> тэгтері HTML тілінде бекітілгенімен, оларды жазбаса да болады, бірақта алдын-ала тұтынушының қандай браузер пайдаланатыны белгісіз. Сондықтан Web-құжат құрылымы олардың толық болуын талап етеді. Әйтпесе кейбір браузерлер Web-құжатты терезеде дұрыс бейнелемеуі мүмкін.

Негізгі бөлімде қолданылатын <body> тэгінің атрибуттарын қарастырайық:

- text="түс" – мәтін түсін анықтау;
- bgcolor="түс" – фон түсін анықтау;

- link="түс" – қаралмаған сілтеме түсі;
- vlink="түс" – қаралған сілтеме түсі;
- alink="түс" – тұтынушы таңдаған уақыттағы сілтеме түсі;
- background="URL-адрес" – фондық сурет адресі.

Мысалы: <body background="files/graf01.gif" text="#0000FF">. Мұнда фондық обой ретінде graf01.gif файлы алынған.

Түсті анықтауда түстердің RGB (red, green, blue) моделінен пайдаланып, түстің ағылшынша атын немесе он алтылық санақ жүйесіндегі мәнін береміз. Түс red-қызыл, green-жасыл, blue-көк үш бояудың араласуынан тұрады. Араласу мөлшері он алтылық сан 00-ден басталып FF-санымен аяқталады. Он алтылық сан алдына # символы жазылады (сурет 4).

Мысалы: <body bgcolor="#008000">. Терезенің фоны ретінде жасыл түсі алынады.

түсі	аты	сандық мәні	түсі
қара	black	#000000	
күміс	silver	#C0C0C0	
күлгін	gray	#808080	
ақ	white	#FFFFFF	
қоңыр	maroon	#800000	
қызыл	red	#FF0000	
сирень	fuchsia	#FF00FF	
сия көк	purple	#800080	
жасыл	green	#008000	
ашық жасыл	lime	#00FF00	
зайтүн	olive	#808000	
сары	yellow	#FFFF00	
көк	blue	#0000FF	
тоқ көк	navy	#000080	
жасыл-күлгін	teal	#008080	
аға көк	aqua	#00FFFF	

Сурет 4 - Негізгі түстер кестесі

1.4 HTML тілінің мәтінді пішімдеу тэгтері

Егер бірнеше қатарлы мәтінді құжаттың ішінде теріп, оны браузермен қарасақ, мәтін бір түрлі пішімді шрифтте жазылып, қатарды тасымалдау амалдары орындалмайды. Мәтін браузер терезесіне сиғанша жазылып, кез-келген жерінен келесі қатарға тасымалданады, яғни мәтін пішімдеуді талап етеді.

Мәтіндік бөлікті пішімдеу тэгтерін қарастырайық. Оларға төмендегі тегтер жатады:

- <hn> (headers)
- <p> (paragraph)
- (division)
- <address>
- <blokequote>
-
 (break)
- <hr> (horisontal rule)
- <pre> (preformatted text)
-

<hn> тэгі мәтін тақырыптарын деңгейге бөледі. Барлығы болып шрифт өлшемімен бөлектенетін алты деңгей бар, яғни $n = 1..6$. Бірінші деңгей ең үлкен өлшемді шрифттермен беріледі. Олардың көмегінде тақырыптар мағынасы бойынша реттеледі. Тақырып </hn> тэгімен жабылады. Тэгтің align атрибуты тақырыпты left - сол жақ шетке, right - оң жақ шетке және center -ортаға туралайды.

Мысалы:

```
<h1 align="center"> тақырып 1</h1>
<h2 align="center"> тақырып 2</h2>
<h3 align="center"> тақырып 3</h3>
<h4 align="center"> тақырып 4</h4>
<h5 align="center"> тақырып 5</h5>
<h6 align="center"> тақырып 6</h6>
```

деп жазсақ, браузер терезесінде төмендегі сөздер пайда болады:

тақырып 1

тақырып 2

тақырып 3

тақырып 4

тақырып 5

тақырып 6

 тэгі жаңа абзацты ұйымдастырады. Мәтінді келесі қатарға көшіріп, абзацты алдыңғы мәтін қатарларынан бөліп тұрады. Абзацты тэгі жабады. Тэгтің align атрибуты абзацты left-сол жақ шетке, right-оң жақ шетке, center-ортаға және justify-екі жақ шетке туралайды.

Мысалы: мәтін .

 тэгі мәтінді логикалық түрде бөлектеу мүмкіндігін береді. Браузер бұл бөлікті логикалық объект деп түсінеді. Ол объектпен динамикалық іс-әрекеттер орындай алады. Мәтінді тэгі жабады. Тэгтің align атрибуты мәтінді left-сол жақ шетке, right-оң жақ шетке, center-ортаға және justify-екі жақ шетке туралайды.

Мысалы: мәтін .

<address> тэгі мәтінді пошталық адрес ретінде сипаттайды. Көп браузерлер мәтінді курсив шрифтпен бейнелейді. Мәтінді </address> тэгі жабады.

Мысалы: <address> Шымкент, 21-мкрн, 14-12 </address>

<blokequote> тэгі мәтінді цитата ретінде сипаттайды. Браузер терезеде цитатаны абзац сияқты бейнелейді. Цитатаны </blokequote> тэгі жабады.

Мысалы: Абайға пародия
 <blokequote>
Қараша, желтоқсан мен сол бір-екі ай
Қыстың басы бірі ерте, біреуі жай
Осы екі айда бір қызбен таныстырып
Ғашық қылып, ажыратқан, құдайым-ай!
</blokequote>

 тэгі қатарды тасымалдау амалын орындайды. Бұл тэг жабылу тэгісіз жазылады. Мысалы: мәтін 1
 мәтін 2
 мәтін 3 .

<hr> тэгі мәтінге горизонталь сызық енгізеді. Тэг жабылу тэгісіз жазылады. Тэг төмендегі атрибуттарға ие:

- align – атрибуты сызықты left-сол жақ шетке, right-оң жақ шетке және center-ортаға туралайды;
- size – атрибуты сызықтың қалыңдығын пиксельде анықтайды;
- width – атрибуты сызық ұзындығын пиксельде немесе терезе енінің пайызында анықтайды. Егер көрсетілмесе 100%, яғни терезенің толық ені алынады;
- noshade – атрибуты сызықты көлеңкесіз шығарады;
- color – сызық түсін түстердің RGB моделінде анықтайды.

Мысалы: <hr align="left" width="80%" size="2" noshade>.

<pre> тэгі абзацты сол жазылған пішімделген күйінде, өзгеріссіз браузер терезесінде бейнелейді. Мәтін шрифті моношрифт типінде, барлық әріптердің ені бірдей өлшемді болады. Абзацтағы бос орындар саны мен сөздерді тасымалдау өзгеріссіз жазылады. Абзацты </pre> тэгі жабады.

Мысалы: <pre> мәтін </pre>.

 тэгі мәтін фрагментін бөлектеп, браузер терезесінде ортаға жазуға арналған. Мәтінді тэгі жабады.

Мысалы: мәтін .

Мәтінді пішімдеу тэгтерін қарастырайық. Оларға төмендегі тегтер жатады:

- <basefont>
-
- <i> (italic)
- (emphasis)
- <b> (bold)
- (stronger emphasis)
- <u> (underline)
- <s> (strike)
- <big>
- <small>
- <sup> (superscript)
- <sub> (superscript)
- <code>, <samp> (sample)
- <tt> (tele type)
- <kbd> (keyboard)
- <var> (variable)
- <cite> (citation)

<basefont> тэгі құжат мәтінінің негізгі шрифтін анықтайды. Бірақта мәтінннің кез-келген жерінде шрифтті тэгімен өзгертуге болады. Тэгтің жабылу тэгі жоқ. Тэг төмендегі атрибуттарға ие:

- size – міндетті атрибут. Негізгі шрифттің өлшемін анықтайды. Қабылдайтын мәндері:
  - абсолют мән: 1-ден 7-ге дейінгі бүтін сандар. Шрифт өлшемі браузерге байланысты түрде атрибуттың мәніне сәйкес үлкейеді;
  - салыстырмалы мән: -7-ден +7-ге дейінгі бүтін сандар. Шрифт өлшемі браузердің негізгі шрифт өлшеміне салыстырмалы түрде өзгереді. Егер + болса үлкейеді, ал – болса кішірейеді;
- face – атрибуты негізгі шрифттің типін анықтайды.

Мысалы: <basefont size=2 face="Times New Roman"> .

 тэгі мәтін шрифтінің типін, өлшемін және түсін өзгертеді. Тэг тэгімен жабылады. Тэг жабылғасын шрифт <basefont> тэгіндегі мәндерді қайтадан алады. Тэг төмендегі атрибуттарға ие:

- size – атрибуты шрифттің өлшемін анықтайды. Мәндері:
  - абсолют мән: 1-ден 7-ге дейінгі бүтін сандар;
  - салыстырмалы мән: -7-ден +7-ге дейінгі бүтін сандар;
- face – атрибуты шрифттің типін анықтайды;
- color – атрибуты шрифттің түсін анықтайды.

Мысалы: мәтін .

Мәтін шрифті Arial типінде қызыл түспен жазылады.

Мысалы: мәтін .

Мәтін шрифті Verdana типінде көк түспен жазылады.

 тэгі мәтінді курсив шрифтпен бейнелейді. Тэг тэгімен жабылады. Мысалы: мәтін .

 логикалық екпін тэгі. Мәтінді логикалық түрде бөлектеу мүмкіндігін береді. Әдетте мәтін браузер терезесінде курсив шрифтпен жазылады. Мағынасы жағынан тэгіне жақын. Тэг тэгімен жабылады.

Мысалы: мәтін .

 тэгі мәтінді жуан шрифтпен бейнелейді. Тэг тэгімен жабылады. Мысалы: мәтін .

 күшейтірілген бөлектеу тэгі. Мәтінді логикалық түрде бөлектеу мүмкіндігін береді. Әдетте мәтін браузер терезесінде жуан шрифтпен бейнеленеді. Мағынасы жағынан тэгіне жақын. Тэг тэгімен жабылады.

Мысалы: мәтін .

<u> тэгі мәтіннің астын сызады. Тэг </u> тэгімен жабылады. Мысалы: <u> мәтін </u>.

<s> тэгі мәтінді жазып, үстінен сызады. Тэг </s> тэгімен жабылады. Мысалы: <s> мәтін </s>.

<big> тэгі мәтіннің өлшемін шамалы үлкейтіп бейнелейді. Тэг </big> тэгімен жабылады.

Мысалы: <big> мәтін </big>.

<small> тэгі мәтіннің өлшемін шамалы кішірейтіп бейнелейді. Тэг </small> тэгімен жабылады.

Мысалы: <small> мәтін </small>.

^{тэгі мәтіннің өлшемін шамалы кішірейтіп, дәреже сияқты жоғарыға бейнелейді. Тэг} тэгімен жабылады.

Мысалы: ^{мәтін}.

_{тэгі мәтіннің өлшемін шамалы кішірейтіп, индекс сияқты төменге бейнелейді. Тэг} тэгімен жабылады.

Мысалы: _{мәтін}.

<code> тэгі формула немесе программа кодын жазуда қолданылады. Тэг </code> тэгімен жабылады.

Мысалы: <code> y=a*x^2+b*x+c </code>.

<tt> тэгі мәтінді моношрифт типінде, барлық әріптердің енін бірдей етіп бейнелейді. Тэг </tt> тэгімен жабылады.

Мысалы: <tt> мәтін </tt>.

<kbd> тэгі тұтынушы пернетақтадан енгізуі қажет болған мәтінді жазуда қолданылады. Мәтін браузер терезесінде моношрифт типінде бейненленеді. Тэг </kbd> тэгімен жабылады.

Мысалы: <kbd> мәтін </kbd>.

<var> тэгінен мәтінде айнымалыны белгілеуде пайдаланады. Мәтін браузер терезесінде курсив түрінде бейненленеді. Тэг </var> тэгімен жабылады.

Мысалы: <var> мәтін </var>.

<cite> тэгі қысқа мәтінді цитата ретінде сипаттайды. Цитата браузер терезесінде курсив түрінде бейненленеді. Тэг </cite> тэгімен жабылады.

Мысалы: <cite> мәтін </cite>.

1.5 HTML тілінде тізімдерден пайдалану

Web-құжатта ақпаратты ықшам түрде бейнелеуде тізімдерден пайдаланады. Тізімдер екі түрге бөлінеді:

- нөмірленген;
- маркерленген.

Тізімдерді құру тэгтерін қарастырайық. Оларға төмендегі тегтер жатады:

- unsorted list
- ordered list
- list item
- <menu> <dir>
- <dl> definition list
- <dt> definition term
- <dd> definition description

 тэгі маркерленген тізім құрады. Тэг ішінде бірнеше <li> тэгтері қолданылуы тиісті. Тэг тэгімен жабылады.

Мысалы:

```
<p> «Гамлет» трагедиясын кім жазған:
<ul> <li> Шекспир </li>
<li> Монтень </li>
<li> Гомер </li>
<li> Рабле </li>
<li> Софокл </li> </ul> </p>
```

Браузер терезесінде төмендегі маркерленген тізім көрінеді:

«Гамлет» трагедиясын кім жазған:

- Шекспир
- Монтень
- Гомер
- Рабле
- Софокл

 тэгі нөмірленген тізім құрады. Тэг ішінде бірнеше <li> тэгтері қолданылуы тиісті. Тэг тэгімен жабылады. Тэг төмендегі атрибуттарға ие:

- start – атрибуты нөмір қай саннан басталатынын анықтайды. Мән ретінде тек қана бүтін санды алады;
- type – атрибуты нөмірлеу силін анықтайды. Қабылдайтын мәндері:
  - "A" – латын бас әріптері: A, B, C, …
  - "a" – кіші латын әріптері: a, b, c, …

- "I" – үлкен рим цифрлары: I, II, III, …
- "i" – кіші рим цифрлары: i, ii, iii, …
- "1" – араб цифрлары: 1, 2, 3, …

Егер type атрибуты берілмесе, онда нөмір араб цифрларымен жазылады.
Мысалы:

```
<p> «Гамлет» трагедиясын кім жазған:
<ol type="a"> <li> Шекспир </li>
<li> Монтень </li>
<li> Гомер </li>
<li> Рабле </li>
<li> Софокл </li> </ol> </p>
```

Браузер терезесінде төмендегі нөмірленген тізім көрінеді:

a. Шекспир
b. Монтень
c. Гомер
d. Рабле
e. Софокл

 тэгі тізім қатарын құрады. Тэг <ul> және <ol> тэгтері ішінде қолданылады. Тэгтің value атрибуты тек қана <ol> тэгінде қолданылады. Атрибут қатар нөмірінің басталу ретін көрсетеді. Тэг тэгімен жабылады.
Мысалы:

```
<p> «Гамлет» трагедиясын кім жазған:
<ol type="a"> <li value="3"> Шекспир </li>
<li> Монтень </li>
<li> Гомер </li>
<li> Рабле </li>
<li> Софокл </li> </ol> </p>
```

Браузер терезесінде төмендегі нөмірленген тізім көрінеді:

c. Шекспир
d. Монтень
e. Гомер
f. Рабле
g. Софокл

<menu> және <dir> тэгтері HTML 4.0 тілі тэгтеріне жатпайды. Олар нөмірленген тізім құруда <ol> тэгінің орнына пайдаланылуы мүмкін.

<dl> тэгі анықтамалар тізімін құрады. Анықтамалар термин (<dt> тэгі) және оның сипаттамасынан (<dd> тэгі) тұрады. Тэгтің compact атрибуты

тізімді ықшам түрде бейнелейді. Атрибуттың мәні жоқ. Тэг </dl> тэгімен жабылады.

Мысалы:

```
<dl compact>
<dt>  HTML <dd> гипермәтінді белгілеу тілі
<dt>  WWW <dd> бүкіләлемдік өрмек
</dl>
```

<dt> тэгі анықтамалар тізімінің терминін анықтайды. Ол <dl> тэгінің ішінде қолданылады. Тэг жабылу тэгісіз жазылады.

<dd> тэгі анықтамалар тізімінің сипаттамасын анықтайды. Ол <dl> тэгінің ішінде қолданылады. Тэг жабылу тегісіз жазылады.

1.6 Web-құжатқа сілтемелер, сурет және объекттер енгізу

HTML тілі Web-құжатқа суреттерді, кестелерді енгізу мүмкіндігін береді. Сонымен қатар желідегі басқа Web-құжаттармен байланысуда гиперсілтемелерден пайдаланады. Web-құжатқа сілтеме, сурет, объект, апплет және скрипттер енгізуді қарастырайық. Оларға төмендегі тэгтер жатады:

- <a> (anchor)
- <img> (image)
- <embed>
- <noembed>
- <applet>
- <param>
- <object>
- <marquee>

<a> тэгі гиперсілтемелерді құруда және пайдалануда қолданылады. Тэг тэгімен жабылады. Тэг төмендегі атрибуттарға ие:

- name – атрибуты бастапқы және жабылу тэгтері арасындағы мәтінді сілтеме жасау объекті деп анықтайды. Мән ретінде қайталанбайтын кез-келген сөзді латын әріптерімен жазуға болады.
- href – атрибуты бастапқы және жабылу тэгтері арасындағы мәтінді немесе суретті гиперсілтеме деп анықтайды. Қабылдайтын мәндері:
  - "http://..." – Web-құжатқа сілтеме жасайды;
  - "ftp://..." – ftp-сайтқа немесе құжатқа сілтеме жасайды;

        - "mailto:..." – пошталық клиент-программаны жүктейді. Ашылған өріске электрондық адрес пен жіберілетін мәтін жазылады;
        - "news:..." – жаңалықтар серверінің конференциясына сілтеме жасайды;
        - "telnet://…" – қашықтағы компьютерге telnet-программасымен байланысқа сілтеме жасайды;
        - "wais://…" – wais-серверге сілтеме жасайды;
        - "gopher://..." – gopher-серверге сілтеме жасайды.
    - title – атрибуты ағымдағы түсініктемені анықтайды.
    - target – атрибуты сілтемеде көрсетілген құжаттың бейнелену фрейм терезесін анықтайды. Қабылдайтын мәндері:
        - "_self" – href атрибутында анықталған құжат ағымдағы фреймде бейнеленеді;
        - "_parent" – href атрибутында анықталған құжаттың ағымдағы фреймнің аналық фреймінде бейнеленуі қажет екенін көрсетеді. Басқаша айтқанда ағымдағы фреймді қосушы frameset тегі сипатталған терезені көретеді;
        - "_top" – href атрибутында анықталған құжаттың бастапқы фреймде бейнеленуі қажет екенін көрсетеді;
        - "_blank" – href атрибутында анықталған құжаттың жаңа терезеде бейнеленуі қажет екенін көрсетеді.

Егер байланыс түрі мен URL-адрес көрсетілмесе бастапқы адрес ретінде ағымдағы бума алынады. Бұл салыстырмалы сілтеме жасауға мүмкіндік береді:

Мысалы: Информатика .

Бұл жерде ағымдағы буманың files ішкі бумасындағы text10.html файлына сілтеме жасалған.

Мысалы: соңына
тэгін құжат басына, ал
 соңы
тэгін құжат соңына жазайық. Браузер терезесінде құжат басында асты сызылған соңы сөзі басқа түспен шығып тұрады және ол құжаттың соңына сілтеме жасайды.

Мысалы: құжаттың кез-келген жеріне
 информатика
тэгі жазылсын, онда
 информатика

тэгі сол «информатика» сөзі жазылған бөлімге сілтеме жасайды. Браузер терезесіндегі басқа түспен бөлектенген информатика сөзін тышқан көрсеткішімен басу арқалы, сол бөлімге келеміз.

Мысалы: Информатика

Бұл жерде ағымдағы буманың files ішкі бумасындағы text10.html файлына сілтеме жасалған. Аталған файл жаңа терезеде бейнеленеді.

<img> тэгі Web-құжатқа суретті енгізуде қолданылады. Сурет jpg, gif немесе png форматтарының бірінде болуы тиіс. Тэг жабылу тэгісіз жазылады. Тэг төмендегі атрибуттарға ие:

- src – міндетті атрибут. Енгізілетін суреттің URL-адресін анықтайды;
- height – атрибуты суреттің биіктігін пиксельде анықтайды;
- width – атрибуты суреттің енін пиксельде анықтайды;
- hspace – атрибуты суреттің құжаттағы басқа объекттен горизонталь бойынша қашықтығын пиксельде анықтайды;
- vspace – атрибуты суреттің құжаттағы басқа объекттен вертикаль бойынша қашықтығын пиксельде анықтайды;
- align – міндетті атрибут. Құжатта суретті браузер терезесіне туралау түрін көрсетеді. Қабылдайтын мәндері:
- left – суретті сол жақ шетке туралайды. Ағымдағы мәтін суретті оң жағынан қоршайды;
- right – суретті оң жақ шетке туралайды. Ағымдағы мәтін суретті сол жағынан қоршайды;
- top – суреттің жоғарғы шетін ағымдағы мәтіннің негізгі сызығына туралайды;
- bottom – суреттің төменгі шетін ағымдағы мәтіннің негізгі сызығына туралайды;
- middle – суреттің ортасын ағымдағы мәтіннің негізгі сызығына туралайды;
- absmiddle – суреттің ортасын ағымдағы мәтіннің ортасына туралайды;
- absbottom – суреттің төменгі шетін ағымдағы мәтіннің төменгі сызығына туралайды;
- name – атрибуты суретті сілтеме жасау объекті деп анықтайды. Мән ретінде қайталанбайтын кез-келген сөзді латын әріптерімен жазуға болады;
- alt – атрибуты қосалқы мәтінді анықтайды. Егер браузер сурет файлын таба алмаса, онда терезеде суреттің орнында қосалқы мәтін пайда болады;
- border – атрибуты сурет рамкасының қалындығын пиксельде анықтайды;

- usemap – атрибуты суретті <map> тэгінде анықталған навигациялық картамен байланыстырады. Атрибуттың мәні <map> тэгінің name атрибуты мәнімен сәйкес болуы тиіс.
- ismap – атрибуты суретті сервер өңдейтін навигациялық картамен байланыстырады. Егер тұтынушы суретті тышқанмен шертсе, суреттің сол жерінің x, y координаталарын пиксель түрінде құжат орналасқан серверге жіберіледі. Сервер жіберілген координаталарды өңдеп, қажетті құжатты немесе суретті көрсетеді. Атрибуттың мәні жазылмайды;
- title – атрибуты ағымдағы түсініктемені анықтайды.

Мысалы: <img border="1" src="files/graf10.gif" width="700" height="79" align="left"> – браузер терезесінің сол жақ шетіне files бумасындағы graf10.gif суреті шығады.

Мысалы: <img border="1" src="files/graf10.gif" width="700" height="79" align=left hspace="10"> – браузер терезесінде мәтін files бумасындағы graf10.gif суретінің оң жағынан 10 пиксель ара қашықтық тастап жазылады.

Мысалы: <img src="files/graf10.gif" border="1" width="700" height="79" align=center> – браузер терезесінде graf10.gif суреті files бумасындағы text00.html құжатына сілтеме жасайды.

<embed> тэгі құжатқа басқа объекттерді (html-емес құжаттар) енгізуде қолданылады. Бірақта көптеген браузерлер бұл тэгті қолдамайды. Тэг </embed> тэгімен жабылады. Тэг <img> тэгі сияқты орындалады. Сондықтан src, height, width, hspace, vspace, align, name, alt, border атрибуттары бірдей. Басқа қосымша атрибуттары:

- pluginspace – атрибуты енгізілген объектті өңдейтін плагин (plug-in) орналасқан URL-адресті анықтайды. Мысалы: mp3 файлдары үшін қажетті плэйерді (player) көрсету қажет;
- palette – атрибуты енгізілген объект үшін түс палитрасы түрін анықтайды. Бұл атрибут тек қана Windows жүйесінде 256 түсті палитрада ғана қолданылады. Қабылдайтын мәндері:
  - background – объектті background-палитрасымен бейнелейді (үндеместен қолданылады);
  - foreground – объектті foreground-палитрасымен бейнелейді;
- hidden – атрибуты енгізілген объектті көрсету қажет пе, жоқ па, соны анықтайды. Қабылдайтын мәндері:
  - true – объектті көрсетпейді;
  - false – объектті көрсетеді (үндеместен қолданылады);

- type – атрибуты браузерге объектті өңдеуші плагинді анықтау мақсатында, енгізілетін объект файлының типін ашық көрсетеді. Атрибуттың мәні ретінде тіркелген MIME-тип файлы жазылады;
- title – атрибуты ағымдағы түсініктемені анықтайды.

<noembed> тэгі, егер <embed> тэгі жұмыс істемесе пайдаланылатын мәтінді анықтайды. Тэг </noembed> тэгімен жабылады.

Мысалы: <embed src="music.mid" hidden="true" width="700" height="100"> <noembed> Сіздің браузеріңізде бұл тэг орындалмайды; </noembed> </embed>.

<applet> тэгі құжатқа Java-апплет (Java тілінде құрылған программалар) енгізуде қолданылады. Тэг </applet> тэгімен жабылады. Браузерде Java-апплет құжатта енгізілген сурет сияқты бейнеленіп, орындалады. Сондықтан <img> тэгі сияқты src, height, width, hspace, vspace, align, name, alt, border атрибуттарының қолданылуы бірдей. Басқа қосымша атрибуттары:

- code – міндетті атрибут. Сілтеме жасалған Java-апплеттің атын анықтайды;
- codebase – атрибуты Java-апплет файлының негізгі URL-адресін анықтайды. Егер атрибут жазылмаса ағымдағы құжаттың URL-адресін қабылдайды.
- align – міндетті атрибут. Java-апплетті суретті браузер терезесіне туралау түрін көрсетеді;
- title – атрибуты ағымдағы түсініктемені анықтайды.

Мысалы:
<applet code="JumpingGirl.class" width="30" height="40" align="left">
мәтін </applet> – браузер терезесінде JumpingGirl.class файлы орындалады.

<param> тэгі <applet> тэгінің ішінде Java-программаға мән енгізуде қолданылады. Тэг жабылу тэгісіз жазылады. Мысалы:

```
applet code="JumpingGirl.class" width="30" height="40" align="left">
<param name="width" value="10">
Java-апплет
 </applet>
```

<object> тэгі құжатқа басқа объекттерді (html-емес құжаттар) енгізуде қолданылады. HTML 4.0 тілінде бұл тег <embed> және <applet> тэгтерінің орнына енгізілген.

<marquee> тэгін құжатқа жүгіруші мәтін қатарын енгізуде пайдаланады. Тэг </marquee> тэгімен жабылады. Тэг төмендегі атрибуттарға ие:

- behavior – атрибуты мәтіннің браузер терезесінде көріну режимін анықтайды. Қабылдайтын мәндері:
- alternate – мәтін терезе шетіне келіп артқа қайтады;
- scrool – мәтін терезеде қайта-қайта айналып жүгіріп отырады (үндеместен қолданылады);
- slide – мәтін терезенің сол жағына келіп тоқтайды.
- bgcolor – фон түсін анықтайды;
- loop – мәтіннің қайталану санын көрсетеді. Қабылдайтын мәндері:
  - 0 - 9 – қайталану саны;
  - infinite – шексіз (үндеместен қолданылады);
- direction – жүгіруші мәтіннің бағытын көрсетеді. Мәндері:
  - down – жоғарыдан төменге;
  - left – оңнан солға (үндеместен қолданылады);
  - right – солдан оңға;
  - up – төменнен жоғарыға;
- lang – мәтіннің негізгі тілін анықтайды. Бұл ақпарат іздеу серверлерінде, дыбыстық синтезаторларда пайдаланылады. Мысалы: en – ағылшынша; ru – орысша;
- scrolldelay – жүгіруші мәтіннің жылдамдығын миллисекундда анықтайды;
- scrollamount – жүгіруші мәтіннің жылдамдығын пиксельде анықтайды.

