

ҚАЗАҚСТАН РЕСПУБЛИКАСЫ БІЛІМ ЖӘНЕ ҒЫЛЫМ МИНИСТРЛІГІ  
Ш. ЕСЕНОВ АТЫНДАҒЫ КАСПИЙ МЕМЛЕКЕТТІК ТЕХНОЛОГИЯЛАР  
ЖӘНЕ ИНЖИНИРИНГ УНИВЕРСИТЕТІ

**Ы.А. ШАХИМОВА**

**Интернет -технологиялары**

(Оқу құралы 040705-Есептеуіш техникасы және бағдарламалық  
қамтамасыз ету)

Ақтау, 2011

ӘОЖ 004.04 (075.8)  
ББК 32.973я73  
И 68

Пікір жазғандар: т.ғ.д. Ускенбаева Р.К.  
т.ғ.д. Туркпенбаева Б.Ж.  
т.ғ.д. Сәуірбаев Б.С.

Интернет-технологиялары: оқу құралы /Құраст.- аға оқытушы Ы.А. Шахимова.  
– Ақтау:И68 2011. – 111 бет.

ISBN 978-601-7276-86-7

Оқу құралында «Интернет-технологиялары» пәнінің негізгі түсініктері, Интернеттің құрылымы ұйымдастырылуы және жұмыс жасау принциптерінің технологиясы және Интернет желісінде жұмыс жасауға арналған қосымшаларды жобалау қарастырылған.

Оқу құралы ЖОО-ң «Есептеуіш техникасы және бағдарламалық қамтамасыз ету» мамандығына оқытылатын «Интернет-технологиялары» курсының типтік бағдарламасы мен мемлекеттік стандартқа сәйкес келеді.

ӘОЖ 004.04 (075.8)  
ББК 32.973я73

Ш.Есенов атындағы Каспий мемлекеттік технологиялар және инжиниринг университетінің оқу-әдістемелік кеңесінің шешімі бойынша баспаға ұсынылған.

© Ш.Есенов атындағы КМТЖИУ 2011

ISBN 978-601-7276-86-7

## Интернет технологияларына кіріспе

Интернет, біртұтас басқару орталықтары жоқ, бірақ бірыңғай ережелер бойынша жұмыс істейтін және өз қолданушыларына біртұтас қызметтер жиының көрсететін әлемдегі үлкен желі. Интернетті әрбірінің қызметтерін тәуелсіз оператор-жабдықтаушысы (ISP, Internet Service Provider) жүргізетін "желілір желісі" ретінде қарастыруға болады.

Интернет қолданушылар тұрғысынан, әр түрлі желілер бойынша: ISP - желі, корпоративтік желі, үй қолданушыларының жеке компьютерлері және желін қоса бытыраңқы ақпараттық қорлардың жиыны болып табылады. Әрбір жеке компьютер (host ағылшын терминінен) осы желіде хост деп аталады.

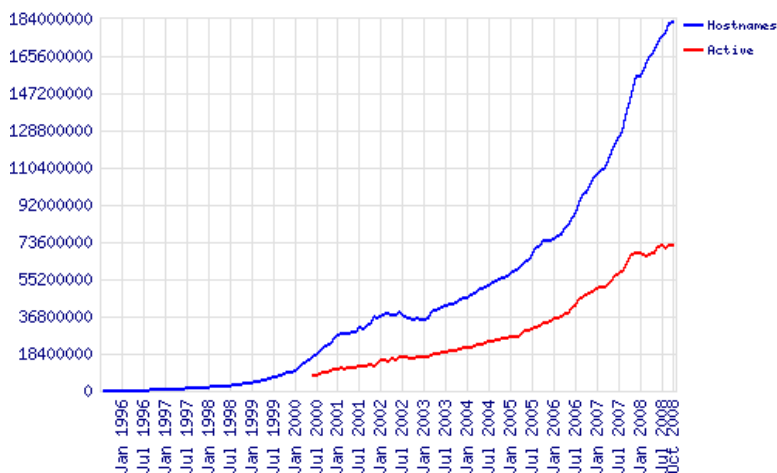
Бүгінгі интернет (кесте 1.1) өзінің пайда болуына сол кезде жаңа пакеттердің коммутацияның технологиясына сыпайы тәжірибе ретінде басталған біріккен ARPANET желісіне міндетті. ARPANET желісі 1969 жылы айқара ашылып басында төрт түйінді пакеттердің коммутациясы бар жинағынан тұратын болды. Алғашқы түйіндер жинағы жалғастырған байланыс торабтары 50 Кбит/с жылдамдықтарында жұмыс істеді. ARPANET желісі АҚШ қорғаныс министрлігінің ARPA (Advanced Research Projects Agency ) ғылыми-зерттеу жұмыстарын болашақты жоспарлауды басқарумен қаржыландырды және технологияның зерттеуі және кооперативтік таралған есептеулер үшін қолдануға болатын пакеттердің коммутациясының хаттамалары үшін арнады.

Кесте 1.1. (1966 жылдан 2000 жылға дейін) интернеттің дамытуын хронологиясы

ЖЫЛ	Оқиға
1966	RPАның басқаруының пакеттерінің коммутациясымен тәжірибе
1969	ARPANETтың желісінің алғашқы іскер түйіндері
1972	Таралған электрондық почтаның өнертабысы
1973	ARPANETтың желісіне АҚШтың сыртына қосылған алғашқы компьютерлер
1975	ARPANET желі АҚШ қорғаныс министрлігінің байланысының басқаруын жүргізуге тапсырылған
1980	TCP/IPмен тәжірибелерді бастайды
1981	Әрбір 20 күндердің желілеріне жаңа хост жамалады
1983	TCP/IPге өткел бітірген
1986	NSFnet-шы магистраль жасалған
1990	ARPANET желі болу тоқтатты
1991	Gopher пайда болу
1991	Бүкіләлемдік өрмекшінің торының өнертабысы. РGPның жүйесі босатқан. Mosaic пайда болу

1995	Интернеттің магистральсінің жекешелендіруі
1996	(155 Мбит/с) БЖ-ші магистраль құрастырылған
1998	Тіркеулі домен аттарының саны 2 миллион шектен шықты.
2000	Индекстелетін веб-беттерді саны 1 миллиард шектен шықты.

1.1 суретте хосттардың санның өсуін динамикасын көрсететін график кескінделген



**Сурет 1.1.** Хосттардың санның өсуін динамикасы

Интернетте біртұтас техникалық саясаттың формасы, техникалық стандарттардың үйлесімді жиыны, интернетке кіретін компьютерлер және желілердің аттары мен мекенжайларын тағайындауы орталықтанудың жанама түрдегі формалары айқындала алады.

Демек Интернет кемшіліктері де өз артықшылығы да бар децентрализациялаңған желі болып табылады.

#### 1. Артықшылығы:

- екі ISP аралық келісім жасасу жолымен интернеттің ұзартуын жеңілдету.

#### 2. Кемшілігі:

- Қызметтердің барлық жабдықтаушыларын күштерінің үйлесімділігі керек болатындығынан, интернеттің қызметтерінің және технологияларының жаңғыртуының күрделілігі.
- Интернеттің қызметтерінің аласалау сенімділігі.
- Желінің жеке сегменттерінің жұмысқа қабілеттілігіне жауапкершілік интернеттің қызметтерінің жабдықтаушыларына тапсырылады.

Интернеттің қызметтерінің жабдықтаушыларының әр түрлі түрлері бар болады:

- интернеттің қызметтерінің жабдықтаушысы жай ғана пайдаланушылар үшін көлік функцияны орындайды - интернеттің қызметтерінің басқа жабдықтаушыларының желісіндегі олардың трафигінің берілуін;
- контенттің интернет-сі жабдықтаушы меншікті информациялық-анықтама қорларды веб-сайт түрдегі олардың мазмұны іліге алады;

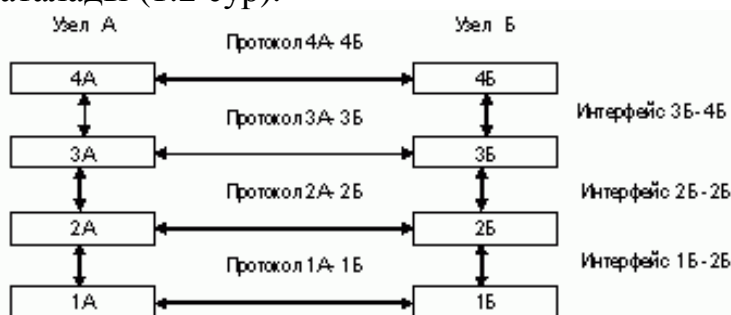
- хостингтің қызметтерінің жабдықтаушысы сыртқы контенттің орналастыруы үшін өз бөлмелері, байланыс каналы және серверлерді ілігеді;
- контенттің жеткізуі бойынша қызметтердің жабдықтаушысы мәліметке қолданушыларының рұқсаттың жылдамдығының жоғарылатуының мақсаты бар көп қол жеткізу нүктелерінде контенттің тек қана жеткізуімен шұғылданады;
- мысалы, қосымшалардың қолдауы бойынша қызметтердің жабдықтаушысы SAP R3 ірі әмбебап программалық өнімдерге рұқсат клиенттерге ілігеді
- биллинг қызметтерінің жабдықтаушысы интернет бойынша есептердің төлеуін қамтамасыз етеді;

### *Интернеттегі стандартизацияның ролі*

Интернет өте күрделі желі болып табылады, сәйкесінше желі құрылымдардың арасындағы өзара әрекеттесуді ұйымдастыру күрделі міндет болып саналады. Сондай міндетті шешу үшін декомпозицияны қолданылады, яғни күрделі мәселені бірнешеге бөліктеулерарқылы шешу. Декомпозицияны іске асыратын тұжырымдамалардың бірі көп деңгейлі жол болып табылады. Мұндай жол басқа деңгейлерден тәуелсіздің әрбір жеке деңгейдің түрлендіруін өткізуге, өндеуді, тестеу мүмкіншілігін береді.

Иерархиялық декомпозицияны төменнен жоғары деңгей бағытында шешілетін есептің қарапайым түрін алуға мүмкіндік береді.

Желілік өзара әрекеттесуді көп деңгейлі ұсыныстың ерекшелігі хабар алмасу процесінде үшін, аппаратты - программалық құралдардың екі иерархияының үйлесімді жұмысты қамтамасыз ететін, кемінде екі тарап қатысады. Деңгейлердің әрқайсысы меншікті құралдар иерархияның жоғары және төменде жатқан деңгейлерің интерфейсін қолдауы. Интерфейстің осы түрі хаттама деп аталады (1.2 сур).



**Сурет. 1.2.** Интернеттің желісіндегі иерархиялық декомпозициясы кезінде иерархияның деңгейлерінің арасындағы өзара әрекеттесуді ұйымдастыру

Желі түйіндерінің өзара әрекеттесуіне жеткілікті иерархиялық ұйымдастырылған хаттамалар жиыны хаттамалар стегі деп аталады. 80-ші жылдардың басында халықаралық стандартизациялау ұйымдары ISO (International Organization for Standardization), ITU (International

Telecommunications Union) және басқалар ашық жүйелердің өзара әрекеттесулерінің стандартты үлгісін OSI (Open System Interconnection) жасады. Үлгінің міндеті желілік өзара әрекеттесудің құралдарының біріктірілген көрінісінен тұрады. Сонымен бірге оны желі мамандарның әмбебап тілі (анықтама үлгісі) ретінде қарастыруға болады.

Егер екі желі ашықтық қағидаларың орындалуымен құрастырылса, бұл келесі артықшылықтарды береді:

- Аппаратты және стандарт ұстаған әр түрлі өндірушілердің программалық құралдарының желінің құрастыруын мүмкіндігі;
- Желінің жеке компоненттерінің басқа жетілген түрімен мүлтіксіздік алмастыруы;
- Бір желінің екіншімен түйіндесінің жеңілдігі.

OSI үлгілер шеңберінде өзара әрекеттесу құралдары жеті деңгейге жіктеледі: қолданбалы, ұсыныстар, сеанстық, көліктік, желілік, арналық және физикалық. Бағдарламашылардың қарамағында ең жоғарғы қосымшалардың деңгейіне сұрау салуларымен айналуға мүмкіндік беретін қолданбалы программалық интерфейс жеткізіліп беріледі.

Интернет желісі ашық жүйе қағидалармен дәлме-дәлдікте жасақталды. Бұл желінің стандарттарының өңдеуінде ЖООның, ғылыми ұйымдар және серіктестіктерден мыңдаған желінің маман-қолданушылары араласты. Стандартизация бойынша жұмыстың нәтижесі RFC-тың құжаттарында іске асады.

RFC (ағыл. Request for Comments) —бүкіләлемдік желі кең қолданылатын техникалық спецификациялар және стандарт тұратын, нөмерленген ақпараттық құжаттар топтамасының құжаты. 1.2 кестеде RFC-тың өте белгілі құжаттарының кейбірі келтірілген.

№ RFC	Тақырып
RFC 768	UDP
RFC 791	IP
RFC 793	TCP
RFC 822	электрондық пошта форматы, RFC 2822 ауыстырылған
RFC 959	FTP
RFC 1034	DNS — концепция
RFC 1035	DNS — ендіру
RFC 1591	домендік аттар құрылымы
RFC 1738	URL
RFC 1939	POP хаттамасының 3 нұсқасы ( <a href="#">POP3</a> )
RFC 2026	Интернеттегі стандартизация процесі
RFC 2045	MIME
RFC 2231	символ кодтау

RFC 2616	HTTP
RFC 2822	электрондық пошта форматы
RFC 3501	<a href="#">IMAP</a> 4 нұсқа 1 басылым (IMAP4rev1)

Интернет стандартизациялар бойынша жұмысын басқаратын негізгі ұйымдастыру бөлімшесі (Internet Society ) ISOC болып табылады. Оған осы желіні дамытудың әр түрлі тұрғыларымен шұғылданатын 100 мың шамасында қатысушы біріктіреді. ISOC екі топ тұратын IAB (Internet Architecture Board ) жұмысына жетекшілік етеді:

- IRTF (Internet Research Task Force). TCP/IP жататын ұзақ мерзімді зерттеу жобаларын басқарады;
- IETF (Internet Engineering Task Force). Интернеттің келесі стандарттары үшін спецификация анықтайтын инженерлік топ
  - 1994 жылдан бастап веб желілер үшін стандарттардың өңдеуімен, Бернерс Тим-Ли құрған және осы күнге дейін басшылық ететін W3C (World Wide Web Consortium ) консорциумы шұғылданады.
  - W3C консорциумы - интернет және WWW үшін технологиялық стандарттар өңдейтін және енгізетін ұйым.

## **Тарау 1. Интернеттің ұйымдастырылуы**

### **1.1 Интернет негізгі түсініктері**

*Интернет өзі* Interconnected networks (біріктірілген желілер) терминінен пайда болады, яғни техникалық көзқарастар тұрғыдан- бұл үлкен және кіші желелердің ауқымды түрде біріктірілуі. Толығырақ айтқанда- дүние жүзіндегі миллиондаған компьютерлердің мәліметтер алмастыруына мүмкіндік беретін ақпараттық орта. Интернет - өзіне көптеген таңғаларлық арсеналды мүмкіншіліктерді жинаған жаңа технология. Интернет - әрі ең қуатты және тәуелсіз ақпараттық ресурс, әрі сенімді және оперативтік байланыс жүйесі, сонымен қатар ақпараттық технологияның дамуына және дүние жүзіндегі миллиондаған адамдардың творчестволық көзқарасымен бөлісуге мүмкіндік беретін орта.

Интернеттің басты қызметі тәулік бойы сапалы байланыс қызметін көрсету болып табылады. Интернетке қосулы тұрған кез – келген екі компьютер бір – бірімен қалаған уақытында байланыса алады. Алдағы уақытта «Желі» ұғымын Интернет сөзінің синонимі ретінде қолданып, ол желі түсінігі интернет арқылы байланысқан компьютерлер екендігін ескертіледі. Интернетке қосылған әрбір компьютер желінің бір бөлігі болып табылады.

#### **1.1.1 Серверлерлік компьютерлер және клиенттік - компьютерлер**

Интернетке қосылған барлық компьютерлерді екі типке бөлуге болады: серверлер және клиенттер. Бір компьютерде сервер де, клиент те қатар орналасқан болуы мүмкін. Локалдық компьютерде веб – сервер қосып, сонымен қатар дәл сол компьютерде браузермен және электронды почтамен жұмыс істеуге болады.

Басқа компьютерлерге сервистік анық қызмет көрсететін компьютер – сервер(ағылш., *to serve* – қызмет көрсету), ал сол қызметті пайдаланушы компьютер - клиент болып табылады. Көп жағдайларда үй желісі арқылы Интернетке кіріп отырған клиенттер тек керек жағдайларда ғана бұл қызметті пайдаланады. Ал серверлік компьютерлер, керісінше, әрқашан Интернет желісіне жедел каналдар арқылы қосулы тұрады, сондықтан оларға кез – келген ақпаратқа сұраныс жасауға болады.

#### **1.1.2 Серверлерлік - қосымшалар және клиенттік - қосымшалар**

Компьютерлерді клиенттер және серверлер деп атаумен қатар, клиент және серверлерді бағдарламалық тұрғыдан қарастырған дұрыс. Бір бағдарлама клиент ретінде, ал екіншісі сервер ретінде қосымшалардың іс – әрекет етуі клиент – серверлік архитектура деп аталады.