Мысалы: <marquee direction="right"> жүгіруші мәтін қатары </marquee>

2 Web-құжатты безендіру және CSS технологиясы

2.1 Web-құжатта кестелерден пайдалану

Web-құжаттармен жұмыс істеу барысында реттелген ақпаратты жиі пайдаланамыз. Осындай реттелген, ықшам ақпаратты сақтаудың ең қолайлы тәсілі кестелер болып табылады. Кестедегі қатар мен баған қиылысқан орынды ұяшық деп атаймыз. Ұяшықтың ішінде мәтінмен қатар, сурет, шағын кесте де болуы мүмкін. HTML тілінің кестелермен жұмыс істейтін тэгтерін қарастырайық. Оларға төмендегі тэгтер жатады:

- <table>
- <caption>
- <tr> (table row)
- <td> (table data)
- <th> (table head)

<table> тэгі Web-құжатқа кесте құру үшін қажет. Тэг </table> тэгімен жабылады. Кестенің ұяшықтары <caption>, <tr>, <td> және <th> тэгтерімен құрылады. Тэг төмендегі атрибуттарға ие:

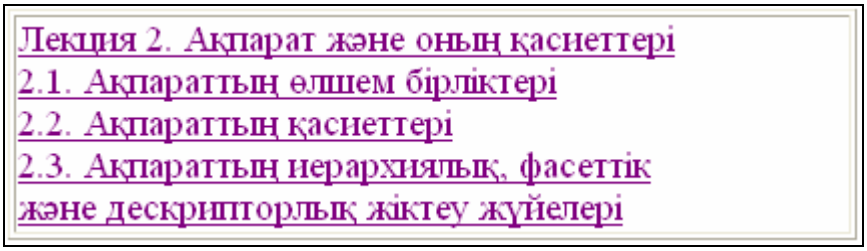
- align – атрибуты кестені браузер терезесінде left-сол жақ шетке, right-оң жақ шетке және center-ортаға туралайды. Үндеместен left мәні қолданылады;
- valign – атрибуты кестені браузер терезесінде тігінен туралайды. Көптеген браузерлер бұл атрибутты бейнелемейді. Мәндері:
  - top – төменнен;
  - bottom – жоғарыдан;
  - middle – ортадан (үндеместен қолданылады);
- border – атрибуты кесте рамкасының қалындығын пиксельде анықтайды. Егер атрибут жазылмаса, онда рамка көрінбейді;
- cellpadding – атрибуты кестенің әр бір ұяшығындағы мәтіннің ұяшық рамкасына дейінгі қашықтықты пиксельде анықтайды;
- cellspacing – атрибуты кестенің жалпы рамкасы мен әр бір ұяшығының рамкасы арасындағы қашықтықты пиксельде анықтайды;
- width – атрибуты кестенің енін анықтайды. Кестенің ені пиксельде немесе браузер терезесіне сәйкес пайызда беріледі. Егер атрибут жазылмаса, кестенің ені енгізілген мәтін ұзындығы бойынша анықталады;
- height – атрибуты кестенің биіктігін анықтайды. Кестенің биіктігі пиксельде немесе браузер терезесіне сәйкес пайызда беріледі. Егер атрибут жазылмаса, кестенің биіктігі енгізілген мәтін бойынша анықталады;

- bgcolor – атрибуты кестенің фон түсін анықтайды. Түс RGB (red, green, blue) моделімен беріледі;
- background – атрибуты кесте фоны ретінде суретті алады. Атрибуттың мәні ретінде суреттің URL-адресі беріледі;
- title – атрибуты ағымдағы түсініктемені анықтайды.

Мысалы:

```
<table width="50%" border="1">
<tr> <td><p>
<a href=files/text10.html>Лекция 2. Ақпарат және оның
қасиеттері</a></br>
<a href=files/text11.html>2.1. Ақпараттың өлшем бірліктері</a></br>
<a href=files/text12.html>2.2. Ақпараттың қасиеттері</a></br>
<a href=files/text13.html>2.3. Ақпараттың иерархиялық, фасеттік
</br>
және дескрипторлық жіктеу жүйелері</a></p></td></tr>
</table>
```

Браузер терезесінде «Ақпарат және оның қасиеттері» кестесі пайда болады (сурет 5):



Сурет 5 – «Ақпарат және оның қасиеттері» кестесі

<caption> тэгі кестенің тақырыбын анықтайды. Тақырып тек қана мәтіннен тұрады. Тэг </caption> тэгімен жабылады. Тэгтің align атрибутының top мәні (үндеместен қолданылады) тақырыптың кесте басында, ал bottom мәні соңында болуын анықтайды.

<tr> тэгі кестенің ұяшықтар қатарын анықтайды. Тэг </tr> тэгімен жабылады. Ұяшықтар <td> және <th> тэгтерімен құрылады. Тэг төмендегі атрибуттарға ие:

- align – атрибуты ұяшық мәтінін left –сол жақ шетке, right –оң жақ шетке және center –ортаға туралайды. Үндеместен left мәні қолданылады;
- valign – атрибуты ұяшық мәтінін браузер терезесінде тігінен туралайды. Қабылдайтын мәндері:
  - top – төменнен;
  - bottom – жоғарыдан;
  - middle – ортадан (үндеместен қолданылады);

- bgcolor – атрибуты кесте ұяшығының фон түсін анықтайды.

<td> тэгі кестенің ағымдағы қатарында ұяшық құрады. Тэг </td> тэгімен жабылады. Тэг төмендегі атрибуттарға ие:

- align – атрибуты ұяшық мәтінін браузер терезесіне туралайды. Егер атрибут жазылмаса, онда <tr> тэгінің align атрибутының мәні үндеместен алынады. Қабылдайтын мәндері:
  - left – сол жақ шетке;
  - right – оң жақ шетке;
  - center – ортаға;
- valign – атрибуты ұяшық мәтінін браузер терезесінде тігінен туралайды. Егер атрибут жазылмаса, онда <tr> тэгінің valign атрибутының мәні үндеместен алынады. Қабылдайтын мәндері:
  - top – төменнен;
  - bottom – жоғарыдан;
  - middle – ортадан (үндеместен қолданылады);
- width – атрибуты ұяшықтың енін анықтайды. Ұяшықтың ені пиксельде немесе кесте еніне сәйкес пайызда беріледі;
- height – атрибуты ұяшық биіктігін анықтайды. Ұяшықтың биіктігі пиксельде немесе кесте биіктігіне сәйкес пайызда беріледі;
- colspan – атрибуты кесте ұяшығы қанша бағанды біріктіретінін анықтайды. Егер жазылмаса атрибуттың мәні үндеместен бірге тең;
- rowspan – атрибуты кесте ұяшығы қанша қатарды біріктіретінін анықтайды. Егер жазылмаса атрибуттың мәні үндеместен бірге тең;
- nowrap – атрибуты ағымдағы ұяшықта мәтінді тасымалдауға тиім салады;
- bgcolor – атрибуты кесте ұяшығының фон түсін анықтайды;
- background – атрибуты ұяшықтың фоны ретінде суретті алады. Атрибуттың мәні ретінде суреттің URL-адресі беріледі.

<th> тэгі <td> тэгі сияқты кестенің ағымдағы қатарында ұяшық құрады, тек ұяшық мәтінін тақырып деп анықтайды. Мұндай анықтама браузерге тақырып пен мәтінді бөлектеу мүмкіндігін береді. Тэгтің атрибуттары <td> тэгінің атрибуттарына ұқсас. Тэг </th> тэгімен жабылады.

Кесте құруға мысалдар қарастырайық. Қарапайым кесте құрамыз:

```
<table width="50%" border="1">
<tr>
   <td> 1 </td>
   <td> 2 </td>
   <td> 3 </td>
```

```
</tr>
<tr>
    <td> 4 </td>
    <td> 5 </td>
    <td> 6 </td>
</tr>
</table>
```

Браузер терезесінде келесі қарапайым кесте пайда болады (сурет 6):

1	2	3
4	5	6

Сурет 6 – қарапайым кесте

Кестеге жалпы фон түсін, ал кейбір ұяшықтарға басқа фон түсін беруге болады. Кейбір ұяшықтардың мәтінін пішімдейік.

```
<table width="50%" border="1" >
<tr>
    <td align="center"> 1 </td>
    <td align="center"> 2 </td>
    <td align="center"> 3 </td>
</tr>
<tr>
    <td bgcolor="#ff0000"> 4 </td>
    <td bgcolor="#ffffff"> 5 </td>
    <td bgcolor="#0000ff"> 6 </td>
</tr>
</table>
```

Браузер терезесінде келесі пішімделген кесте пайда болады (сурет 7):

1	2	3
4	5	6

Сурет 7 – пішімделген кесте

Кесте ұяшықтарын бір-біріне қосуға болады. Кейбір ұяшық мәтіндерінің түсін өзгертіп көрейік.

```
<table width="50%" border="1" bgcolor="#e0e0e0">
<tr>
    <td align="center" colspan="2"> <font color="#ff0000"> 1+2
    </font></td>
    <td align="center" rowspan="2" width="25%"
```

```
    bgcolor="#0000ff">3+6
    </td>
</tr>
<tr>
    <td bgcolor="ff0000"> 4 </td>
    <td bgcolor="#ffffff" > 5 </td>
</tr>
</table>
```

Браузер терезесінде келесі ұяшықтары қосылған кесте пайда болады (сурет 8):



Сурет 8 – ұяшықтары қосылған кесте

2.2 Web-құжатта қалыптардан пайдалану

Web-құжатта тұтынушымен ақпарат алмасу қажет болса, онда қалыптан пайдаланады. Әдетте қалыптан тұтынушыны тіркеуде, әр-түрлі мәліметтер жинауда, сұранысты қабылдауда пайдаланады. Қалып тұтынушыға арнайы тек толтыруға рұқсат етілген өрістерге деректер енгізуді ұсынады. Енгізілген деректерді Web-сервердегі арнайы программа (скрипт) өңдейді. HTML 4.0 тілінің қалыптармен жұмыс істейтін тэгтерін қарастырайық. Оларға төмендегі тэгтер жатады:

- <form>
- <textarea> (text area)
- <select>
- <option>
- <input>

<form> тэгі қалып құрушы негізгі тэг. Тэг </form> тэгімен жабылады. Тэгтің ішінде HTML тілінің көптеген тэгтерінен пайдалануға болады. Тэг төмендегі атрибуттарға ие:

- name – атрибуты қалыптың өзіндік атын анықтайды. Атрибуттан құжатта бірнеше қалып болғанда пайдаланады;
- action – міндетті атрибут. Қалыпқа енгізілген деректерді өңдеуші скрипттің URL-адресін анықтайды;
- method – атрибуты қалыптың мазмұнын сервердегі скриптке жіберу әдісін таңдайды. Қабылдайтын мәндері:
    - get – параметрлерді ашық жіберу (үндеместен қолданылады);

    - post – параметрлерді жасырын жіберу;
- enctype – атрибуты қалыптың мазмұнын сервердегі скриптке жіберудің post әдісіндегі деректердің MIME-типін анықтайды;
- target – атрибуты қалып мазмұнын өңдеген сервердегі скрипттің нәтижелері бейнеленетін фреймнің атын анықтайды. Ал егер <base> тэгі қоданылса, онда үндеместен қолданылатын фреймнің атын анықтайды. Қабылдайтын мәндері:
    - _self – нәтиже ағымдағы фреймде бейнеленеді;
    - _parent – нәтиже ағымдағы фреймнің аналық фреймінде бейнеленуі қажет екенін көрсетеді;
    - _top – нәтиже бастапқы фреймде бейнеленуі қажет екенін көрсетеді; (үндеместен қолданылады);
    - _blank – нәтиже жаңа фреймде бейнеленуі қажет екенін көрсетеді;
    - нақты көрсетілген фреймнің аты;
- accept-charset – атрибуты құжатқа енетін нәтиженің код стандартын көрсетеді. Қабылдайтын мәндері:
    - ISO-8859-1 – жалпы стандарт;
    - Windows-1250 – ағылшын тіліндегі мәтіндер үшін;
    - Windows-1251 – орыс тіліндегі мәтіндер үшін;
- dir – атрибуты мәтіннің бейнелену бағытын көрсетеді. Қабылдайтын мәндері:
    - ltr – солдан оңға;
    - rtl – оңнан солға;
- lang – мәтіннің негізгі тілін анықтайды. Мысалы: en – ағылшынша; ru – орысша;
- title – атрибуты ағымдағы түсініктемені анықтайды.

<textarea> тэгі қалыпта мәтіндік аймақты анықтайды. Тэг <form> тэгінің ішінде жазылып, </textarea> тэгімен жабылады. Тэг төмендегі атрибуттарға ие:

- name – міндетті атрибут, өрістің өзіндік атын анықтайды;
- rows – атрибуты өріс терезесінде мәтіннің көрінуге тиісті қатар санын анықтайды;
- cols – атрибуты өріс терезесінде мәтіннің көрінуге тиісті баған санын анықтайды;
- wrap – атрибуты мәтіннің сөздерін тасымалдау әдісін анықтайды. Қабылдайтын мәндері:
    - off – мәтін сөздері тасымалданбайды;
    - virtual – мәтін сөздері тасымалданады, бірақта сервердегі скриптке тасымалданбай жіберіледі;
    - physical – мәтін сөздері тасымалданады;
- title – атрибуты ағымдағы түсініктемені анықтайды;
- lang – мәтіннің негізгі тілін анықтайды;

- dir – атрибуты мәтіннің бейнелену бағытын көрсетеді. Қабылдайтын мәндері:
  - ltr – солдан оңға;
  - rtl – оңнан солға;
- align – атрибуты мәтінді браузер терезесіне туралайды. Қабылдайтын мәндері:
  - left – сол жақ шетке;
  - right – оң жақ шетке;
  - baseline – негізгі сызыққа;
  - top – жоғары шетке;
  - bottom – төменгі шетке;
  - center – ортаға;
  - justify – еніне;

Мысалы:

```
<form action="mailto:us-ibr@rambler.ru" method=post>
<textarea name="comments" wrap="virtual" cols="40" rows="3">
</textarea><br>
<input type="submit" value="ok">
</form>
```

Браузер терезесінде келесі қарапайым қалып пайда болады (сурет 8):



Сурет 8 – қарапайым қалып

Қалыпқа енгізілген мәтінді, браузер элекрондық поштаға жібереді. Көрсетілген мысалда mailto: – скрипттің аты, us-ibr@rambler.ru – электрондық адрес.

<select> тэгі қалыпта мәзірді анықтайды. Тұтынушы мәзір ішінен қажеттісін таңдай алады. Тэг <form> тэгінің ішінде жазылып, </select> тэгімен жабылады. Тэг ішінде бірнеше <option> тэгі жазылуы тиіс. Тэг төмендегі атрибуттарға ие:

- name – атрибуты мәзірдің өзіндік атын анықтайды;
- title – атрибуты ағымдағы түсініктемені анықтайды;
- lang – мәзірдің негізгі тілін анықтайды;
- dir – атрибуты мәтіннің бейнелену бағытын көрсетеді. Қабылдайтын мәндері:
  - ltr – солдан оңға;

        - rtl – оңнан солға;
    - align – атрибуты мәзір мәтінін браузер терезесіне туралайды. Қабылдайтын мәндері:
        - left – сол жақ шетке;
        - right – оң жақ шетке;
        - baseline – негізгі сызыққа;
        - top – жоғары шетке;
        - bottom – төменгі шетке;
        - center – ортаға;
        - justify – еніне;
    - multiple – атрибуты мәзірдің бірнеше тақырыбын таңдау мүмкіндігін береді. Бірнеше тақырыпты таңдауда Ctrl пернесін басып тұру қажет. Атрибуттың мәні жоқ. Егер атрибут жазылмаса, мәзірдің тек қана бір тақырыбын таңдау мүмкін.

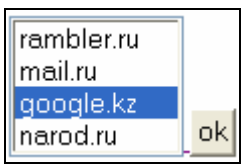
<option> тэгі қалыптағы мәзірдің тақырыптарын анықтайды. Тэг <select> тэгінің ішінде жазылып, жабылу тэгі болмайды. Тэгтің атрибуттары:

- value – атрибуты ағымдағы тақырыпқа берілетін мәнді анықтайды. Егер атрибут берілмесе мән ретінде тақырыптың мазмұны алынады;
- selected – логикалық атрибуты ағымдағы тақырыпты бөлектеп, мазмұнын бастапқы мән ретінде қабылдайды.

Мысалы:

```
<form action="mailto:us-ibr@name" method=post>
  <select name="firstname" multiple>
    <option value="rumbler">rambler.ru
     <option value="mail">mail.ru
     <option value="google" selected>google.kz
     <option value="narod">narod.ru
   </select>
  <input type="submit" value="ok">
</form>
```

Браузер терезесінде келесі тақырыптары анықталған қалып пайда болады (сурет 9):



Сурет 9 – тақырыптары анықталған қалып

<input> тэгі қалыптың өрісін анықтайды. Өрістің мазмұнын тұтынушы анықтайды немесе орындауға жібереді. Тэг <form> тэгінің ішінде жазылып, жабылу тэгі болмайды. Тэгтің атрибуттары:

- name – атрибуты өрістің өзіндік атын анықтайды;
- type – атрибуты өрістің типін анықтайды. Қабылдайтын мәндері:
  - text – мәтіндік өріс;
  - textarea – мәтіндік аймақ;
  - file – қалыпқа файлды біріктіреді;
  - password – мәтіндік өріс. Енгізілген мәтіннің орнына * символы шығады;
  - checkbox – чекбокс, логикалық типті (ия/жоқ) немесе бірнеше пернелі тізімді өріс. Бірнеші өрісті қатар белгілеуге болады. Таңдалған әрбір өрістің өзіндік name/value мәнін серверге жібереді. Осы типті өрістің міндетті түрде name, value және міндетті емес checked атрибуттары болуы тиіс;
  - radio – берілген тізімнің біреуін таңдау өрісі. Тізімдегі барлық пернелер (radio buttons) бірдей аталуы тиіс, олардың тек қана біреуі белгіленеді. Таңдалған өрістің name/value мәні серверге жіберіледі. Осы типті өрістің міндетті түрде name, value және міндетті емес checked атрибуттары болуы тиіс;
  - submit – перне өрісі. Құрылған пернені бассақ, қалып деректері серверге жіберіледі. Value атрибутының мәні перне үстіндегі жазуды көрсетеді;
  - image – графикалық перне өрісі. Графикалық суреттің URL-адресі src атрибутының мәні ретінде көрсетіледі. Серверге тышқан көрсеткішімен шертілген суреттегі орынның x, y координаталары жіберіледі. Координаталар өсі терезенің сол жақ жоғары бұрышынан басталады. Серверге жіберілетін координаталар name.x/value және name.y/value атауына ие.
  - reset – перне өрісі; Құрылған пернені бассақ, қалып өрістеріндегі мәтіндер өшеді, деректер серверге жіберілмейді. Value атрибутының мәні перне үстіндегі жазуды білдіреді;
  - hidden – жасырын өріс, терезеде бейнеленбейді. Өрістің мазмұны name/value басқа мәндер сияқты, серверге жіберіледі. Әдетте тұтынушыға білдірмей, скрипттен скриптке мәлімет жіберуде қолданылады;
  - button – перне өрісі, скрипттің функционалдық перне қызметін атқарады. Value атрибутының мәні перне үстіндегі жазуды білдіреді;
- value – атрибуты өрістің атын, перне үстіндегі жазуды анықтайды;
- checked – логикалық атрибуты тізімдегі тақырыпты бөлектеп, мазмұнын бастапқы мән ретінде қабылдайды;
- size – атрибуты мәтіндік өріс өлшемін символда анықтайды;

- maxlength – атрибуты өріске енгізілетін мәтіннің ұзындығын анықтайды;
- src – атрибуты графикалық перненің URL-адресін анықтайды;
- align – атрибуты графикалық пернені браузер терезесіне туралау түрін көрсетеді. type="image" атрибутымен бірге қолданады. Қабылдайтын мәндері:
  - left – графикалық пернені сол жақ шетке туралайды. Ағымдағы мәтін пернені оң жағынан қоршайды;
  - right – графикалық пернені оң жақ шетке туралайды. Ағымдағы мәтін пернені сол жағынан қоршайды;
  - top – графикалық перненің жоғарғы шетін ағымдағы мәтіннің негізгі сызығына туралайды;
  - bottom – графикалық перненің төменгі шетін ағымдағы мәтіннің негізгі сызығына туралайды;
  - middle – графикалық перненің ортасын ағымдағы мәтіннің негізгі сызығына туралайды;
  - absmiddle – графикалық перненің ортасын ағымдағы мәтіннің ортасына туралайды;
  - absbottom – графикалық перненің төменгі шетін ағымдағы мәтіннің төменгі сызығына туралайды;
- accept – атрибуты файлдың өрістің MIME-типін анықтауда, type="file" атрибутымен бірге қолданылады. Мысалы: accept="text/html", "image/png", "image/gif", "text/tcl", "text/javascript", "text/vbscript"
- title – атрибуты ағымдағы түсініктемені анықтайды.
- alt – атрибуты қосалқы мәтінді анықтайды. Егер браузер файлды таба алмаса, онда терезеде өрістің орнында қосалқы мәтін пайда болады;
- dir – атрибуты өрістегі мәтіннің бейнелену бағытын көрсетеді. Қабылдайтын мәндері:
  - ltr – солдан оңға;
  - rtl – оңнан солға.

2.3 Web-құжатта фреймдерден пайдалану

Web-құжатта фреймдерден пайдалану, құжатты рәсімдеу мүмкіндіктерін арттырады. Фреймдер браузер экранын бірнеше терезеге бөледі. Әр бір терезеде құжаттың жеке бір парағы бейнеленеді. Осы технология негізінде браузерге бірнеше Web-құжатты бір уақытта жүктеу мүмкін. HTML 4.0 тілінің фреймдермен жұмыс істейтін тэгтерін қарастырайық. Оларға төмендегі тэгтер жатады:

- <frameset> (frame set)
- <frame>

- <noframes>

<frameset> тэгі құжаттың фреймдік құрылымын сипаттайды. Тэг фреймдердің өлшемін, олардың экрандағы орнын анықтайды. Тэг </frameset> тэгімен жабылады. Бұл тэг құжатта <body> тэгінің орнына, яғни құжаттың негізгі бөлімі ретінде қолданылады. Тэгтің ішінде <frame> және <noframe> тэгтері жазылады. Тэг төмендегі атрибуттарға ие:

- rows – атрибуты браузер терезесіндегі горизонталь фреймдер саны мен өлшемдерін анықтайды. Фреймдер өлшемі тізбектеліп, төмендегі тәсілдердің бірімен беріледі:
  - браузер терезесі биіктігіне сәйкес пайыз – % түрінде. Мысалы: "30%, 30%, 40%";
  - браузер терезесінің барлық көлемі – * түрінде. Мысалы: "25%, 25%, *";
  - пиксельде – бүтін сан түрінде. Мысалы: "200, *";
- cols – атрибуты браузер терезесіндегі вертикаль фреймдер саны мен өлшемдерін анықтайды. Фреймдер өлшемі тізбектеліп, төмендегі тәсілдердің бірімен беріледі:
  - браузер терезесі еніне сәйкес пайыз – % түрінде. Мысалы: "30%, 30%, 40%";
  - браузер терезесінің барлық көлемі – * түрінде. Мысалы: "25%, 25%, *";
  - пиксельде – бүтін сан түрінде. Мысалы: "200, *";
- border – атрибуты фреймдердің рамкасын пиксельде анықтайды;
- frameborder – атрибуты фреймдердің рамкасы бар немесе жоқтығын анықтайды. Мәндері: yes, no немесе 1, 0;
- framespacing – атрибуты фреймдер арасының қашықтығын (бос аймақ) пиксельде анықтайды;

<frame> тэгі фреймді және оның қасиеттерін анықтайды. Тэг жабылу тэгісіз, <frameset> тэгінің ішінде жазылады. Тэг төмендегі атрибуттарға ие:

- src – міндетті атрибут, фреймде бейнеленетін html-файлдың URL-адресін анықтайды;
- name – атрибуты фреймнің өзіндік атын анықтайды. Бұл атау басқа құжаттан сілтеме жасалғанда target атрибутының мәні ретінде беріледі;
- marginwidth – атрибуты фрейм мазмұнының терезеден сол жақ және оң жақ шеттен ара қашықтығын пиксельде анықтайды;
- marginheight – атрибуты фрейм мазмұнының терезеден жоғарғы және төменгі шеттен ара қашықтығын пиксельде анықтайды;
- scrolling – атрибуты фрейм терезесінің айналдыру сызғышын анықтайды. Қабылдайтын мәндері:
  - yes – айналдыру сызғышы бейнеленеді;
  - no – бейнеленбейді;

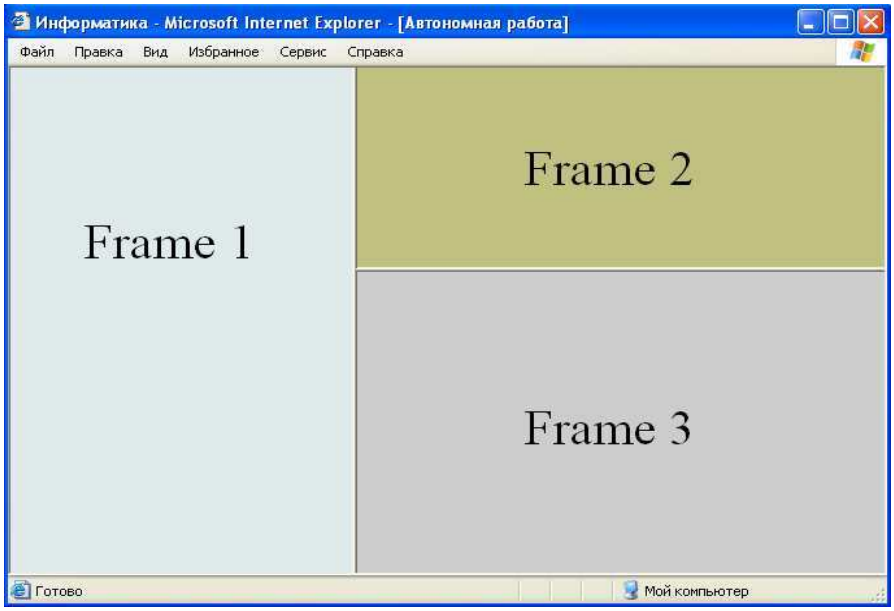
- auto – қажет болса бейнеленеді;
- noresize – атрибуты фрейм өлшемдерін өзгертуге тиім салады;
- frameborder – атрибуты фреймнің рамкасын анықтайды. Мәндері: yes, no немесе 1, 0;

<noframe> тэгі, егер браузер фреймді қолдамаса, терезеге шығатын мәтінді анықтайды. Тэг </noframe> тэгімен жабылады. Тэг <frameset> тэгінің ішінде атрибутсыз жазылады.

Мысалы:

```
<frameset frameborder="1" framespacing="0" border="1" cols="250,*">
  <frame src="info1.html" name="page1">
    <frameset rows="160,*">
      <frame src="info2.html" name="page2" marginwidth="0">
      <frame src="info3.html" name="page3" marginwidth ="0">
    </frameset>
     <noframes> фреймдер </noframes>
</frameset>
```

Браузер терезесінде келесі 3 фреймді терезе пайда болады (сурет 10):



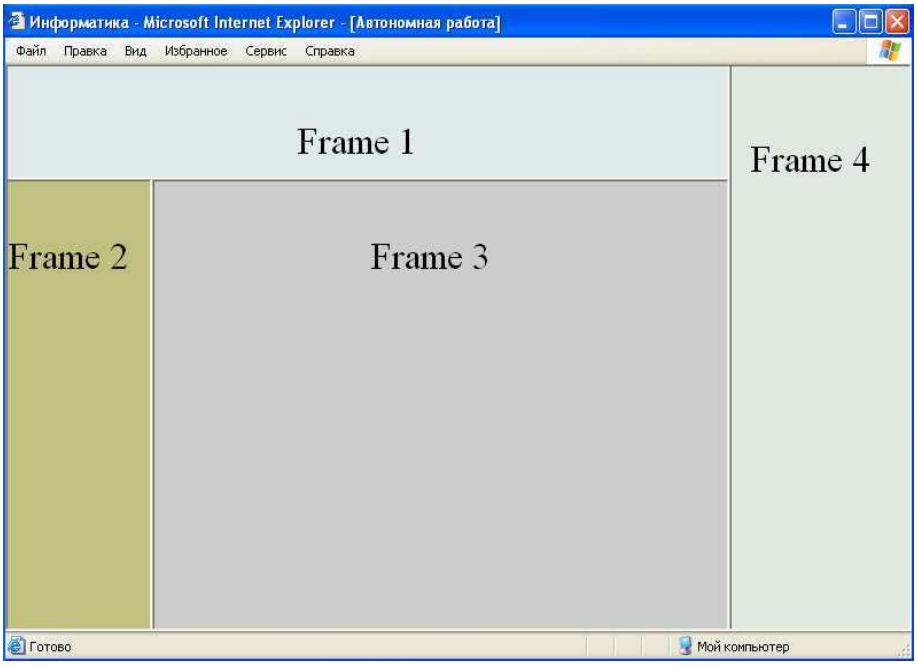
Сурет 10 – үш фреймді терезе

Мысалы:

```
<frameset frameborder="1" framespacing="0" border="1" cols="80%,*">
  <frameset rows="20%,*">
    <frame src="info1.html" name="page1" marginwidth="0">
      <frameset cols="20%,*">
       <frame src="info2.html" name="page2" marginwidth ="1">
       <frame src="info3.html" name="page3" marginwidth ="1">
      </frameset>
  </frameset>
  <frame src="info4.html" name="page4">
     <noframes> сіддің браузер фреймді көрсетпейді </noframes>
```

</frameset>

Браузер терезесінде келесі фреймдер пайда болады (сурет 11):



Сурет 11 – төрт фреймді терезе

2.4 CSS технологиясы

Интернетте көп тараған HTML 3.2 гипермәтін құжаттарды «физикалық форматтау» әдісімен ұйымдастырады. Web-форматтаудың бұл әдісінде тэгтер мен олардың атрибуттары әрбір жаңа абзацта қайталанады. Бұл Web-құжаттардың бастапқы кодтағы мәтін көлемінің ұлғаюына әкеледі. «Физикалық форматтауда» Web-құжаттың құрылымына талдау жасау, браузерлерге, ақпаратты іздеу жүйелеріне көп уақыт талап етеді, нәтижеде Web-құжат монитор терезесінде өте жай бейнеленеді.

Қазіргі таңда Web-құжат құруда «логикалық форматтау» әдісін қолданған қолайлы. Бұл әдісті Интернет консорциум ұсынған HTML 4.0 тілінде қарастырамыз. «Логикалық форматтау» Web-құжаттың құрылымы мен сипаттауын бөлектеуге әкеледі. Нәтижеде Web-құжаттың көлемі мен оның жедел жадыға жүктелу уақыты қысқарады және Интернеттегі ақпаратты іздеу жүйелері оларды жеңіл талдай алады, қажетті гипермәтін мультимедия-жобаға берілген форматта жеткізіледі.

Интернет немесе локалдық жүйеде Web-сайттар ұйымдастыруда Web-құжаттардан пайдаланамыз. Егер Web-құжаттар бір-біріне тәуелсіз компьютерлерде сақталатынын және оларды виртуал жинақтау керектігін еске алсақ, онда Web-құжаттарға жалпы стильдерді, шрифттерді, форматтарды, фильтрлерді, т.с.с. енгізуге қажеттілік туады. Web-құжаттарда барлық гипермәтіндер бір түрлі форматта, стильде болуы

тұтынушыға қолайлылық тұғызады. Web-құжаттардағы дизайн мен анимация да көп роль ойнайды.

Әрине бұл айтылғандар Web-құжаттарды құруда «классикалық әдіс» болып есептеледі. Бірақта Интернетте Web-сайттарды құруда, коллекциядағы гипермәтіндерді жинақтауда, басқа әдістер қолданған жөн. Web-кеңістікті дұрыс пайдалану мәселесіне де көңіл бөлу қажет. Визуалды Web-редакторлар Web-құжаттарды құруда тэгтерден өте көп пайдаланады, бірақ оларды тиімді пайдаланбайды. Web-құжаттардың бастапқы кодтағы мәтіндерін тұтынушының өзі қарапайым редактормен тиімділейді.

Web-құжаттың құрылымы мен сипаттауын бөлектеуді CSS (Cascade Style Sheets – каскадты стильдер кестесі) технологиясын қолдана отырып орындаймыз. CSS технологиясы Web-құжатты сипаттауда қосымша файлдардан, шаблондар мен стильдерден пайдалануға көмектеседі. Бұл технологияны NetsCape Navigator мен Internet Explorer браузерлері қолдайды.

CSS технологиясы HTML тілінің кез келген тегіне өзіндік стильді визуалды (inline style) енгізе алады. Егер енгізілген стиль HTML-дің <body> тегіне орналасса, онда бұл стильден басқа абзацтар мұрагерлік қасиетімен автоматтық түрде пайдалана алады. Егер абзацта <font> тегі және оның colour, size атрибуттары айқын сипатталса, онда абзац полиморфизм қасиетімен сипатталған стильді аладі.

Әдетте Web-құжаттардың негізгі бөлімдерінде жалпы стильдерден пайдаланылады. Мысалы: гипермәтіннің барлық абзацтары 12 пункт өлшемді, Times New Roman шрифтты, жасыл реңді болсын. Бұл үшін негізгі форманың <body> тегіне Style атрибутын енгіземіз:

```
<body
style="font-family: "Times New Roman",serif;
font-size: 12 pt;
colour: green;">
```

Жоғарыдағы мысалда Times New Roman шрифты Serif эффектімен берілген. Егер пайдаланушының компьютерінде бұл шрифт болмаса, онда ол браузермен Serif эффекті бар басқа шрифтпен алмастырады. Браузер Times New Roman шрифтіне сәйкес шрифтты автоматтық түрде таңдайды.

Енді Web-құжаттарда бірнеше стильді қолдану мәселесін көрейік. Бұл үшін стильдер кестесін (CSS) құрамыз.

```
<head>
<style>
body{
font-family: "Times New Roman", serif;
font-size: 12 pt;
color: red;
font-weight: bold;
font-style: italic;
```

```
}
.fon1
{
background-image: url("files/fon1.gif");
background-color:"#e0e0e0";
font-family: "Times New Roman", shadow;
font-size: 12 pt;
color: "#000080";
}
h2
{
font-family: Helvetica, shadow;
font-size: 14 pt;
colour: yellow;
font-weight: normal;
}
</style>
</head>
```

Көрсетілген мысалда негізгі мәтін 12 пунктті Times New Roman шрифтімен, бас тақырып 18 пунктті Arial шрифтімен, ал II-сатыдағы тақырып 14 пунктті Helvetica шрифтімен берілген. CSS-кестесі Web-құжаттардың бастапқы кодының басына енгізіледі. CSS-кестесінің ішіндегі h1, h2 таңдамасы деп, ал фигуралық жақшанің ішіндегілері оның анықтамалары деп аталады.

Жоғарыдағы әдісті CSS технологиясын HTML-ге енгізу деп атаймыз. Енді CSS технологиясын HTML-ге енгізу және байланыстыру әдістерін қарап шығайық. Бұл әдіс сипаттау стилін бірнеше құжатта қолдану қолайлы. Мұнда CSS-кесте .css файл кеңеймесімен файлда сақталады.

Мысалы: my.css файлының мазмұны:

```
body
{
font-family: "Times New Roman", serif;
font-size: 12 pt;
colour: green;
}
.fon1
{
background-image: url("files/fon1.gif");
background-color:"#e0e0e0";
font-family: "Times New Roman", shadow;
font-size: 12 pt;
color: "#000080";
}
```

```
h1
{
font-family: Arial, sans-serif;
font-size: 18 pt;
colour: red;
font-weight: bold;
font-style: italic;
}
h2
{
font-family: Helvetica, shadow;
font-size: 14 pt;
colour: yellow;
font-weight: normal;
}
tizim
{
font-family: "Times New Roman", shadow;
font-size: 12 pt;
color: "#000080";
}
```

Байқасаңыз <style> тегі CSS-кестесінде көрсетілмейді. Браузер файл атының кеңеймесіне қарап, стильдер кестесін автоматтық түрде таниді. CSS-файлын бір уақытта бірнеше Web-құжаттармен байланыстыруға болады. Бұл үшін Web-құжатты стильдер кестесімен төмендегідей байланыстыру қажет:

```
<head>
<link rel="style sheet" type = "text/css" href="my.css">
</head>
```

Бұл қатарда документтің стильдер кестесін (rel="stylesheet") файл форматымен (tyre = "text / CSS") ағымдағы бумадан (href = "my. CSS") байланыстыру көрсетілген. Бірақта my.css файлының адресі басқа URL-адресте болуы да мүмкін.

Web-құжатқа стильдер кестесін басқа құжаттан енгізу үшін төмендегі қатарды қолданамыз.

```
<head>
<style>
@import: url-адрес (my.com);
</style>
</head>
```

Бұл жерде енгізілетін стильдер кестесімен қатар өзіміз де қалағанша таңдама мен оның анықтамаларын қоса аламыз.

Браузерлер пайдаланылатын стильдер тізбегін олардың орындалу приоритетіне қарай төмендегіше қолданады:

- inline-стильдер - құжатқа <style> тегінің атрибуттарымен енген стильдер;
- енгізілген стильдер - құжатқа <body> тегімен енген стильдер;
- импортталған стильдер - құжатқа <style> тегінің @import қасиетімен енген стильдер;
- байланысқан стильдер - құжатқа <link> тегімен енген стиль.

Браузерлер Web-құжаттарды талдау жасағанда мәтінді форматтау стильдерін осы приоритет бойынша таңдап алады. Сондықтан бас серверде мультимедия-жоба үшін тек қана бір CSS-кste құрылады.

Біз сонымен Web-сайттарды ұйымдастыруда Web-құжаттар үшін HTML 4.0 версиясын қолданып, CSS-технологиясын пайдаланудың кейбір мәселелерін қарастырдық. CSS-технологиясы Web-құжаттарды графикалық пернелерсіз қарапайым құрастырады. Нәтижеде Web-құжаттардың көлемі кішірейіп, жедел жадыға жүктелу уақыты азаяды. Web-сайттарда мыңдаған гипермәтіндерден пайдалану мүмкіндігін қарастырсақ, бұл айтарлықтай жетістік.

2.5 CSS технологиясының қасиеттері

Web-құжаттың құрылымы мен сипаттауын бөлектеуді CSS (Cascade Style Sheets – каскадты стильдер кестесі) технологиясын қолдана отырып орындаймыз. CSS технологиясы Web-құжатты сипаттауда қосымша файлдардан, шаблондар мен стильдерден пайдалануға көмектеседі.

Стильді анықтау құжаттың түрін стиль қасиеттерін енгізу арқалы өзгертетеді. Стиль қасиеттерінің мәні ретінде кілттік сөздер алынады. Кілттік сөздер арасында кездесетін | символы, қасиеттің осы немесе келесі мәнді қабылдайтынын білдіреді. Төменде осы технологияның негізгі қасиеттерін қарастырайық.

Font-family: [шрифттер тізімі] – Шрифттер немесе олардың типтері үтірмен бөлетеніп көрсетіледі. Браузер шрифттерді көрсетілген ретпен компьютерде кездескенше іздеп, табылған шрифтті пайдаланады. Қасиеттің шрифт типтері түрінде төмендегі мәндер алынуы мүмкін:

- serif – түртпесі бар шрифттер, мысалы: Times;
- sans-serif – түртпесі жоқ шрифттер, мысалы: Helvetica;
- cursive – курсивті шрифттер, мысалы: Zarf-Chencery;
- fantasy – декоратив шрифттер, мысалы: Western;
- monospace – моношрифттер, мысалы Courier.

Егер шрифт типі бірнеше сөзден тұрса, онда шрифт типін тырнақша ішінде жазу қажет, мысалы: “Times New Roman Cyr”.

Егер бірнеше шрифт │ белгісімен көрсетілсе, браузер олардың біреуін үндеместен алу принципі бойынша таңдайды.

Font-style: normal │italic│ oblique – шрифт стилдерін көрсетеді. Үндеместен normal алынады. Мұндағы:

- normal – қалыпты;
- italic – курсив;
- oblique – қиғаш.

Font-size: [абсолют]│[салыстырмалы]│[нақты]│[пайыз] – шрифт өлшемін анықтайды. Қасиеттің мәні ретінде абсолют, салыстырмалы, нақты өлшемдер немесе пайыз көрсетіледі. Үндеместен normal алынады.

Font-weight: normal│bold│bolder│lighter – шрифттің енін анықтайды. Үндеместен normal алынады. Мұндағы:

- normal – қалыпты;
- bold – жартылай жуан;
- bolder – жуан;
- lighter – жіңішке.

Font-variant: normal │small-caps – шрифттің бас қаріпінің өлшемін анықтайды. Үндеместен normal алынады. Мұндағы:

- normal – қалыпты;
- small-caps – кіші бас қаріп.

Font: [font-style]‖[font-variant]‖[font-weight]‖[font-size]‖[font-family] – жоғарыда сипатталған қасиеттердің жалпыламасы, яғни шрифт қасиеттерін стильдер кестесінің бір жерінде анықтау мүмкіндігін береді. Қасиеттердің қабылдайтын мәндері жоғарыдағыдай. Егер кейбір қасиеттер көрсетілмесе, онда олар үндеместен алу принципі бойынша таңдалады.

Мысалы:

```
h1 {
font-family: Arial;
font-size: 30 pt;
font-weight: bold;
font-style: italic
}
p {
font: bold italic large Times New Roman, serif
}
```

word-spacing: normal │[ұзындық] – сөздер арасындағы аралық беріледі. Үндеместен normal алынады.

Letter-spacing: normal │[длина] – қаріптер арасындағы аралық анықталады. Үндеместен normal алынады.

Text-decoration: underline │overline │line-through │blink │none – Қаріп гарнитурасы көрсетіледі. Мұндағы:

- underline – астын сызу;
- overline – төбесін сызу;
- line-through – үстін сызу;
- blink – жанып-өшу;
- none – алдынғы орнатылған мәндерді жою.

Text-transform: capitalize │uppercase │lowercase │none – мәтінді бейнелеуде регистрді түрлендіруді анықтайды. Үндеместен none алынады. Мұндағы:

- capitalize – әрбір сөздің бірінші қаріпі бас қаріпке түрленеді;
- uppercase – барлық әріптер бас қаріпке түрленеді;
- lowercase – барлық әріптер кіші қаріпке түрленеді;
- none – алдынғы орнатылған мәндерді жояды.

Text-align: left │right │center │justify – мәтінді терезе шеттеріне теңестіреді. Мұндағы:

- left – солға;
- right – оңға;
- center – ортаға;
- justify – екі шеттен.

Text-indent: [абсолют бірліктер] │[пайыз] – мәтіннің шегінуін көрсетеді. Әдетте шегіну абзацтың бірінші қатарының сол жағына ғана тиісті. Егер бірлік мәні сол таңбалы болса, онда салыстырмалы түрде солға қарай жылжыйды. Абсолют бірліктер төмендегідей анықталады:

- px – логикалық пиксельдер;
- pt – логикалық пункттер;
- em – шрифт өлшемінің бірліктері;
- cm – сантиметрлер;
- mm – миллиметрлер.

Мысалы: text-indent: -3em.

Vertical-align: baseline │sub │super │top │text-top │middle │bottom │text-botton │[пайыз] – элементтің тік орналысуын анықтайды. Үндеместен baseline алынады. Мұндағы:

- baseline – элементті негізгі элементтен төменге орналастырады;
- middle – элементтің орта сызығын «негізгі + жартылай биіктікке» теңестіреді;
- sub – элементті қатар астына теңестіреді (әдетте font-size: smaller қасиетімен бірге беріледі);
- super – элементті қатар төбесіне теңестіреді (әдетте font-size: smaller қасиетімен бірге беріледі);
- text-top – элементің жоғарғы шетін негізгі элементтің төбесімен теңестіреді;
- text-bottom – элементтің төменгі шетін негізгі элементтің астымен теңестіреді;
- top – элементтің төбесін негізгі элементтің ең жоғарғы шетімен теңестіреді;
- bottom – элементтің астын негізгі элементтің ең төменгі шетімен теңестіреді.

line-height: normal | [сан] | [ұзындық] | [пайыз] – қатар биіктігін көрсетеді. Үндеместен normal алынады.

color:[түс] – құжат мәтінінің түсін көрсетеді. Үндеместен құжатта анықталған түсті алады. Түстің мәні бірнеше тәсілмен анықталады:

- түс аттары бойынша, мысалы: red (қызыл), blue (жасыл);
- он алтылық сан түрінде, мысалы: #000000 (қара);
- нөлдері алып тасталған он алтылық сан түрінде, мысалы: #fff (ақ);
- RGB моделі бойынша, мәндері 0-ден 255-дейін, мысалы: rgb(255,255,255) (ақ);
- RGB моделі бойынша пайызбен берілген, мысалы: rgb(100%, 0%, 0%) (қызыл).

Егер түс атымен берілсе, онда бразерлерде әр түрлі бейнеленуі мүмкін.

background-color: [түс] | transparent – құжаттың фон түсін анықтайды. Фонның түсін анықтаумен қатар, transparent арқалы фон асты түсін де көрсетуге болады. Үндеместен transparent алынады.

background-image: url ([url]) | none – фон ретінде суретті файлдан алады. Мысалы: background-image: url (“foto.jpg”). none – фондық сурет берілмейді.

background-repeat: repeat | repeat-x | repeat-y | no-repeat – фондық суретті қайталайды. Үндеместен repeat алынады. Мұндағы:

- repeat-x – фондық сурет көлделеңінен қайталанады;
- repeat-y – фондық сурет тігінен қайталанады;

- no-repeat-y – фондық сурет бір-ақ рет көрсетіледі, қайталанбайды;
- repeat – фондық сурет көнделеңінен де, тігінен де қайталанады.

background-attachment: scroll │fixed – фонды мәтінге байланыстырады. Үндеместен scroll алынады. Мұндағы:

- scroll – егер мәтін терезеге симаса, онда фон мәтінмен қоса жүреді;
- fixed – фон орнында тұрады, ал мәтін оның үстінде қозғалады.

background-position:top │center │bottom │left │center │right │[пайыз] [ұзындық] – фондық суреттің басталу координаталарын анықтайды. Фондық сурет координаталары кілттік сөздермен тігінен және көлденеңінен беріледі.

Фондық суреттің тігінен басталу координаталары төмендегіше беріледі:

- top – жоғарыдан;
- center – ортадан;
- bottom – төменнен.

Фондық суреттің көлденеңінен басталуы төмендегіше беріледі:

- left – солдан;
- center – ортадан;
- right – оңнан.

Координаталар тігінен бастап, сонан соң көлденеңінен көрсетіледі.

Мысалы: Суретті жоғарғы оң жақтағы бұрышқа орналастырайық, онда background-position: top right.

Координаталар пиксельде – (px), сантиметрде – (cm) немесе пайызда – (%) берілуі де мүмкін.

Мысалы: background-position:50px 50px немесе background-position: 50%50%.

background: [background-color] ╟ [background-image] ╟ [background-repeat] ╟ [background-attachment] ╟ [background-position] – жоғарыда сипатталған қасиеттердің жалпыламасы, яғни фондық сурет қасиеттерін стильдер кестесінің бір жерінде анықтау мүмкіндігін береді. Қасиеттердің қабылдайтын мәндері жоғарыдағыдай. Егер кейбір қасиеттер көрсетілмесе, онда олар үндеместен алу принципі бойынша таңдалады.

Мысалы:

```
body (background-image:ur1(“suret.jpg”);
background-repeat: repeat-y; background-attachment: fixes;
background-position: 1004 1004 )
p (background: ur1(suret.gif) red 50% repeat fixed).
```

2.6 Web-құжаттың интерактивтік интерфейсі

Бұл бөлімнің мақсаты Web-құжаттың интерактивтік интерфейсін құру әдістерімен танысу болып табылады. Бөлімді сипаттауда WWW және HTML негіздерінен пайдаланамыз.

Тұтынушының интерактивтік интерфейсі - тұтынушы мен программаның өзара қарым-қатынасын ұйымдастыратын жүйе. Ол WWW-де тұтынушы интерфейсін сипаттайтын HTML-құжаттардың тізбегі түрінде анықталады.

Интерактивтік интерфейсті HTML-құжатты құру үрдісі шартты түрде екі түрге бөледі:

- статикалық;
- динамикалық.

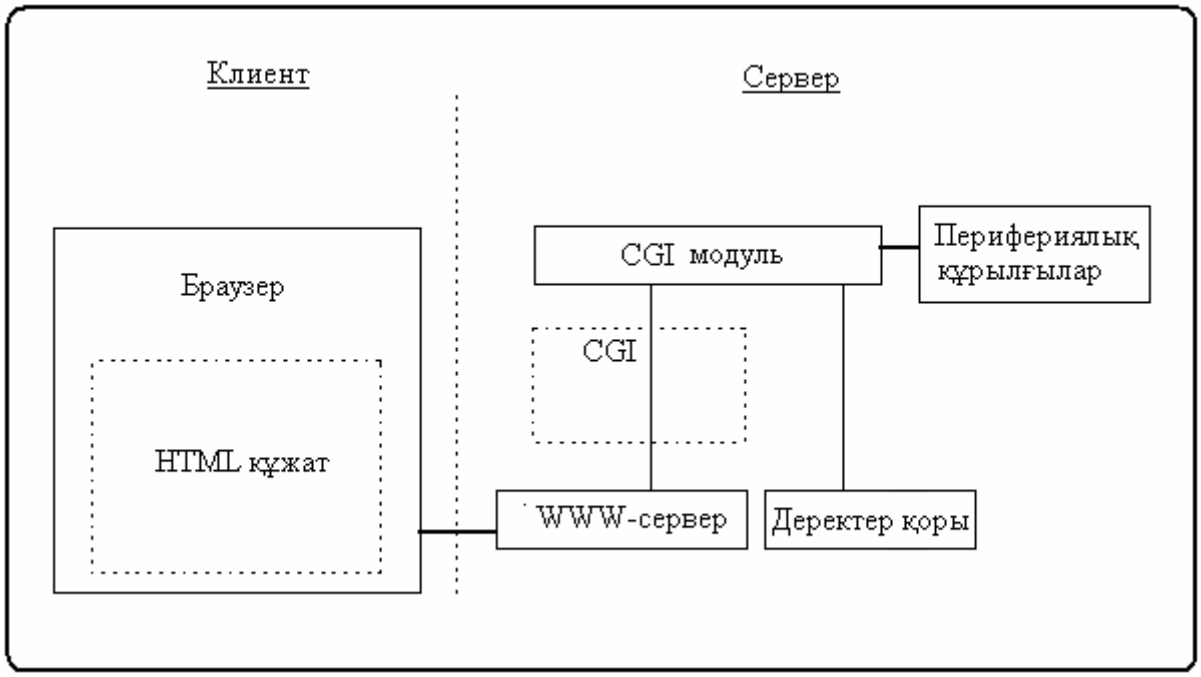
Статикалық интерфейс ретінде мәтіндік немесе HTML-редактормен құрылған HTML-құжатты айтамыз. Әдетте бұл интерфейс жұмыс барысында тұрақты түрін сақтайды.

Динамикалық интерфейсте HTML-құжатқа арнайы программа (cgi-модуль) енгізіліп, ол серверлік компьютерде орындалады. Нәтижеде тұтынушы интерфейсі жұмыс барысында өзгеріп отырады.

Сонымен, WWW үшін интерактивтік интерфейс анықтамасын төмендегі түрде анықтауға болады, яғни тұтынушының интерактивті интерфейсі ретінде статикалық немесе динамикалық түрде құрылған, деректер алмасуға арналған HTML-құжаттар тізбегі айтылады. Іс жүзінде клиенттік компьютер мен сервер арасында деректер алмасатын кез-келген HTML-құжат интерактивтік интерфейс құрумен байланысты. Интерактивтік интерфейс мәселеріне ретінде SQL-сервермен байланысу, деректер қорына кіру, перифериялық құрылғылырдан ақпарат алу, клиентпен қарым-қатынас орнату т.с.с. жатады. Бұл айтылғандардың барлығы CGI-программалау технологиясы арқылы орындалады.

CGI (Common Gateway Interface) WWW-сервермен байланыс орнатушы қолданбалы программаның стандарт интерфейсі болып есептеледі. Тұтынышының интерактивтік интерфейс құру үрдісі екі бөлімнен тұрады (сурет 12):

- клиенттік;
- серверлік.



Сурет 12 - Тұтынушының интерактивтік интерфейсі

Клиенттік бөлім. Интерактивтік интерфейстің клиенттік бөлімін құру үшін тұтынушы интерфейсі сипатталған HTML-құжат құру қажет. Ол HTML тілінде қалыпты құру тэгінен пайдаланамыз. Әдетте қалыптан тұтынушыны тіркеуде, әр-түрлі мәліметтер жинауда, сұранысты қабылдауда пайдаланады. Қалып тұтынушыға арнайы тек толтыруға рұқсат етілген өрістерге деректер енгізуді ұсынады. Енгізілген деректерді WWW-сервердегі арнайы программа (cgi-модуль) өңдейді. HTML 4.0 тілінің қалыптармен жұмыс істейтін тэгтерін қарастырайық. Оларға төмендегі тэгтер жатады:

- <form>
- <textarea> (text area)
- <select>
- <option>
- <input>

<form> тэгі қалып құрушы негізгі тэг. Тэг </form> тэгімен жабылады. Тэгтің ішінде HTML тілінің көптеген тэгтерінен пайдалануға болады.

<textarea> тэгі қалыпта мәтіндік аймақты анықтайды. Тэг <form> тэгінің ішінде жазылып, </textarea> тэгімен жабылады.

<select> тэгі қалыпта мәзірді анықтайды. Тұтынушы мәзір ішінен қажеттісін таңдай алады. Тэг <form> тэгінің ішінде жазылып, </select> тэгімен жабылады. Тэг ішінде бірнеше <option> тэгі жазылуы тиіс.

<option> тэгі қалыптағы мәзірдің тақырыптарын анықтайды. Тэг <select> тэгінің ішінде жазылып, жабылу тэгі болмайды.

<input> тэгі қалыптың өрісін анықтайды. Өрістің мазмұнын тұтынушы анықтайды немесе орындауға жібереді. Тэг <form> тэгінің ішінде жазылып, жабылу тэгі болмайды.

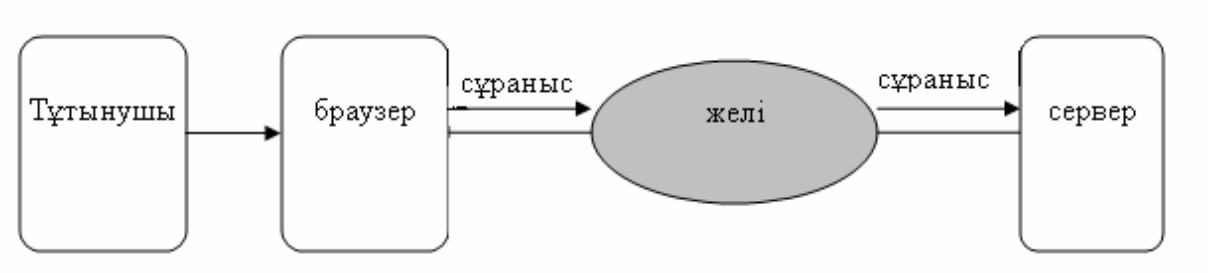
Серверлік бөлім. Серверлік бөлім клиенттік бөлімнен түсетін деректерді өңдеу сұранысын орындаушы модульден тұрады. Ол HTML тілінде жауапты өңдейді. Мұндай модуль cgi-модуль деп аталады. HTML-құжат WWW-сервермен «клиент-сервер» байланыста болуы үшін, олардың өзара қандай HTTP әдісті пайдалануын анықтау қажет. Жалпы алғанда, сұраныс – бұл клиенттің серверге жіберген хабарлары.