Сервердің басты қызметі – барлық уақытта жұмыс істеп тұру немесе клиент қандай да бір сервиске сұраныс жасағанға дейін күту режимін қабылдау.



Сервер шамадан тыс сұраныстардың көп болу әсерінен нақты бір сервистік сұраныстардың орындалуына кідіртүі мүмкін. Серверге сұраныс желі арасындағы компьютерлік байланысты қамтамсыз ететін стандарттар жинағы - белгілі бір протоколдар шегінде жүзеге асады. Серверлік бағдарламалар клиенттерге қызмет ету мақсатында компьютердің аппараттық ресурстарын пайдаланады. Клиент – бағдарлама сұраныс жасақтап, оны желіге белгілі бір мекен – жай бойынша жібереді және алдын ала келіскен протокол арқылы сервер – бағдарламамен қарым – қатынаста болады. Бір компьютерде бірнеше серверлік бағдарламалар орналасуы мүмкін. Клиенттік қосымша сервер орналасқан компьютерде орнатулуы мүмкін немесе желімен байланысқан оқшау компьютерде орналасуы мүмкін. Оқшау орналасқан жағдайда жауап алу мөлшері көбірек болуы мүмкін.

Әрбір сервер – бағдарламасына сай өзіндік клиент – бағдарламасы болады. Осылайша веб - клиент веб - серверге, почтолық клиент - почталық серверге т.с.с. сұраныс жасайды. Серверлік - бағдарлама сұраныстарды орындауға әрқашан дайын болуы керек, сондықтан да серверлік бағдарлама жұмыс істеп отырған компьютерлерге аппараттық жоғары талаптар қойылады. Клиенттік компьютерге ондай қатаң талаптар қойылмайды. Жоғарыда айтылған қызмет көрсетулер (клиенттік-серверлік архитектура) дербес компьютерлер қолданушысына өз жұмыс столынан Интернет желісіне кіруге мүмкіндік береді.

### **1.1.3 Провайдерлер және олардың желілері**

Интернетте жұмыс жасағанда біз, интернет-провайдер қызметін жиі пайдаланамыз ISP(Internet Service Provider). Әдетте, ISP – жеке (магистральды) желісі бар көптеген клиенттер қосылатын мамандандырылған ұйым. Провайдер желісі бүкіл жер шарының әр жерінде орналасқан желілерге қосылып планетамыздың кез келген нүктесімен байланысуға мүмкіндік береді. ISP-провайдерлер – әр ауданда Интернетке кіруге мүмкіндік беретін аппараттық құрылғылары орналасқан өз қатынас ету нүктелері (POP - Point of Presence) бар ірі компаниялар. Ірі провайдерлердің әртүрлі қалаларда орналасқан ондаған қатынас ету нүктелері және мыңдаған клиенттері бар. Ірі провайдерлермен қатар кішігірім қатынас нүктелері бір қалада орналасқан провайдерлер де болады. Провайдерлер ретінде жеке меншік арнайы маманданған фирмалар да ірі телефон компаниялары да қызмет істей алады.

ISP мен телефон желісі арқылы байланыс орнату механизмі: ДК қолданушысы провайдерге телефон шалып, онымен және оның модемімен байланыс орнатады. Қолданушы өзінің ISP қосылған соң провайдер желісінің құрамына енеді. Өз серверінде провайдер қолданушыларына әртүрлі сервис қызметін көрсетеді: электрондық почта (e-mail), желі жаңлықтар қызметі (Usenet) т.с.с. Провайдердің магистральді желісін бэкбоун (ағылш. Backbone — жон омыртқа) немесе тірек (опорная) желі деп те атайды. Провайдер желілері көптеген қолданушыларға қызмет көрсететіндіктен олардың желілері үлкен

жылдамдықты желіге және ауқымды трафикке иеленуі керек. Сол себепті провайдерлер ондай желілерді ірі коммуникационды компаниялардан арендаға алады немесе жеке өзінікін тарттырады. Ірі коммуникациялық компаниялар жеке үлкен жылдамдықты желілерге ие.

### **1.1.4 Провайдер желілерінің біріктірілуі**

Бір провайдерлердің клиенттері, мысалға ISP-A өз желісінде бір бірімен қарым – қатынаста бола алады, ал басқа компания ISP-B клиенттері өз желісінде бір бірімен қарым – қатынаста бола алады. Егер ISP-A және ISP-B желілері байланыспаған болса, онда А компаниясының клиенттері В компаниясының клиенттерімен байланыса. Оларды біріктіру үшін А және В компаниялары арасында желілік байланыс нүктелері (NAP - Network Access Points) арқылы тіке байланыс орнатады. Осылайша барлық провайдерлердің магистральды желілеріне қосылып нәтижесінде жоғары дәрежелі желілер жиының бірігу процесі қалыптасады.

Интернетте жүздеген ірі интернет-провайдерлер қызмет етеді, олардың әр қалада орналасқан магистральды желілері NAP арқылы жұмыс істейді және көптеген мәліметтер жиыны NAP-түйіндер арқылы түрлі желілерге жеткізіліп отырады.

Үлкен және кіші интернет желілерін біріктіру негізінде келісім шарттар тізбегі орналастырылады. Әрбір клиент өз локалдық желісінің провайдер желісіне қосылғандығы жөнінде келісімшартқа тұрады.

### **1.1.5 Провайдер желісінің иерархиясы**

Дамыған елдерде жүздеген провайдерлер болады, олар халықаралық, мемлекеттік және регионалды болып бөлінеді.

Регионалды провайдерлер жедел жылдамдықты каналдар арқылы мемлекеттік провайдерлермен байланысқан. Мысалға АҚШта стандартты жеткізу жылдамдығы 1,544 Мбит/с Т1 линиясы немесе жеткізу жылдамдығы 44,74 Мбит/с. Т3 линиясы Халықаралық провайдер желісі мұнан да жедел каналдар арқылы біріктіріледі.

Осылайша сіз провайдер желісіне қосылу арқылы Интернетке кіріп отырған жер шарының әр жерінде орналасқан компьютермен байланысқа түсе аласыз.

#### **Бақылау сұрақтары:**

1. Клиент-серверлі архитектурасы желісінің негізгі элементтерін атаңыз?
2. Желінің бағдарламалық қамтамасыз етуі қандай принциптермен бөлінеді?
3. Провайдердің магистральді желісі термині қандай мағына береді?
4. Провайдер желілерін біріктіру процесі қалай жүзеге асады?
5. Интернетте пайда болатын желілілер иерархисы қандай ?

## 1.2 Интернет коммуникациялардың модельдері.

Интернет жүйесінің коммуникациялық сипаттамасын қарастырайық. Дәстүрлі коммуникациялық модель негізіне "біреуі бірнешеге" процессі тұжырымдалған. Коммуникация құралына байланысты ақпарат статистикалық түрде (текст, графика) немесе динамикалық түрде (аудио, бейнеклип, анимация) берілуі мүмкін. Дәстүрлі құралдар қарым қатынасының негізінің басты ерекшелігі интерактивті қарым қатынас болмауында.

Бұл модельден айырмашылығы Интернет негізіне екі принцип орналасқан. Біріншіден, Интернет коммуникациясы кезінде өзара әрекеттесу өзара әрекеттесуге түбегейлі үлесті кіргізетін оның ортасы арқылы болады. Бұл модель бастапқы қарым – қатынас ақпаратты алушы мен жіберуші арасында емес, қолданушы мен коммуникация орта арасында болып жатқандығын көрсетеді. Мұнда диалогқа түсушілер ақпарат жіберуші де ақпарат алушы да бола алады. Осы модельде жай қабылдаушы мен алушы арасындағы ақпарат алмасудан ақпараттық орта құруға қадам жасалынған.

Интернет "бірнешеу-бірнешеуге" атты коммуникациялық процессті ұсынады, мұнда желідегі әр бір қолданушы өзінің атынан немесе өзінің тобы атынан жекелеген немесе топталған абоненттермен қарым – қатынас жасауға болады. Бұл модельде коммуникация құралы болып арнайы жекелендірілген компьютерлік желі болып табылады, ал ақпарат басқа да мәліметтермен бірге гипермедиалық түрде жеткізіледі. Берілген модель көрсетіп отырғандай интерактивтік қарым – қатынасты Интернет сервисіндегі қолданушымен де, осы қызмет түрін ұсынып отырған ортамен де мүмкін екенін көрсетеді. Осындай қарым – қатынас негізінде тасымадаушы әрі тұтынушы, әрі қызмет көрсетуші бола алады. Мұндай модельде ақпараттар жіберушіден тұтынушыға беріліп қана қоймай өзінің қолданушылары арасында ақпараттық ортаны құрып та отырады. Дәстүрлі жалпылама коммуникациялар құралдарының коммуникациялық моделі қайта байланысу контурына ие емес, ал Интернет желісіндегі модель орталықтары үшін қайта байланысу контуры мүмкін. Тұтынушылар арасындағы қайта байланысу контурының орындалуы ретінде электрондық почта, "cookie" файлдары, Веб – серверлерге тіркелу мысал бола алады. Қайта байланысу коммуникациялық құралдарды қолдану тиімділігін арттырумен қатар ішкі және сыртқы ортадағы оқиғаларға сәйкес жауап қайтаруға мүмкіндік береді.

### 1.2.1 Интернет коммуникациясының Pull- и Push-моделдері

Коммуникация тұрғысынан қарасақ тұтынушыларға бұқаралық ақпарат құралдары (теледидар, радио и т. д.) ақпаратты жеткізудің push-модель орындап отыр; бұл жағдайда тұтынушы пассивтік роль атқарып, тек шектеулі ақпараттар арналарын жиының таңдауға ғана ие.

Интернет желісінде push-модельден басқа, pull-модель де бар; ол ақпаратты тек сұраныс бойынша ғана жіберіп отырады (demand pull). Бұл Интернет ортаның ерекшелігі – тұтынушылар активті роль ойнайды себебі олар ақпараттар мен навигацияларға іздеу механизмдерін және URL (uniform resource locator) басқа қорларын пайдаланып өздері сұраныс жасайды. Сонымен қатар Интернетте push-модельмен жұмыс жасау мүмкіншілігі бар. Интернет бұл модельді push-технологиясы арқылы жүзеге асырады, яғни қолданушылар қажетті ақпаратты Интернет желісінде іздемей-ақ, өзін қызықтырушы тематика каналына тіркелсе, қажетті ақпарат автоматты түрде канкалға жазылғандардың компьютеріне жеткізіліп отырады.

Интернеттегі технологиялар push- и pull-модельдері арасында тығыз байланыста дамып келеді. Интернет желісінің жоғары функционалдығына байланысты тұтынушының ақпараттарды, компьютерлік ресурстарды пайдаланулары үшін навигациялауына әрқашан мүмкіндік бар. Келесі бір жағынан қолданушының ақпараттармен жабдықтаушыларды және қабылданатын push-каналдардың тематикаларды таңдауға еркі болады.

### **1.2.2 Интерактивтеу**

Интернет жүйесіндегі маңызды сипаттамалардың бірі - интерактивтеу. Интерактивтеу – коммуникациондық хаттардың қарым – қатынасын анықтайтын коммуникациондық процесстің жүзеге асу сипаты. Интерактивтік қарым – қатынас сипаттамасына сәйкес келіп түскен хатқа жауап берілуі қажет және де жауап алдыңғы хаттардың контекстіне сай болуы керек. Жоғарыда айтылғандарды қорытындылай келе, Интернетті интерактивтеу диалогке қатысушыларға келіп түскен хаттарға жауап беру қабілеттілігі. Мұнымен, интерактивтеу диалогке қатысушының іс-әрекетін толық жеткізуге, компьютердің функциясын кеңейтеді. Интернет ортадағы интерактивтілік енді, орта арқылы дербестен деңгейде қарым-қатынас емес, ортамен өзара әрекеттесу деңгейде тікелей іске асады.

Өзара әрекеттесуді интерактивті сипаты коммуникацияның қатысушыларының өзара әрекеттесуін тиімділікті жоғарылатуға айтарлықтай мүмкіндік береді. Мысалы, іс жүзінде бұл мәміленің шешімі немесе сатып алу үшін қажетті мәліметтің алуы үшін тиісті уақыт қысқарта алады. Сонымен бірге, ортаның интерактивті сипаты жеке клиенттің ерекшеліктеріне байланыстың құрал-сайманының мүмкіндігін туралауды ұсынады.

### **1.2.3 Гипермәтін**

Интернет желісіндегі тағы ерекшелік оның гипермәтіндік табиғаты болып табылады. Дүние жүзіндегі ең алғашқы гипермәтіндік жүйені 50 жыл бұрын, бірінші ЭВМ-нің негізін қалаушы, Ванневар Буш ұсынды. Бұл жүйе Метех деп аталады және былай түсіндірілді: " Қолданушы өз кітапшаларын, жазбаларын

және коммуникацияларын сақтайтын құрылғы . Ол әрі ыңғайлы, әрі жедел іске қосылуы тиіс". 1967 жылы Нельсонның сипаттамасы бойынша Гипермәтін-сілтемелер мен ассоциалар желісі. Ол адам, өз жадында, ақпараттың әр фрагментің қалай біріктірсе, дәл солай жүзеге асады. Ал Бомман 1993 жылы Гипермәтін анықтамасын былай көрсетеді: "Гипермәтін- қолданышуның желілер мен байланыстар арқылы біріктірілген ақпараттық фрагменттер концепциясы . Гипермәтінде ақпараттық жүйе хаттамалар мен сілтемелер түрінде ұсынылады.

Гипермәтін сызықтық емес ақпараттар жиынының формасы ретінде ұсынылады. Олардың әрқайсысы жек фрагменттердің құралған. Әрқайсысында байланыс түрінің типіне байланысты келесі фрагментке сілтеме көрсетілген. Гипермәтіндік технологиясының орнына емес, ыңғайлылығы, әрбір қолданушы, біліміне және қабілеттігіне қарай материалды қолдану әдісін өзі анықтайды.

Гипермәтін мәліметті ғана емес, оның тиімді іздестіруін қамтамасыз ететін аппарат та болады. Сайып келгенде, мәліметтің гипермәтін ұсынысы кәдімгі әдістің алдында артықшылықтары мәліметтің менгеруі мен өндіріп алуға тиімді әсер етеді.

#### **1.2.4 Мультимедиа**

Мультимедианың пайда болуына байланысты статикалық (текст, графика, сурет) және динамикалық ( аудио, видео, анимация) құралдардың компьютерлік интеграциялау мүмкіншілігі пайда болады. Мультимедиа құралдары мен гипертекст, камбинациясы жаңа ортаның пайда болуына себеп болды.

Гипермедиа – ақпараты бейнелеу және оған рұқсаталу әдісі. Оның концепциясы, түйіндері ақпаратты сақтайтын граф түрінде, семантикалық байланыстары графаның доғалары түрінде көрсетілген, ақпараттық орта моделінде қалыптасады. Гипермедиа жүйесінде сақталған ақпараттар қазіргі заманғы компьютерлер оқитын барлық формада ұсынылуы қажет. Осылайша гипермедиа өзінде гипермәтіндік байланыстар арқылы радионы да (аудио), теледидарды да (динамикалық бейнетаспа), прессаны да (текст, сурет, фото) және компьютерді (бейнетерминал) біріктіреді.

#### **1.2.5 Қатысу эффектісі**

Коммуникациондық Интернет моделінің тағы бір негізгі элементі "Қатысу эффектісінің" болып табылады . Ол клиент пен қоршаған ортаның қарым-қатынасын сипаттайтын процесс, және компьютерлік гиперқұралдар құрайтын, яғни жұмыс орнынан, қоршаған ортадан, бөлмеден т.с.с тұрады. Осы екі жағдайдың клиентке әсер етуі "қатысу эффектісінің" интерактивті ортамен және компьютерлік гиперортамен тікелей байланысты басты ерекшелігі, "қатысу эффектісінің" клиент үшін "айқындылығы" Ол диалогтің қарапайым және жан бітірілген түрде өтуіне көп мүмкіншілік тугызады.

## 1.2.6 Желілік навигация

Желілік навигация компьютерлік гиперортада өзіндік анықтау процесі ретінде сипатталады. Желілік навигацияға негізделген сызықтық емес іздеу мен ақы сұранысы клиентке бұл процестің үстінен қағалап отыруға мүмкіншілік береді

### 1.2.7 Интернет сервисіне коммуникациялық сипаттама.

Интернет әртүрлі сервистер жиынтығын ұсынатындықтан, оның әрқайсысының жеке сипаттамасын білу керек. Таблица да интернет сервис коммуникациясы анализ қорытындысының қысқаша мағлұматтар ұсынылған.

**Кесте 1.3 Интернет сервисінің коммуналдық сипаттамасы**

Сервис	Аралық қарым-қатынас	Машиналық қарым-қатынас	Коммуникация моделі	Ақпараттың жеткізілу түрі	Қайта байланысу симметриясы	Интерактивтік қарым-қатынастар мүмкіндігі
Электрондық пошта	Ия	Ия	Біреуден-біреуге-бірнешеуден-бірнешеуге	Мәтін, графика, дыбыс	Ия	Жоқ
Таттар тізімі	Ия	Ия	Бірнешелер-бірнешеуге	Мәтін	Ия	Жоқ
Конференциялар	Ия	Ия	Бірнешелер-бірнешеуге	Мәтін	Ия	Жоқ
Өйлесу бөлмелері	Ия	Ия	Бірнешелер-бірнешеуге	Мәтін	Ия	Ия
WWW	Ия	Ия	Бірнешелер-бірнешеуге	Мәтін, графика, дыбыс, баспа	Нет	Ия
Дыбыстық байланысу бағдарламасы	Ия	Ия	Біреуден-біреуге-	Дыбыс	Ия	Ия
Аудио және баспа конференциялар	Ия	Жоқ	Бірнешелер-бірнешеуге	Дыбыс, сурет	Ия	Ия

Берілген мағлұматтар интернет орта коммуналда диапазон түрінің кең екендігін дәлелдейді.

Қажетті сервис- құралын анықтаған соң, оның әрқайсысына анализ жасап, бағау беруге болады.

#### **Бақылау сұрақтары:**

- 1) Интернет жүйесінде қандай коммуналдық модельдер жұмыс істейді?
- 2) Гипертекст ұғымын қалай түсінесін?

- 3) Интернет- ортада интерактивтеу дегеніміз не?
- 4) Интернет коммуникациясын Рим және PUSH модельдерінің айырмашылығы неде?
- 5) Интернет ортада " қатысу эффе́ктісі" деген ұғым қандай мағына береді.

### 1.3 Қолданушылардың интернетке қосылу мүмкіндігі

Бұлардың көпшілігі провайдер секілді желіге жоғары жылдымқты қосылуды қамтамасыз етпейді. Абонент пен провайдер арасындағы байланысу технологиясы " ең соңғы миль" атағын алды. Көп жағдайда желіден қолданушы компьютеріне ақпараттарды жеткізудің " соңғы нүктесі" болып каналдар тізбегіндегі ең жіңішке жері табылады.

*"Соңғы миль" технологиясының* интернетке қосылу әдісі әртүрлі болу: "Қосылу жылдамдығы" жоғарыболса, оған төленетін төлемақты да жоғары. Бұрыннан пайдаланылып жүрген инфоструктурамен де жұмыс істейтін технологиялар бар: телефон линиялары, кабельдік теледидар, желісі және т.б. Интернет қосылу әдісінің әртүрлі технологиялары; телефон линиясы арқылы модемдық қосылу радиоканал арқылы қосылу, кабельдік телевидения арқылы қосылу тартылу желісі арқылы және, ең соңғысы, спутник каналдары арқылы қосылу.

Интернет қосылу әдісі көп қолданушы үшін айырмашылығы тек қосылу тұрақтылығында, жылдамдығында және жауап қайтару мерзімінде.

*Интернет ресурстарының барлығы да, клиент үшін, провайдер арқылы қол жеткізімді болады. Интернет қосылу әдісінің әрқайсысына жеке тоқталып өтейік:*

#### 1.3.1 Телефон линиясы арқылы.

Коммутациялық каналдар желісі және телефон линиясы арқылы интернет қосылудың әрқайсысында тоқталып өтейік. Сіз біреуге телефон соққанда коммутаторлар арқылы жалғанатын физикалық байланыспайда болады. Әрбір абонент участкісі бірінші коммутаторға дейін бөлшектелмейді, жеке қолданады. Ал басқа коммутаторлар бөлшектеледі және әр уақытта қолданушылармен байланыса алады. Егер бір участок басқа қолданушымен байланысып отырса, онда келесі участок мұнымен байланысқа түсе алмайды. Коммутаторлық телефон линиясы арқылы байланыс- уақытша (сианстық) байланыс. Телефон линиясы - телефонмен өйлесумен немесе модем арқылы мағлұматтар жіберілумен бос болмауы мүмкін " Модем" өзі "Модулятор" және ДЕМодулятор" өздерінен шыққан. Модем телефон линиясы арқылы мағлұматтарды жіберу үшін қолданылады.

Жалпы айтқанда, модем- сандық кодты екі арнада екі дыбыс ретінде жіберетін, және керісінше, дыбысты сандық ақпаратқа айналдыратын құралғы.

Екі модем байланыс орнатқан соң модулятор негізгі сигналды тұрақты арнаға қосып (ақпарат тасушы сигнал), екінші сигналды қосады (ақпаратты

өзгертуші сигнал) Демулятор келіп түскен сигнал мен тасушы сигналдың айырмашылығын анализдеп, өзгерістерге қарай негізге сигналды іске қосады. Басқаша айтқанда модем тасымалдаушы сигналды модульдеп, жоғары адресат сигнал қайтадан қажеті сандық формаға ДЕМОДУЛЬдейді. Бұл процесс ASCII (American Standard code for Information Interchange) кодтауға сай болуы керек.

ASCII- латын алфавитінен, сандардан, және т.б. тұратын 128 таңбалық кодтау 7 битті қамту к.к., ал 8 бит - кодтың дұрыстығын қамтиді.

Егер екі түрлі тасушы сигналды қолданса, мағлұматтар бір мезгілді 2 арнаға да беріле алады. Алғашқы модем VM Laboratories зерттеу орталығында ойлап табылды, VM 103 есіміне ие болды.

Бұл құрылғылар 2 жұп арнаны қолданды. 1 жұп арна 1 модемге жіберуші модем мәліметтерді 1,07 және 1,27 КГц арнасында жіберген.

VM 103 модемдері 300 бит /с жылдамдықта жұмыс істеген, шамамен 1 с 30 таңба.

VM 103 модемдер 300бит/с жылдамдықта жұмыс істеген, шамамен 1 с 30 таңба жіберіліп отырылған. Мұндай жылдамдық тек мәтіндік мәліметтермен жұмыс істейтін кезде қанағаттандырған, ол, тіпті, адамның текст оқу жылдамдығынан жоғары болған.

Жылдамдық 300бит/с модемдері 1980 ж. басына дейін тіршілік етті. Адамдар сурет секілді мәліметтерді жібере бастаған кезден-ақ, бұл жылдың жеткіліксіз болады.

1980 ж. ортасынан бастап жылдамдығы 1200бит/с болатын модемдер пайда болды. 1990 жылдың басынан бері қарай модем жылдамдығы 9,6 -дан бастап, жоғарылай түсті: 19,2; 288; 33,6; Кбит/с

Қазіргі модемдер - күрделірек құрылғылар, олар модуляцияның күрделі - схемаларын қолданып, мағлұматтарды сығады. Бірақ жұмыс істеу принципі өзгерген жоқ. Клиент компьютері кез-келген мәліметтер мен командаларды жібере алатын нольдер мен и 1-лерден құралған коммуникациондық порт ұсынады. Модем оларды қабылдап, командалар мен ақпараттарға жіктейді, сонан соң телефон мениясына жібереді.

Қолданушы модем арқылы ортақ қолданысағы телефон жиесіне қосылады. Ал ISP, басқа модем арқылы сандық сигналдық анықтап, интернетпен байланыс жасайды.

Қазіргі коммунитарлық линияларға арналған модемдер ішкі модемдер және кеңейту құрылғысы ретінде қолданылады, ол сыртқы модем-өзіндік автономды блок питаниясы бар, жеке құрылғы ретінде жұмыс істейді.

Қазіргі модемдердің қабылдау жылдамдығы - 56 Кбит/с, ол жіберу жылдамдығы 33608 Кбит/с жоғары. Яғни, VM 103 модемдерімен салыстырғанда 200 есе жылдам жұмыс істейді.

#### **Бақылау сұрақтары:**

- 1) "Соңғы миль" Интернет технологиясы қандай мағына береді?
- 2) Модем функциясы (қызметі) қандай?
- 3) ASCII- кодтау дегеніміз не?
- 4) Модемдердің жылдамдығылары қандай?



## 1.4 Интернет желісіне қосылу технологиялары

Қарапайым телефон тек төменарналық диапазон линиясын қолданады. Екіарналы телефон проводынан ақпаратты жіңішке "телефондық" жіберу аралығымен емес, одан да кеңдеу аралық қолданса, ақпаратты бұдан да көп жіберу мүмкіншілігі туады.

Жіберу аралығы және оның аумағы дегеніміз - ақпарат жіберіп жатқан канал диапазонындағы ең жоғарғы және ең төменгі арналардың айырмашылығы.

Сигналдық телефон линиясы арқылы аналогті жіберілуі 300Гц - тен 3,4кГц аралығындағы диапазонды қамтиды. Сандық сигнал үлкен диапазондық арнаны талап етеді. Жіберу жылдамдығы, жоғары болған сайын, диапазон арнасының кеңеюін қажет етеді. Сандықтан да кеңаралықты қосылу ұғымын кезінде компьютер желісінің жоғарғы шегі аталады.

Жоғары жылдамдықты ақпарат жіберу деп бірнеше сигналдардың бір физикалық арнада бірнеше арналарға топталуы негізінде жіберілуі мультиплексирлеу д. а. Қысқаша айтқанда, мультиплексирлеу - арналардың кішігірлім арналықтарға жіктелуі.

"Кеңаумақты қосылу" терминін транспорттық магистраль арналогиясы негізінде тусіндіруге болады. Яғни, машина жолдары көп болған сайын, сол арқылы бір мезгілде көбірек транспорт жүріп өтеді. "Кеңаумақты қосылу" термині тек дыбысты ғана жіберуге мүмкіншілігі бар каналдарды көрсетеді. Кейде "кеңаумақты емес қосылу" термині де кездеседі.; бұл- мәліметтердің 64Кбит/с каналына диін ғана жіберілуі.

Қолдануына байланысты кеңаумақты емес қосылу әртүрлі анықтаманы көздейді. Мысал: Jupiter communications өздігі оны 256 Кбит/с кем емес канал ретінде анықтайды. Әдетте кеңаумақты қосылу бір канал арқылы сигнал жіберуге де телефон және интернетпен қолдануға мүмкіндік береді.

DSL технологиясы сандық мәліметтерді жоғары жылдамдықты жіберу мақсатында телефон желісін қолдану

DSL (Digital Subscriber Line) аббревиатурасы - сандық абоненттік линия. Бұл технология телефон сұхбатына зиян келтірместен жоғары арналарда ақпарат жіберуге қолданылады. XDSL деп аталатын бұл технологияның көптеген топтамалары (X-бұл технологияның нақты бір түрін нұсқайды). DSL технологиясы пермпективті болып табылады. Ол бір мезгілде интернетке қосылуға да, телефонмен сұхбаттасуға да мүмкіндік береді. DSL жылдамдығы басқа модемдермен салыстырғанда біршама жоғары болады. DSL үшін жаңа проводтар қажет емес, өйткені ол телефон линиясы тікелей жалғанады.

### 1.4.1 Ассиметриялық DSL (ADSL)

Интернет жүйесінде жұмыс істеген кезде ақпарат көбінесе желіден қолданушыға жіберіледі, ал желігі бірнеше есі аз мәліметтер көлемі келіп түседі. Мысалы: сіз, веб-беттерді қарағанда, сіз тек кішігірім сұраныс

жасайсыз, ал желіден текстеп бірге сурет те келіп түседі. Яғни, ақпарат алмасу асимметриялық болып табалады. Сондықтан мұндай трафикке асимметриялық канал қажет. Мұндай асимметриялық алмасуды айтқанда автомагистраль аналогиясын елестетуге болады: бір жаққа азмашина ағымы қозғалуда, ал екінші жаққа – кеіріснше. Бұл жағдайда кіру аралығын – жіңішке, ал шығу аралығын – кең етіп орнатқан тиімді.

ADSL – мәліметтерді қолданушының жіберген сұраныстың жылдам жіберген сигнал желіден қолданушы сигналына қарағанда төмен болады.

Қазіргі жаңа технологиялар арқылы мәліметтерді ADSL – де жіберу он есе арттырып отыр: шамамен абоненттен желіге – 1Мбит/с.

Сонымен қатар, бір линияны сандық және дыбыстық мәліметтерді жіберу үшін параллель қолдануға болады. Жоғары деңгейлі жылдамдық, веб-сайттармен жұмыс істеуге, үлкен ауқымды файлдарды тарттыруға және интерактивті приложенияларды толығымен қолдануға қолайлы болып келеді.

ADSL басты бір ерекшелігі – оны қондыру болып табылады: телефон станциясы мен қолданушыны байланыстырып тұрған телефон кабеліне жалғанады. ADSL –дің өз кемшілігі де бар: қашықтық шектеулігі.

ADSL технологиясы қолдануда ақпараттар арнасының кері бағыты қашықтыққа тәуелді болады. Егер абонентке қарай ақпаратты 3 км арақашықтықта 8 Мбит/с тек 1,5 Мбит/с жылдамдықпен алуға болады. Стандартты ADSL 2 технологиясында «тығу» жылдамдығы 3 км қашықтыққа 10 Мбит/с – ті қамтиды, ал «шығу» жылдамдығы 1Мбит/с

Қолданушының ДК ADSL – модеміне жалғанады, ADSL – модемінің қызмет принципі былай тұжырымдалады:

24 кГц – тен 1100 кГц интервалында диапазон арнасы 4000 Гц/тік арнашықтарға жіктеледі, әр арнада виртуалды модем өз диапазон арнасында жұмыс істейді. ADSL – модем дара бөлгішке қосылады. Арналық бөлгіні төмен арналы фильтрді ұсынады, ол төменарналы телефон байланысы мен ADSL – сигналын ажыратып отырады. Арналық бөлу блок ретінде орындалады. Ол 3 ұяшықтан тұрады:

Бірінші – ADSL модемі қосу үшін;

Екінші – телефон құрылғысын қосу үшін;

Үшінші – ADSL линиясына жалғау үшін.

Арналық бөлігі бір линияға компьютерді де, телефонды да қосуға мүмкіншілік береді. Осылайша, бір линияда компьютерлік сигналдар да, аналогтық сигналдар да жіберіле алынады. Мұндай арналық бөлгіш арқылы телефон станциясы абоненттік линиялық келесі жағында да жоғары және төмен арналық сигналдарды ажыратуға болады.

Дыбыстық (аналогтық) сигнал телефон желісіне бағытталады, ал сандық сигнал DS LAM (Digital subscriber Line Access multi plexer) мультиплексеріне бағыттталып, сол жерден провайдер арқылы интернет желісіне жіберіледі.

Қосылу мультиплексоры – барлық DSL абоненттерінің бір жоғары арнаға қосылуын қамтамасыз ететін телефон компаниясының ғимаратындағы механизм.

ADSL – экономды технология, ал тұтынушыға аналогтық қосылуға карағанда арзанырақ түседі.

#### 1.4.2 ISDN – технологиясы.

Бөлінген телефон линиясы – бұл арендаға алынған 2 абонентті қосатын телефон линиясының байланысы. Бөлінген телефон линиясының ең көп тараған технологиясының бір түрі – ISDN( Integrateal Service Digital Network).

ISDN – сандық мәліметтер жіберілудің стандартты кез – келген ISDN – линияның негізгі компоненті – бір бағыттағы bearer – канал немесе В – канал, жіберу жылдамдығы 64 кбит/с. Бұл канал арқылы сандық мәліметтер және сандық формат аударылған аудио және видеомәліметтер жіберіле алады. Арнаны кеңейту үшін В-каналдары екі-екіден тапталып, оған Д-каналы қосылады – мәліметтердің жеткізілуін қадағалайды. Ақпарат жіберілуі қарапайым проводтар арқылы жүзеге асады. Дербес қолданушылар модемі орнына ISDN – адаптерін қондырып, интернетке 128 кбит/с жылдамдықпен қосыла алады. ISDN адаптердің линияны 2 бетінде де қосылуын талап етеді.

ISDN – адаптерін ISDN – модемі деп те атайды. Ескере кететін жағдай, бұл термин нақты емес, өйткені ISDN – адаптерлер модуляция және демодуляция функцияларын орындамайды, ISDN – линиясы сандық болып табылады. ISDN – каналы телефон компаниясымен ұсынылады. ISDN линиясы арқылы бір мезгілде телефонмен өйлесуге де, интернетте отыруға да болады. Әдетте бөлінген канал ұғымы тұрақты интернет линиясы деген мағыналы түсіндіреді. Алдағы уақытта, бұл өз дәл осы ұғымда қолданатын болады.

Төмендегі кестеде интернет технологиясының әр түрлі жылдамдығы көрсетілген.

Интернет қосылудың әр түрлі технологияларды пайдаланғандығы максималды жылдамдығы:

Қосылу технологиясы	Максималды жылдамдығы
Коммутирондық телефон линиясы	56 кбит/с
ISDN	128 кбит/с
Кабельді теледидар	36 кбит/с
ADSL 2	1 Мбит/с (сұраныс) 10 Мбит/с (жауап қайтары)
2,4 ГГц диапазондағы радиоканал	54Мбит/с
Спутник арқылы қосылу0	Жүйеге байланысты бірнеше Мбит/с
T 1	1,544 Мбит/с
T 3	44,74 Мбит/с
GSM - data	9,6 кбит/с
GPRS	160 кбит/с (теориялық максимум)
UMTS	2,048 Мбит/с

### **Бақылау сұрақтары:**

1. DSL – технологиясы арқылы ақпарат жіберудің ерекшелігі неде?
2. ISDN – технологиясы арқылы ақпарат жіберудің ерекшелігі неде?
3. ADSL – технологиясы арқылы ақпарат жіберудің ерекшелігі неде?
4. ISDN және DSL технологияларын қолдану үшін қандай құрылғы қажет?