WWW-сервер – бұл желі тұтынушыларына серверде орналасқан гипермәтіндік құжаттарды пайдалану мүмкіндігін беретін бүкіләлемдік немесе корпоративтік желінің бір бөлігі. Тұтынушы WWW-сервермен байланысу үшін арнайы бағдарлама - браузер қажет.

Web–браузерлер HTML тілінде Internet желісіне арнап жазылған Web-сайттар мен Web-құжаттарды парақтап көру, оқу мүмкіндігін береді. Web–браузерлер ішінде ең көп таралғандары Internet Explorer (Microsoft), Opera (Opera Software ASA) және Mozilla FireFox (Mozilla Corporation).

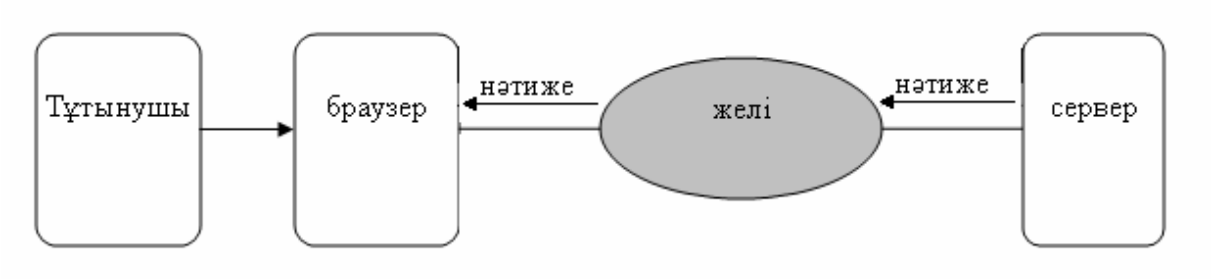
Енді WWW-сервердің жұмыс үрдісін қарап шығайық. Ол төменде көрсетілген:

1. Тұтынушы браузерді жұмысқа қосады. Ол төмендегі іс-әрекеттерді орындайды:
   - сервермен байланыс жасайды;
   - қажетті HTML-құжатты іздейді;
   - табылған HTML-құжатты браузер терезесінде бейнелейді;
   - тұтынушының HTML-құжатқа іс-әрекеттерін орындайды;
   - тұтынушыға сервистік қызметтер ұсынады;
   - WWW-серверге тұтынушы сұранысын жібереді (сурет 13).



Сурет 13 - Клиенттен серверге сұраныс жіберу

2. WWW-сервер тұтынушы сұранысын орындап, сұраныс бойынша нәтижені браузерге қайтарады (сурет 14).



Сурет 14 - Серверден клиентке нәтижені қайтару

3. Браузер құжатты қабылдап, тұтынушыға бейнелейді және оның іс-әрекеттерін күтеді: Тұтынушының мүмкін іс-әрекеттері:
    - жаңа HTML-құжаттың адресін енгізеді;
    - ағымдағы HTML-құжат бойынша іздеу, баспаға шығару т.с.с. амалдар орындайды;
    - ағымдағы HTML-құжат бойынша басқа құжатпен байланыс (link) орната алады.

Ескерту: Бұл жерде бірінші және үшінші жағдайларда әр-түрлі құжат қолданылады.

3 JavaScript тіліне кіріспе

3.1 JavaScript тілінің қарапайым элементтері

JavaScript бағдарламалау тілін 1995 жылы Брендан Айх (Sun Microsystems) жасап шығарды. Тілдің синтаксисисін жазуда C++ және Java тілдері пайдаланылған. Бұл тіл алғашқыда Netscape Navigator браузеріне арналған болатын. Қазіргі таңда JavaScript тілін барлық браузерлер қолданады.

JavaScript тілі – динамикалық Web-құжаттарды дайындауға арналған объектті-оқиғалы тіл. Бұл тіл негізінен клиенттік компьютерде орындалатын скрипт құруға арналған.

Скрипт (Script) – HTML тілінің мүмкіндіктерін кеңейтуге арналған, арнайы бағдарламалау тілінде жазылған ықшам бағдарлама. Кейбір әдебиеттерде сценарий термині де кездеседі, яғни бұл екі сөз синоним.

Скрипттер HTML-құжаттың HTML тілінде жазылған кодының ішіне тікелей орналасады. Олар компиляцияны қажет етпейді, яғни компилятордың орнына интерпретатор жеткілікті. Бұл бағдарлама құруды жеңілдетіп, уақытты үнемдейді. Клиенттік скрипттердің негізгі қасиеті, олар клиент компьютерінде, яғни браузерде HTML-кодпен бірге орындалады.

Біз төменде JavaScript бағдарламалау тілін қысқаша қарастырамыз [5-8].

Алфавит. JavaScript тілінің алфавиті төмендегі белгілерден тұрады:

- 26 латын алфавиті әріптері: A, B, C, D, E, F, G, H, I, J, K, L, M, N, O, P, Q, R, S, T, U, V, W, X, Y, Z.
- 10 араб цифрлары: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9.
- арнайы белгілер:

+	қосу	~	тильда
-	алу	!	леп белгісі
*	жұлдызша	@	коммерциялық эт
/	қисық сызық	#	нөмір белгізі
=	тең	$	ақша бірлігі
,	үтір	%	пайыз
.	нүкте	^	көрсеткіш
:	қос нүкте	&	амперсанд
;	нүктелі үтір	|	тік таяқша
>	үлкен	\	кері қисық сызық
<	кіші	()	жақша
?	сұрақ белгісі	[]	тік жақша
_	сызықша	{}	фигуралық жақша
␣	бос орын		
‘	апостроф		
“	тырнақша		

JavaScript интерпретаторы регистрды есепке алады, яғни бас және кіші әріптерді бөлек деп қабылдайды. Сонымен қатар орыс немесе қазақ алфавиті әріптерін тек қана мәтін қатарында қолдануға болады.

Алфавит белгілерінен қызметші сөз, идентификатор, литерал, оператор т.с.с. тіл элементтері құрылады.

Қызметші сөздер деп JavaScript тілінің жұмысына қажетті нұсқауларды айтады. Әрбір сөздің өз мағынасы, атқаратын қызметі бар. Қызметші сөздерді жұмыс барысында біртіндеп мағынасы қарай түсіндіріп отырамыз.

Идентификатор деп әріп немесе _(сызықша) белгісінен басталатын әріп, цифр және _ белгілерінің тізбегін айтады. Идентификатор - литерал, оператор, функция сияқты қарапайым элементтерді атауға қажет. Идентификаторды жазуда бас және кіші әріптер әртүрлі деп есептеледі. Сонымен қатар идентификатор ретінде қызметші сөздерді қолдануға болмайды. Мысалы:

Дұрыс жазылған идентификаторлар:

- Java_Script – сызықша орынды қолданылған;
- _a1 – сызықшадан басталады;
- X – латын әрібі қолданылған.

Қате жазылған идентификаторлар:

- Java-Script –басқа белгі қолданылған;
- For – қызметші сөз;
- 1a – цифрден басталған;
- Скрипт –орыс әріптері қолданылған.

Литерал деп әдетте JavaScript тілінде қолданылатын деректерді айтады. Литерал *тұрақты* және *айнымалы* деп екіге бөлінеді. Тұрақты литерал деп скриптте мәні өзгермейтін деректі айтамыз. Ал мәні өзгеретін деректі айнымалы литерал дейміз. Литералдың атауы идентификатормен беріледі.

Деректердің типтері. JavaScript тілінде қолданылатын деректер типтерін қарастырайық:

- бүтін;
- нақты;
- логикалық;
- қатарлық;
- null.

Бүтін типті деректерге бүтін сандар жатады. Сандар ондық, сегіздік және он алтылық санақ жүйесінде берілуі мүмкін. Сегіздік санақ жүйесіндегі сандардың алдына 0 (нөл) таңбасы, ал он алтылық сандардың алдына 0x таңбасы жазылады. Мысалы: -100, 0175, 0xA1.

Нақты типті сандар екі түрде жазылады:

- тұрақты нүктелі;
- жылжымалы нүктелі.

Тұрақты нүктелі сандар, нүктемен бөлектенген бүтін және бөлшек түрінде жазылады. Санда бөлшек бөлім болмаса да, нүкте қойылады. Мысалы: 3.1415, -100.

Жылжымалы нүктелі сандар, E (үлкен әріп) символымен бөлектенген мантисса және дәреже түрінде жазылады. Мысалы: 1E-5, 3.14E3.

Логикалық типті деректер тек қана true (ақиқат, 1) false (жалған, 0) мәндерін ғана қабылдайды.

Қатарлық типке мәтін жатады. Мәтін апостроф (‘) немесе қос тырнақша (“) ішіне жазылады. Мысалы: “TurboPascal тіліне ұқсас”.

JavaScript тілінде қатарлық типке арнайы қызметші және баспаға шықпайтын символдарды енгізуге болады.

\b	бос орын (backspace)
\f	жаңа бетке өту (PageDown)
\n	жаңа қатарға өту
\r	енгізу (enter)
\t	табуляция (tab)
\’	апостроф
\”	тырнақша
\\	кері қисық сызық (backslash)

Мысалы:
my_file -идентификатор;
12345 -бүтін сан;
012345 -сегіздік бүтін сан;
0x12345 -он алтылық бүтін сан;
2.71828 -тұрақты нүктелі нақты сан;
21. -тұрақты нүктелі нақты сан (бөлшек бөлігі жоқ);
3.14E5 -жылжымалы нүктелі нақты сан ($3{,}14 \cdot 10^5$);
1.2E-4 -жылжымалы нүктелі нақты сан ($1{,}2 \cdot 10^{-4}$);
“Сауле” -қатар.

Регуляр өрнектер. Регуляр өрнек деп қатардан берілген символдар тізбегін іздеуге арналған шаблондарды айтады. Олардың қарапайым іздеу функцияларынан айырмашылығы, регуляр өрнектің іздеу операциясының күрделілігінде. Регуляр өрнектер / (слэш) символдары ішіне жазылады.

Мысалы: file*.txt - шаблоны атауы file сөзінен басталатын барлық txt кеңеймесі бар файлдарды іздейді. Осы шаблонды регуляр өрнек түрінде жазайық.

file*.txt → / [fF] ile . * \. txt /

Жазылған регуляр өрнек [fF] - f немесе F символдарынан басталатын, ile – тізбегін және .* - кез келген символдар тізбегін, \. – нүктені, txt – тізбегін іздеу мағынасын береді.

Регуляр өрнектерде кездесетін литералдарды қарастырайық:

\	арнайы символдарға жатады, келесі символ литерал емес деген мағынаны білдіреді, мысалы: \n –қатардың соңы, жаңа қатарға өту. Сонымен қатар регуляр өрнектерде литералды символ түрінде жазуда қолданылады, мысалы: \(-регуляр өрнекте жақша кездеседі.
^	қатардың басы
$	қатардың соңы
*	алдыңғы символ көп кездеседі немесе кездеспейді
+	алдыңғы символ міндетті түрде бір рет немесе одан көп кездеседі
?	алдыңғы символ тек қана бір рет кездеседі немесе кездеспейді
{n}	алдыңғы символ міндетті түрде n -рет кездеседі
\w	әріп, цифра немесе _ (сызықша белгісі)

Мысалдар:

/ (\w+) @ ([\ w \ . _] +) / - кез-келген электрондық поштаның адресін білдіреді. Адресті тұтынушы аты және сервердің атауы ретінде екіге бөледі.

/ (\w + : \ / \ /) ? ([^ /] +) (. *) ? / -кез-келген HTML-құжаттың URL-адресін білдіреді. Адресті хаттама, домен және файл аты деп үш бөлікке бөліп, жедел жадыда сақтайды. Сонымен қатар, хаттама мен файл атын көрсетпесе де болады.

RegExp сыныбы қатарды регуляр өрнектер көмегімен өңдеуде қолданылады. Жалпы форматы:

RegExp ({регуляр өрнек} [, {флагтар}])

бұл жерде: регуляр өрнек міндетті компонент;

флагтар: g -глобальды іздеу;

i -символдар регистрін есептемеу;

m -көп қатарлы іздеу.

3.2 JavaScript тіліндегі амалдар

JavaScript тілінде амалдар (операторлар) төмендегі түрлерге бөлінеді:

- арифметикалық;
- логикалық;
- салыстыру;
- биттік (екілік);
- меншіктеу;
- қатарлық;

Арифметикалық амалдар. Бүтін және нақты типті деректер үстінде төмендегі арифметикалық амалдар орындалады:

+	қосу
-	алу
/	бөлу
*	көбейту
-	инверсия – таңбаны өзгерту
%	арифметикалық модуль – бүтін бөлу амалының қалдығы
++	инкремент – операнд мәнін бір бірлікке арттыру
--	декремент - операнд мәнін бір бірлікке азайту

Мысалы:c = 2 * a + b; - бірнеше арифметикалық амал қолданылған;
c = c++; - инкремент амалы операндтан кейін жазылған;
c = ++c, - инкремент амалы операндтан алдын жазылған;
x = -a; - инверсия амалы;

Арифметикалық амалдағы нәтиженің типі, амалда қатысатын деректердің типімен анықталады.

Логикалық амалдар. Логикалық амалдар тек қана логикалық *true* және *false* мәндері үстінде орындалады. Логикалық теріске шығару амалы бір аргументті, ал қалған амалдар екі аргументті.

!	теріске шығару
&&	конъюнкция (логикалық көбейту)
||	дизъюнкция (логикалық қосу)

a	b	!a	!b	a && b	a || b
false	false	true	true	false	false
false	true	true	false	false	true
true	false	false	true	false	true
true	true	false	false	true	true

Мысалы: (x <= 1) && (x >= 7) амалдарын нәтижесін табайық.

- x = 0 болса, нәтиже *false;*
- x = 5 болса, нәтиже *true;*
- x = 9 болса, нәтиже *false;*

Салыстыру амалдары. Деректер үстінде төмендегі салыстыру амалдары орындалады:

>	үлкен
>=	үлкен немесе тең
<	кіші
<=	кіші немесе тең
=	тең
!=	тең емес
= = =	қатаң түрде тең
!= =	қатаң түрде тең емес

Салыстыру амалының нәтижесі *true* немесе *false* мәндеріне ие. Бұл амалдарда деректер кез келген типке, ал нәтиже тек қана логикалық типке тиісті.

Мысалы: *a = 5; b = 3; болсын, онда*
a >= b - true;
a = b - false болады.

Егер салыстырылатын деректер әр түрлі типке ие болса, онда қатаң түрде тең (тең емес) амалдарын қолданған жөн.

Биттік амалдар. JavaScript тілінде биттік амалдар төменгі деңгейлі программалауда қолданылады.

~	инверсия – теріске шығару
&	конъюнкция – логикалық көбейту
|	дизъюнкция – логикалық қосу
^	логикалық алу
<<	солға жылжыту
>>	оңға жылжыту
>>>	оңға таңбасыз жылжыту

Мысалы:
c = ~a; - инверсия амалы;
c = a & b; - конъюнкция амалы;
c = a << b; - a-ны солға b-разрядқа жылжыту;
c = a >> b; - a-ны оңға b-разрядқа жылжыту;
c = a >>> b; - a-ны оңға b-разрядқа жылжыту (үлкен разрядқа нөлдер жазылады).

Биттік амалдардың мәндері:

a	b	~a	~b	a & b	a | b	a ^ b
0	0	1	1	0	0	0
0	1	1	0	0	1	1
1	0	0	1	0	1	1
1	1	0	0	1	1	0

Меншіктеу амалы (=) литералға мән беруде қолданылады. Мән ретінде тұрақты немесе өрнек жазылады. Мысалы: a = 1; a = a++;

JavaScript тілінде меншіктеу амалының қысқартылған түрлері кездеседі. Мысалы: a = a +10; және a += 10; өрнектерінің мәні бірдей.

жазылуы	*мағынасы*
a += b;	a = a + b;
a -= b;	a = a - b;
a *= b;	a = a * b;
a /= b;	a = a / b;
a &= b;	a = a & b;
a >>= b;	a = a >> b;

Қатарлық амалдар. Қатарлық амалға конкатенация (+) амалы жатады. Бұл амал қатарларды қосуда қолданылады.

Мысалы: a = ”Java”; b = ”Script”; c = a + b;

Өрнектер - деректер үстінде амалдарды және олардың орындалу тәртібін анықтайды. JavaScript тілінде өрнек литералдар және амалдар тізбегінен тұрады. Өрнектер арифметикалық және логикалық болып екіге бөлінеді.

Арифметикалық өрнектер - арифметикалық амалдармен бөлектелген тұрақтылар, айнымалылар және стандарт функциялар тізбегінен тұрады. Өрнектің нәтижесінің типі деректердің типімен анықталады.

Мысалы: 2*a + b*x + c

Логикалық өрнектер – логикалық деректер, логикалық және салыстыру амалдары тізбегінен тұрады. Салыстыру амалдарында арифметикалық, логикалық және символдық типті деректер кездеседі. Логикалық амалдардың нәтижесі true және false мәндеріне ие.

Мысалы: (x > 0) && (x < 13)

Typeof амалы өрнектің типін анықтауда қолданылады. Егер тип бүтін немесе нақты болса «number», қатарлық болса «string» және логикалық болса «boolean» мәндерін қабылдайды. Егер өрнектің типі анықталмаған болса «undefined» мәні шығады.

Мысалы: t = typeof (b);

Өрнектердегі амалдардың орындалу тәртібі амалдар приоритетімен анықталады. Амалдар приоритетін жақшалармен өзгертуге болады. Өйткені жақша ішіндегі амалдар бірінші орындалады. Өрнектегі приоритеті бірдей амалдар солдан оңға қарай орындалады.

приоритет	*аталуы*	*амалдар*
I	унарлық	-, ++, --, ~
II	мультипликатив	*, /, %, &&, &
III	аддитив	+, -, ||, |
IV	салыстыру	=, >, <, >=, =, !=, = = =, != =
V	меншіктеу	=, +=, *=, …

Өрнектерді құруда төмендегі ережелерді орындау керек:

- өрнекті бір қатарға жазу;
- тек қана қарапайым жақшалардан пайдалану;
- қатарынан екі амалды жазбау.

JavaScript тілінде бірнеше өрнекті жинақтап жазуға болады. Жинақталған өрнектер фигуралық жақшалар {} ішіне жазылады.

Мысалы:

```
{
i = i++;
a += i;
}
```

Мысалы:

```
{
i = ++i;
a += i;
}
```

JavaScript тіліндегі бағдарлама кодына түсініктеме төмендегідей жазылады:

```
// түсініктеме
/* түсініктеме */
```

3.3 Шартты операторлар

Шартты оператор деректерді өңдемейді, өңдеуді басқарумен айналысады. JavaScript тілінде шартты оператор төмендегі түрлерге бөлінеді:

- if-else тармақталу операторы;
- ? шартты операторы;
- switch таңдау операторы.

if-else тармақталу операторы логикалық шарттың мәніне қарай бағдарлама кодының бір бөлігін орындайды. Оператордың форматы:

a) if шарт өрнек1 else өрнек2;

b) if шарт өрнек1;

бұл жерде: if, else –қызметші сөздер; шарт –логикалық өрнек;

Егер шарттың мәні true болса, онда өрнек1 орындалады, ал кері жағдайда, яғни шарттың мәні false болса, онда өрнек2 орындалады.

Мысал 1:
```
if (x>0 && x<13)
f = 1
else
f = 0;
```

Мысал 2:
```
if str1==”Нурдаулет” str2=”Сәлем Нурдаулет”;
```

Мысал 3:
```
if (x==0)
{
f  = 11;
k = 22;
}
else
{
f = 33;
k = 44;
}
```

Мысал 4:
```
if (x>=1)
{
f  = 5;
k = 6;
}
else
{
```

```
f = 7;
k = 8;
}
```

? шартты операторы логикалық шарттың мәніне қарай орындалады. Операторды форматы:

```
шарт ? өрнек1 : өрнек2;
```

бұл жерде: ? –қызметші сөз; шарт –логикалық өрнек;

Егер шарттың мәні true болса, онда өрнек1 орындалады, ал кері жағдайда, яғни шарттың мәні false болса, онда өрнек2 орындалады.

Мысалы: `f = (x>0 && x<13) ? 1 : 0;`

switch таңдау операторы бірнеше тармақталу операторларын біріктіру мүмкіндігін береді. Операторды форматы:

```
switch (өрнек)
{
case мән 1  :  оператор 1;
break;
case мән 2  :  оператор 2;
break;
case мән 3  :  оператор 3;
break;
case мән 4  :  оператор 4;
break;
- - - - - - - - - - - - - - - - -
case мән n  :  оператор n;
break;
default  :  оператор;
}
```

Операторды орындалуы switch қызметші сөзінен кейінгі өрнекті өңдеуден басталады. Сонан соң өрнектің мәні case қызметші сөздерінен кейінгі мәндермен салыстырылады. Қай мән сәйкес келсе, сол қатардағы оператор орындалады. Егер ешқандай мән сәйкес келмесе, онда default қатарындағы оператор орындалады.

Мысалы: берілген day айнымалының мәні, егер 1-ге тең болса, онда out айнымалысы ”дүйсенбі” мәнін қабылдайды, ал егер 2-ге тең болса, онда out айнымалысы ”сейсенбі” мәнін қабылдайды, т.с.с. Ал егер day айнымалының мәні басқа сан болса, онда out айнымалысы ”апта күні емес” мәнін қабылдайды.

```
switch (day)
```

```
{
case 1  :  out = ”дүйсенбі”;
break;
case 2  :  out = ”сейсенбі”;
break;
case 3  :  out = ”сәрсенбі”;
break;
case 4  :  out = ”бейсенбі”;
break;
case 5  :  out = ”жұма”;
break;
case 6  :  out = ”сенбі”;
break;
case 7  :  out = ”жексенбі”;
break;
default  :  out = ”апта күні емес”;
}
```

Таңдау операторындағы default қызметші сөзін жазбауға болады. Мұндай жағдайда, switch қызметші сөзінен кейінгі өрнектің мәні case қызметші сөздерінен кейінгі мәндердің ешқайсысына сәйкес келмесе, ешқандай іс-әрекет орындалмайды.

Сол сияқты, break қызметші сөзін де жазбауға болады. Мұндайда, switch қызметші сөзінен кейінгі өрнектің мәні сәйкес келген case қызметші сөзінен кейінгі операторлан бастап, барлығы орындалады.

Мысалы:

```
switch (day)
{
case 1  :  out = ”дүйсенбі”;
case 2  :  out = ”сейсенбі”;
case 3  :  out = ”сәрсенбі”;
case 4  :  out = ”бейсенбі”;
case 5  :  out = ”жұма”;
case 6  :  out = ”сенбі”;
case 7  :  out = ”жексенбі”;
default  :  out = ”апта күні емес”;
}
```

Бұл операторлың нәтижесінде out айнымалысы тек қана ”апта күні емес” мәнін қабылдайды. Себебі day айнымалының мәні апта күніне сәйкес келседе, соңында default қызметші сөзінен кейінгі опреатор орындалады. Сондықтан break қызметші сөзін жазған дұрыс.

3.4 Цикл операторлары

Цикл операторы бағдарлама кодының кейбір бөлігін бірнеше рет қайта-қайта орындауды ұйымдастырады. JavaScript тілінде цикл операторы төмендегі түрлерге бөлінеді:

- for - цикл операторы;
- do-while цикл операторы;
- while цикл операторы;
- break және continue операторлары.

for цикл операторы қайталаулар саны белгілі болғанда қолданылады. Цикл бағдарлама кодындағы логикалық шарт true мәнін қабылдаған кезде орындалады. Оператордың форматы:

for (a, v, b) {<цикл денесі>}

мұндағы: for –қызметші сөз;
a -есептеуішті инициалдау;
v –логикалық шарт;
b -өрнекгі инкременттеу немесе декременттеу.

Бұл оператор бағдарлама кодының бірнеше рет қайта-қайта орындауын ұйымдастыру үшін қажет. Сондықтан кодтың неше рет орындалғанын анықтайтын бүтін типті айнымалы *есептеуіш* деп аталады. Ал, бағдарлама кодының бірнеше рет қайта-қайта орындалушы бөлігін *цикл денесі* атайды. Цикл денесі логикалық шарт false мәнін қабылдағанша орындалады.

Мысалы: S = 1 + 2 + … + 10.

а) for (i = 1; i <= 10; i ++) S += i;

б) for (i = 10; i > 0; i --) S += i;

Көрсетілген мысалда есептеуіш мәнін өзгерту үшін а)-да инкремент және б)-да декремент амалы қолданылған.

Мысалы:

```
for ( i = 2;  i < 21;  i += 2)
{
a += 3;
b = 2 * i + 1;
}
```

Көрсетілген мысалда есептеуіш мәні 2, 4, 6, ..., 20 мәндерін қабылдайды, ал цикл 10 рет орындалады.

do-while цикл операторы қайталаулар саны логикалық шартқа байланысты болған жағдайда қолданылады. Бұл оператор for операторына ұқсас, тек мұнда есептеуіш жоқ. Операторды форматы:

```
do
    цикл денесі
while (шарт);
```

мұндағы: do, while –қызметші сөздер; шарт –логикалық өрнек.

Цикл бағдарлама кодындағы логикалық шарт true мәнін қабылдаған кезде орындалады. Логикалық шарт цикл соңында орналасқан, сондықтан цикл шарт тексерілгенше бір рет орындалады.

Мысалы:

```
do
{
a = a * i + 2;
i = i ++;
}
while (i < 100);
```

do-while цикл операторын қайталаулар саны белгілі болғанда да қолдануға болады.

Мысалы: S = 1 + 2 + … + 10.

```
var  i = 1; S = 0;
do
{
S += i;
i = i ++;
}
while (i > 10);
```

while цикл операторы do-while операторына ұқсас, тек қана логикалық шарт цикл басында тексеріледі. Операторды форматы:

```
while (шарт);
    цикл денесі
```

мұндағы: while –қызметші сөз; шарт –логикалық өрнек.

Цикл логикалық шарт true мәнін қабылдаған кезде орындалады. Егер логикалық шарт басынан-ақ false мәнін қабылдаса, онда цикл міндетті түрде орындалмайды.

Мысалы:

```
while (i < 100)
```

```
{
a = a * i + 2;
i = i ++;
}
```

do-while цикл операторын қайталаулар саны белгілі болғанда да қолданса болады.

Мысалы: S = 1 + 2 + … + 10.

```
var  i = 1; S = 0;
do
{
S += i;
i = i ++;
}
while (i > 10);
```

break және continue операторлары. Кейде циклді орындалу барысында тоқтату қажеттілігі туындайды. Мұндай жағдайда JavaScript тілінде break және continue операторларынан пайдаланамыз.

break операторы циклдың орындалуын тоқтатып, басқаруды келесі операторға немесе белгісі бар операторға жібереді. Операторлың форматы:

```
break [белгі];
```

мұндағы: break –қызметші сөз; белгі –идентификатор.

Белгі – бұл бағдарлама кодында кездесетін басқа бір операторға қойылатын идентификатор. Басқару белгі орналасқан операторға жіберіледі. break операторында белгіні қолданбаса да болады.

Мысалы:

```
while ( a < 100 )
{
a = a * i + 2;
if ( a > 50) break;
i = ++ i;
}
```

Көрсетілген мысалда, егер а-ның мәні 50-ден көп болса, цикл тоқтайды (бағдарлама орындала береді).

continue операторы циклдың орындалуын тоқтатып, басқаруды цикл басына немесе белгі орналасқан операторға жібереді. Операторлың форматы:

```
continue [белгі];
```

мұндағы: continue –қызметші сөз; белгі –идентификатор.