## **1.5 Интернеттегі web – серверлер**

### **1.5.1 WEB-сервердің жұмыс жасау механизмі**

Браузерде қажетті URL мекен – жайын терген соң, браузер қолданған протокол туралы ақпарат пен сервер атауын алады.

Браузер DNS – серверге сұраныс жасайды, ол үшін сервер атауын IP – мекен жайға алмастыру керек. Алынған IP – мекен - жай арқылы браузер қажетті Web – сервермен HTTP протоколын қолданып, байланысқа түседі де, қажетті ресурсқа сұраныс береді.

Сервер серверде сақталған HTTP – бетті (сақталған) браузерге жібереді. Браузер HTML – элементтерін реттеп, сіздің компьютеріңізге қажетті бетті көрсетеді. Бұл сіз жасаған сұраныстың жауабы болып табылады. Әдетте, қарапайым веб-беттің өзі әртүрлі типтегі бірнеше файлдан тұрады (текст, графика, аудио т.б.) Браузер бұл файлдарды анықтауы үшін, сервер оны қандай файлдер жіберіліп жатқандығы туралы ескертеді, сонан соң ғана файлдар жіберіледі. Веб – бет бірнеше файлдан тұратындығына қарамастан, ол бір сұраныс кезінде тек 1 файлды ғана жібереді. Яғни, HTML – тексттің құрамынан графикалық элементтер сілтемесін кездестірсе, браузер серверге жаңа сұраныс жасайды. Жаңа HTTP – сұраныс жасауы қазіргі серверлер мен браузерлер көп арналы режимде жұмыс істейтіндіктен, қатарынан бірнеше сұраныс жасай алады. Егер сұраныс жасаған файл бұл мекен жай бойынша табылмаса, web – сервер «You / File not found (404) файл табылған жоқ» деген хат жібереді.

### **1.5.2 Статикалық және динамикалық беттер.**

*Статикалық беттер* web – сервер каталогында орналасқан файлдар көшірмесі, ал қолданушының іс-әрекетінсіз өзгеріске түспейді. Бірақ беттер динамикалық өзгерістерге ұшырауы мүмкін, яғни дискіде дайын файлдар емес, сұраныс беттің құрылуы жолының бірнеше әдісі бар.

*Web – серверге сұраныс жасау негізінде құрылу web – беттің* динамикалық түрде құрылуы үшін серверге қандай файл «қарапайым» ал қайсысы программалық өзгертуге түсу керек екендігін көрсету керек. Беттің сервердің өзі (арнайы командалар арқылы) немесе сыртқы программалар құра алады. Динамикалық беттердің құрылуы туралы командалар сақталған программалық текст скрипт д.о.

### ***Қолданушының компьютерде құру.***

Бұл жағдайда динамикалық бетті құру үшін программалық текст әуелі локалды компьютерге жіберіледі, сонан соң браузер қажетті құрылғылармен байланыс жасайды. Динамикалық бетті құрудың бірнеше технологиялары бар.

### **1.5.3 CGI – технологиясы.**

Веб – беттерге динамикалық элементтерді қосуға мүмкіндік беретін технологиялардың бірі CGI (Common Gateway Interface) технологиясы. Ол екі URL арқылы екі программаны салыстыруға мүмкіндік береді. Мысалы, сіз әр жердегі аудандарға дәл осы уақыттағы ауа райын айту үшін сіз жаңа бетті құрасыз. Осы CGI – технологиясы іске қосылады. Ол қажетті арна арқылы ауаның температурасын, қысымын т.б. туралы ақпарат алады әр кез анықтама жайында бұл адреске қосылу арқылы, сіз ағымдағы ақпарат туралы мағлұмат алып отырасыз. Қысқаша айтқанда сіз іздеу жүйесінде қандай да бір ақпаратқа сұраныс жасап, оған жауап елсе, бұл CGI – (технологиясының) программасы функциясының қорытындысы.

CGI – программасын дәл сол мезгілде іске қосылатын web – сервердің бір бөлігі деп қарауға болады.

Сервер қолданушы сұранысы CGI – бағдарлама жібереді, ал ол өз кезегінде қажетті ақпарат туралы мағлұматтар жинақтап, компьютер экранының бетіне шығарады. Қолданушың көзқарасы бойынша қажетті URL мекен – жайы статикалық құжат немесе CGI – бағдарлама екендігінде еш айырмашылық жоқ. Браузер ақпараттарды бірдей қабылдайды.

CGI термині тек бағдарлама емес, сонымен қатар протокол болып та табылады.

Статикалық құжаттар мен CGI – бағдарламаның форматы бірдей болады. Сервер мен приложение арақатынасындағы CGI протокол HTTP протоколының бір бөлігі болып табылады.

CGI – бағдарламасының көп бөлігін CGI – скриптер қамтиды.

Скрипт – басқа бағдарламамен орындалатын ережелер тобы. Perl, Java Script секілді программаларды скриптік, тілдер ретінде ойлап шығарған. Олар кейде сценарийлер тілі деп те атайды. CGI - программалар скриптік тілде жазылады: C, C++, Delphi.

Microsoft компаниясының Active server Page (ASP) технологиясы да CGI альтернативті технологиясының бір түрі. Ол екеуінің құрылылу принциптері бір: Web – бетке қосылған скрипті бет қолданушыға жіберілгенге дейін серверде орындалады. Динамикалық файлдарды тек серверде ғана емес, клиент web – бетінде де құруға болады.

Бұл жағдайда актив құжаттар web – серверде сақталып, статикалық беттер сияқты локалды компьютерге көшіріледі. Бағдарлама локалды компьютерге түскен, өз қызметін орындап, қорытындысын экран бетіне шығарады. Ақпаратты экранға шығару жылдамдығы жойылған сервер жылдамдығына тәуелді емес, өйткені актив құжатты қорытындылау бағдарлама

локалдык компьютерге кұйылғаннан соң іске қосылады. Кұжат кұру бағдарламасын кұйып алу ұзақ уақытқа созылады. Актив кұжаттарды кұру үшін әртүрлі бағдарламаларды қолдануға болады: Ravascript, Java – апплеттер және ActiveX басқару бағдарламасы.

#### **Бақылау сұрақтары:**

1. Web – сервердің жұмыс жасау механизмі?
2. Статикалық және динамикалық беттердің айырмашылығы неде?
3. CGI және ASP серверлік технологияларының айырмашылығы неде?
4. «Скрипт – бағдарлама» ұғымы нені білдіреді?

### **1.6 Web – қорларды алу технологиялары.**

#### **1.6.1 Web – беттерді кэштеу.**

Егер клиент жұмыс істеп отырған сервер ұзақ қашықтықта орналасып, төменарналы жылдамдықкөзі болса немесе «шамадан тыс» режимде жұмыс жасаса, ол кұжаттардың жіберілуін кешіктіреді.

*«Кұжаттарды қабылдау жылдамдығын қалай жеделдетуге болады?»* деген сұрақ туады. Шешімдердің бірі мәліметтерді жойылған серверлерден бірнеше рет алуға болмайды. Web – арналарда кедергі болған жағдайда ақпаратты қайта кұю керек. Сіз браузерде «Артқа (назад)» командасын берген кезде, қараған бетке қайта ораласыз, бұл жағдайда мәліметтерді серверге қайта кұып керек емес. Сіз оның орнына қарған кұжаттарыңызды компьютеріңіздің арнайы буферлік зонасында (кэш) сақтап, қажет жағдайда қайтадан қарасаңызболады. Қазіргі браузерлер дәл осылай істейді. Мысалы: сіз 1-2-3 бетті қарап отырсыз, 3 бетте, 4бетте сілтеме көрсетілген. Бұл жағдайда 4 беттен (желіден кұю) гөрі 2бетке (кэш-тен кұю) еткен тиімді.

Қазіргі браузерлерде кэштелген кұжаттар сақталған папканың көлемін өзгертуге болады. (Internet Explorer, Netscape Navigator)

#### **1.6.2 Прокси-сервердің жұмыс жасау механизмі**

Кэштің технологиясыныңдаөз шектеулері бар. Мысалы: сіз және сіздің коллегаңыз бір интернет желісіне қосылып отырсыз (ортақ провайдер). Екеуңізде бірдей кұжатқа сұраныс жасадыңыз. Ол кұжатты әуелі досыңыз алады, сосын сіз алсаңыз рационалды емес. Оның орнына оны кэштеген тиімді. Бұл жағдайда провайдер бұл тапсырманы прокси-серверге тапсырады.

*Прокси-сервер* – интернетпен жұмыс істеп тұрған станция арасында «елші» қызметін атқаратын және қорғанысты жоғарылатуға, административтік бақылауға, кэштеу функциясын қолдануға мүмкіндік беретін сервер түрі.

*Прокси-сервер* қолданушыдан қандай да бір интернет-сервистің орындауына тапсырыс алады. Мысалы: Web – бетті қарауға. Егер прокси кэштеу функциясын атқарса, бұл бетке сұраныс бар-жоқтығын локалды кэш-серверден қарайды. Егер мұндай бет бар болса, ол интернет желісіне сұраныс

жасамайды. Ал егер бет жоқ болса, ол өз IP – мекен-жайы арқылы клиент атынан интернетке сұраныс жасайды. Сұратылған бет келген кезде, қолданушыға жіберіледі. Прокси-сервер функциясын кәштеу интернет жұмысын жылдамдатады және желілік трафик көлемін төмендетеді. Web – бетті кәштеу желілік қорғаныс қызметінде атқарады, қажетті Web – хаттаманың жойылып кетуден қорғайды.

Әдетте, прокси-сервер жұмыс істеп тұрған программасы бар компьютерде арнайы дисктік аумақ бөлінеді. Ірі провайдерлер кәштеу кезінде құжаттары 10-100 Гбайт жадты қамтиды.

Осылайша, көп сұранысты құжаттар интернет серверінде ғана емес, қолданушының прокси-серверлі провайдерінде немесе қолданушы компьютерде сақталына алынады.

Провайдердің мыңдаған клиенттері көп жағдайда желіден шектеулі құжаттарға сұраныс жасайды екен. Мыңдаған клиенттермен жұмыс жасайтын ірі ISP прокси-сервер өз дискісіне қолданушыларды 50% сұранысың өз дискісінен қанағаттандырады екен әдетте сервер басқа аудандардың интернет провайдерлерімен байланысады. Осылайша, Кәш – қолданушылардың саны арта түседі. Прокси - серверлер тек ірі компанияларда да арнықтылады.

### **1.6.3 Құжаттардың сақталу мерзімі.**

Ескірген файлдарда кәштемеу үшін арнайы ережелер бар. Олар қайсы сұранысты кәштеу керектігін, қайсысын кәштемеу керектігін анықтап отырады. Бұл ережелердің кейбіреуі HTTP протоколында, қалғандары браузерде прокси – сервер администраторында орнатылған.

Прокси- серверде кәштеу шешімі қабылданатын критериелер қатары бар. Мысалы клиент сұраған құжаттың сақталу мерзімі өтіп кетсе құжат кәштелмейді, ол интернет серверінен жаңа құжатқа сұраныс жасайды. Кейбір конфиденциалды құжаттар да олардың сақталу мерзімі шексіз, сондықтан ол кез- келген уақытта кәштеледі.

WEB - жеделткіш.

Стандартты браузерлер кәштеу техникасын web – беттен құю . Internet Explorers қарағанда процессті жеделдететін арнайы бағдарламаларда бар. Жеделдету параметрлерді оптималдау және мәліметтерді кәштеу негізінде жүзеге асуы мүмкін. Кейбір бағдарламаларда жоғарыда көрсетілген механизмдердің бірнешеуі бірден жұмсалады.

### **1.6.4 Кәштеу процессін оптималдау негізінде жеделдету**

Web – беттегі мәліметтерді кәштеу кезінде 3 бағытты көрсетуге болады.

- IP – мекен-жайларды кәштеу;
- Локалды компакттегі дисктік кәштеу;
- Алдын ала оқу.

Әрбір бағытқа жеке тоқталып кетейік. Есіңізде болса браузерге қайсы ресурстың URL айтсаңыз да, ол DNS – сервердің бірнеше тізіміне сұраныс жасау- күнделікті оқиға әдетте IP – мекен-жай (кәштелмейді) қолданушы компактінде кәштелмейді, сондықтан сол минуттан алдын ала сұраныс жасаған IP – мекен-жайға қайтадан сұраныс жасалынады. Бұл да уақытты алады. Осылайша, Web – бетті көрсетуде алдын ала DNS –серверге жедео сұраныс жасайтын бағдарлама – IP – мекен-жайды кәштей бағдарламасын ұйымдастыру керек.

IP – мекен-жайды қабылдаған соң, бірінші шалушы қажет бетті компьютер кәшінен іздейді, егер сақталған болса оның жарамдылығын тексереді, жарамды болса қолданушыға ұсынылады. Бұл процессте уақытты қажет етеді, ал оны жеделдету – бөлек тапсырма. Көпбағдарламалар бұл жағдайда өзінің жеке кәшін құрады. Ол көп орын алғанымен, тезірек жұмыс істейді. Екінші бағыт сипаты осындай.

Ал, 3-ші механизмді түсіндіру үшін қайтадан қарапайым браузер қызметіне тоқталамыз. Құжат браузерде көрсетілген соң, пауза болады. Қолданушы құжатты оқып болғанша, құю процессі тоқтатылады. Алдын ала оқу пауза кезінде байланысты үзбей, құжаттың одан әрі құйылып, көрсетілген бетте сілтеме пайда болуын туғызады. Бұл құжаттарды құйып алу фондық режимде болуы бұл процесстің жедел өтуін қамтамасыз етеді. 3-ші механизмнің сипаты осы алдын ала оқу процессінде немесе префетчинг процессінде (ағылшынның prefetch – алдын ала әкелу) кейбір кезде бұл механизмді алдын ала құю механизмі деп те аталады. (ағылшынның pre-loading).

#### **Бақылау сұрақтары:**

1. Web - бетті кәштеу процессінің қызметі қандай?
2. Прокси-сервер жұмысының механизмі қандай?
3. Web – бетті кәштеу оптималының механизмі қандай?
4. Web – бетті кәштеудің 3 негізгі бағыты қандай?

### **1.7 Интернетте іздеу технологиясы.**

#### **1.7.1 Web-те іздеу**

Интернетте миллиондаған сайттар орналасқан. Онда актуалды ақпараттармен қатар, ескірген ақпарат ресурстары да бар. Интернет – белгілі бір басқаушысы жоқ демократиялық ақпараттар көзі. Әркім желіде өз ресурсын орналастыра алады. Нәтижесінде, ақпаратты дубликаттау және стандартты бұзу көп кездеседі. Сондықтан желіде барлығы бар, бірақ ол жерден қажетті ақпаратты табу оңай емес. Ақпаратты табу үшін іздей білу керек. Төменде интернетте іздеудің әдістері көрсетілген.

Интернетте іздеуге арналған құралдардың бірнеше түрі бар: іздеу машиналары (ізденушілер), индекстендірілген каталогтар (рубрикаторлар), рейтингтер, мета іздеуіш жүйелер және тематикалық тізімдер сілтемесі, онлайндық энциклопедиялар және анықтамалар.



Әр құрылғының өз ерекшеліктері мен кемшіліктері бар. Әрқайсысына жеке тоқталайық.

### **1.7.2 Индекстендірілген каталогтар.**

Каталог – иерархиялық құрылыс ретінде тақырыптары бойынша топтылған мәліметтер жиынын ұсынады. Бірініші деңгейлі топтамалар кең көлемде таралған тақырыптарды қамтиды (спорт, ғылым, магазин т.б). Мұндай әрбір топама бөлімдерге жіктеледі. Осылайша қажет бағытта қозғалып, сіз өзіңізге керек ақпаратты таба аласыз. Мысалы: оқу орнын іздегенде тізім мынадай болуы мүмкін: Білім алу – Оқу орындары – Жоғары оқу орындары – Институттар. Қажетті каталогқа жеткен соң сілтемелер жиынтығын кездестіресіз, әдетте, каталогтағы сілтемелер профильді болады, өйткені оларды программалар емес, адамдар құрған. Егер сіз жалпыландырылған ақы түрін іздесеңіз каталогқа жүгінсеңіз болды, ал нақты ақы қажет болса, каталог – тиімсіз әдіс болып табылады. Жалпы профильді каталогтардан басқа, арнайы каталогтарда болады. Егер жеке тақырып ішінде ресурстар көлемі тым көп болса, қосымша жіктелуді қажет етеді. Мысалы: Яндекс каталогында жіктелу цитирлеу индексі негізінде жүзеге асады. (цитирлеу индексі – желінің басқа сайттарынан сіздің сайтыңызға келіп түскен сілтемелер саны).

Желіде каталогтармен қатар рейтингтер де болады. Екеуінің айырмашылығы: рейтингте – ақпараттар сипаттамасын ақпаратты сұраушы жасайды, ал каталогте олардың авторлары.

Сілтемелердің тематикалық сілтемелері – дара коллекционерлермен немесе мамандандырылған топтармен құрылған тізімдер. Көп жағдайда кішігірім арнайы тақырыпты ірі каталог қызметкерлер топтамасына қарағанда, бір маман тез таба алады.