Мысалы:

```
while ( a < 100 )
{
i = ++ i;
if ( i > 9 && i < 21) continue;
a = a * i + 2;
}
```

Көрсетілген мысалда, a-ның мәні цикл есептеуіші i-дің 10 мен 20 мәндері аралығында есептелмейді.

JavaScript тілінде *қабаттасқан циклден* де пайдалануға болады. Қабаттасқан цикл сыртқы және ішкі цикл болып бөлінеді. Ішкі цикл сыртқы циклдің ішіне орналасады. Сыртқы цикл мен ішкі цикл қиылыспайды, сыртқы циклдің әр бір қадамында ішкі цикл толық орындалады.

Мысалы:

```
for ( i = 1; i <= 10; i++ )
{
for ( j = 1; j <= 10; j++ )
s = s + i + j;
}
```

Мысалы: Қабаттасқан циклда continue операторын қолданайық.

```
outer:
for ( i = 1; i <= 10; i++ )
{
  inner:
  for ( j = 1; j <= 10; j++ )
  {
    if ( j = =2 ) continue outer;
    else
    s = s + i + j;
  }
}
```

Көрсетілген мысалда, j айнымалысының мәні 2-ге тең болса, басқару қайтадан сыртқы циклге жіберіледі. Егер continue операторында белгіні қолданбасақ, онда басқару қайтадан ішкі циклге жіберіледі.

3.5 Web-құжатқа скриптті орналыстыру әдістері

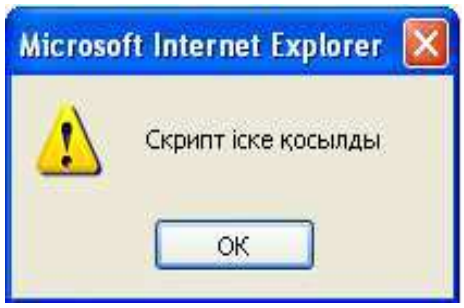
1. *Скрипт коды Web-құжаттың ішінде орналасқан.* Web-құжаттың тақырып бөліміне коды орналасқан қарапайым скриптті жүктеуді қарастырайық. Бағдарлама листингі мен скрипт коды:

```
<html>
<head>
<script language='JavaScript'>
function f1()
     {
      alert("Скрипт іске қосылды");
      }
</script>
</head>
<body>
       <input type="button"  value="Скрипт"  onclick="f1()">
</body>
</html>
```

Скрипт жазылған Web-құжатта төмендегі батырма пайда болады.



Батырмасы бассақ төмендегі хабарлама шығады (сурет 15).



Сурет 15 – Скрипт іске қосылғаны туралы хабарлама

Берілген скриптті талдайық. Скрипт коды мен оқиғаны өңдеу батырмасы бір құжатта берілген. Скриптті іске қосу үшін батырма (button) объекті және батырманы басу (onclick) оқиғасы пайдаланылған. Батырманы басқанда f1() функциясы жұмысқа қосылады. Бұл функция өз кезегінде alert() функциясын шақырып, терезеге мәтінді («Скрипт іске қосылды») шығарады. Скрипт коды Web-құжат бағдарламасынан <script> және </script> тэгтері арқылы бөлектенген.

Сипатталған скрипт кодын Web-құжаттың кез келген жеріне орналыстыруға болады. Скриптті ашу тэгінде скрипт тілі (language='JavaScript') көрсетілуі қажет.

2. *Скрипт коды бөлек файлда орналасқан*. Скрипт коды file.js файлына жазылған болсын:

```
function f1()
{
alert("Скрипт іске қосылды");
}
```

Скриптке баратын жол мен файл атын scr атрибутында көрсетеміз. Скрипт жазылған файлды Web-құжатқа төмендегіше байланыстырамыз.

```
<script language='JavaScript' scr=”file.js”> </script>
```

Скриптті жұмысқа қосу үшін батырмадан пайдаланамыз.

```
<input type="button"  value="Скрипт"  onclick="f1()">
```

Сонымен Web-құжат коды төмендегідей жазылады:

```
<html>
<head>
   <script language='JavaScript' src='file.js'> </script>
</head>
<body>
   <input type="button"  value="Скрипт"  onclick="f1()">
</body>
</html>
```

Скрипт жазылған Web-құжатта төмендегі батырма пайда болады.



Батырмасы бассақ жоғарыда көөрсетілген хабарлама шығады (сурет 15).

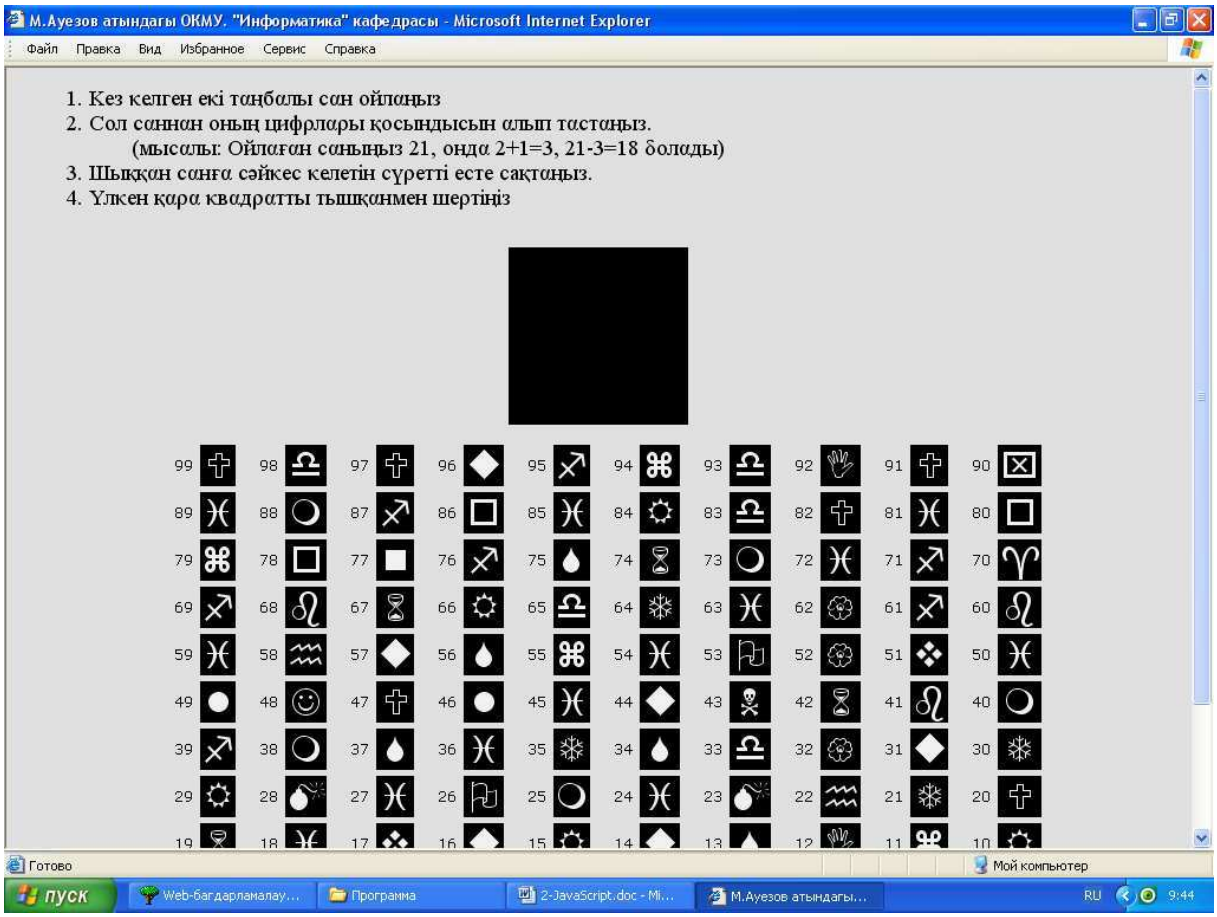
3.6 JavaScript-ке мысалдар

1. «Қара квадрат» скрипті. Скрипт орындауға дайын түрде Web-құжатқа жазылған. Файлды орындау үшін оны html кеңеймесімен сақтау керек. Кейде браузер JavaScript-ті бірден орындамайды, яғни оны орындауға сұраныс жасау керек болады.

```
<!DOCTYPE HTML PUBLIC "-//W3C//Dtd HTML 4.0 Transitional//EN">
<HTML><head><title>М.Ауезов атындагы ОҚМУ.  "Информатика"
кафедрасы</title>
<meta http-equiv=Content-Type content="text/html; charset=windows-1251">
</head>
<body bgcolor="e0e0e0"> <P style="font-size: 13px">
<style type=text/css>.num {
font-size: 12px; width: 30px; color: #000; font-family: verdana; text-align:
right}
.symbols {
font-size: 30px; background: #000; color: #f1f1f1; font-family: wingdings; text-
align: center}
.viewsymbol {
font-size: 100px; background: #000; width: 150px; color: #f1f1f1; font-family:
wingdings; height: 150px}
</style>
<script language=javascript type=text/javascript>
<!--//
ax=0;
function viewtable() {
ax=Math.round(Math.random()*26);
alphaArray=new Array("a", "n", "b", "d", "f", "h", "{", "i", "l", "v", "x", "z", "I",
"J", "M", "N", "o", "O", "R", "S", "T", "U", "m", "6", "^", "u", "_", "[", "]");
table="<table align=\"center\" border=\"0\" cellspacing=\"5\"
cellpadding=\"1\"><tr>";
j=1;
for ( i = 99 ; i >= 0 ; i-- ) {
a=Math.round(Math.random()*26);
if(i%9 == 0 &&  i < 89)
a=ax;
table+="<td class=\"num\">"+i+"</td><td
class=\"symbols\">"+alphaArray[a]+"</td>";
if(j%10 == 0) table+="</tr><tr>"; j++;
}
table+="</table>";
sym.innerHTML=table
sh.innerHTML=""
}
```

```
function show() {
sh.innerHTML=alphaArray[ax]
sym.innerHTML="<center>&#927;сы сурет пе -? <a
href=\"javascript:viewtable()\"><b>&#954;&#945;йт&#945;л&#945;у</b></a
></center>";
}
//-->
</script>
<table cellSpacing=0 width="100%" border=0>
<tbody>  <tr>
<td vAlign=top>
<dd>1. &#922;ез &#954;елген е&#954;і т&#945;&#1187;б&#945;лы
с&#945;н &#959;йл&#945;&#1187;ыз
<dd>2. С&#959;л с&#945;нн&#945;н &#959;ны&#1187;
циф&#961;л&#945;&#961;ы &#1179;&#959;сындысын &#945;лып
т&#945;ст&#945;&#1187;ыз.
  <dd>           (мыс&#945;лы:
&#927;йл&#945;ғ&#945;н с&#945;ны&#1187;ыз 21,
&#959;нд&#945; 2+1=3, 21-3=18 &#948;&#959;л&#945;ды)
<dd>3. Шы&#1179;&#1179;&#945;н с&#945;нғ&#945; сәй&#954;ес
&#954;елетін сү&#961;етті есте са&#1179;т&#945;&#1187;ыз.
<dd>4. Үл&#954;ен &#1179;&#945;&#961;&#945;
&#954;в&#945;д&#961;&#945;тты тыш&#1179;&#945;нмен
ше&#961;ті&#1187;із
<dd> 
<table style="cursor: hand" onclick=show() cellSpacing=10 align=center
border=0>
<Tbody> <tr> <td class=viewsymbol id=sh
align=middle> </td></tr></Tbody></table></dd></td></tr>
<tr>
<td id=sym vAlign=top height=300></td></tr></Tbody></table>
<script language=javascript type=text/javascript>
<!--//
viewtable();
//-->
</script>
</P>
<P style="font-size: 12px" align=right></P>
<HR align=left width="100%" color=#000000 noShade SIZE=1>
<script language=javascript><!--
bmQ='<img src=http://c.bigmir.net/?s75479&t5'
bmD=document
bmD.cookie="b=b"
if(bmD.cookie)bmQ+='&c1'
```

```
//--></script> &#917;сте б&#959;лсын: &#954;&#945;ме&#961;&#945;
жұмыс &#236;степ тұ&#961;, ми&#954;&#961;&#945;ф&#959;н
к&#959;сулы ... 
      &#927;йын &#945;лг&#959;&#961;итм&#236;н
т&#945;пқ&#945;нғ&#945; Сем&#948;иев
&#927;&#961;д&#945;&#948;&#945;й
&#945;ғ&#945;й 100$ &#948;е&#961;ед&#236;!
<script language=javascript1.2><!--
bmS=screen;bmQ+='&d'+(bmS.colorDepth?bmS.colorDepth:bmS.pixelDepth)+
"&r"+bmS.width;
//--></script>
<script language=JavaScript1.1 type=text/javascript><!--
a+=';j='+navigator.javaEnabled()
js=11//--></script>
<script language=JavaScript1.2 type=text/javascript><!--
s=screen;a+=';s='+s.width+'*'+s.height
a+=';d='+(s.colorDepth?s.colorDepth:s.pixelDepth)
js=12//--></script>
<script language=JavaScript1.3 type=text/javascript><!--
js=13//--></script>
</body></HTML>
```



Сурет 16 – Скрипт терезесі

4 PHP тіліне кіріспе

4.1 PHP тілінің қарапайым элементтері

PHP (Hypertext Preprocessor – гипермәтінді препроцессор) тілі серверде орындалатын сценарий жазуға арналған тіл. Браузер URL-адресте (Uniform Resource Locator – ықшамдалған ресурстар көрсеткіші) көрсетілген PHP тілінде жазылған Web-құжатты орындау үшін Web-серверге сұраныс жасайды. Web-сервер PHP-процессорын жүктеп, PHP-кодты орындайды ды, нәтижені Web-серверге қайтарады. Ал Web-сервер өз кезегінде нәтижені браузерге терезеде бейнелеу үшін жібереді.

PHP тілін 1994 жылы Расмус Лердорф (Rasmus Lerdorf) жасап шығарды. Оның негізгі мақсаты жеке сайттарда интерактивтік әдістер қолдану болатын. Расмус тілдің негізгі синтаксисі мен интерпретаторын жазып, оны PHP – Personal Home Page атады.

1995 жылы Расмус интерпретаторға қалыпты өңдеу FI (Form Interpretator) бөлімін қосып, интерпретаторды PHP/FI 2.0 деп атайды. Ал, 1998 жылы Зива Суракси мен Энди Гутманс (Zeev Suraski and Andi Cutmans) интерпретатор коды ашық болғандықтан (open source), оны қайта өңдеп, PHP 3.0 (Hypertext Preprocessor) версиясын шығарды. Осы кезден бастап PHP технологиясын дамытумен айналысатын PHP Group пайда болады. Сонымен, PHP тілі қолдануға оңай, түсінуге жеңіл, мүмкіндіктері көп болғандықтан жылдам тарап кетті. Қазіргі кезде тілдің 2000 жылы шыққан PHP 4.0 және 2004 жылы пайда болған PHP 5.0 версиялары белгілі. Төменде тілдің объектке бағытталған PHP 5.0 версиясын қарастырамыз [9-11].

PHP тілі Web-сервер мен пайдаланушы арасында деректер алмасуда, яғни Web-құжатта форум, қонақ кітабы, тіркелу анкетасы, хабарландыру тақтасы, чат құруда қолданылады.

Әдетте PHP тілінде жазылған құжатты PHP-скрипт деп атайды. Скрипттер .php кеңеймесі бар мәтіндік файлдар. Оларды құруда қарапайым мәтіндік редакторлардан пайдалануға болады. Мысалы: MS Windows ортасында Notepad (Блокнот) редакторы. Бірақта скрипттерді арнайы HTML-редакторларда құру ыңғайлы. Мысалы: 1st Page 2000 [12], PHPEdit [13], Web Development Studio [14].

PHP-скрипттер екі жолмен орындалуы мүмкін: Web–сервермен скрипт түрінде және консольдық программа сияқты. Біздің мақсатымыз Web–беттерді дайындау болғандықтан, біз бірінші жолын қарастырамыз.

Браузер бетті .php кеңеймесімен сұрағаннан кейін, Web-сервер программаны PHP-препроцессор арқылы өңдейді. PHP-скрипттер HTML-құжатқа енгізілетін PHP-кодтарымен беріледі. HTML-құжатқа PHP файл нұсқауын енгізу үшін оларды арнайы тэгтермен байланыстыру қажет. Оларды төмендегі төрт түрмен ажыратады.

1. // 1 – түрі – тэг XML стилімен

```
<? php
. . .
?>
```

2. // 2 – түрі – тэг SGML стилімен
```
<?
. . .
?>
```

3. // 3 – түрі – тэг ASP стилімен
```
<%
. . .
%>
```

4. // 4 – түрі – тэг HTML стилімен
```
<scrip language = “ php ”>
. . .
</ script>
```

Біз Web-құжаттарда көп тараған SGML стилін ұстанамыз. Стильдің тағы бір варианты бар, яғни мына төменде қарастырылған тэг түрінде пайдалануға болады.

```
<? =
. . .
?>
```

Мұндай түрде жазылған тэгтерде бірінші олардағы мағыналары орындалады, ал тэгтің өзі болса орындалған мағыналардың нәтижесімен ауысады:

бес қосу бес тең болады <? = 5 + 5>

Нәтижесінде браузер терезесінде біз мынаны көруіміз керек:

бес қосу бес тең болады 10- ға.

Бұдан былай түсінікті болу үшін тэгтердің қысқартылған жазуын қолданамыз, ал шығаруда echo функциясын пайдаланамыз. Бұл функцияның жұмысы әр түрлі ақпаратты (мәтін, амалдар нәтижесі, HTML тэгтері, т.с.с) браузерге шығару үшін қолданады.

Мысалы: бес қосу бес тең болады <? echo (5 + 5); ?>

PHP тілінің қарапайым элементтеріне түсініктемелер, қызметші сөздер, тұрақтылер, айнымалылер және функциялар жатады. Енді осыларды қысқаша қарастырайық.

Түсініктемелер. PHP тілі бізге түсініктеме қоюдың бірнеше түрін ұсынады. Олардың ішінен екі қисық сызықтан (//) кейін түсініктеме жазу ең қарапайым болып есептеледі. Осы екі сызықтан кейін PHP процессоры осы қатардың соңына дейін өңдемейді.

Түсініктемені UNIX тілінің скрипт түсініктемелері сияқты нөмір (#) символының көмегімен де беруге болады. Сонымен қатар, C++ тілі стиліндегі (/*..*/) түсініктемелерді де қолдануға болады. Түсініктеме берудің түрлі жолдары төмендегі мысалда келтірілген.

```
<?
echo (“Hello”); // бұл түсініктеме
echo (“Hello”); # UNIX тілі стиліндегі түсініктеме
/* C++ тілі стиліндегі түсініктеме */
?>
```

PHP түсініктемелері PHP-кодының қысқартылған тэгтерінде ғана жұмыс істейді. Егер түсініктеме символдары қысқа тэг құрамында орналаспаса, онда олар әр түрлі мәтін сияқты браузерде көрсетілуі мүмкін.

Мысалы:

```
<?
echo (“Hello”); //
?>
```

Түсініктемені тек қана оператордан кейін емес, мына түрде де қоюға болады.

```
<?
$a = “Hello, world”;
echo strstr (// бұл функцияны кейінірек қарастырамыз
$a,  “H”);
?>
```

Қызметші сөздер деп JavaScript тілінің жұмысына қажетті нұсқауларды айтады. Әрбір сөздің өз мағынасы, атқаратын қызметі бар. Қызметші сөздерді жұмыс барысында біртіндеп мағынасы қарай түсіндіріп отырамыз.

Тұрақтылар. Тұрақтылар PHP тілінде define (): функциясы көмегімен жазылады. Функцияның жалпы форматы:

```
define (constant, value)
```

Бұл функцияның бірінші параметрі – тұрақтының атауы, ал екіншісі - оның мәні. Пайдалану алдында тұрақтының мәні оның атына меншіктеледі:

```
<?
define (name, 15);
define (t, “\x20”); // бос орынның коды
define (data, “Hello);
echo (name);
echo (t);
echo (data);
?>
```

PHP тілінде тұрақтыларға қатысты тағы бір defined () функциясы қолданылады. Бұл функция тұрақты анықталған ба, жоқ па соны тексереді:

```
defined (constant)
```

Бұл функцияның бір ғана параметрі бар, ол анықталатын тұрақтының атауы. Егер тұрақты анықталған болса, онда функция true мәнін қабылдайды, кері жағдайда false мәніне ие болады.

Мысалы:

```
<?
define (constant, “Hello”);
if (defined (“constant”))
{
echo (“constant is defined”);
}
?>
```

Айнымалылар. PHP тілінде айнымалы $ (ақша бірлігі) белгісімен басталатын әріп, цифр және _ (сызықша) белгісінің тізбегін айтады. Бірақта ($) кейінгі символ әріп болуы шарт, яғни сан, не болмаса арнайы белгі болмауы керек.

Мысалы: $a, $x1, $user_Func_1

Қызметші сөздерге қарағанда айнымалылардың аты PHP тілінде регистрге сезімтал болады. PHP процессоры регистрды есепке алады, яғни бас және кіші әріптерді бөлек деп қабылдайды.

Мысалы:

```
$user_func
$User_func
$USER_FUNC
```

Бұлардың жазылуы бір болғанымен регистрінде айырмашылық бар. Оларды PHP процессоры әр түрлі айнымалылар деп қабылдайды.

PHP тілінің басқа тілдерден тағы бір ерекшілігі, айнымалыны пайдаланудан алдын оның типін көрсету шарт емес. Сонымен бірге, айнымалы скрипт орындалу барысында түрлі типтер қабылдау мүмкін.

4.2 Деректердің типтері және амалдар

PHP тілінде қолданылатын деректер типтерін қарастырайық:

- Integer
- String
- Float (double, real)
- Array
- Boolean
- Object

PHP тілінде айнымалының типін сипаттау үшін settype функциясы қолданылады. Ол айнымалының типін типін белгілейді.

Мысалы:

```
<?
$var = “15”;
echo(gettype($var));
settype ($var, integer);
echo “<br>”;
echo(gettype($var));
?>
```

PHP тілінде айнамалының типін анықтау үшін gettype функциясы қолданылады. Бұл функцияның бір ғана параметрі бар, ол типі анықталатын тұрақтының атауы.

Мысалы:

```
<?
$var_str = “name”;
$var_num = 15;
echo (gettype ($var_str));
echo “<br>”;
echo (gettype ($var_num));
?>
```

Бағдарлама коды орындалғаннан соң, мына нәтижені аламыз:

```
String
Integer
```

Сондай-ақ, PHP тілінде айнымалының типін өзгерту үшін gettype функциясымен қатар, типті сипаттау операторы да қолданылады. Жаңа тип жақшаның ішінде көрсетіледі:

```
$var = (int) $var;
```

Мысалы:

```
<?
$var = “5”; // string типі
$var = (int) $var; //  int
echo (gettype ($var));
?>
```

PHP тілінде амалдар (операторлар) төмендегі түрлерге бөлінеді:

- меншіктеу
- арифметикалық
- биттік
- салыстыру
- логикалық

Меншіктеу амалы немесе меншіктеу операторы = (теңдік) белгісімен беріледі. Әдетте меншіктеу амалы тек қана бір айнымалы мәнін меншіктеу керек болса пайдаланылады. Айнымалының аты сол жақта, ал оның мәні оң жақта жазылады.

Мысалы:

```
<?
$var = 5;
echo ($var);
?>
```

Арифметикалық амалдар. Сандық деректер үстінде төмендегі арифметикалық амалдар орындалады:

+	қосу
-	алу
/	бөлу
*	көбейту
%	арифметикалық модуль – бүтін бөлу амалының қалдығы
++	инкремент – операнд мәнін бір бірлікке арттыру
--	декремент - операнд мәнін бір бірлікке азайту

Арифметикалық амалдағы нәтиженің типі, амалда қатысатын деректердің типімен анықталады.

Мысалы:

```
$c = 4*$a-$b;   - бірнеше арифметикалық амал қолданылған;
$c = $c++;      - инкремент амалы қолданылған;
$c = $c--,      - декремент амалы қолданылған;
```

PHP тілінде кейде амалдарды қысқартып жазуға болады. Мысалы, $a = $a + $b өрнегін қысқартылған түрде былай жазуға болады, $a + = $b. Мұндай әрекеттер жоғарыда көрсетілген барлық амалдарға рұқсат етілген.

Арифметикалық амалдарды операндтардың санына қарай екіге бөледі, унарлы және бинарлы. Унарлы амал бір операндпен жұмыс істейді, ал бинарлы екі операндпен жұмыс істейді. Инкремент және декремент амалдарынан басқа барлық арифметикалық амалдар бинарлы болып есептеледі.

Енді инкремент және декремент амалдарын қарастырайық. Бұл амалдарды префиксті және постфиксті деп екіге бөледі. Инкременттің префикс амалында операнд мәнінің өсуі, оны қолданғанға дейін орындалады. Ал постфиксті инкремент амалында операнд мәнінің өсуі, оны қолданғаннан кейін орындалады. Осылай инкремент және декремент амалдарын екі түрде қолдануға болады.

Мысалы:

```
<?
// префиксті инкремент амалы
$var = 1;
++$var;
// постфиксті инкремент амалы
$var = 1;
$var++;
?>
```

Келтірілген мысалда префикстік және постфикстік жазулар арасындағы айырмашылық үлкен емес, екеуінің де нәтижесі бірдей. Бірақта бұл амалдармен жұмыс барысында мұқият болған жөн. Инкремент амалының дұрыс қолданбағанына мысал келтірейік.

Мысал:

```
<?
// 1 - жағдай – префиксті инкремент амалы
$var = 1;
Echo (+ + $var);
// 2 – жағдай – постфиксті инкремент амалы
$var = 1;
echo ($var + +);
?>
```

Бірінші жағдайда біз 2 санын аламыз. Себебі инкременттің префикстік амалы бірінші амалды орындап, одан кейін алынған мәнді шығарады. Екінші жағдайда нәтиже ретінде біз 1 санын аламыз.

Инкремент және декремент амалдарында операнд тұрақты болмауы қажет. Мысалы, ++1 жазуы дұрыс емес, сол сияқты, ++($a + $b) жазуы да қате, яғни операнд ретінде екі айнымалылы өрнекті қолдануға болмайды. Инкремент және декремент амалдарында операнд тек қана біреу болуы тиіс.

PHP тілінде инкремент және декремент амалдарын тек қана integer типті бүтін айнымалыларға ғана емес, басқа типтерге де қолдануға болады. Инкремент амалын string типіндегі айнымалыға қолданып көрейік.

Мысалы:

```
<?
$var = “aaa”;
echo (+ + $var);
?>
```

Нәтижеде біз aab қатарын аламыз, себебі алфавитте b әрпі a әрпінен кейін келеді.

Биттік амалдар арифметикалық амалда сияқты сандық мәндерді өңдейді.. Бұл жерде бүтін сандар биттер қатары болып көрсетіледі.

~	инверсия – теріске шығару (not)
&	конъюнкция – логикалық көбейту (and)
|	дизъюнкция – логикалық қосу (or)
^	логикалық алу (xor)
<<	солға жылжыту
>>	оңға жылжыту

Биттік амалдардың мәндері:

a	b	~a	~b	a & b	a | b	a ^ b
0	0	1	1	0	0	0
0	1	1	0	0	1	1
1	0	0	1	0	1	1
1	1	0	0	1	1	0

Мысалы:

```
<?
echo ( 4 << 2 ); // нәтиже 16-ға тең болады
echo “<br>”;
echo ( 5>>1 ); // нәтиже 2-ге тең болады
echo “<br>”;
echo ( 6 & 5 ); // нәтиже 4-ге тең болады
echo “<br>”;
echo ( 6^5 ); // нәтиже 3-ке тең болады
echo “<br>”;
```

```
?>
```

Осы мысалдарды талдайық, 4 саны үшін екілік код 100-ге тең, екі разряд солға қарай жылжытсақ, 100 коды 10 000 болады, бұл ондық кодта 16 санына тең. Әдетте, n позиция солға қарай жылжыту 2^n –ге көбейткенге тең, ал оңға қарай жылжыту осы мәнді 2^n рет кішірейтеді және нәтиженің бөлшек бөлігін тастайды (сондықтан 5>>1 амалы 2-ге тең). Қалған мысалдар осы сияқты логикалық жолмен қарастырылады.

Салыстыру амалдары. Деректер үстінде төмендегі салыстыру амалдары орындалады:

>	үлкен
>=	үлкен немесе тең
<	кіші
<=	кіші немесе тең
= =	тең
!=	тең емес

Салыстыру амалының нәтижесі *true* немесе *false* мәндеріне ие. Бұл амалдарда деректер кез келген типке, ал нәтиже тек қана логикалық типке тиісті.

Логикалық амалдар. Логикалық амалдар тек қана логикалық *true* және *false* мәндері үстінде орындалады. Логикалық теріске шығару амалы бір аргументті, ал қалған амалдар екі аргументті.

!	теріске шығару (not)
&&	конъюнкция (логикалық көбейту) (and)
| |	дизъюнкция (логикалық қосу) (or)

a	b	!a	!b	a && b	a | | b
false	false	true	true	false	false
false	true	true	false	false	true
true	false	false	true	false	true
true	true	false	false	true	true

Өрнектердегі амалдардың орындалу тәртібі амалдар приоритетімен анықталады. Амалдар приоритетін жақшалармен өзгертуге болады. Өйткені жақша ішіндегі амалдар бірінші орындалады. Өрнектегі приоритеті бірдей амалдар солдан оңға қарай орындалады.

приоритет	*аталуы*	*амалдар*
I	унарлық	-, ++, --, ~, !
II	мультипликатив	*, /, %, &&
III	аддитив	+, -, ||
IV	салыстыру	>=, >, <, >=, = =, !=

4.3 Операторлар

Қарапайым және құрама операторлар. Жоғарыдада көрсетілген Echo (5 + 5) функциясы қарапайым операторға мысал болады. PHP-де барлық оператор ; (нүктелі үтір) белгісімен аяқталады.

Қарапайым оператор бірнеше қатарда жазылуы да мүмкін. Мысалы, мына екі кодтың мағынасы бірдей.

```
<?
echo (5 + 5);
echo (5 - 2);
echo (“Hello”);
?>

<?
echo (5   +
5); echo (5 –
2); echo (“Hello”
);
?>
```

Қарапайым операторлардан басқа тағы құрама оператор да бар, олармен операторлар тізбегін анықтауға да болады. Кейде құрама операторларды көп қатарлы деп те атайды. Құрама операторлардың қарапайым операторлардан айырмашылығы, ол нүктелі үтірмен аяқталмайды, оның аяқталғанын фигуралық жақша білдіреді.

If шартты операторы. Шартты оператор деректерді өңдемейді, өңдеуді басқарумен айналысады. Операторлардың жалпы форматы:

If (шарт) {оператор1} else {оператор2}

Егер шарт орындалса, яғни логикалық шарттың мәні true болса, онда оператор1 орындалады. Ал кері жағдайда, яғни логикалық шарттың мәні false болса, онда оператор2 орындалады.

Мысалы:

```
<?
$flag = true;
if ($flag)
{
echo “<p>” flag айнымалысы ақиқат мәнді қабылдады. “</p>”
}
else
{echo “<p> flag айнымалысы жалған мәнді қабылдады “</p>”
}
?>
```

Көрсетілген мысалда шартты оператор алдымен flag айнымалысын тексереді. Егер шарт қанағаттанарлық болса, онда бірінші фигуралық жақшаның ішіндегі оператор орындалады, ал егер шартты қанағаттандырмаса, онда екінші фигуралық жақшаның ішіндегі оператор орындалады.

Шартты операторды қысқартылған түрде де жазуға болады:

If (шарт) {оператор}

Мысалы:

```
<?
$flag = true;
if ($flag)
{
echo “<p>” flag айнымалысы ақиқат мәнді қабылдады. “</p>”
}
?>
```

Егер шартты операторда тек қана жалғыз шарт қарастырылса, онда логикалық шарт онша күрделі болмайды. Ал егерде екі немесе одан да көп шарттар қарастырылса, онда шарт құрауда логикалық амалдар қолданылады.

Мысалы:

```
<?
$flag1=true
$flag2=true
if ($flag1 && $falg2)
{
echo “<p>” Екі шарттың мәні ақиқат “</p>”;
}
else
{
echo “<p>” Екі шарттың кемінде біреуі жалған“</p>”;
}
?>
```

Switch таңдау операторы. Таңдау операторы айнымалыға берілген бірнеше мәннің ішінен біреуін таңдауда қолданылатын оператор. Осы таңдалған мәнде жазылған операторлар тізбегі орындалады. Switch операторының жалпы форматы:

```
Switch (expression)
{
case value:
statements;
```

```
break;
case value:
statements;
break;
default;
statements;
}
```

Таңдау операторы алдымен expression айнымалысының мәнін қабылдайды, содан кейін case операторындағы тұрақтылардың бірін таңдайды да, сол блоктағы операторларды break операторы кездескенше орындайды.

Цикл операторлары. Цикл операторы бағдарламаның бір бөлігінің бірнеше рет қайта-қайта орындалуын қамтамасыз етеді. PHP тілінде цикл операторының төрт түрі бар:

- while
- do…while
- for
- forreach

Алдыңғы үш цикл операторлары C++ тілінен алынған болса, ал соңғы оператор Perl тілінен алынған.

While цикл операторы қайталаулар саны логикалық шартқа байланысты болған жағдайда қолданылады. While операторында басында логикалық шарт тексеріліп, содан кейін операторлар тізбегі орындалады. While операторының жалпы форматы:

```
While (шарт)
{
операторлар тізбегі;
}
```

Циклдың орындалу тәртібі мынадай, алдымен логикалық шарт тексеріледі. Егер шарт ақиқат болса, онда операторлар тізбегі орындалады.

Мысал:

```
<?
$var = 5;
$i = 0;
while (++$i <= $var)
{
echo($i);
echo('<br>');
}
?>
```

Циклдан шығу үшін break операторы қолданылады. Бұл оператор орындалу барысында кездессе, онда циклдің орындалуы тоқтайды.

Мысал:

```
<?
$var=7;
$i=0;
while (++$i<=$var)
echo($i);
echo(‘<br>’);
if ($i==3) break;
}
?>
```

Егер тек қана орындалып жатқан оператордан шығу керек болады, онда continue операторы қолданылады.

Мысалы:

```
<?
$var = 7
$i = 0;
while (++$i <=$var)
{
if ($i==5)
{
continue;
}
echo($i);
echo(‘<br>’);
}
?>
```

Do…while операторы. Операторды жалпы форматы:

```
Do
{
операторлар тізбегі;
}
While (шарт);
```

Бұл оператор while операторына ұқсас, тек қана логикалық шарт цикл соңында тексеріледі. Сондықтан операторлар тізбегі кемінде бір рет орындалады. Цикл соңында шарт тексеріледі, егерде шарт қанағаттанарлық болса, онда цикл жалғасады.

Мысал:

```
<?
$var = 5;
$i = 0;
do
```

```
{
echo($i);
echo('<br>');
}
while(++$i<=$var)
?>
```

for цикл операторы қайталаулар саны белгілі болғанда қолданылады. Оператордың жалпы форматы:

```
For (мән1, мән2, мән3)
{
операторлар тізбегі;
}
```

мұндағы: мән1 - циклдің бастапқы мәні; мән2 – логикалық шарт; мән3 – операторлар тізбегі.

Мысал:

```
<?
$var = 5;
$i = 0;
for ($i=0; $i<=$var; $i++)
{
echo($i);
echo('<br>');
}
?>
```

Бағдарлама кодында циклдар кейде бір-бірінің ішінде орналасуы мүмкін. Мұндай жағдайда ішінде орналасқан циклды ішкі цикл деп, ал сыртында орналасқан циклді сыртқы цикл деп айтады.

Мысал:

```
<?
For ($i=2; $i<100; $i++)
{
for ($j=2; $j<$i; $j++)
{
if (($i%$j)!=0)
continue;
else
{
$flag=true;
break;
}
}
if (!$flag)
```

```
{
echo($i);
echo(“ ”);
}
$flag=false;
}
?>
```

4.4 Функциялар

PHP тілінде функция бағдарламаның өз алдына орындалатын бөлігі ретінде қарастырылады. Функцияны бағдарламаның қажет жерінде, қалағанша шақырып орындатады. Қарапайым түрде функция бірнеше айнымалылардан тұрады. Функция осы берілген айнымалыларды пайдаланып қандайда бір іс-әрекетті орындайды және нәтижені бағдарламаға қайтарады.

PHP тілінде функцияны сипаттаудың жалпы форматы:

```
Function Myfunction (айнымалы)
{
операторлар;
returm айнымалы;
}
```

мұндағы, Function - қызметші сөз; Myfunction – функцияның атауы.

Егер функцияда айнымалылар бірнешеу болса, онда олар үтірлермен бөлектеніп жазылады.

Мысалы:

```
Function Myfunction ($var, $var1, $var2)
```

Егерде функцияда нәтиже негізгібағдарламаға қайтарылатын болса, онда функцияда міндетті түрде return операторы болуы керек.

Мысалы:

```
<?
Function get sum()
{
$var=5;
$var1=10;
$sum=$var+$var1;
return $sum;
}
echo (get sum());
?>
```

Көрсетілген мысалда екі санның қосындысы шығады. Бұл функция айнымалылардың мәнін қабылдайды да, осы екі айнымалының қосындысын нәтиже ретінде негізгі бағдарламаға қайтарады. Сонан соң echo операторын шақырады да, нәтижені браузер терезесіне шығарады.

Осы мысалды жалғастырайық, енді алынған нәтиже негізгі бағдарламаға қайтарылмай, тікелей браузерге шығарылсын:

```
<?
Function get sum ()
{
$var=5;
$var1=10;
$sum=$var+$var1;
echo $sum;
}
get sum ();
?>
```

Функцияда $var және $var1 айнымалыларын бағдарлама ішінде сипаттау қажет емес.

```
<?
Function get sum ($var ,$var1)
{
$sum =$var+$var1;
echo $sum;
}
get sum (5,2);
?>
```

Егер функцияда айнымалы аргумент арқылы берілсе, онда ол параметр функциясы деп аталады. Мысалы:

```
<?
Function get sum ($var)
{
$var=$var+5;
return $var;
}
$new var =20;
echo(get sum($new var)); // 25 шығады
echo (“<br>$new var”); // 20 шығады
?>
```

Мысалда көрсетілгендей 5 және 20 сандары аргумент болады, ал айнымалылар $var және $var1 - функция параметрлері деп аталады.

Сонымен қатар, айнымалыларды локальды және ауқымды деп екіге бөледі. Локалды айнымалылар тек қана осы функция ішінде, ал ауқымды айнымалылар бағдарламаның кез-келген жерінде анықталады.

Локальды айнымалыны ауқымды етуге болады. Ол үшін айнымалыны global қызметші сөзімен сипаттайды, сонда оны кез келген жерден шақыруға болады. Айнымалының өмірі деп бағдарламаның орындалуы уақытын атайды. Мысалы:

```
<?
Function get sum ()
{global $var;
$var=5;
echo $var;
}
$var =10;
echo (“$var<br>”);  // 10 шығады
get sum (); // 5 шығады
?>
```

Локальді айнымалы шақырылу кезінде жаңадан сипатталады. Мысалы:

```
Function   counter ()
{
$counter = 0;
retutrm ++ $counter
}
```

Сонымен қатар, локальды айнымалы қайтадан шақыруда өз мәнін сақтап қалу үшін, оны сипаттауда static қызметші сөзінен пайдаланады.

```
Function   counter ()
{
static $counter = 0;
retutrm ++$counter
}
```

PHP тілінде көптеген стандарт функциялар анықталған. Олар PHP-скрипттерде көп қолданылады. Оларды PHP-скрипт кезекпен орындайды. Енді осы стандарт функиялардың жиі кездесетіндерін қарастырайық.

Мысалы: PHP-скрипт тұтынушы сайтқа кіргенде, оның IP-адресін, хосттағы браузер типін анықтау, мәліметтерді сұрыптау, файлға мәлімет енгізу сияқты іс-әрекеттерді орындау.

```
<?
Function get brows() // тұтынушы браузерінің типін анықтау
{
…
}
Function get  os () // тұтынушы операциялық жүйесін анықтау
```

```
{
…
}
Function get ip () // тұтынушының IP-адресін анықтау
{
…
}
Function get host () // хост атын анықтау
{
…
}
Function is search engine ($referer) // іздеу жүйесімен сайтқа кіру
{
…
}
Function displeyDate () // мәліметтерді сұрыптау және
// оларды браузерге шығару
{
…
}
Function main ()   // сипатталған функцияның орындалуы
{
get brows ();
get os ();
get referer ();
get ip ();
…
createDate ();
getdate ();
displeyDate ();
}
main ();
// main () функциясын шақыру
?>
```

Strcmp функциясы. Жалпы форматы:

```
Int strcmp (string str1, string str2)
```

Strcmp функциясы екі қатарды салыстырады:

- егер қатарлар бір-біріне сәйкес келсе, онда 0 саны шығады;
- егер str1 қатары str2 қатарынан үлкен болса, онда 1 саны шығады;
- егер str1 қатары str2 қатарынан кіші болса, онда -1 саны шығады.

Мысалы:

```
<?
$str1= “ttt”;
$str2= “tttttttt”
echo (“($str1, $str2): -салыстыру нәтижесі”);
echo (strcmp (str1, str2)); echo (“<br>”);
echo (“($str2, $str1): -салыстыру нәтижесі ”);
echo (strcmp (str2, str1)); echo (“<br>”);
echo (“($str1, $str1): -салыстыру нәтижесі ”);
echo (strcmp (str1, str1));
?>
```

Нәтиже:

```
(ttt, tttttttt): -1 -салыстыру нәтижесі
(tttttttt, ttt):1 -салыстыру нәтижесі
(ttt, ttt): 0 -салыстыру нәтижесі
```

Strncmp функциясы. Егер қатарды регистрді есептемей салыстыру қажет болса, онда Strcasecmp () функциясын қолдансақ болады. Функцияның жалпы форматы:

```
Int strncmp (string str1, string str2, int len)
```

Бұл функцияның strncnp функциясынан айырмашылығы, функция қатар басын салыстырады. Дәлірек айтсақ, бірінші len байттарын салыстырады. Егер len ең кіші қатар ұзындығынан да кіші болса, онда қатарлар толық салыстырылады.

4.5 PHP және Apache локальді серверін орнату

Локальды серверлерді пайдаланудың себебі көп. Мысалы, PHP-ді үйрену үшін керек болады немесе Интернеттегі хостингде Web-жобаларды тестілеу қымбатқа түседі. Мұндай жағдайларда дербес компьютерде Apache+PHP байланыстары қажет болады.

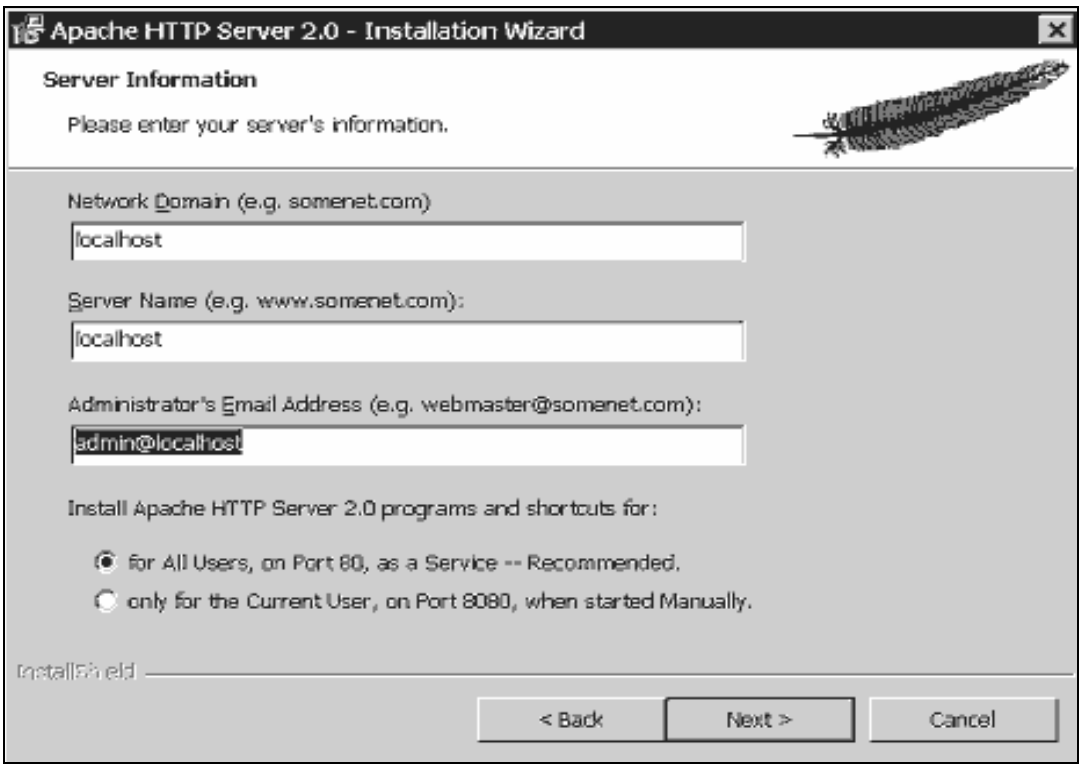
Ең алдымен Интернеттен Apache архивін көшіріп алу керек. Оны Apache ресми сайтындағы мына беттен алуға болады [15]. Іздестіруде мынаны есте сақтаған жөн. Apache өзінің UNIX тегіндегі домен аты бойынша httpd деп те аталуы мүмкін. Әдетте сайт бетінде әртүрлі файлдар көп, мысалы: httpd-2.0.49-win 32-src.zip httpd-2.0.49. tar. Gz, apache 2.0.50-win 32-x86-no ssl-exe.

Ескерту: Apache тасымалдаушыларының бинарлы кодтары әртүрлі нұсқада таралады. Мысалы: *exe және *msi кеңеюімен және де hettpd версия win 32 *msi түріндегі атауға ие болуы мүмкін. Шатасып қиналмау

үшін оны мына сайттан алуға қажет [16]. Екінші және үшінші версияда айырмашылық болуы мүмкін, сондықтан ең соңғы версияны таңдаған абзал. Өйткені онда алғашқы версиядағы кездескен қателіктер жойылған.

PHP 5 бағдарламасын оның ресми сайтынан табуға болады [17]. Сайтта PHP екі формада берілген болады. Алғашқы кодта (Complete Source Code) және компилицияланған нұсқада (Windows Binsries). Бізді екі нұсқада таралатын компиляцияланған версия қызықтырады: орнату түрінде (php-5.0.0-installer exe) және zip архив түрінде (php-5.0.0-Win 32. zip). Орнату түрі ыңғайлы, бірақ PHP версиясы шектеулі. Оның үстіне автоматты инсталляторды пайдаланудан Apache серверінің конфигурация файлын құру қажеттілігі туындайды. Сондықтан да zip-архивті жүктеу ұсынылады. Сіз PHP сайтынан PHP-ге арналған құжаттаманы алуыңызға да болады. MS Windows операциялық жүйесін пайдаланушылар үшін chm-формат ыңғайлы.

Сонымен, Apache Web-серверінің орнатушы бағдарламасын жұмысқа берейік. Экранда лицензиялық келісімі бар терезе пайда болады. Одан Apache версиясының жаңа кіріспесі туралы ақпараты бар келесі терезеге өту керек. Келесі терезе сервер туралы ақпарат енгізуге мүмкіндік береді, олар: сервердің домендік аты, сервердің аты, тұтынушының электрондық поштасы. Егер орнату локальды машинада жүретін болса, домендік атау мен сервер атауына өріске localhost жазуын енгізу керек болады. Терезенің төменгі бөлігінде сервер сұранысты қабылдайтын порт нөмерін таңдау керек болады (8080 порт). Ол төмендегі суретте көрсетілген (сурет 17). Localhost – бұл IP-адреспен байланысты локалды машинада серверді пайдалануға арналған атау.



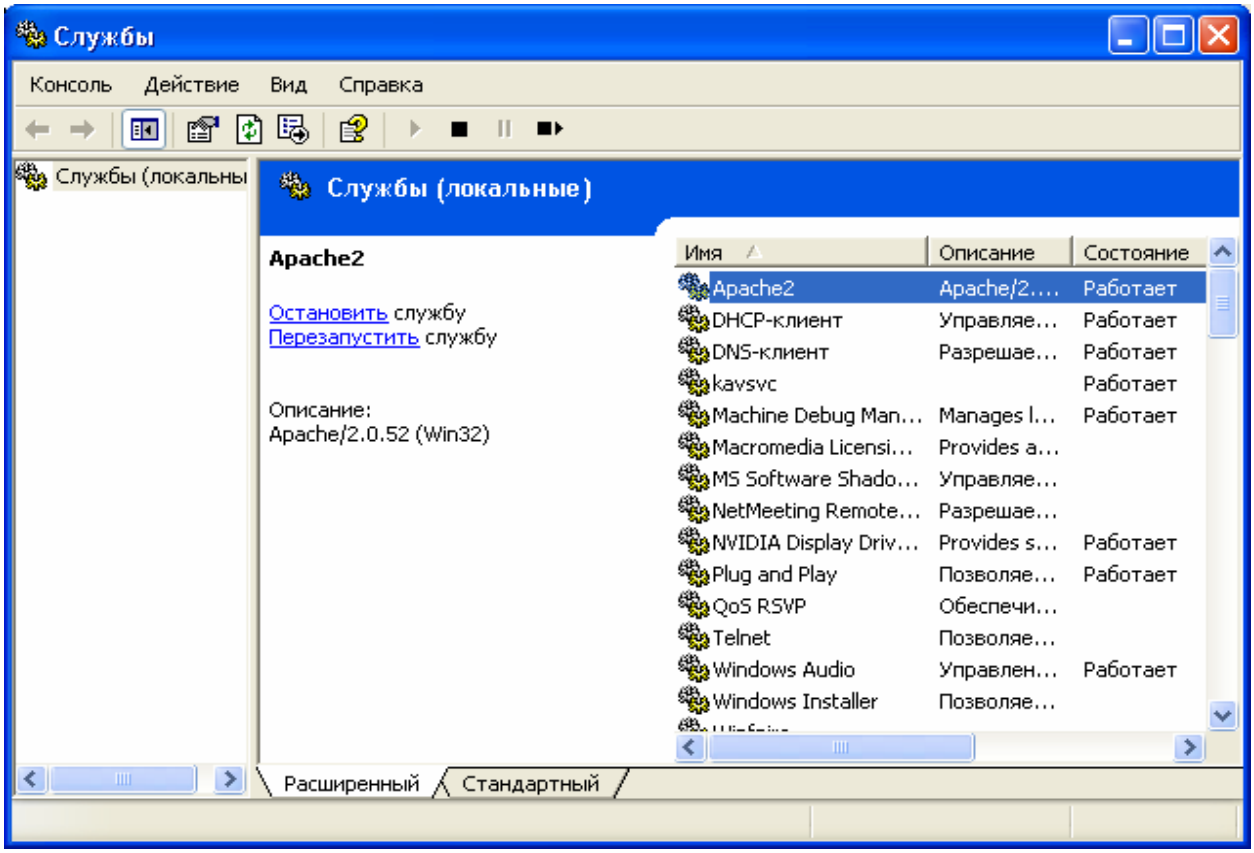
Сурет 17 - Apache Web-серверінің терезесі

Әрі қарай Apache серверін орнату әдістері ұсынылады. Стандартты (Typical) немесе таңдаушы (Custom). Бұл әдістер сервер компонентін қолмен таңдауға мүмкіндік береді. Келесі терезе серверді орнатудың каталогын таңдауға арналған. Әдетте C: Program Files Apache Group, дегенмен біз басқа каталогты таңдауды ұсынамыз. Мысалы C: www. Бұдан кейін орнату шебері орнату үрдісіне дайын екендігін хабарлайды.

Install түймесін басқан соң сервер файлдарын көшіру жүзеге асырылады. Егер орнату сәтті шықса, Windows автоматты түрде Apache серверін жұмысқа қосады. Орнату кезіндегі сәтті инсталляциядан кейін браузер терезесіне http://localhost/ немесе http://127.0.0.1/ сервер бетіне жүктелуі керек.

Енді Apache-ні басқаруды үйрену керек. Атап айтсақ серверді іске қосуды, жұмыс істеуді және тоқтатуды үйрену керек. Бұл іс-әрекеттерді орындаудың бірнеше тәсілдері бар. Олар: ApacheMonitor терезесі, Windows сервистерін басқару консолы, Apache серверінің Start мәзірі. Басқару консолын жіберу үшін мына нұсқауды орындау қажет.

Пуск -> Настройка->Панель управления-администрирование-> Службы.



Сурет 18 - Apache 2 серверін басқару

Нұсқаулар орындалғаннан соң «Службы» терезесі пайда болады (сурет 18). Терезенің пайда болған бетінде, берілген суретте көрініп тұрғандай

Apache 2 сервисін таңдау керек. Тышқан манипуляторының оң жақтағы пернесін басу арқылы ашылған жанама мәзір сервистің жіберуін, тоқтатуын, жүктелуін жүзеге асыруға мүмкіндік береді.

Windows қызметтері жүйенің жүктелуінде фондық қосымшаларды шығаруды жүзеге асырады. Бұл үшін «Свойства» терезесіне өту керек. Сервистің жанама мәзірінен «Свойства» қатарын таңдап, пайда болған терезеде “Тип запуска” тізімінен “Авто” қатарын таңдау керек.

Енді C: /www/ Apach 2/conf. бумасында орналасқан Apache–http.conf. Web-серверінің конфигурациялық файлын орнатуға кірісу керек. Ескерте кетейік, бұл файлда жүзеге асырылған өзгерістер серверді қайта жүктегеннен кейін ғана орындалады.

Жалпы, Apache сервері ISO кодында жұмыс істейді, яғни сервер осы кодта жазылған файлдарды ғана өңдейді. Сондықтан ISO кодын WinCode кодына өзгерту керек. Ескерте кетейік, Қазақстанда WinCode кодының 1251 парағы (Ресей парағы) қолданылады. Сонымен, кодты өзгерту үшін, http.conf файлын ашып, мына қатарды табу керек:

AddDefaultCharset ISO-8859-1

Осы қатарды мынаған өзгерту керек:

AddDefaultCharset –Windows-1251

Әрі қарай, кеңейтілген файлдар орналасқан бумасын құру керек болады. Оларды мына мен сайттан табуға болады [18].

Сонымен, Apache сервері құрылды. Әрі қарай PHP-ді құруға кірісу керек. Сондықтан C: /PHP бумасын құрамыз және осы бумаға php-5.0.0-Win32.zip. архивін ашамыз. Ал, PHP-ді орнату бумасы ретінде бөлек бума құру қажет. Ол үшін, мысалы: Internet бумасын құрып алып, Пуск нұсқауын орындап, subst E: C:\Internet нұсқауын береміз. Нәтижеде бізде тағы да виртуалды E дискісі пайда болады. Енді сол дискте usr бумасын құрып, содан кейін usr бумасының ішіне php бумасын құрамыз.

Орнату файлы ретінде екі файл бар, біріншісі exe, ал екіншісі zip файл. Алдымен exe файлды жүктеп алып, пайда болған сұраныс терезелерінде Next батырмасын басып отыру қажет. Сұраныста Standart типін орнатқан жөн. Орнату жолына e:\usr\php қатарын көрсету керек. Әрі қарай SMTP серверін және электронды пошта адрестерін енгіземіз. Содан кейін бізге PHP жұмыс істейтін серверді таңдау қажет болады. Ол сервер ретінде біз Apache серверін аламыз.