### **1.7.3 Домендік атау арқылы іздеу.**

*Каталог* – ыңғайлы іздеу жүйесі, бірақ сізге Intel немесе IBM компанияларының серверлеріне түсу қажет болса, сізге каталогқа жүгіну қажет емес. Олардың қажетті сайтына түсу қиын емес: [w.w.w.intel.com](http://w.w.w.intel.com). [w.w.w.IBM.com](http://w.w.w.IBM.com).

Егер сізге дүние жүзінің ауа райы туралы мәлімет керек болса, оны [w.w.w.weather.com](http://w.w.w.weather.com) серверінде іздеген қолайлы көпжағдайда кілттік өз арқылы іздеуге салған тиімдірек. Танымал емес компаниялар сайтыын іздегенде, интуициялық іздеу (компания атауы) қолайлы.

*Іздеу машиналары.*

Сіз сұраған мәліметтер жауабына сізге, тіптен, қажет емес ақпарат тізімдерімен соқтығасыз. Мұндай құжаттар релеванттық емес құжаттар деп аталады (ағылшынның relevant – сай келетін, іске қатысы бар).

Дұрыс сұраныс жасауда релеванттық құжаттар сілтемесінің саны арта түседі.

Іздеу машинасы арқылы табылған релеванттық құжаттар тізімін нақты іздеу деп аталады. Нерелеванттық құжаттарды «шумовые» деп аталады. Барлық табылған құжаттар релевантты болса, іздеу нақтылығы – 100%.

#### **1.7.4 Іздеу машиналарының жұмыс жасау механизмі**

Іздеу жүйесін анықтама қызметімен салыстыруға болады, олардың агенттері мәліметтер барысында ақпарат жинау мақсатында өнеркәсіптерді аралап шығады. Клиент ақпараттар қызметіне сұраныс жасағанда, қажетті құжатты мәліметтер базасынан алып береді. Мәліметтер базасы ескіріп тұрады, сондықтан оны уақыт өте жаңартып жаңартып тұру қажет. Ал кейбір өнеркәсіптер ақпараттарды өздері жіберіп отырады. Осылайша анықтама қызметі 2 функция атқарады:

1. Мәліметтер базасын құру және оны жаңарту;
2. Клиент сұранысы бойынша ақпараттарды іздеу.

Іздеу машинасы 2 бөліктен тұрады:

1. Робот – барлық көрсетілген серверлерді мәліметтер базасын құру мақсатында аралау.
2. Іздеу механизмі.

«Робот» -терминінің синонимдері көп: Желілікагент, ол желіде «өрмелейтінін» ескере отырып «кұрт» немесе «өрмекші» т.с.с.

Робот базасын негізіндеоның өзі құрып отырады (робот өзіне жаңа ресурстарды өзі тауып отырады). Кей жағдайда – ресурстар иесі, өз сайттарын іздеу машиналарында құру арқылы құрады.

Мәліметтер базасын құратын роботпен қатар, табылған сілтемелерді рейтингін анықтайтын бағдарламалар да бар.

Іздеу машиналарының қызметін тұжырымдайтын болсақ, ол қолданушы ұсынған кілттік өз арқылы ішкі каталогын аралып, қажет сілтемелерді көрсетіп отырады.

Айта кететін жағдай, жүйесі ішкі функциясы болып табылады. Іздеу машинасының мәліметтер базасы жаңартылып тұратынына қарамастан, іздеу машинасының ақпараттар жинағы мен желі ресурстарын салыстыруға болмайды. Онда ескірген ақпарат болуы немесе тіптен ақпараты жоқ болуы мүмкін. Проблема тек ішкі ресурстардың шектеулігінде ғана емес, сонымен қатар робот жылдамдығының шектеулігі екендігінде де. Ішкі ресурстар көлемін үлкейту бұл мәселені шешпейді, өйткені аралау жылдамдығы шектеулі. Сонымен қатар іздеу машиналарында интернеттен алынған ақпараттардың барлығы емес, тексанаулы бөлігі ғана сақталады. Оны индекстелген тізім немесе индекс деп аталады.

Индекс құрылымы мынадай ережелер бойынша құрылады.: база көлемі барынша кішкентай, бірақ іздеу жылдамдығы жоғары және ақпарат көлемі барынша үлкен болуы керек. Бұл тізіммен бірге қағазды аналогта болғаны жөн, конкорданс өздігі беп аталады, яғни белгілі бір автормен қолданылған өздердің

индекстері көрсетілген өздер жинағы, сонымен қатар оларға сілтемелер мен арналар да көрсетілген.

### 1.7.5 Индексті құрастыру.

Желілік агенттер немесе робот-өрмекшілер желіде «өрмелеп», Web –бетті тұжырымдап, олар туралы ақпарат жинайды, яғни қайсы ақпарат қайсы беттен табылғаны туралы қажетті HTML – беттерді тапқан іздеу жүйелері өздерді, өзіне керегінше өзгертеді, суреттерді, сілтемелерді, скрипттардыжәне т.б. элементті өздерді өзгерткенде оның беті ғана емес, орналасқан орны да өзгеріске ұшырайды, яғни өздер қай жерде орындалғандығы: тақырыбы (title), тақырыпша (subtitles), метотэгтер (meta tags) және т.б. әдетте негізгі өздер өзгеріске ұшырайды, алшылаулар жайылып кетеді (ал, бірақ, немесе т.б.). Метатэгтер қолданушыға индестелетін беттің кілттік өзі мен тақырыбын өзі таңдауға мүмкіншілік береді. Бұл кілттік өз көп мағыналы болғанда өте тиімді. Метатэг бұл өздің ішінде тек 1дұрыс мағыналы өзді тауып береді. Алайда, метатэгтердің қызметі дұрыс қолданылатын өздер кезінде ғана тиімді. Кейбір жағдайда Web – сайтының иелері өз ресурстарына көбірек қолданушылады тарту мақсатында тематикаға қатысы жоқ танымал өздерді орынсызжерге жазып қояды. Бұл кезде метатэгтің көмегі жоқ. Бұл секілдісайттарды желіден жою жақсы іздеу машиналарының тағы бір қызметі. Әрбір роботтардың мұндай жағдайда қолданылатын өз «жазалау әдісі» бар.

Индексті құрған кезде дубликаттар да көп кездеседі. Ал ең бастысы, тақырыбы әртүрлі, бірақ текстті бірдей дубликаттар мұндай құжаттар желіде өте көп. Мысалы:біреу рефератты көшіріп алып, тақырыбын өзгертеді де, өз сайтына шығарып қояды. Қазіргі іздеу машиналары бұл мәселенің бәрін шеше алады.

### 1.7.6 Индекс бойынша іздеу

Индекс бойынша іздеу кезінде қолданылатын сұранысты өзі құрып, іздеу машинасына жібереді. Бірнеше кілттік өздерді қолданғанда булерлік операторлар негізіндегі бағдарламаларға сұраныс жасаған жөн.

Көп қолданылатын булерлік операторлар:

- AND – «AND» шылауы бар барлық құжаттар көрсетілуі тиіс. Кейбір іздеужүйелері «AND» орнына «+» таңбасын қолданады.
- OR – іздеуге түскен құжатта кемінде бір «or» шылауыболуы керек.
- Not – «Not» кілттік өзі ізделінген құжатта болмауы керек. «Not» орнына кейде « - » таңбасы қолданылады.
- FOLLOWED BY – бір-бірінен кейін жүретін кілттік өздер.
- NEAR – орын ауыстыруға келмейтін кілттік өздер үшін.

Тырнақша («») – тырнақша ішіндегі өздер, құжаттар немесе файлдарда табылатын өздер фрагменті.

Логикалық комбинациялар негізінде тексерілетін текст іздеу бірлігі деп аталады. Мұндай іздеу жүйесі тек индексі анық көрсетілген өйлемдерге ғана тән.

**Бақылау сұрақтары:**

1. Қажетті ақпаратты Интернеттен іздеудің жалпы принциптері?
2. Іздеу машинасының жұмыс жасау механизмі қалай?
3. Индекс түсінігі ақпаратты Интернеттен іздеу технологияларында неменені білдіреді?
4. Индексті құрастыру принциптері қалай?

## Тарау 2. Интернет сервистері

### 2.1 Интернет сервистері, электрондық пошты.

#### 2.1.1 Электрондық почта мекен-жайы.

Интернетке тіркелу кезінде ISP қолданушыға пошталық жәшікке арналған дискілі корын, пошталық жәшік мекен-жайын (E-mail Account Address), қолданушы есімін (E-mail Account Login Name) және қорғау мақсатында пароль (E-mail Account Password) ұсынады. Электрондық пошты мекен-жайының өз форматы болады.: қолданушы\_ есімі @ домен атауы. Мысалы: Sanat @ ntu.almaty.kz. «@» таңбасының сол жағындағы бөлім – пошты атауы, пошталық жәшіктің қай жерде орналасқанын көрсетеді. Қолданушы әр серверде бірнеше электрондық мекен-жайын тіркей алады. Қазіргі таңда электрондық пошты қызметін тегін көрсететін серверлер көп, сондықтан әркім қалауы бойынша дүние жүзінің қалаған жерін электрондық пошта мекен-жайын тіркей алады.

Сондықтан, электрондық пошты мекен-жайы сіздің компаниялардың мекен-жайын емес, сіз хат алатын сервер мекен-жайын көрсетеді. Мысалы: бір қолданушының 2 мекен-жайы бар: **sasha @ ny-server.com.** және **sasha @ almaty-server.ru.**

Әр жәшіктен ол өзінің үйдегі компьютері арқылы хат алып отырады. Біреуі Нью-Йорктегі серверде, екіншісі Алматыдағы серверден келеді.

#### 2.1.2 Екі сервис – екі хаттама

Электрондық пошты клиент-сервер архитектурасы бойынша құрылған. Қолданушы клиент бағдарламасымен әрекеттеседі, ал ол өз кезегінде пошталық сервермен әрекетке түседі. Поштаны қабылдау және жіберу тұлғаны идентификациялауды талап етеді, яғни кәдімгі пошты секілді, сіз өз хатыңызды алу үшін кілт керек. Электрондық поштыда кілттің орнына авторизациялау құралы қолданылады, яғни сіздің поштыны қарауға құқыңыздың бар немесе жоқ екендігін қадағалайды. Ал хат жіберуде авторизация қажет емес. Хатты қабылдау және жіберуге байланысты тұлғаны идентификациялау дәрежесі де 2 түрлі болады. Осыған байланысты 2 протокол болады – хатты жіберуге және хатты қабылдауға. Хат жіберу кезінде авторизация маңызды еместігіне қарамастан, қазіргі таңда провайдерлер, бұл бағдарламаны қорғаныс мақсатында өндіруді талап етеді.

Хаттарды жіберу үшін SMTP (Simple Mail Transfer Protocol – хат жіберу почтасының қарапайым протоколы) протоколдар және SMTP – серверлер қолданылады. Көбінесе, хат жіберу сіздің интернет провайдеріңіздің пошталық сервері арқылы жүзеге асады. Әдетте SMTP – серверлер идентификацияны талап етпейді де, сондықтан хатты мұндай кез-келген сервер арқылы жібере аласыз. Ал хаттарды қабылдау үшін қазіргі таңда POP 3 (Post Office Protocol – пошталық офис протоколы) протоколы қолданылады. Бұл тек нақты қолданушының хатын тасымалдайды, сондықтан авторизацияны талап етеді.

### 2.1.3 Пошталық клиенттердің негізгі қызметтері

Барлық пошталық клиенттердің базалық қызметіне – хаттарды қабылдау, олардың автономды режимде қаралуы, хаттарды реттеу, жауап жазу хаттарын құру және мекен-жай кітапшасын сақтау жатады.

### 2.1.4 Хатты жіберу және жеткізу

Сіз интернетке қосылған соң, клиенттік бағдарлама пошталық сервермен байланысып, оған жіберушінің мекен-жайы мен текстті жібереді. Жіберу кезінде клиент бағдарламасы поштасымен сыртқа пошта байланысып, (SMTP – сервері, SMTP протоколы бойынша) 25 нөмеріндегі портпен әрекеттеседі. Пошта жіберілетін

SMTP – серверінің атауы алдын ала клиенттік бағдарлама поштасына енгізілуі тиіс. Бұл процессті толығырақ қарастырайық. Хат жіберушінің серверіне жеткізілген соң, SMTP – сервер қабылдаушының серверімен байланысуы керек. Оның мекен-жайын білу үшін, сервер DNS – серверімен байланысып хат алушының SMTP – серверінің IP – мекен-жайын сұрайды. Қандайда бір себеппен жіберушінің SMTP – сервері мен қабылдаушының SMTP – сервері байланысқа түсе алмаса, бұл хат кезекке тұрады.

Әдетте әрбір 15 минут сайын хатты қайта жіберу мүмкіншілігі орындалып орырылады. Шамамен 4 сағаттан соң бұл мәселе жайында жіберушіге хат жолданылады да, 5 күннен кейін бұл процесс орындалмаса, хат иесіне қайтарылады. Сондықтан, хат тектен-тек жоғалыпкетпейді.

Пошта серверге келіп түскен соң алушы бұл хатты өз компактсіне құйып алады. Сіздің пошталық жәшігіңізде жиналған хаттарды алу үшін клиенттік бағдарлама POP 3 протоколы бойынша POP 3 серверімен байланысады. Сіз интернетке кірерден алдын пароль мен логиніңді көрсететіндіктен, хат алу кезінде олар қажет емес. Сіздің почтаңыз POP 3 – серверімен байланысып, хатты сіздің компактсіңізге құю туралы бұйрық береді.

Егер клиент поштасында «сақтау» бұйрығы көрсетілмесе, серверде хаттар қабылданғаннан соң жойылады. SMTP – серверде, POP 3- серверде қабылданушының почтасымен «сұхбаттасқан» кезде интуициялық командалар жинағын қолданады; user (IORG – ң FD енгізіңіз), pass ( парольді енгізіңіз, quit POP 3 серверден шығыңыз), dele (хатты жойыңыз) және т.б.

Хатты қабылдау үшін IMAP – протоколы да қолданылуы мүмкін. Келіп түскен хаттарды таңдаулы түрде өз компактсіңізге құя аласыз, папкалармен жұмыс істей аласызжәне т.с.с.

Электронды хатты адресатына жеткізілмей қалуына бірнеше себептер болуы мүмкін.

Мүмкін хат алушы жіберуші серверінен хат қабылдамайды. Сонымен қатар сервер шамадан тыс болған кезде қайтарылып жіберіледі. Егер жіберуші хат жіберіп отырған сервер атауын қате жазса, хат кері қайтарылады және

соңғысы, көрсетілген адресат болғанымен, оның почталық жәшігінде бос орын болмаған жағдайда.

### **2.1.5 Хатқа енгізілген арнайы таңбалар жіберу**

SMTP протоколы ASCII таңбаларының барлығын жіберуге мүмкіндік бермейді, тек ағылшын алфавитіне сай және ASCII кодтау кестесін базасы анықтайтын таңбаларды жібереді. ASCII кодтаудың 2 кестесін анықтайды. 7 биттік – базалық (0-127 нөмерлік таңбалар) және 8 биттік кеңейтілген (128-255)

Кириллица символдары кеңейтілген кестеге жатады. SMTP протоколына арнайы таңбалар болып табылады. Арнайы таңбаларды жібергенде әдетте MIME стандарты (Multipurpose Internet Mail Extensions - интернет стандартының көпбағытты кеңейтілуі) қолданылады, арнайы таңбаларды жіберуге және электрондық поштада жіберілетін тексттік емес ақпаратты жіберуге қолданылады. Мысалы: сіз хатқа енгізген графикалық файлдар әуелде электрондық пошта хат жіберу мақсатында қолданылған, сондықтан файлды жіберу үшін оны тексттік хат аналогына алмастыру керек.

MIME стандарты файл тақырыпшасы арқылы оның қандай типті мәлімет екендігін анықтайды (текст, графика, музыка т.б.).

MIME стандартын қолданғанда кириллицияны кодтау оңай жүзеге асады.

#### **Бақылау сұрақтар:**

1. E-mail-де ISP тіркеу қалай жүзеге асады?
2. Электронды почтаның жіберу-қабылдау қызметі қалай жүзеге асады?
3. E-mail сервисінде қандай протоколдар қолданылады?
4. Арнайы таңбаларды жіберу ерекшелігі қандай?

## **2.2 Файлдар мен алмасу.**

### **2.2.1 FTP (FILE TRANSFER PROTOCOL) файлдарды тасымалдау хаттамасы.**

FTP (file Transfer Protocol – файлдарды жіберу протоколы) 1971 жылы құрылып, осы күнге дейін кеңінен қолданылды. FTP TCP IP базасының протоколы базасына негізделеді және интернет пен компьютер арасында файл алмастыру әдісін ұсынады.

FTP-ның басты қызметі – интернеттегі файлдарды жойылған компьютерді локалды компьютерге (Down land) және керісінше (Upload) жіберуге, көшіруге арналған. FTP- протокол арқылы жойылған компактте өз файлдарыңды кетіруге және өзгертуге болады.