Енді бізге қосымша модульдерді орнату мен Apache серверіне PHP кодтарын түсінуді үйрету керек болады. PHP-ді Apache-ге келтіру үшін E:/usr/apache/conf бумасынан Apache-дің httpd.conf файлын кез-келген мәтіндік редактормен ашамыз. Файлдан

AddType application/x-httdp-php .php

қатарын іздейміз. Осы қатардан түсініктеме (#) белгісін алып тастау керек. Сосын осы қатардан кейін

```
ScriptAlias /_php/ “PHP жолы”
Action application /x-httpd-php “/_php/php.exe”
```

қатарларын жазамыз. Мұндағы, «PHP жолы» қатары біздің жағдайымызда e:/sr/php/. Бұдан кейін файлды сақтап, жабамыз.

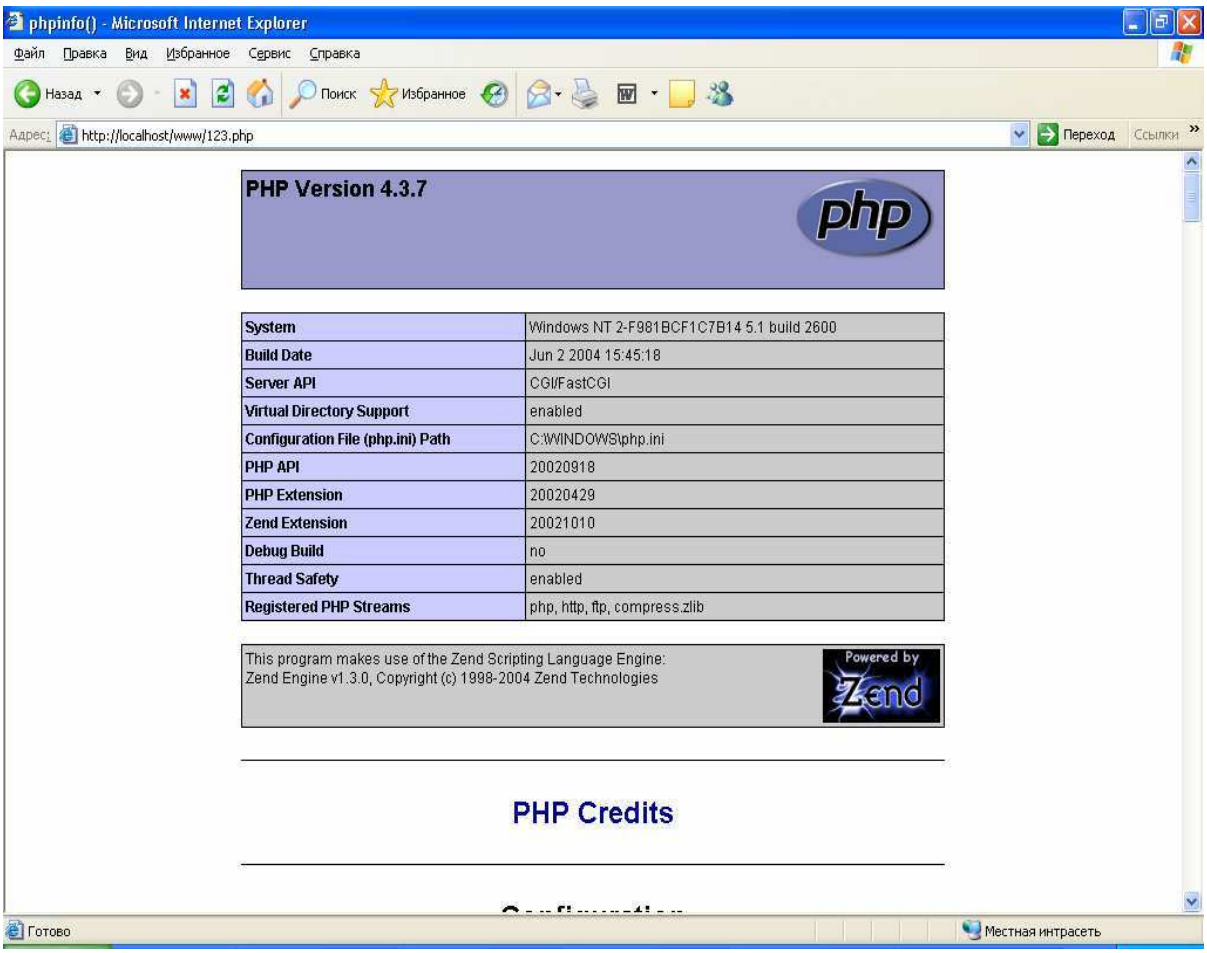
Apache серверінде бір компьютерде бірнеше виртуалды хосттар немесе жүздеген виртуалды серверлер болуы мүмкін. Сондықтан өзімізге сондай бір виртуалды сервер жасау, бірнеше сайт құру қажет болады. Ол үшін біз тағы да сол httpd.conf файлын ашамыз және файлдың соңына келесі қатарларды жазамыз:

```
<VirtualHost 127.0.0.1 >
ServerAdmin admin@server
rname.com
Server
erName www.servername.com
DirectoryIndex index.php3 index.phtml index.php
Index.htm Index.html Index.shtml Index.shtm
DocumentRoot “e:/usr/public_html/host/virtual”
ScriptAlias /cgi-bin/ “e:/usr/public_html/host/virtual/cgi-bin/”
ErrorLog e:/usr/php
ublic_html/host/virtual/logs/error.log
CustomLog e:/usr/public_html/host/virtual/logs/access.log common
</VirtualHost>
```

PHP скрипттері жұмыс істейтіндігін тексеру үшін алдымен e:/usr/public_html/host/virtual бумасына test.php файлын құрып, ішіне келесі кодты жазамыз:

```
<? phpinfo(); ?>
```

Браузердің адрестік қатарына http://127.0.0.1/test.php деп жазамыз. Егер экранда суретте көрсетілгендей PHP туралы ақпарат пайда болса, онда PHP процессоры жұмыс істеп тұр деген сөз (сурет 19).



Сурет 19 – PHP процессорын орнату

4.6 PHP-скрипттерге мысалдар

Web-сайттарда жиі қолданылатын PHP-скрипттерді қарастырайық.

1. «Санақ» скрипті тұтынушының сайтқа қанша рет кіргені туралы ақпаратты сақтау үшін қолданылады.

Негізгі Web-құжатта орналасатын PHP-скрипт коды:

```
<center><a href="info.php" target='_blank'><img src="counter.php"
width="88" height="31" border=0> </a></center>
```

Web-сайтқа кіргендер санын көрсететін PHP-скрипт коды:

```
<?
#error_reporting (E_ALL);
include "config_counter.php";
$ldate="0";
if (is_file("$data/last.dat")) {$lline=file("$data/last.dat"); $li=count($lline); if
($li>0) {$ldate=$lline[0];}}
$datescribe=$ldate+7*86400;
```

```
$today=mktime();
if ($today>$datescribe) {include("infomail.php");}
function addSpace($num) {$strlen=17-strlen($num); $space=null; while($strlen)
{$space.=" "; $strlen--;} return $space.$num;}
function read_file($path)
{if(!is_file($path))return false;
elseif(!filesize($path))return array();
elseif($array=file($path))return $array;
else while(!$array=file($path))sleep(1);
return $array;}
function normal_numeric($number)
{if(!isset($number))return false;
else{$strlen=strlen($number);
$new=null;
for ($i=$strlen-1;$i>-1;$i--)
{$n = $i;$n++;if(strstr($n/3,"."))$new.=$number[$strlen-1-$i];
else if($n!=$strlen)$new.=    " ".$number[$strlen-1-$i];
else$new.=$number[$strlen-1-$i];}
return $new;}}
if(!extension_loaded("gd")) {error("",date("Дата: d.m.Y. ya&#1179;ыт:
H:i:s",time())); exit;}
$ip=(isset($_SERVER['REMOTE_ADDR']))?$_SERVER['REMOTE_ADDR']:
0;
$a=$b=$c=null;
$today=date('d.m.Y',time());
$times=mktime();
if (isset($_SERVER["HTTP_REFERER"]))
{$from=$_SERVER["HTTP_REFERER"];} else {$from="";}
$who="";
if (isset($_SERVER["SCRIPT_NAME"]))
{$gde=$_SERVER["SCRIPT_NAME"];} else {$gde="";}
if (!is_file("$data/$today.dat")) {$OpenToday=fopen("$data/$today.dat","a");
fclose($OpenToday);}
$itogo=read_file("$data/all.dat"); if(!isset($itogo[0])) $itogo[0]=0;
$fp=fopen("$data/all.dat","w"); flock ($fp,LOCK_EX);
fwrite($fp,($a=$itogo[0]+1)); flock ($fp,LOCK_UN); fclose($fp);
@chmod("$data/all.dat", 0644);
$ft=fopen("$data/$today.dat","a"); flock ($ft,LOCK_EX);
fwrite($ft,"$ip|$times|$gde|$who|$from|\r\n"); flock ($ft,LOCK_UN);
fclose($ft); @chmod("$data/$today.dat", 0644);
$newlines=read_file("$data/$today.dat");
for ($i=0;$i < count($newlines); $i++) {$dt=explode("|", $newlines[$i]);
$lines[$i]=$dt[0];}
$b=count($lines); $c=count(array_unique($lines));
```

```
if(strlen($a)>9||!isset($a)) $a="?"; if(strlen($b)>9||!isset($b)) $b="?";
if(strlen($c)>9||!isset($c)) $c="?";
header("Expires: Mon, 01 Apr 2012 05:00:00 GMT");
header("Last-Modified: ".gmdate("D, d M Y H:i:s")." GMT");
header("Cache-Control: no-store, no-cache, must-revalidate");
header("Cache-Control: post-check=0, pre-check=0", false);
header("Pragma: no-cache");
header("Content-type: image/png".chr(10).chr(10));
$image=ImageCreateFromPNG("./images/".$image);
$color1=ImageColorAllocate($image,$s1r,$s1g,$s1b);
$color2=ImageColorAllocate($image,$s2r,$s2g,$s2b);
$color3=ImageColorAllocate($image,$s3r,$s3g,$s3b);
ImageString($image,1,0,2, addSpace(normal_numeric("$a")),$color1);
ImageString($image,1,0,12, addSpace(normal_numeric("$b")),$color2);
ImageString($image,1,0,21, addSpace(normal_numeric("$c")),$color3);
ImagePNG($image);
?>
```

Web-сайтқа кірген тұтынушылыр туралы жалпы ақпаратты монитор экранында диаграмма түрінде көрсететін PHP-скрипт

```
<?
#error_reporting (E_ALL);
include "config_counter.php";
function prcmp ($a, $b) {if ($a==$b) return 0; if ($a<$b) return -1; return 1;}
$months=array("&#1178;а&#1187;тар","А&#1179;пан","Наурыз","С&#1241;
у&#1110;р","Мамыр","Маусым","Ш&#1110;лде","Тамыз","&#1178;ырк&#1
199;йек","&#1178;азан","&#1178;араша","Желто&#1179;сан");
$deldt=mktime()-$days*86400;
$i=0; if ($handle = opendir($data)) {
while (($file = readdir($handle)) !== false)
if (!is_dir($file)) {$lines[$i]=$file; $i++;}
closedir($handle);
} else {print 'Бума бос!';}
$itogo=count($lines); $k=0; $text=null;
do {
$fline=file("$data/$lines[$k]");
$fitogo=count($fline);
if ($fitogo!=0 and $lines[$k]!="all.dat" and $lines[$k]!="last.dat" and
$lines[$k]!="mainbase.dat" and $lines[$k]!=".htaccess")
{
$thendayx=str_replace(".dat","",$lines[$k]);
$dt=explode(".",$thendayx);
@$then=mktime(0,0,0,$dt[1],$dt[0],$dt[2]);
$tekdate=date("d.m.Y",$then);
```

```
if ($deldt>$then) {unlink ("$data/$lines[$k]");}
usort($fline,"prcmp");
$numip="0"; $numsys="0"; $hi=0; $ab="0"; $ac="0";
do {
$dt=explode("|",$fline[$hi]);
if ($ab!=$dt[0]) {$ab=$dt[0]; $numip++;}
if (isset($dt[2])) {if ($ac!=$dt[2]) {$ac=$dt[2]; $numsys++;}}
$hi++;
} while ($hi<$fitogo);
$text.="$then|$fitogo|$numip|$numsys|\r\n";
}
unset($fitogo);
unset($fline);
$k++;
} while ($k<$itogo);
$fp=fopen("$data/mainbase.dat","w");
flock ($fp,LOCK_EX);
fputs($fp,"$text");
fflush ($fp);
flock ($fp,LOCK_UN);
fclose($fp);
@chmod("$data/mainbase.dat", 0644);
$lines=file("$data/mainbase.dat");
$maxi=count($lines);
usort($lines,"prcmp");
print"<HTML><head><META content='text/html; charset=windows-1251'
http-equiv=Content-Type></head>
<body text=#000000 leftMargin=0 topMargin=0 rightMargin=0
bottomMargin=0 marginheight=0 marginwidth=0><center>
<B>A&#1179;парат</B>
<table border=0 cellSpacing=0 cellPadding=0 width=\"98%\"><tr><td
valign=top>";
if (sizeof($lines)>=1) {
if ($gtype=="1") {
$xdaym="</tr><tr><td align=center><small>Дата</td><td> </td>\r\n ";
$graph1=""; $graph2="";
$g1shapka="<table border=0 cellSpacing=0 cellPadding=0 align=center><tr
align=center valign=bottom><td
valign=middle><small>X<BR>И<BR>T<BR>T<BR>E<BR>P<BR>-
<BR>C<BR>A<BR>H<BR>Ы</td><td><img src='images/v1scale.gif'
border=0></td>\r\n";
$g2shapka="<table border=0 cellSpacing=0 cellPadding=0 align=center><tr
align=center valign=bottom><td
valign=middle><small>X<BR>O<BR>C<BR>T<BR>T<BR>A<BR>P<BR>-
```

```
<BR>C<BR>A<BR>H<BR>Ы</td><td><img src='images/v2scale.gif'
border=0></td>\r\n";
for ($i=0; $i<$maxi; $i++)  {
$dtt=explode("|",$lines[$i]);
$dttn1=round($dtt[1]*$scale1);
$dttn2=round($dtt[2]*$scale2);
$graph1.="<td><table cellSpacing=0 cellPadding=0><tr><td width=24
align=center><small>$dtt[1]</small></td></tr><tr><td align=center><img
src='images/v1.gif' height=$dttn1 width=20></td></tr></table></td>\r\n";
$graph2.="<td><table cellSpacing=0 cellPadding=0><tr><td width=24
align=center><small>$dtt[2]</small></td></tr><tr><td align=center><img
src='images/v2.gif' height=$dttn2 width=20></td></tr></table></td>\r\n";
if (!isset($m0)) {$m0=date("m",$dtt[0])-1;}
if ($i==$maxi-1) {$m1=date("m",$dtt[0])-1;}
$xday=date("d",$dtt[0]);
$xdaym.="<td align=center><small>$xday</small></td>\r\n";
}
$mm1=$months[$m0]; $mm2=$months[$m1]; if ($mm1==$mm2) {$mm1="";}
else {$mm1.=" - ";}
$msdat="</tr></table><BR><center>Кезе&#1187;:   <B>$mm1
$mm2</B></center>\r\n";
print"$g1shapka $graph1 $xdaym </tr></table>";
print"$g2shapka $graph2 $xdaym $msdat <BR>";
}  else  {
$g1shapka="<table cellSpacing=0 cellPadding=0 align=center><tr align=center
valign=bottom><td valign=middle><small>Дата</td><td>ХИТТЕР-
САНЫ</td></tr><tr><td> </td><td><img src='images/g1scale.gif'
border=0></td>\r\n";
$g2shapka="<table cellSpacing=0 cellPadding=0 align=center><tr align=center
valign=bottom><td valign=middle><small>Дата</td><td>ХОСТТАР-
САНЫ</td></tr><tr><td> </td><td><img src='images/g2scale.gif'
border=0></td>\r\n";
$xdaym=""; $graph1=""; $graph2="";
for ($i=0; $i<$maxi; $i++)  {
$dtt=explode("|",$lines[$i]);
//$dtt[0]=date("d.m.y",$dtt[0]);
if (!isset($m0)) {$m0=date("m",$dtt[0])-1;}
if ($i==$maxi-1) {$m1=date("m",$dtt[0])-1;}
$xday=date("d",$dtt[0]);
$dttn1=round($dtt[1]*$scale1);
$dttn2=round($dtt[2]*$scale2);
$graph1.="<tr><td align=center><small>$xday</small></td><td><table
cellSpacing=0 cellPadding=0><tr><td><img src='images/g1.gif' height=16
width=$dttn1></td><td align=center> 
<small>$dtt[1]</small></td></tr></table></td></tr>";
```

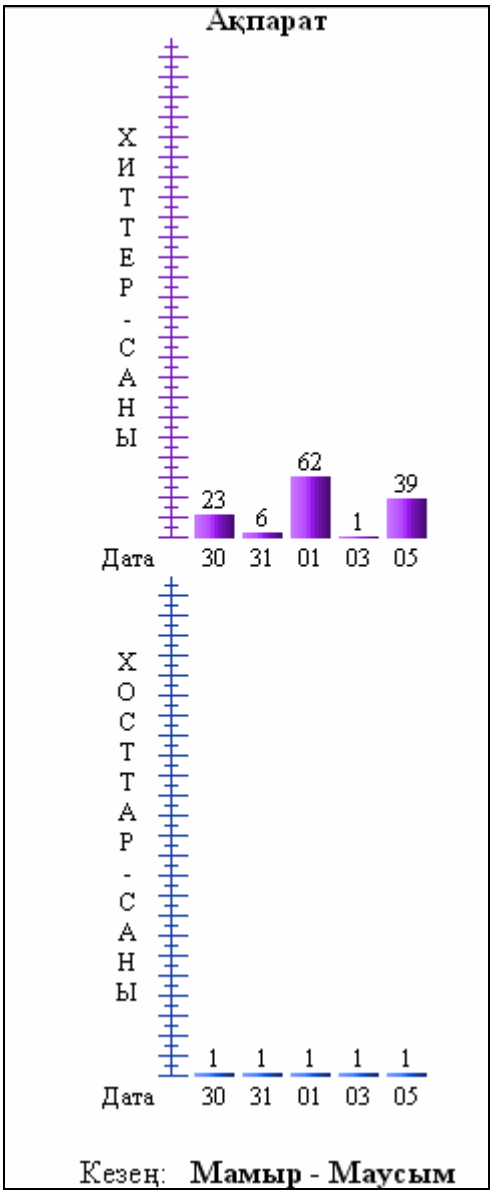
```
$graph2.="<tr><td align=center><small>$xday</small></td><td><table
cellSpacing=0 cellPadding=0><tr><td><img src='images/g2.gif' height=16
width=$dttn2></td><td align=center> 
<small>$dtt[2]</small></td></tr></table></td></tr>";
}
$mm1=$months[$m0]; $mm2=$months[$m1]; if ($mm1==$mm2) {$mm1="";}
else {$mm1.=" - ";}
$msdat="</tr><tr><td colspan=2><center><small>Кезе&#1187;:</small>
<B>$mm1 $mm2</B></center>\r\n";
print"<BR>$g1shapka $graph1 $xdaym </tr></table></td><td width='50%'
valign=top>";
print"<BR>$g2shapka $graph2 $xdaym </tr></table></td>$msdat<BR>";
} // else ($gtype)
print"</center>P.S. <B>Хиттер</B> - санауыш орнатыл&#1171;ан веб-
беттерге кіру саны;<BR>
       <B>Хосттар</B> - IP-адрестер
саны.<BR>";
}
else {print"<center><br><br>Бос<br> К&#1110;рулер орын
алмады<br><br>";}
?>
```

Скрипт нәтижесі келесі суретте көрсетілген (сурет 20).

2. Web-сайтта «Дауыс беру» PHP-скрипті. Негізгі беттегі PHP-скриптті іске қосу коды төмендегіше жазылады:

```
<script language=JavaScript><!--
function gosub()
{WRSub=window.open('vote.php','WRGolos','width=450,height=350,left=250,t
op=100'); WRSub.focus();}
function gorez()
{WRSub=window.open('rezult.php','WRRezultGolos','width=450,height=350,le
ft=250,top=100'); WRSub.focus();}//-->
</script>
<table cellPadding=3 align=center border=0><tbody><tr><td vAlign=top
align=middle>
<table bgcolor=navy cellSpacing=1 cellPadding=0 align=center
border=0><tr><td>
<table bgcolor=#ffffff  style="BACKGROUND-IMAGE:
url(images/fonlogo3.jpg); BACKGROUND-REPEAT: no-repeat"
cellSpacing=0 cellPadding=1 align=center border=0>
<form name=wrvote action='vote.php?id=1' method=post target='WRGolos'>
```

<tr><td colspan=3 align=middle bgColor=#FFFFFF><B> Веб-бет сізге ұнады ма? </B></td></tr>

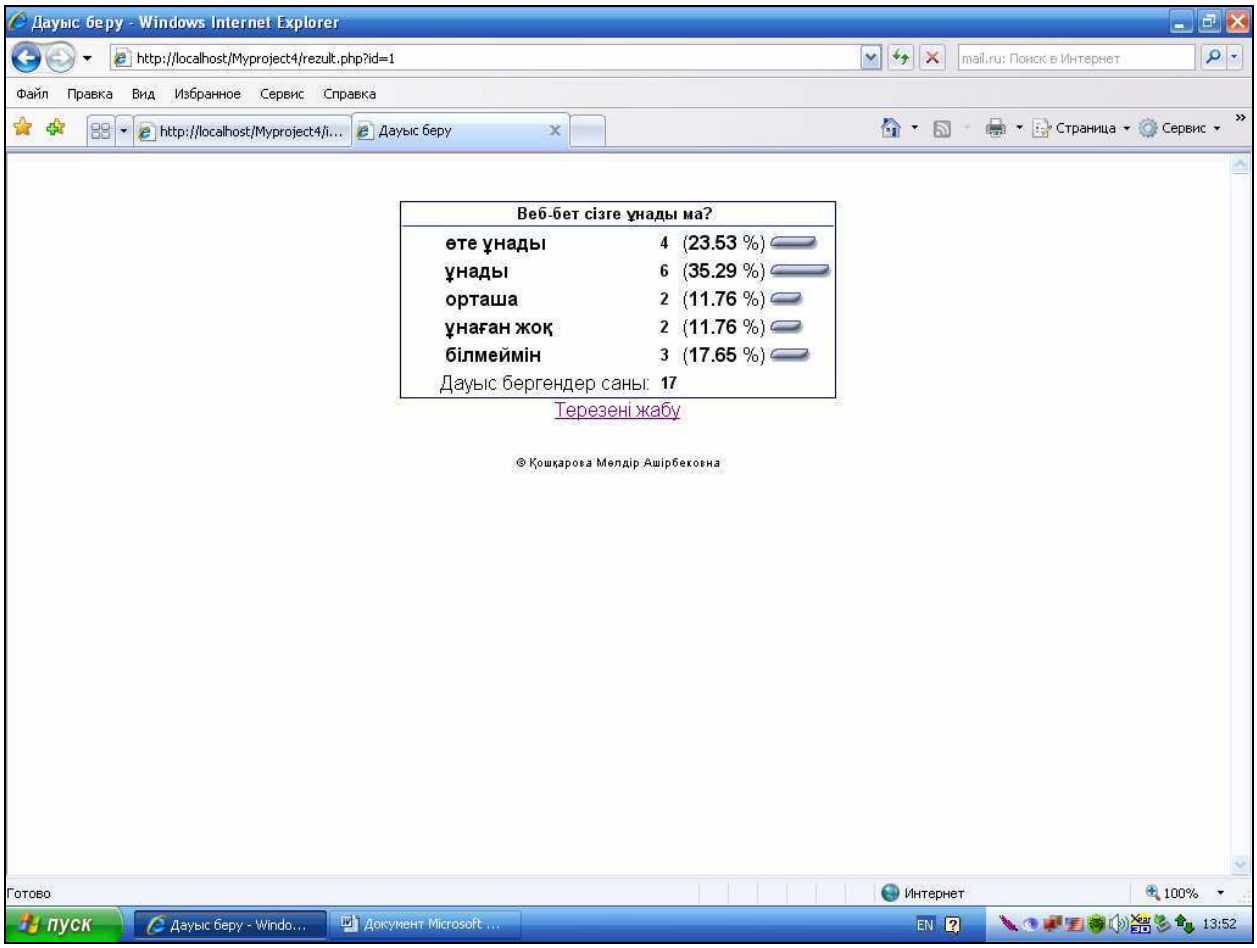


Сурет 20 – «Санақ» скриптінің нәтижесі

<tr><td colspan=3><img src='images/l1.gif' width=100% height=1></td></tr>
<tr><td><table border=0 cellSpacing=0 cellPadding=2 width=100%><tbody><tr>
<td width=15> </td><td><input name='votec' type=radio value='1'></td>
<td><B> өте ұнады</B></td>
</tr><tr>
<td width=15> </td><td><input name='votec' type=radio value='2'></td>
<td><B> ұнады</B></td>
</tr><tr>

```
<td width=15> </td><td><input name='votec' type=radio value='3'></td>
<td><font face=arial size=2><B> орташа</B></font></td>
</tr><tr>
<td width=15> </td><td><input name='votec' type=radio value='4'></td>
<td><font face=arial size=2><B> &#1201;на&#1171;ан
жо&#1179;</B></font></td>
</tr><tr>
<td width=15> </td><td><input name='votec' type=radio value='5'></td>
<td><font face=arial size=2><B> білмеймін</B></font></td>
</tr><tr><td colspan=3 align=center>
<input type=image alt='Дауыс беру' src='images/vote.gif'  onClick='gosub();'
border=0></td></tr>
</form></tbody></table>
</tbody></table>
</tbody></table>
<A href='rezult.php?id=1' onClick='gorez();'
target='WRRezultGolos'>H&#1241;тижелер</A>
</td></tr></table>
<!—Дауыс беру скриптінің соңы  -->
```

Скрипт нәтижесі келесі суретте көрсетілген (сурет 21).