WWW – сервер мен FTP – сервердің айырмашылығы; WWW – тек оқуға арналған ақпараттарды ұсынады, ал FTP – серверлер ақпаратты алумен қатар, серверге жаңа ақпарат қосуға мүмкіндік береді. Көбіне, Web – бетте құрастырылатын файлдарды жіберу үшін FTP – протокол қолданылады. Ақпарат өзгерту үшін бұл ақпарат иесінің келісімі қажет. FTP – серверге

сұраныс жасағанда логин мен пароль қажет, бірақ бұл желіде көпшілікке арналған сервер жоқ деген өз емес.

### 2.2.2 FTP – ресурстар мекен-жайының форматы.

URL FTP – ресурсының форматы төмендегідей:

Ftp: // <user> : <password> @ <host> : <port> / <url – path>, мұнда

- user – қолданушы есімі (логин)
- password – пароль
- host – домендық атау немесе сервердің IP – мекен-жайы.
- url - path – файлға жол.

Әдетте порт қолданылмайды, мұндай жағдайда 21 стандарты қолданылады.

Сервермен байланысқанда қолданушы есімінің орнына anonymous, ал пароль орнына почталық мекен-жайын көрсетуге болады.

Анонимдік FTP үшін URL-де жеңілдетілген синтаксис қолданылады. Ftp: // <host> / <url-path>, яғни атауы көрсетілмеген жағдайда автоматты түрде anonymous қолданылады.

### 2.2.3 FTP – клиенттер

FTP қызметі клиент-серверлік архитектурада құрылған. Клиенттік компьютерде бағдарлама – клиент жіберіледі, ол сервермен байланысып, файлдарды жібереді және қабылдайды. FTP – серверге қосылу әртүрлі FTP – клиенттер көмегімен жүзеге асады:

- консольды клиент - Windows жүйесімен қондырылатын – ftp.exe . бағдарламасы.
- Браузерге енгізілген FTP – клиент ( мысалы, Internet Explorer Netscape Commonder).
- HTML –редакторға енгізілген FTP –клиент (Homesite, Dreamweaver Front Page)
- Мамандандырылған бағдарламалар (мысалы, Cute FTP немесе WSFTP)

### 2.2.4 ftp. exe көмегімен қосылу

ftp.exe – графикалық интерфейсi жоқ консольды клиент. Бұл пайдалануға қолайлы емес. Мұны тек ftp.exe қызметін түсіндіру үшін қолданамыз.

ftp.exe – ті қосу инетерактивті сеанспен байланысады:бағдарлама сізге командалау жолын көрсетіп, сұраныс жасауыңызды күтеді. Диалогты енгізу үшін ондаған командалар түрі бар, солардың кейбіреуі төменде көрсетілген.

- Cd [жойылған -директория] – жойылған машинада каталог құру.
- Mkdir [директория- атасуы] – жойылған машинада каталог құру.
- Get [жойылған- файл] – сіздің компьютеріңізге жойылған файл көшірмесін жіберу.



- reget [жойылған\_файл] [локалды\_файл] – локалды машинаға жартысы құйылған жойылған файлды соңына дейін құю. Команда көлемді файлдарды құю кезінде қолайлы.
- put [локалды\_файл] [жойылған\_файл] – жойылған жүйеге файлды жіберу. Егер жойылған файл атауы көрсетілмесе, ол локалдық жүйедегі файл атауына сай келеді;
- delete [жойылған\_файл] - атаулы файлды жою;
- close – жойылған сервермен FTP-сеансты ұзу және оны командалық интерпретаторға қайтадан әкелу;
- bye — FTP-сервермен жұмысты аяқтау (интерпретатордан шығуға әкеледі).

Толық командалар тізімін және қосымша мәліметтер RFC бағдарламасында алу мүмкін (RFC шешімі Request for Comments (түсінікге сұраныс) – құжаттар сериясы, Internet Engineering Task Force (IETF) ұйымымен басылатын, және әр түрлі мәселелерді қозғайды Интернеттің дамуы және оның технологиялары).

FTP-хаттаманы айтып кету керек интерактивті ұзақ қосылым жасайды, диалог жүргізу мүмкіндігін береді, қосылымды сақтауға, және байланыс бітті деген команда берілмесе жұмыс істей алады.

FTP-хаттама екі түрлі файлдарды түсінеді – мәтінді және екі түсті, басқа мәтінді емес файлдарды екі түсті қылып қабылдайды. FTP-қосылым арқылы жұмыс. Мәтінді хабарлармен ауысу ретінде түсінеді (қызметтік командалармен), және нақты сондай файлдармен екі компьютер арасында жүретін. Консольды жұмыс істеу формасы қолданушыларға онша ыңғайлы емес, бірақ қандай командаларды компьютер орындап отырғандығын жақсы түсіндіреді.

### **Бақылау сұрақтары:**

1. FTP – Интернет сервисінің мақсаты не де?
2. FTP адресінің форматы – Интернет ресурста қандай?
3. Қандай FTP – клиенттер Интернетте қолданылады?
4. Консольды FTP – клиентпен жұмыс жасағанда қандай командалар қолданылады?

## **Тарау 3. Интернетте ақпаратты қорғау.**

### **3.1 Қауіпсіздік мәселелері.**

Мысал қарастырайық: Электрондық поштамен фирма А-ның бастығы фирма В-нің бастығына маңызды құжат жібереді. Коммуникация қауіпсіздігі жағынан қандай мәселелер туындайды? Осы хатты алғаннан соң фирма В-нің бастығында келесідей сұрақтар туындайды. Осы құжатты расында А фирмасының бастығы жіберді екен? (жіберушінің идентификациясы). Бұл хат В фирмасының басқа кім оқыды екен (құпияны сақтау). Белгілі хатты құпия сақтау, тек хабарды шифрлеумен ғана құпиялық сақтау мүмкін. Шифрлеу базасында жоғарыда айтылған үш есептіде шешуге болады, және де осындай сұрақтармен криптография айналысады.

#### **3.1.1 Криптография және интернет**

**Криптография** (грек.тіл Cryptos - құпиялы) – бұл маңызды ақпаратты өзгертуден және авторизацияланбаған рұқсат алудан қорғау және шифрлеу технологиясымен ғылымы. Криптография тек қана мәтінді оқылмайтын шифрленген түрге аудару қызметін емес, және қолданушылардың аутентификациялауының және де идентификациялау желіде жұмыс жасағанда есептерін шешеді.

Криптография – коммуникациялар қауіпсіздігінің негізі болып табылады. Себебі, желіле біз тек қана адамдармен қарым-қатынаст жасау емес, және де әр түрлі қызметтермен жұмыс жасаймыз. Мысалы: қандай да бір серверден бағдарламаны көшіріп алатын болсақ, бізге де маңызды білу керек бұл сервер жасаған фирманы екендігін, ия пират-фирманыкі екендігін, себебі пират – фирмасы біздің компьютерге вирус жұқтыруы мүмкін. Жолдауды жеке және бұқаралық кілтімен шифрлеуге болады, ал бұл жолдауды тек қана қос кілттердің екінші кілтімен ғана ашып алуға болады. Яғни жеке кілтпен шифрленген жолдауды бұқаралық кілтпен ашуға болады және де керісінше. Жеке кілт тек қана иеленушіге белгілі және де оны ешкімге беруге болмайды, ал бұқаралық кілт барлық корреспонденттерге ашық таратылады.

#### **3.1.2 Симметриялық және асимметриялық кілттер.**

*Қос кілттерді – жеке және бұқаралық – аутентификациялау және құпиялық есептерін шешу үшін қолдануға болады.*

Екі тұлғаның бір-бірімен құпиялы хабарландырулар алмасуы үшін міндетті түрде екі кілт болуы қажет.

Қос кілттердің көмегімен шифрленген кезде, барлық корреспонденттерге өзіндік бұқаралық мкілтіңізді жіберу міндет емес. Сізге ыңғайлы ашық рұқсатпен сол кілтті желіде сервермен орналастырған. Сонда сол кілтті кез-келген адам көшіріп алып сізге құпиялық хабар жібере алады, бірақ сол хабарды сізден басқа ешкім де оқып алмайды.

### ***Симметриялық және ассиметриялық кілтпен шифрлеу.***

Ассиметриялық шифрлеу алгоритмі тиімділік жағынан симметриялық шифрлеу алгоритіміне қатты ұмтылады, себебі көптеген бар шифрлеу жүйелер ассиметриялық және дәстүрлі симметриялық шифрлеу жүйелерінің түзбелерін қалданады, берілген ақпаратты шифрлеу үшін қолданады.

### **3.1.3 Цифрлік қолтанба**

Цифрлық қол қоюдың механизмін тусіндіру үшін бір жақты хеш-функция түсінігін кіргізу керек.

*Бір жақты хеш-функция*, әр түрлі ұзындығы бар басылған жолдауды, шегі бар символдар ұзындығына және жолға өзгертеді, былай айтылғанда жолдаудың дайджесті. 16-байттық хеш-функцияны қолданғанда, шығында сіз 16-байты ұзындығы бар жазылу аласыз.

*Хеширлеу - бұл бір жақты, қайтпайтын, шифрлеу мен салыстырғанда, өзгеріс.*

Жолдаудың дайджесті бойынша бірінші кезіндегі хабарландыруға өзгерту мүмкіндік жоқ, бірақ оны идентификациялауға болады.

Мысалы А хеш-функция қолданып, код түрінде дайджест алды деп айтайық (**код1**). Содан соң өзінің жеке кодын қолданып А дайджесті шифрлейді, бұл құжаттың қол қою аналогы болып қалады. Содан соң А ашық мәтіннің жолдауын және цифрлық қолын қоюды В-ға жібереді. Абонент В бұқаралақ кілтпен қол қоюды шифрлеп ашады, және өзі дәлелденеді хатты А абонентінен алғандығына. Сонымен аутентификация жасалады. Содан соң, абонент В бұл жолдау жеткізілуі процессінде өзгертілмегендігіне куәлану керек. Шифрлеп ашу бағытында абонент В жолдаудың дайджестін алады - **код1**. Содан соң абонент В, хеш-функция қолданып отырып, А абоненті қолданған, жолдаудың мәтінін хештейді де, бір түрлі **код 2** түріндегі дайджесті алады. **Код1** және **код 2** сәйкес болса В абоненті жолдаудың мәтіні жеткізу этапында өзгерілмегендігі екенінде дәлелденеді.

*Цифрлық қол қою немесе электронды қол қою* - бұл жіберушінің аутентификациялайтын немесе қол қоюдың авторы, құжаттың мазмұны өзгермегендігі туралы дәлелдейтін әдіс. Цифрлық қол қою шифрленген түрде, және жолдаудада қойла алады.

### **Бақылау сұрақтары:**

1. Ақпаратты Интернетке өткізгенде қандай ақпаратты қорғау механизмдері қолданылады?
2. Жіберушінің идентификациясы, жолдаудың аутентификациясы, құпияны сақтау түсініктердің мағынасы неде?
  1. Симметриялық және ассиметриялық шифрлеу түсініктердің мағынасы қалай?
  4. Цифрлық қол қоюдың механизмінің қолдану алгоритмі қандай?

## 3.2 Интернетте қолданушыларды идентификациялау

### 3.2.1 Cookies механизмі

Cookies – үлкен емес мәтінді жазулар, сервердің сұрауы бойынша қолданушының компьютерінде браузер арқылы сақталады.

Бұл мәтіндік ақпарат қолданушының компьютерінде жойылғанына дейін сақталады. Cookies қолданушысының Web серверге хабарласқан тарихы сақталады және сол хабарды көптеген серверлер жібереді. Бұл сайтқа хабарласқандарды анық тексеру мүмкіндігін береді, және сайт қолданушыларға және олардың мұрагерлеріне қосымша сервис береді. Мысалы: сайт мұрагерлеріне қанша адам сайтты қарады, кім сайтқа бірінші рет келді, кім қайталап келді екенін білу үшін маңызды. Бұл есеп Cookies технология арқылы өте оңай шешіледі. Web сайттың бағдарламалық қамтамасыз етуі әрбір сайтқа хабарласқан тұлғаның өзіндік номерін көшіріп қолданушыға жібереді. Ол қолданушының компьютерінде сақталып тұрады. Қайталап хабарласқанда хабарласқан мәліметер оқылып, Web сайт хабарласқан адамдарды таниды.

Келешекте осы бетке хабарласқанда сіздің Cookies-тегі ақпарат өзгеріп тұрады.

Соның ішінде атап кету керек, әртүрлі Cookies-ті сервер әртүрлі Web сайттың бөлшектерімен салыстырады. Cookies серверлер түйіннің жұмыс жасау қабілетті дайындайды. Келесіде қолданушыны жеке қабылдайды.

Cookies технологиясы арқылы жұмыстану қолданушыны есіне сақтау оның файлынан атын көшіру және жеке бұраулар танийды. Күнделікті және уақытша Cookies файлдары болады.

Cookies уақытша файлдар қолданушылардың компьютерінде браузердің сеансы уақыттықта сақталады, ал күнделікті файлдар қолданушының компьютерінде сақталады. Cookies технологиясы көмегімен сайттың қожайыны қолданушының сайтта жүрісін байқайды. Оның ерекшеліктерін және сұраныстарын байқайды және оны жарнамалық мақсаттарда пайдаланады. Cookies-те қолданушы туралы оның анықтамалық жеке қолданылатын бұрауларын білуге болады. Ол тұындыға хабарласқанның және жеке идентификациялар мәліметтер білуге болады. Өзі толтырған регистрациялық түрлерінде. Атап кету керек, Cookies тек қолданушының өзі серверге жіберген мәліметтерді сақтайды. Cookies мәліметтерін сақтау үшін браузер қолданушының компьютерінде анықтамалы мәліметтер базасын жүргізу керек.

MS Internet Explorer мен қолдансаңыз, онда сіздің компьютеріңізде ол мәліметтер базасы файлдар папкасымен көрсетіліп тұрады, аттары қолданумен және серверден тұрады. Мысалы: файл `sanata@spylog[2].txt` осындай түрі болады:

VISIDSVWDOBAOFOTIYBYDIPSPYLog.com/10241362503808295671022  
16316196829530892.

Қызметтік ақпарат. Жалпы браузер Cookies максималды сыймдылықтарына тиім салынады, жалпы Cookies саны 500 ден аспайды және Cookies максималды саны бір серверде сақтау үшін 20 ға жақын.

Cookies жалпы сыймдылығы онша үлкен емес және мәліметтердің үлкен көлемін сақтамайды. Жалпы қолданушы сайтқа хабарласқан, қандай жарнама оған көрсетілді, сұраныс қылған тауарлырдың тізімі серверде сақталады. Ал Cookies сол ақпаратты іздеу үшін қолданылады. Cookies технологиясы персоналды беттерді жасау мүмкіндігін береді.

Электронды коммерциялар сайты Cookies-ті сатып алушы қапшығын құру үшін пайдаланады. Сіздің қапшығыңызға қосылған барлық тауарлар туралы ақпарат. Сіз шот алғыңыз келгенде сіз ұсынған тауарлардың барлығы белгілі, сондықтан серверге мәліметтер базасынан, олардың тұратын бағалары туралы ақпарат алып, сізге шот қою қиын емес.

Сіздің компьютеріңіздің жұмыс жасау қабілеттілігіне байланысты Cookies-ті жою қауіпсіз, бірақ есте сақтау керек. Сіз жиі хабарласатын сайттың Cookies-ін жасаңыз, сол Cookies-пен бірге сіздің бұрауларыңыз жоғалады. Бірнеше адамдар қандай да бір сайтқа бір жеке компьютермен хабарласып тұрса, сол хабарласқан Web сервер әртүрлі келушілерді бір тұлға деп шатысып, олардың жеке бұрауларында шатастырады. Сіз интернетке қосылуға әртүрлі компьютерлер қолдансаңыз.

Сайтта тіркелгендер жоғарыда айтылған мәселелерден құтылуға болады. Осындай жағдайда сервер сіздің Cookies-пен бірге сіздің логинмен пароль сақталады. Сондықтан әртүрлі компьютерден хабарласып рұқсат алыңыз. Сізге дұрыс ақпарат беріліп отырады.

Cookies мәтінді түрде сақталатын болғандықтан қолданушы оны жойып немесе өзгертіп жібере алады. Сондықтан Web толықтырушы жұмыстары Cookies-те маңызды ақпарат сақталмайды және Web сервердің жұмысын қолданушының дискісіне тыйым салынады. Cookies-ті орналастырып, жұмыс жасайтын мүмкіндігі бар. Дискіде Cookies-тің бары қолданушының қай сайтқа хабарласқандығын қорытындылауға мұрша береді және қолданушының анықты сайтқа жіберілген жеке мәліметтерді білуге болады.

### **Бақылау сұрақтары:**

1. Cookies термині интернетте нені білдіреді?
2. Cookies жұмыс жасау механизмі қалай?
3. Cookies-ті қолданып, қандай есептер шешіледі?
4. Cookies технологиясын қолданғанда қандай ерекшеліктер бар?

## Тарау 4. Web-қосымшалар құру технологиясы

### 4.1. Интернет-қосымшаларын құру

#### 4.1.1 HTML тіліне жалпы түсінік

Интернетке түрлі қызметтердің жиынтығы әсер етеді. Солардың бірі ең көп тараған қызметтер қатарына Web (World Wide Web(бүкіләлемдік өрмекші)) енеді. Web – тораптардағы құжаттардың негізгі форматы HTML болып табылады. HTML (Hyper Text Maker Language) ағылшын тілінен аударғанда гипермәтінді белгілеу тілі деген мағынаны білдіреді.

*Гипермәтін* – бұл қосымша элементтерді басқару мақсатында ішіне екпінді элемент орналасқан мәтін түрі.

#### HTML тілінің құрылымы

<HTML>

<HEAD> <TITLE> құжат тақырыбы </TITLE> </HEAD>

<BODY>

құжат тұлғасы

</BODY>

</HTML>

Көп жағдайларда HTML документтердің тақырыптары болады. Олар <Hn></Hn> тегтер арқылы жасалады, *n*–нің орнына 1-ден 6-ға дейін сан жазамыз. Осы тегтер ортасына текстімізді орналастырсақ, сізге белгілі көлемді тақырып шығады.

<H1>Тақырып</H1>

<H2>Тақырып</H2>

<H3>Тақырып</H3>

<H4>Тақырып</H4>

<H5>Тақырып</H5>

<H6>Тақырып</H6>

Ал енді жаңа абзацқа шығару үшін <P>, ал жаңа жолға шығу үшін <BR> тәгі жазылады. Бұл тәгтерді жабу керек емес. Әрине, егер Сіз <P> тәгінде ALIGN абзацты түзеу элементін қолданбасаңыздар:

<P ALIGN=LEFT>Сол жақ шетке түзеу</P>

<P ALIGN=CENTER>Ортаға келтіру</P>

<P ALIGN=RIGHT>Оң жақ шетке түзеу</P>

Ал мәтінге қалың, курсивті, сызықты және т.с.с. түр беру үшін, оны сай келетін тәгтер ортасына жазу керек:

<B> Қалың мәтін </B>

<I> Курсивті </I>

<U> Асты сызылған </U>

<STRIKE>Сызылып тастаған</STRIKE>

<SUP> Жоғарғы индекс </SUP>

<SUB> Төменгі индекс </SUB>

Кейбір тегтер белгілі бір параметрлермен қолданыла алады немесе қолдану тиіс, олар ашылған тэг элементтерінде жазылады (бір тэге бірнеше параметр беруге болады). Мысалы, ашылып жатқан <FONT> тэгінің (</FONT> жабылатын тег жазылу тиіс) бірнеше атрибуттары болуы мүмкін:

**SIZE** – шрифттің өлшемін береді (стандартты шрифт өлшемі –«3»). Текстті <FONT SIZE=**n**></FONT> тэгтерінің ортасына қойсақ (бұл жердегі **n** – цифр)

Сіз оған керекті өлшем бересіз:

<font size=»1»>Мысал 1</font>

<font size=»2»> Мысал 2 </font>

<font size=»3»> Мысал 3 </font>

<font size=»4»> Мысал 4 </font>

<font size=»5»> Мысал 5 </font>

<font size=»6»> Мысал 6 </font>

<font COLOR="white"> Красный </font>

<font COLOR="FF0000"> Қызыл </font>

Сонымен түс туралы басынан айтып кетейік. <BODY> тэгінің ішіне **BGCOLOR** параметрін жазып және оған түс атын немесе оның 6 орынды кодын беру керек. Берілген екі мысал бетті қызыл түспен бояйды.

<BODY BGCOLOR=»RED»> (түстің аты қолданылған)

<BODY BGCOLOR=»FF0000»> (түс 6 орынды кодпен берілген)

HTML документіне сурет салу үшін келесі құрылымды жасау керек. (толық көрсетілген):

<IMG SRC=»Сурет аты» BORDER=»0» ALIGN=» түзеу» WIDTH=»Ені» HEIGHT=»Биіктігі» HSPACE=» 1-орын қалу» VSPACE=»2-орын қалу» ALT=»сұқпат»>

### Кестелер

Кестелерді қолдана отырып, бірнеше бағана құруға және суреттің фон мен үйлесуін сәндірек етуге болады т.с.с.

Кесте <TABLE> ашылатын тегімен басталып </TABLE> тегімен аяқталады. <TABLE> тегінде мынадай атрибуттары болуы мүмкін:

Кестенің құрлымын біліп алу үшін, қарапайым мысалды алып қарайық:

1 ұяшық	2 ұяшық
3 ұяшық	4 ұяшық

Мұндай кесте былай жасалады:

<TABLE BORDER=»2» WIDTH=»200»

BGCOLOR=»#00FF00»>

<TR>

<TD>1 ұяшық</TD>

<TD>2 ұяшық</TD>

</TR>

```

<TR>
<TD> 3 ұяшық</TD>
<TD> 4 ұяшық</TD>
</TR>
</TABLE>

```

Ескеру: Берілген мысалда кестенің ені қолдан берілген (WIDTH=»200» пиксель), бірақ пайызбен берген ыңғайлы, себебі бұл жағдайда терезенің көлеміне қарай өзгеріп отырады.

Кестеде тақырып орналасуы мүмкін, ол <CAPTION></CAPTION> тегтерінің ортасында орналасады. Ол міндетті түрде <TITLE> тегінен кейін орналасу керек. Тақырыпқа ALIGN атрибутын қолдануға болады, ол кестеде орналасуын анықтайды:

**TOP** – тақырып кестенің жоғарғы жағының ортасына жазылады.

**LEFT** – тақырып кестенің жоғарғы жағының сол жақ шетіне жазылады.

**RIGHT** – тақырып кестенің жоғарғы жағының оң жақ шетіне жазылады.

**BOTTOM** – тақырып кестенің төменгі жағының ортасына жазылады.

Ал енді кестенің жолдары мен ұяшықтары туралы. Кестенің жолдары <TR> тегімен басталады және </TR> тегімен жабылады, ал әрбір ұяшық <TD> тегімен басталып </TD> тегімен аяқталады.

**Тізімдер құру.** Бұл жағдайда тізімнің әрбір элементтің алды автоматты түрде нөмірленіп шығады. Нөмірленген тізім <OL> тегінен басатылып, </OL> тегімен аяқталады. Мысалы:

<OL>		
<LI>	Бір	Бір
<LI>	Екі	Екі
<LI>	үш	үш
<LI>	Төрт	Төрт
<LI>	Бес	Бес
</OL>		

<OL> тегінің мынадай параметрлері болуы мүмкін:

**TYPE** – санау түрі:

**A** – үлкен латын әріптері

**a** – кіші латын әріптері

**I** – үлкен рим цифрлары

**i** – кіші рим цифрлары

**1** – кәдімгі цифрлар

Келесі мысалда тізім үлкен рим санының сегізінші цифрінен басталып нөмірленген:

<OL	TYPE=»I»	Сегіз
START=»8»>		Тоғыз
<LI>	Сегіз	Он
<LI>	Тоғыз	Он бір



<LI>		Он	Он екі
<LI>	Он	бір	
<LI>	Он	екі	
</OL>			

### Нөмерленбеген тізім:

Номерленбеген тізім үшін әдетте тізімдегі элементті белгілеуге маркерді қолданады. Ал маркердің түрін, заң бойынша, қолданушы таңдайды.

Номерленбеген тізім <UL> тегімен ашылып, </UL>тегімен жабылады. Тізімдегі әрбір элемент <LI> тегімен басталады. Мысалы:

<UL>		
<LI>	Бір	Бір
<LI>	Екі	Екі
<LI>	үш	үш
<LI>	Төрт	Төрт
<LI>	Бес	Бес
</UL>		

<UL> тегі өзіне **TYPE** параметрін қоса алады. Оның **disc**, **circle**, **square** деген белгілері бар.

<UL TYPE=disc><LI> disc </UL>	disc
<UL TYPE=circle><LI> circle </UL>	circle
<UL TYPE=square><LI> square </UL>	square

### 4.1.2 Динамикалық HTML

Internet Explorer және Netscape Navigator нұсқалары пайда болғанға дейін құжаттарды өзгерту сыртқы бетте ғана write, document арқылы мүмкін еді. Содан соң орындалатын сценарийлер браузерге жаңа бетті құю, алдын-ала ескерту терезесін құю т.б. секілді командаларға сұраныс жасауға болады, бірақ бет компьютерге құйылған соң, HTML-код өзгертілмейді. Егер қолданушының сұранысына сай келмесе, Web-беттерді қалай интерактивтеуге болады. Netscape-тің 4 бағытымен және Internet Explorerдің 4 бағытымен құрылған динамикалық HTML бұл кедергілерін жоғалтады. Динамикалық HTML сценарийі арқылы HTML блоктарды енгізу, оларды жоюға немесе өзгертуге немесе объектілердің қасиеттерін өзгертуге болады. Браузер жаңа қасиеттерді және жаңа HTML-кодты автоматты түрде жаңартады.

Динамикалық HTML 2 принциптен құрылады:

- Браузердегі объектінің атауын немесе ұйымдастыру әдісін сипаттайтын құжаттың объекті моделі.
- Сценарийдік жіберу қасиетін қодағалайтын әдісін сипаттайтын модель.

World Wide Web жедел түрде дамып келеді. Дәстүлі Web- сайттар бұрынғыша статистикалық ақпарат беттерімен қолданушылар арасында көп сұранысқа түспейді. Сондық Web дизайнерлер өз беттеріне ерекше сипат аударуға тытысады. Ол үшін әртүрлі графика элементтері мен 6 форматтағы анимациондық файлдарды қолданады. 6 tf.jpg 6 pn бдәстүрлі форматтар арқылы суретте өзгерту сайтқа көрнекті болғанымен құю кезінде көп уақытты алады.

#### **4.1.3 Macromedia Flash бағдарламасындағы Action script тілінің қызыметі**

Анимацияны құрудың альтернативті құралы- macromedia flash продуктысы, Web- өзгертуге алға қойған мақсатқа жетуге болады басқа профикалық приложениялармен салыстырғанда өзгертіледі.

- Негізінде математикалық формулалар тізбегі арқылы құжат көлемін түсін, орналасуға формасын анықтайтын векторлық графиканы қолдану. Векторлық графика – түзулерден тұрады, расторлық графика – пиксельдерден тұрады.

Web үшін, көлемі кішкентай файлдарды сақтау және графиканы көрсету математикалық формулаларды қолдану өте маңызды болып келеді.

- Дыбыстарды графиканы анимацияны қысу мүмкүндігі. Мұндай процесс арқылы Web – бетте көрсетілуін, ал Web- браузер бір графиканы сайт көрсетіп тұрады, ол екінші графиканы және анимацияны құюға болады.

Flash-ге интерактивтеу әртүрлі бөлімдерден тұруы мүмкін.

Қарапайым графиканы шексіз қолдану графиканы, анимацианы, ірі тақырыптарды, кез- келген объектіні өзгертіп сайтқашығарған соң навигациялық егізуге болды, кнопкаларды, мәзір анимация бөлшектері және т.б. Сонан соң интерактивті фильмдер құруға болады. Осы жерден интерактивті басқару элементері FLASH- ға ACTION SCRIPT –тілді құру сценарий арқылы жүзеге асады.

FLASH- тен веторлық графика нәтижесінде құрылған суреттін және текісттін сапасы жоғары болады. FLESH – қолданушының сурет салуына мүмкүндік береді. Негізгі, FLASH құру процессі кезінде геометриялық формаларды анықтай алады. Сонымен қатар ол сурет салу барысында қолданушының жасаған қателерін векторлық формалар арқылы түзейді. FLASH тіке және толқынды түзеулер құруға да көмектеседі. FLASH құрамына сапалы FLASH –фильмдерді құрауғанегіз болатын құралдар кіреді. Бұл құрал - ACTION SCRIPT FLASH-ге ACTION SCRIPT –сіздің бетіңізді интерактивті қылады. Сіз белгілі бір кадр жүріп жатқан кезде, бұл программамен тышқан немесе клавиатура арқылы әрекеттесе аласыз. ACTION SCRIPT бағдарламасын үйрену үшін бұрыннан да бағдарлаумен жұмыс істеуді білген жөн-(C++Java script). Flash тілінің бір ерекшелігі сіз арнайы мамандарылған болмасаңыздар FLASH-те ACTION SCRIPT сапалы код жаза аласың, тек ACTION SCRIPT бғдарламасын толық меңгерсеңіз болғаны. Сіз тек өзіңізге қажет бөлігімен ғана жұмыс істей аласыз.

### Бақылау сұрақтары:

1. Dynamic HTML құрастырудың негізгі принциптері қандай?
2. Инт-те FLESH- технологиясын қолданудың ерекшелігі неде?
3. FLESH-тегі ACTION SCRIPT тілінің қызыметі қандай?

### Практикалық жұмыс № 1

**Тақырып:** HTML- құжаттарды құру негізі (фреймдер, тізімдер, кестелер)

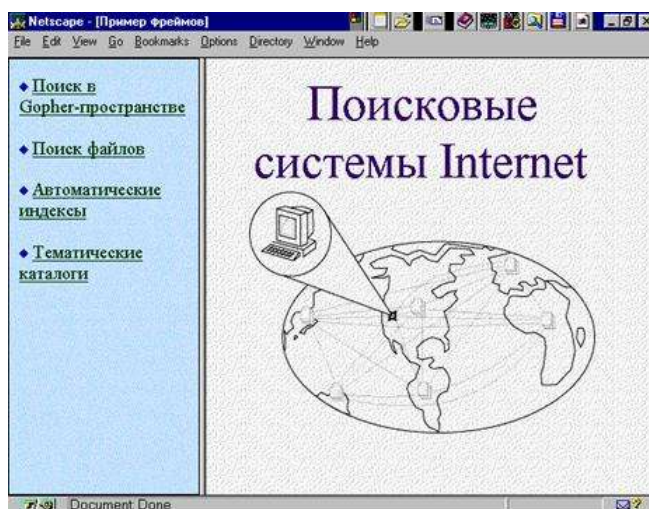
**Мақсаты:** Кестелер мен тізімдерді қолдану арқылы және фреймдер көмегімен HTML- құжаттарды құру

#### Теориялық мағлұматтар:

HTML- құжаттарды құру үшін 2 тэг қолданылады: <frameset> және <frame>.

<frameset> тәгі кадрдегі жолдар мен бағандардың бастапқы саны мен көлемін енгізуге мүлік береді. Ал экранды көлденеңінен бөлу «row» атрибуты арқылы орындалады. Атрибуттер түрінде көрсетіледі. Ал тігінен бөлу «cols» атрибуты арқылы орындалады. Атрибуттер саны пайыз немесе пиксельдер түрінде көрсетіледі.

«Frameset» тегінің ішінде күрделі беттер құрамына кіретін құжаттардың барлығына сілтемелер болуы керек. Бұл < Frame > тегінің көмегімен орындалады. Web- беттерге арналған екі вертикальды фрейм ұсынылған.



Сол жақтағы кадрда тақырып атауы, ал оң жақтағысында фрейманың құрамындағы барлық сілтемелер көрсетіледі. Сол жақтағы кадрға Frame 1, htm енгізіледі.

</ HtmL>

Төменде құжаттардың бастапқы коды көрсетілген:

< HtmL>

< head>

< title>

< /head>

< frameset cols= 1"30 %, \*">

< Frame name= toc "src= " Frame 2 htm">

```
< Frame name= "main "src=" Frame 1 htm">
</ frameset >
</Html>
```

"Frame" тегіндегі "name" атрибутындағы кадрді меншіктену атауы көрсетіледі. Ол еренсілтеме құрамын көрсету үшін қолданылатын бағытталған кадрды нұсқау кезінде қажет етіледі. Еренсілтеме арқылы алынған құжат көрсетілген кадрға құйылуы үшін енгізіледі.

```
< Ahref = "Url- құжат main target= " кадр атауы">
```

Төменде Frame 2 htm құжатының бастапқы жеңілдетілген коды ұсынылған; мұнда барлық құжаттарды сілтемелер көмегімен шығаруға "main " атаулы кадр қолданылады.

```
< Html>
< head>
< title> файл Frame 2 htm <Title>
< /head>
<Bodi>
<A HREF = "file. Htm "Target ="main" >файлды іздеу </ A>
< A HREF= index. Htm " Target ="main" > Автоматты индекстер
< A HREF " tema Htm " Target ="main" >матикалық каталогтар
```

### **Практикалық тапсырмалар:**

1. Желі ресурстарында сілтемелер бетін құру, міндетті түрде фреймаларды қолдану арқылы. Файл атаулары кішкентай латын әріптерінен тұруы қажет.

Сол жақтағы фреймде рубрика құру, құрылған сілтемені тізім ретінде тіркеу ( Тег <Uh>)

- іздеу жүйелері;
- каталогтар;
- әдебиеттер;

Оң жақтағы фреймде атаулы сілтемелерді құру және ресурстарды қысқаша сипаттау:

- іздеу жүйелері үшін, сілтемелерді 2 бағанды және бірнеше жолды кестелер түрінде құру. Сол жақтағы бағанда іздеу жүйелерінің сурет-логатиптерін орналастыру, оң жағында – осы жүйеге еренсілтемелер құру. (сілтемелер w.w.w. yandex.ru , w.w.w. rambler.ru, w.w.w. aport.ru, w.w.w. google.com, file-search.ru).

Мысал:



2. Кестенің мысалы:

- каталогтар үшін құрылған сілтемелерді нөмірленген тізімдер түрінде тіркеу (w.w.w. list.ru, w.w.w. rambler.ru).