Сурет 21 – «Дауыс беру» скриптінің нәтижесі

PHP-скрипттің негізгі коды:

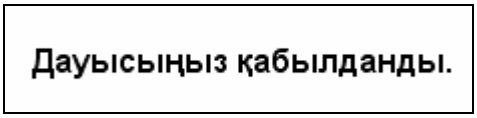
```
<?
error_reporting (E_ALL);
include "config.php";
$shapka="<html><head><title>Дауыс беру</title><META HTTP-
EQUIV='Pragma' CONTENT='no-cache'><META HTTP-EQUIV='Cache-
Control' CONTENT='no-cache'><META content='text/html; charset=windows-
1251' http-equiv=Content-Type></head><body>";
if (isset($_GET['id'])) {$id=$_GET['id']; if ((!ctype_digit($id)) or
(strlen($id)>2)) {exit("<B>Жауаптар саны 1-9 дейін!!!</B>");}} else {$id=1;}
$lines = file("$datadir/vote$id.dat");
$itogo=count($lines); $i=1; $glmax=0;
// Дауыстарды санау
do {$dt=explode("|",$lines[$i]); $glmax=$glmax+$dt[1]; $i++; }
while($i<$itogo); $i=1; $all=$glmax;
$vdt = explode("|",$lines[0]);
print"$shapka
<center><table cellPadding=3 align=center border=0><tbody><tr><td
vAlign=top align=middle>
<table border=0 bgcolor=navy cellSpacing=1 cellPadding=0
align=center><tr><td>
<table border=0 bgcolor=#ffffff  style=\"BACKGROUND-COLOR: white;
BACKGROUND-REPEAT: no-repeat\" cellSpacing=0 cellPadding=1
align=center border=0>
<form name=wrvote action='submit.php' method=post>
<tr><td colspan=3 align=middle bgColor=#FFFFFF><font face=arial
size=2><B> $vdt[0] </B></font></td></tr>
<tr><td colspan=3><img src='images/l1.gif' width=100% height=1></td></tr>
<tr><td><table border=0 cellSpacing=0 cellPadding=2 width=100%><tbody>";
do {$dt = explode("|",$lines[$i]);
if ($glmax==0) {$glmax=0.1;}
$glpercent=round(10000*$dt[1]/$glmax)/100;
$hcg=round($glpercent);
if ($glpercent<1) {$hcg=1;} if ($glpercent>100) {$hcg=100;}
print"<tr>
<td width=25> </td><td><B> $dt[0]</B></td>
<td><font face=arial size=2><B> $dt[1]</B></font></td>
<td>(<B>$glpercent</B> %)</td>
<td><img src='g1/lg.gif' border=0><img src='g1/cg.gif' width=$hcg height=11
border=0><img src='g1/rg.gif' border=0></td>
</tr>";
$i++;
} while($i<$itogo);
```

```
print"<tr>
<td> </td><td>Дауыс бергендер саны:</B></td>
<td><font face=arial size=2><B> $all</B></font></td>
<td> </td>
<td> </td>
</tr>
</form></tbody></table>
</tbody></table>
</tbody></table>
<a href='rezult.php' onClick='self.close()'>Терезені жабу</b></a>
</td></tr></table>";
?>
```

Дауыс берілгендігін, оның қабылданғандығын көрсететін скрипт. Нәтижеде, экранға



немесе



дауыс қабылданғаны туралы хабарламасы шығады.

```
<?
error_reporting (E_ALL);
include "config.php";
if (isset($_POST["votec"])) {$numv=$_POST["votec"];} else {print
"<BR><BR><BR><center> Сіз <B>  ешбір жауапты
та&#1187;дамады&#1187;ыз</B> </center>"; exit;}
$ip=(isset($_SERVER['REMOTE_ADDR']))?$_SERVER['REMOTE_ADDR']:
0;
if (isset($_GET['id'])) {$id=$_GET['id']; if ((!ctype_digit($id)) or
(strlen($id)>2)) {exit("<B>Жауаптар саны 1-9 дейін!!!</B>");}} else {$id=1;}
$lines = file("$datadir/vote$id.dat");
if (is_file("$datadir/ip$id.dat")) {
$iplines=file("$datadir/ip$id.dat"); $sizef=count($iplines);
if ($sizef > 1) { $itip=$sizef;
do {$itip--; $idt=explode("|",$iplines[$itip]);
if ($ip==$idt[0]) { $dayx=date("d.m.Y в H:i:s",$idt[1]); $stime=$idt[1];
$today=mktime();
}
$allredy="<B>Сіз дауыс берді&#1187;із  $dayx!</B></center>";}
```

```
 while ($itip>0); } }
if ($ipblok!="1") {$allredy=""; unset($allredy);}
if (!isset($allredy)) {$allredy="<B>Дауысы&#1187;ыз
&#1179;абылданды.</B>";
$mkdate=mktime();
$lines=file("$datadir/vote$id.dat");
$itogo=count($lines); $i=$itogo;
do {$i--; if ($numv==$i) {$vote=$i;}} while ($i>0);
$i=$itogo;
do {$i--; $dt=explode("|",$lines[$i]);
if ($vote==$i) {$dt[1]++;}
$lines[$i]="$dt[0]|$dt[1]|\r\n";
} while ($i>0);
$fp=fopen("$datadir/vote$id.dat","a+");
flock ($fp,LOCK_EX);
ftruncate ($fp,0);
for ($i=0; $i<$itogo; $i++) {fputs($fp,$lines[$i]);}
fflush ($fp);
flock ($fp,LOCK_UN);
@chmod("$fp", 0644);
$fp=fopen("$datadir/ip$id.dat","a+");
flock ($fp,LOCK_EX);
fputs($fp,"$ip|$mkdate|\r\n");
fflush ($fp);
flock ($fp,LOCK_UN);
@chmod("$fp", 0644);
}
print "<center><script language='Javascript'>function reload() {location =
'rezult.php?id=$id'}; setTimeout('reload()', 1500);</script>
<BR><BR><BR> $allredy";
?>
```

Жеке жұмысқа тапсырмалар

Тапсырма №1. Төменде берілген тақырыптарға сипаттама беріп, талдап, мысалдармен түсіндіріңіз. Тапсырманы орындауда график, кесте, диаграмма т.с.с. пайдалануға болады. Тапсырма презентация түрінде қабылданады (вариант бойынша):

1. Интернет компоненттері;
2. Интернет қызметтері (WWW, FTP, Usenet, Telnet, E-mail);
3. Интернет хаттамалары (TCP/IP, HTTP, FTP);
4. WWW қызметі;
5. Домендік адрестер;
6. Іздеу серверлері;
7. Интернетке қосылу әдістері (Қазақстан провайдерлері);
8. HTML тілі тэгтері;
9. ASCII-редакторлар;
10. WYSIWYG-редакторлар;
11. Web–браузерлер;
12. IP-адрес және URL-адрес;
13. Web-құжат құрылымы;
14. RGB (red, green, blue) моделі;
15. CSS –стильдер кестесі.

Тапсырма №2. HTML тілін пайдаланып өзіңізге резюме жазыңыз (Блокнот редакторын пайдалану қажет). Тапсырманы орындаған кезде әртүрлі шрифттерді қолдана отырып, олардың түсін, өлшемін және шрифт түрін өзгертіңіз, фотосуретті пайдаланыңыз.

Резюме төмендегі бөлімдерден тұрады:

- тегі;
- аты;
- әкесінің аты;
- туылған жері;
- ұлты;
- білімі;
- кәсіби біліктілігі;
- білетін тілдері;
- мекен-жай;
- байланысу телефондары.

Тапсырма №3. HTML тілін пайдаланып браузер экранына келесі тізім шығатын Web-құжат құрыңыз (вариант бойынша):

1. Топ студенттерінің тізімі;
2. Қалааралық телефон кодтарының тізімі;
3. Оңтүстік Қазақстан облысындағы аудандар тізімі;
4. Оқылатын пәндер тізімі;

5. Қазақстан университеттерінің тізімі;
6. Курстық жұмысты жазу үшін қолданылған әдебиеттер тізімі;
7. Компьютерлік бағдарламалар тізімі;
8. Браузерлер тізімі;
9. Антивирустар тізімі;
10. Қазақстан қалаларының тізімі;
11. Бағдарламалау тілдерінің тізімі;
12. Футбол бойынша Қазақстан командаларының тізімі;
13. Көп қолданылатын сайттар тізімі;
14. Ұялы телефон шығаратын корпорациялар тізімі;
15. Іздеу серверлерінің тізімі.

Тапсырма №4. HTML тілін пайдаланып кесте орналасқан Web-құжат құрыңыз. Тапсырманы орындаған кезде әртүрлі шрифттерді қолдана отырып, олардың түсін, өлшемін және шрифт түрін өзгертіңіз, суреттерден пайдаланыңыз. Кестенің ішіне берілген тақырып бойынша деректер шығатын болсын (вариант бойынша):

1. Сабақ кестесі;
2. Футбол чемпионаты ойындарының кестесі;
3. Шахмат турнирі ойындарының кестесі;
4. Информатика пәні оқытушысының сабақ кестесі;
5. Компьютерлік сыныпта өтетін сабақ кестесі;
6. Қала маңындағы жүретін автобустар қозғалысының кестесі;
7. Қазақстан бойынша ұшақтар қозғалысының кестесі;
8. Темір жол вокзалынан жүретін пойыздар қозғалысының кестесі;
9. Календарь;
10. Компьютер дүкенінің прайс-листі;
11. Университет оқу үрдісінің кестесі;
12. Мекеменің жұмыс кестесі;
13. Студенттердің рейтинг кестесі;
14. Сессия нәтижелері кестесі;
15. Студент анкетасы.

Тапсырма №5. HTML тілін пайдаланып браузер экранына берілген тақырып бойынша жарнама мәтіні және баннер шығаратын Web-құжат құрыңыз (вариант бойынша):

1. Жоғары оқу орнына түсуге шақыратын жарнама (мамандықтар тізімі, оқу бөлімдері – күндізгі және сырттай оқу, жеңілдіктер, қажетті құжаттар, қабылдау мерзімі, мекен-жайы, байланыс телефоны және т.б.);
2. Атақты әншінің концертіне шақыратын жарнама (концерт өтетін уақыты мен орны, билет бағасы, өлеңдер тізімі мен орындаушының фотосуреті және т.б.);
3. Компьютер сататын дүкеннен тауар сатып алуға шақыратын жарнама (дүкенде сатылатын компьютерлер, олардың

модельдері, техникалық сипаттамасы, түрлері, бағалары, шығарған фирманың аты, мекен-жайы, байланыс телефоны және т.б.);

4. Кинотеатрға кино көруге шақыратын жарнама (көрсетілетін кинофильмдер тізімі, сеанстар уақыты, билет бағасы, танымал киноактерлардың фотосуреті және т.б.);
5. Театрға спектакль көруге шақыратын жарнама (спектакльдер тізімі, уақыты мен орны, билет бағасы, актерлар фотосуреті және т.б.);
6. Супермаркеттен тауар сатып алуға шақыратын жарнама (тауарлар тізімі, бағалары, жаңа тауарлар, жеңілдіктер мерзімі, тауар шығаратын атақты фирмалар, супермаркет бөлімдері және т.б.);
7. Спорт мектебіне шақыратын жарнама (спорттық секциялар тізімі, тренердің аты-жөні, спорт мектебінің атақты түлектері, мекен-жайы, байланыс телефоны және т.б.);
8. Спорт жарысына шақыратын жарнама (мысалы – футбол, қатысатын командалар тізімі, жарыстың өтетін уақыты мен орны, демеушілер тізімі, атақты ойыншылардың фотосуреті және т.б.);
9. Ұялы телефон сататын дүкеннен тауар сатып алуға шақыратын жарнама (дүкенде сатылатын телефон түрлері, олардың бағалары, модельдері, шығарған фирманың аты, мекен-жайы, байланыс телефоны және т.б.);
10. Ән байқауына қатысуға шақыратын жарнама (байқауға қатысатын атақты әншілердің аты-жәні, жюри мүшелері, алдыңғы байқау жеңімпаздары, байқаудың өтетін уақыты мен орны, демеушілер тізімі, байланыс телефоны және т.б.);
11. Айтысқа шақыратын жарнама (қатысатын ақындар тізімі, жеңімпаздарға арналған сыйлықтар, уақыты мен орны, билет бағасы, демеушілер тізімі және т.б.);
12. Әсемдік салонына шақыратын жарнама (көрсетілетін қызметтер тізімі, бағалары, мекен-жайы, байланыс телефоны және т.б.);
13. Әйелдер киімін сатып алуға шақыратын жарнама (тауарлар тізімі, бағалары, жеңілдіктер мерзімі, тауар шығаратын атақты фирмалар, дүкен бөлімдері, мекен-жайы, байланыс телефоны және т.б.);
14. Құрылыс материалдарын сатып алуға шақыратын жарнама (тауарлар тізімі, бағалары, тауар шығаратын атақты фирмалар, дүкен бөлімдері, мекен-жайы, байланыс телефоны және т.б.);
15. Банк қызметіне арналған жарнама (қызметтер тізімі, кредит және депозит түрлері, пайызы, жеңілдіктер мерзімі, банк филиалдары, мекен-жайы, байланыс телефоны және т.б.).

Тапсырма №6. HTML тілін пайдаланып берілген тақырып бойынша Web-сайт құрыңыз. Тапсырманы орындаған кезде түрлі-түсті шрифттерден, қалыптардан, кестелерден, фреймдерден, баннерлерден, суреттерден пайдаланыңыз (вариант бойынша):

1. АҚ «Қазақтелеком»;
2. АҚ «Шымкент шина»
3. АҚ «Су ресурстары - Маркетинг»;
4. АҚ «Қазпошта»;
5. «Астана моторс» автосалоны;
6. ЖШС «Шымкент жарық»;
7. «Рамстор» дүкендер желісі;
8. «Қала әкімшілігі»;
9. «Интернет-дүкен»;
10. №1 облыстық дарынды балалар лицейі;
11. №69 экономика және құқық лицейі;
12. «Информатика» кафедрасы;
13. «Жиһаз дүкені»;
14. «Әсемдік салоны»;
15. «Рахат» кондитер фабрикасы;
16. Еркін тақырып (студент сұранысы бойынша).

Тапсырма №7. Құрылған Web-сайтқа арнап, Блокнот редакторында my.css каскадты стильдер кестесі файлын құрыңыз. Құрылған my.css файлын гипермәтіндік құжатқа енгізіңіз (вариант бойынша).

Тапсырма №8. Құрылған Web-сайтқа объекттер (Java-скрипттер, клиптер, апплеттер) енгізіп, сайтты безендіріңіз (вариант бойынша).

Тапсырма №9. Apache жүйесін және PHP процессорын орнатыңыз (Denwer немесе Joomla жүйелерін де қолдануға болады). Құрылған Web-сайтқа PHP-скриптті енгізіп, браузерде жұмысын қадағалаңыз (вариант бойынша).

Тест сұрақтары

1. Web-құжатының файл кеңеймесі (extension) қандай?
   a) .ppt
   b) .xls
   c) .bmp
   d) .doc
   e) .html

2. TCP/IP деген не?
   a) деректер алмасуға арналған стандарт хаттама
   b) техникалық, программалық және информациялық компонент
   c) физикалық адрестерге сәйкес домендік адрестер жүйесі
   d) гипермәтіндік белгілеу тілі
   e) Internet ресурстарын қарау браузері

3. Internet қызметтері:
   a) www, ftp, telnet, E-mail
   b) us, ru, kz, uk
   c) http, ftp, imap, pop3
   d) com, edu, gov, net
   e) Internet Explorer, Netscape Navigator

4. URL-адресті көрсет:
   a) www.ukgu.kz
   b) 192.168.32.1.
   c) nurlan@reiting.ball.kos
   d) http://www.
   e) esbol@agai.kele.jatir

5. IP-адресті көрсет:
   a) 192.168.32.1.
   b) www.ukgu.kz
   c) nurlan@ bes.koyip.ber
   d) http://www.
   e) esbol@shpargalkani.jasir

6. Аймақтық домен аттары:
   a) us, ru, kz
   b) com, edu, gov
   c) http, ftp, imap
   d) www, ftp, telnet
   e) E-mail, Usenet, ISQ

7. Қызмет бабына негізделген домен аттары:
   a) com, edu, gov, net
   b) us, ru, kz, uk
   c) http, ftp, imap, pop3
   d) www, ftp, telnet, E-mail
   e) Internet Explorer, Netscape Navigator

8. Іздеу серверлері:
   a) www.rambler.ru www.yandex.ru
   b) www.referat.ru www.bazar.net
   c) www.nursat.kz www.telecom.kz
   d) www.mail.ru www.ok.kz
   e) www.khabar.kz www.president.kz

9. Браузерлер:
   a) Internet Explorer, Opera, Mozilla FireFox
   b) us, ru, kz, uk
   c) http, ftp, imap, pop3
   d) www, ftp, telnet, E-mail
   e) com, edu, gov, net

10. Электрондық пошта адресі:
   a) nurlan@mail.ru
   b) www. kazatk.kz
   c) 192.168.32.1.
   d) http://www.
   e) esbol@.jayabi.kaisi

11. Графикалық файлдың форматын көрсет:
   a) jpeg, gif, bmp
   b) doc, rtf, dot
   c) rar, zip, arj
   d) exe, com, bat
   e) xls, txt, tmp

12. Пиксел деген не?
   a) графикалық режимдегі экрандағы ең кіші элемент
   b) кішкене төртбұрышты элемент
   c) растрлық элемент
   d) графикалық режимдегі нүкте
   e) графикалық режимдегі сәуле

13. Растр деген не?
a) пикселдер матрицасы
b) монитордағы жарық шығарушы нүктелер
c) бейненің форматы
d) суреттің форматы
e) бейне сапасы

14. Интернет желісінде электрондық поштаның адресі берілген: user_name@mail.ru Тұтынушының логині қандай?
a) user_name
b) ru
c) @user_name
d) mail.ru
e) user_name@ mail.ru

15. HTML тегі деген не?
a) HTML тілі нұсқаулары
b) HTML -құжат беттері
c) HTML -гипермәтінді белгілеу тілі
d) HTML -құжаттар іздеу сервері
e) HTML -деректер алмасуға арналған стандарт хаттама

16. Қандай атрибут түсті анықтайды?
a) color
b) size
c) width
d) height
e) title

17. <html>…</html> тэгі не үшін қолданылады?
a) HTML-құжатты ашып жабу үшін
b) ені бойынша түзету
c) тақырыпты бейнелеу үшін
d) абзацты бейнелеу үшін
e) шрифттармен жұмыс істеу үшін

18.
 тэгі нені білдіреді?
a) жаңа жолға өту
b) абзацтың бейнеленуі
c) тақырыптың бейнеленуі
d) шрифт өлшемі
e) көлденең түзу сызықты салады

19. Web-құжатқа сурет қандай тэг арқылы қойылады?
a) <img>
b) <html>
c) <body>
d) <head>
e)

20. Web-құжатқа кесте қандай тэг арқылы қойылады?
a) <table>
b) <head>
c) <title>
d) <body>
e) <img>

21. … - тэгтері нені анықтайды?
a) абзац
b) сол жақтан бос орын қалдыру
c) құжат тақырыбыі
d) тэг атрибуты
e) шрифт өлшемі

22. JavaScript-те (!=) амалы нені білдіреді?
a) тең емес
b) кіші
c) үлкен
d) қатаң түрде тең
e) false

23.PHP-скрипт нәтижесін көрсетіңіз

```
<?
Function get sum ($var)
{
$var=$var+5;
return $var;
}
$new var =20;
echo(get sum($new var));
?>
```

a) 25
b) 20
c) 0
d) 5
e) false

24. <h1 align = center>…</h1>. Бұл жазу нені білідіреді?
a) центрге түзету
b) еніне түзету
c) оң жаққа түзету
d) сол жаққа түзету
e) жалпы түзету

25. <body>…</body> тэгі не үшін қолданылады?
a) құжат денесінің басы мен аяғын анықтайды
b) ені бойынша түзету үшін
c) абзацтарды сипаттау үшін
d) суреттерді орналастыру үшін
e) құжат тақырыбын орналастыыру үшін

26. <hi>…</hi> тегі не үшін қолданылады?
a) тақырыпты бейнелеу үшін
b) абзацті бейнелеу үшін
c) шрифттармен жұмыс істеу үшін
d) суреттерді орналастыру үшін
e) ені бойынша түзету

27. Бас шрифтті анықтау үшін қандай тэг қолданылады?
a) <basefont>
b) <html>
c) <body>
d) <head>
e) <title>

28. Түсті анықтау қандай символдан басталады?
a) #
b) ?
c) *
d) $
e) @

29. bgcolor атрибуты нені береді?
a) фон түсін анықтау
b) шрифтті орнату
c) абзацты анықтау
d) тақырыпты бейнелеу
e) графикалық объекттерді орналастыру

30. Гиперсілтеме қандай тэг арқылы қойылады?
a) <a>
b) <html>
c) <body>
d) <head>
e) <img>

31. RGB моделіне 0,255,0 параметрлері қойылса, экранда қандай түс пайда болады?
a) жасыл
b) қызыл
c) ақ
d) көк
e) сары

32. RGB моделінде негізгі түстер ретінде:
a) қызыл, жасыл, көк
b) көгілдір, күлгін, сары
c) қызыл, көгілдір, сары
d) күлгін, сары, қара
e) қызыл, сары, қара

33. Модемнің ең маңызды параметрі:
a) жылдамдық
b) өнімділігік
c) биттер саны
d) такт жиілігі
e) разрядтылық

34. Модем жылдамдығының өлшем бірлігі:
a) бит/сек
b) файл/сек
c) символ/сек
d) МГц
e) пиксел/дйюм

35. Web-құжатты көруге арналған бағдарламаның жалпы атауы:
a) Browser
b) Usenet
c) E-mail
d) WWW
e) FTP

Пайдаланылған әдебиеттер

1. Снелл Нед. Интернет. Пер. с англ. –М.: «Вильямс», 2006. –384с.
2. Комягин В.Б., Печников В.Н. Создание Web-страниц и Web-сайтов. Учебное пособие. –М.: Триумф, 2003. –496 с.:ил.
3. Ибрагимов У.М., Абдиева З.А. HTML & CSS. Электронный учебник / Свидетельство о гос. регистрации объекта интеллект. собст. №288 от 14.04.2009
4. Ибрагимов У.М., Изтаев Ж.Д. Web-программалау: HTML & CSS технологиясы. 050602 –«Информатика» мамандығы күндізгі және сыртқы бөлім студенттеріне арналған әдістемелік нұсқау. –Шымкент: ОҚМУ, 2009. –58 б.
5. http://javascript.internet.com/
6. Гарнеев А.Ю. Web–программирование на Java и JavaScript. –СПб.: БХВ–Петербург, 2002. –1040 с.
7. Макинтош Дж. Java & JavaScript: библиотека программиста.–Спб.: Питер, 2003. –208 с.
8. Ибрагимов У.М., Оразов И. Web-дизайн: HTML+JavaScript // Свидетельство о гос. регистрации объекта в Комитете по правам интеллект. собст. МЮ РК. №679 от 29.04.2011.
9. Мазуркевич А. PHP: настольная книга программиста. –М.: Новое знание, 2004. –479 с.
10. Макинтош Дж. Perl & PHP: библиотека программиста. –Спб.: Питер, 2004. –218 с.
11. Орлов А.А. PHP: Полезные приемы. –2-е изд. перераб. и доп. –М.: Горячая линия – Телеком, 2006. –272 с.: ил.
12. http://www.evrsoft.com
13. http://phpedit.com.ua
14. http://delphiworld. narod.ru/wds.html
15. http://www.apache.org/dyn/closer.egi
16. http:// www.apache.rin.ru/dist/httpd/binaries/win32/
17. http://www.php.het/downloads.php
18. http://www.localhost/index.html

Қосымшалар

Қосымша 1 – Аймақтық домен аттары

AF Afghanistan	GR Greece	PE Peru
AL Albania	GL Greenland	PH Philippines
DZ Algeria	GD Grenada	PN Pitcairn
AS American Samoa	GP Guadeloupe	PL Poland
AD Andorra	GU Guam	PT Portugal
AO Angola	GT Guatemala	PR Puerto Rico
AI Anguilla	GN Guinea	QA Qatar
AQ Antarctica	GW Guinea-Bissau	RE Re'union
AG Antigua and Barbuda	GY Guyana	RO Romania
AR Argentina	HT Haiti	RU Russian Federation
AM Armenia	HM Heard and McDonald Islands	RW Rwanda
AW Aruba	HN Honduras	SH Saint Helena
AU Australia	HK Hong Kong	KN Saint Kitts and Nevis
AT Austria	HU Hungary	LC Saint Lucia
AZ Azerbaijan	IS Iceland	PM Saint Pierre and Miquelon
BS Bahamas	IN India	VC Saint Vincent and the Grenadines
BH Bahrain	ID Indonesia	WS Samoa
BD Bangladesh	IR Iran	SM San Marino
BB Barbados	IQ Iraq	ST Sao Tome and Principe
BY Belarus	IE Ireland	SA Saudi Arabia
BE Belgium	IL Israel	SN Senegal
BZ Belize	IT Italy	SC Seychelles
BJ Benin	JM Jamaica	SL Sierra Leone
BM Bermuda	JP Japan	SG Singapore
BT Bhutan	JO Jordan	SK Slovakia
BO Bolivia	KZ Kazakhstan	SI Slovenia
BA Bosnia-Hercegovina	KE Kenya	SB Solomon Islands
BW Botswana	KI Kiribati	SO Somalia
BV Bouvet Island	KP Korea	ZA South Africa
BR Brazil	KR Korea (Republic of)	SU Former Soviet Union
IO British Indian Ocean Territory	KW Kuwait	ES Spain
BN Brunei Darussalam	KG Kyrgyzstan	LK Sri Lanka
BG Bulgaria	LA Lao People's	SD Sudan

	Democratic Republic	
BF Burkina Faso	LV Latvia	SR Suriname
BI Burundi	LB Lebanon	SJ Svalbard and Jan Mayen Islands
KH Cambodia	LS Lesotho	SZ Swaziland
CM Cameroon	LR Liberia	SE Sweden
CA Canada	LY Libyan Arab Jamahiriya	CH Switzerland
CV Cape Verde	LI Liechtenstein	SY Syria
KY Cayman Islands	LT Lithuania	TW Taiwan, Province of China
CF Central African Republic	LU Luxembourg	TJ Tajikistan
TD Chad	MO Macau	TZ Tanzania
CL Chile	?? Macedonia	TH Thailand
CN China	MG Madagascar	TG Togo
CX Christmas Island	MW Malawi	TK Tokelau
CC Cocos(Keeling) Islands	MY Malasia	TO Tonga
CO Colombia	MV Maldives	TT Trinidad and Tobago
KM Comoros	ML Mali	TN Tunisia
CG Congo	MT Malta	TR Turkey
CK Cook Islands	MH Marshall Islands	TM Turkmenistan
CR Costa Rica	MQ Martinique	TC Turks and Caicos Islands
CI Cote d'Ivoire	MR Mauritania	TV Tuvalu
HR Croatia	MU Mauritius	UG Uganda
CU Cuba	MX Mexico	UA Ukraine
CY Cyprus	FM Micronesia	AE United Arab Emirates
CZ Czech Republic	MD Moldova	UK United Kingdom
DK Denmark	MC Monaco	US United States
DJ Djibouti	MN Mongolia	UM United States Minor Outlying Islands
DM Dominica	MS Montserrat	UY Uruguay
DO Dominican Republic	MA Morocco	UZ Uzbekistan
TP East Timor	MZ Mozambique	VU Vanuatu
EC Ecuador	MM Myanmar	VA Vatican City State
EG Egypt	NA Namibia	VE Venezuela
SV El Salvador	NR Nauru	VN Vietnam
GQ Equatorial Guinea	NP Nepal	WF Wallis & Futuna Islands
EE Estonia	NL Netherlands	EH Western Sahara

ET Ethiopia	AN Netherlands Antilles	YE Yemen
FK Falkland Islands	NT Neutral Zone (between Saudi Arabia & Iraq	YU Yugoslavia
FO Faroe Islands	NC New Caledonia	ZR Zaire
FJ Fiji	NZ New Zealand	ZM Zambia
FI Finland	NI Nicaragua	ZW Zimbabwe
FR France	NE Niger	
GF French Guiana	NG Nigeria	
PF French Polynesia	NU Niue	
TF French Southern Territories	NF Norfolk Island	
GA Gabon	MP Northern Mariana Islands	
GM Gambia	NO Norway	
GE Georgia	OM Oman	
DE Germany	PK Pakistan	
VG Virgin Islands (British)	PW Palau	
VI Virgin Islands (U.S.)	PA Panama	
GH Ghana	PG Papua New Guinea	
GI Gibraltar	PY Paraguay	

Қосымша 2 – Қазақша әріп кодтары

Бас әріп	код	Кіші әріп	код
Ә	Ә	ә	ә
Ө	Ө	ө	ө
Ң	Ң	ң	ң
Қ	Қ	қ	қ
Ғ	Ғ	ғ	ғ
Ү	Ү	ү	ү
Ұ	Ұ	ұ	ұ

Ибрагимов Оспанәлі Мусақулұлы

WEB-БАҒДАРЛАМАЛАУҒА КІРІСПЕ

Оқу құралы

Басуға _____ қол қойылды
Пішімі $X^{x}Y$ 1/16
Типографиялық қағаз. Офсеттік баспа. Көлемі _8_ б.т.
Таралымы _____ дана. Тапсырыс _____

© М.Әуезов атындағы Оңтүстік Қазақстан мемлекеттік университетінің баспаханасында шығарылған

М.Әуезов атындағы ОҚМУ баспахана орталығы, Шымкент, Тәуке хан даңғылы, 5