- әдебиеттер үшін, құрылған сілтемелерді нөмірленген тізімдер түрінде тіркеу, сілтемелерді рубрикалар бойынша орналастыру; Мысалы: кітапханалар, кітапханалар, техникалық әдебиет, стандарттар ([w.w.w. citforum.ru](http://w.w.w.citforum.ru), [w.w.w. lib.ru](http://w.w.w.lib.ru), [w.w.w. W3.org](http://w.w.w.W3.org), [http:// pyramidin. Nazod.ru/html//index.htm](http://pyramidin.Nazod.ru/html//index.htm)).

Негізгі беттен осы бетке сілтеме құру.

6. Файлдарда серверде орналастыру және оны браузерде тексеру, барлық сілтемелердің жұмыс істеу қабілеттеілігін тексеру.

7. Әрбір тектер тобы қандай мақсатта қолданылатындығы комментарий ретінде беттің бір шетінде көрсетіңіз.

## **Практикалық жұмыс № 2.**

**Тақырыбы: Құжат «шапкасы», META-мәліметтер, HTML-құжаттардағы суреттер белгісі мен картасы.**

**Жұмыс мақсаты:** HTML - құжаттарды белгілер (метки) көмегімен құру, сурет картасын

құру және де META – мәліметтер мен құжат «шапкасын» қолдану.

**Теориялық мағлұматтар.**

Құжат «шапкасы» мен META – мәліметтер </head> тегтерімен шектеледі. META – ақпараттарды құрайды, яғни бет жөнінде қосымша мәліметтер, беттер атауы, браузер терезесіне шығарылатан тақырыптар және сілтемелер. Тақырыпша </title> және </title> тегтерімен анықталады, бұл міндетті емес, бірақ бағаны жақсы <head> және <title> тегтерінде параметрлер болмайды.

Егер еренсілтемелік құжаттар авторы өз шығармасын оқитын қауымға жеткізгісі келсе, өзінің жұмысын, каталогтар мен іздеу жүйелерінде әрі ерекше , әрі тиімді етіп тіркеуі қажет.

Бұл үшін HTML тілінің командалар жинағын арнайы META инструкциялар тобы қондырылған, ал негізінен іздеу машиналары арқылы құжатты сипаттауға және индекстеуге арналған.

Өзіңіздің сайтыңызды іздеу жүйелері мен каталогтарда тіркеуден алдын дайындау қажет. Болмаса қолданылған күш пен уақыт нәтижесі төмен болады. Берілген мысалдарда біз META-тегтерді қарастырамыз. Олардың рөлі беттерді көрсетуде білінбейді. Бұл командалар Web-серверлер мен браузерге арналған, бірақ маңызды командалар, өйткені META-тегтерді дұрыс қолдану сізге өз артықшылығын береді.

Іздеу жүйелері беттерді тіркеу кезінде META дискрипторларда сақталған ақпараттарда қолданады, олар сіздің сайтыңыздағы <head> немесе </head> тегтерінің арасында орындалады. META барлық іздеу жүйелерінде қолданылмағанымен, оларсыз жұмыс істеу мүмкін емес. META элементтері құжаттардың кейбір қасиеттерін сипаттайтын ауыспалы мағынаны анықтайды: автор жөнінде ақпарат, кілттік өздер тізімі т.б. Name атрибут – ауыспалы, ал content – оның мағыналықұғымы. Есте сақтаңыз, барлық META – тегтер визуалды ақпаратты тасымайды және тек <head> </head> құжаттар акциясында орналасуы қажет.

META – жинағының барлығын қолдану міндет емес, бірақ базаның болуы қажетті. Оларға Title, Keywords, Description, Charset, Author- ді жатқызуға болады.

### **Title – құжат тақырыбы.**

Дұрыс таңдалған тақырып сіздің сілтемеңізге көптеген қолданушыларды еліктіреді. Мысалы: <TITLE> мұнда құжат тақырыбын жазсаңыз </TITLE>.

### **Description – сипаттау.**

Сіздің сайтыңыздың артықшылығын нақты сипаттаудан артық жарнама жоқ. Description сұранысқа жауап беру кезінде сіздің сайтыңызды сипаттауға жауап береді.

Қысқаша сипаттауға мына нұсқаулар қолданылады:

<meta name =” description “ content = “ бет сипаттамасы “>.

Егер сипаттауды орыс тілінде жүргізгіңіз келсе, «lang» атрибутын көрсету керек.

<meta name =” description” lang = “ru” content =” бет сипаттамасы “ >.

“Description” тегіне сіздің сайтыңызда көп қолданылатын барлық терминдерді, өздерді, олардың синонимдерін енгізу керек.

### **Keywords – іздеу өздер мен өйлемдері.**

Өзіңіздің кілттік өзіңізді пайдаланыңыз, олар маңызды. Әртүрлі өздерді және олардың өзгертулерін, төменгі регистрді (әдетте, орфографиялық қателермен жазылады) қолданыңыз, оларды үтірлер мен пробелдер арқылы айырып жазады. Өздердің мағынасы нақты, әрі қатесіз жазылуы маңызды. Сонда сіздің сайтыңызға өз «қажеттіліктері қанағаттандыратын» адамдар табылады.

«Keywords» тегінің құрамындағы таңбалар саны - 1000 таңбадан аспауы керек, бірақ кейбір іздеу құралдары тек 255 таңбаны ғана қабылдайды. (Іздеуіш машиналар нормадан асып кеткен таңбаларды қабылдамайды)

Кілттік өздер мен терминдерді көрсету үшін мыналар қолданылады:

<meta name = “keywords” content = “кілттік өздер“>.

Егер өздерді орыс тілінде көрсеткіңіз келсе, «lang» атрибутын көрсетіңіз:

<meta name = “keywords” lang = “ru” content = “кілттік өз“ >.

Көптеген іздеу машиналары кілттік өздер мағынасы беттің негізгі мәтініне сай келуін тексереді. Бір кілттік өзді бірнеше рет қайталауға болмайды, себебі сізді мәліметтер базасына тіркемеуі мүмкін. Бүкіл тексттік іздеу жүйелері беттің құрамын толықтай қорытындылайды., ал кілттік өздер ақпаратты жоғары сапада іздеуге мүмкіндік береді.

### **Content-Language**

META-тегі пайдаланудың тағы да бір түрі – парақ жасақталған тілді көрсету. Мысалы, браузерге оның парағында орыс тілінде мәтін бар екенін білдірі үшін мынадай код жазуыңыз қажет:

<META HTTP-EQUIV="Content-Language" CONTENT="ru">

### **Charset – кодтау.**

Бұл жазылған бетте кодтауға арналған нұсқаулар, мысалы: сіз браузерге сіздің бетіңіз Windows-1251 кодтауына сәйкес текстті құрайтындығын айтқыңыз келсе, мына кодтарды теруіңіз қажет:

<meta http-equiv="content-type" content="text/html; charset=windows-1251">.

### **Robots – робот.**

Іздеу серверлеріне арналған индекстеуді басқару. Негізгі мағыналары:

ALL - барлығын индекстеу, NONE – индекстемей. Егер тег берілмесе, өзіндік мағынасы барлығын индекстеу.

<meta name="robots" content="ALL">.

### **Revisit – ревизия.**

Жүйе құжатты индекстейтін уақыт аралығы көрсетіледі, берілген мысалда 7 күн аралығы:

<meta name="revisit-after" content="7 days">.

### **Refresh – автоматты түрде қайта бағыттау.**

(URL – қайта бағыттау).

Кейде хостингті өзгертуге тура келеді. Сондықтан бұл жағдайда оған ескі сервер бойынша қосылуға болмайды. Біз өзіміздің қолданушыларымызды жоғалтқымыз келмейді ғой? Сондықтан бұрынғы жерде жаңа сайттың мекен-жайы көрсетілген бір беті орналастырылады. Мұнан гөрі ақпараттарға қосымша браузердің өзі жаңа мекен-жайға көшкені тиімді. Мұны «refresh» параметрлік META – тег көмегімен орындау қиын емес. Мысалы:

<meta http-equiv="refresh" content="10; url=http://www.kstu.ru/">.

Бұл мысалда біз браузерге 10 секундтан соң http://www.kstu.ru/ мекен-жайына автоматты түрде көшірілу командасын бердік. 10 секунд уақыт қолданушының ақпаратты оқып үлгеруі үшін қажет.

Сонымен қатар, бұл ақпараттың көмегімен беттердің автоматты түрде қайта қосылуын орындауға болады.

### **Author – автор.**

«author» параметрлері META-тег бет авторын нұсқау үшін қолданылады және кейбір іздеу жүйелерінде есепке алынады. Негізгі әрбір беттің авторы болу керек, сондықтан бұл тегті қолданудың мәзі бар. Сонымен қатар сіздің сайтыңыздың мекен-жайы мен e-mail-ді енгізсеңіз болады, бұл құжатыңыз сайт шекарасынан шығып кеткен кезде оның қай жақтан екенін тез анықтауға мүмкіндік береді. Owner-ді де қолдануға болады, онда автордың e-mail-ы көрсетілсе болғаны:

<meta name="author" lang="ru" content="Вася Пупкин">.

Осыған ұқсас «generator» параметрлік META-тег әртүрлі HTML – редакторлар мен автоматтарды сақтайды. Бұл тегтен пайда жоқ, бірақ сапалы жарнама. Кейбір авторлар бұл тегтің негізінде «author» META-тегі туралы ақпаратты қайталайды.

### **Copyright – авторлық құқықтыр.**

Авторлық құқықтарды көрсету үшін:

<meta name="copyright" content="ұйым\_атауы">.

Expres – қолдану мерзімі немесе беттерді кәштеуге тыйым салу.

Кейбір тапсырмалар тобы беттің Web-браузер кәшінен емес, серверден күйылуын талап етеді.

Біріншіден бұл динамикалық түрде өзгертін мәліметтерді қоятын сайттар үшін қажет. Мысалы: сайт арқылы іздеу нәтижесі, қазіргі уақыттағы доллар курсы, электронды дүкен немесе қарапайым чат. Бұл кезде кәштегі бет шын өмірдегі ақпаратпен сәйкес екендігіне сенімді болмау керек.

Бұл мәселені шешу үшін сіз браузерге сіздің бетіңізді кәштеуге тыйым салсаңыз болады. Әрине мұны META-тегтің «expires» қасиетін пайдалану арқылы орындауға болады. Ол беттің құрамындағы мәліметтердің ескеру мерзімін көрсетеді. Беттің кәштелмеуі үшін өткен шақ мерзімін қондырсақ болғаны. Мысалы былай:

```
<meta http – equiv = “expires” content = “Mon, 01 Jan 1990 00:00:00 GMT”>.
```

Құжаттың кәштелген көшірмесі қай мезгілге дейін пайдалы екендігін анықтау былай көрсетіледі:

```
<meta http – equiv = “expires” content = “Tue, 22 Aug 2001 12:12:00 GMT”>.
```

«0» таңбасымен белгіленген күн жадысы (дата) «дәл қазір» деген ұғымда қолданылады. Браузер әрбір сұраныс жасаған сайын бұл құжаттың өзгергенін немесе өзгермегенін тексеріп отырады. Бұл құжатты қарағанда кәштегі емес, жаңа ақпаратты ұсынуы үшін қажет.

Tue, 22 Aug 2001 12:12:00 GMT-нің орнына never – ұғымын ешқашан қолдануға болмайды.

Бетіңізде орналасқан ақпаратты айқынырақ көрсету үшін әр бетте жеке мега-сипаттамаларды құруға тырысыңыз.

Бұл жалпылама түрде былай көрсетіледі:

```
<html>
```

```
<head>
```

```
<META NAME = “ Robots” CONTENT = “ALL”>.
```

```
<META HTTP – EQUIV = “content – Language” CONTENT = “ru”>
```

```
<META NAME = “Description” CONTENT = “мұнда бет сипаттамасын жазасыз”>
```

```
<META NAME = “Keywords” CONTENT =“мұнда кілттік өздерді жазасыз”>
```

```
<META NAME = “Author” CONTENT = “сіздің аты-жөніңіз”>
```

```
<META HTTP – EQUIV = “Reply – to” CONTENT = “сіздің E-mail-ңіз”>
```

```
<META NAME = “Generator” CONTENT = “документ құрылған бағдарламалар атауы”>
```

```
<META NAME = “Creation _Date” CONTENT = “күні/айы/жылы”>
```

```
<META NAME = “revisit – after” CONTENT = “қанша күн немесе апта екендігі”>
```

```
<TITLE> мұнда құжат тақырыбын жазасыз
```

```
</TITLE> <head> <link> - берілген құжаттың басқа құжаттармен қарым қатынасы туралы ақпаратты көрсетеді.
```

Авторлар LINK элементін іздеу машиналарына әртүрлі ақпараттарды ұсынғанда қолданса болады, соның ішінде:



- басқа тілдерге арналған альтернативті түрдегі құжат сілтемелеріне;
- басқа тасымалдаушыларға арналған альтернативтік түрдегі құжат сілтемелеріне;
- құжаттар жинағының алғашқы бетіндегі сілтемелерге;

Төмендегі мысал іздеу машинасы арқылы құжатты өңдеуді жеңілдету үшін тіл туралы, тасымалдаушы типтері және сілтемелер типтерінің ақпараттарын біріктірілген болуы мүмкін екендігін көрсетеді.

Мұнда біз құжаттың даттық, португалдық және арабтық түрін іздеу машинасына хабарлау үшін hreslang атрибутын қолданамыз. Оқулықтың араб тіліндегі нұсқасына charset атрибутының қолданылуына көңіл аударыңыз.

Французша оқулыққа Link элементінің title атрибутының lang атрибуты ұғымымен қатар қолданылуына көңіл бөліңіз:

1. <LINK title = “дат тіліндегі оқулық” type = “text /html” rel = “alternate” hreslang = “nl”  
href = “http: // someplace.com / manual / dutch.html”>

2. <LINK title – “португал тіліндегі оқулық” type = “text / html” rel” alternate” hreflang = “pt”. href = “http: // someplace.com / manual / portuguese.html”>.

3. <LINK title = “араб тіліндегі оқулық” type = “text / html” rel = “alternate” charset = “ISO – 8859-6 hreslang = “ar”

Келесі мысалды біз іздеу машинасына басып шығаруға арналған құжат түрін қай жерден іздеу керектігін хабарлаймыз:

<LINK media = “print” title = “postscript-тегі анықтама” type = “application / postscript” rel = “alternate” href = “http: // someplace.com / manual / postscript.ps”>.

Келесі мысалда құжаттар тобының алғашқы бетін қай жерде іздеу керектігін іздеу машинасына хабарлаймыз:

<LINK rel = “start” title = “бірінші бет” type = “text / html” href = “http: // someplace.com / manual / start.html”>

### **Белгілер.**

Жүріп жатқан беттің ішінде бір беттен, екінші бетке көшу керек болса, белгілер қойылады:

<A name =” метка> </A>

Мәтіндегі белгілерге мысал:

<A name = “metka 1”> </A>

<A name = “metka 2”> </A>

Белгіге өту үшін сілтеме келесі шаблондарды қолданады:

<A href = “# белгі ”> шертуге арналған мәтін </A>

Белгілерге арналған сілтемелер мысал:

<A href = “# metka 1”> бірінші белгіге сілтеме </A>

<A href = # metka 2"> екінші белгіге арналған сілтеме </A>.

### **Суреттер картасы.**

HTML спецификацмасы суретті гипертексттік сілтеме түрінде ұсынуға мүмкіндік береді. Берілген суреттің үстінен шертсеңіз сіз көрсетілген құжат бетіне ауысасыз. Егер сіз сілтемелерді бірнеше құжаттар үшін ұйымдастырғыңыз келсе, онда оны 3 немесе 4 ұсақ суреттер (пиксеграмма) көмегімен орындауға болады.

Бірақ бұл процесті орындаудың басқа әдісі де бар – суреттер картасы (Image Map). Бұл жағдайда сілтемелерді ауыстыру қай суретті шерткеніңізге байланысты болады. Осылайша бір сурет басқа суреттерге бағыттайтын векторға айналады.

Суреттер картасын қолдану Web-браузер Web-серверден процеспен әрекеттеседі деген (көзқарасқа) ойға әкеледі, ал шерту кезінде курсордың орналасу координатасын жіберу керек.

Image Map технологиясы әртүрлі облыстарда қолданылады. Бірақ оны көбінесе графикалық мәзірді құру кезінде кездестіруге болады, яғни (бір үлкен сурет) құрамында мәзір элементтері бар үлкен сурет құрылады және суреттің әрбір бөлігіне қандай да бір іс-әрекет бөлініп беріледі. Сонымен қатар бұл технологияны картографиялық мүмкіндіктері бар қарапайым ГИС- типтес жүйелерді құру кезінде кездестіруге болады.

Сурет картасын іске қосу үшін IMG тегіне қосымша параметрлер енгізу керек:

```
<IMG SRC = "url" USEMAP = "url # map name">
```

USEMAP параметрі айтылып отырған сурет картасының қай жерде орналасқандығын көрсетеді.

Сурет картасы қайсы сурет бөлігіне қандай URL сай келетіндігін көрсетіп отырады. Сурет картасына сурет орналасқан құжат ішінде де, басқа құжатта да орналасуы мүмкін.

Сурет картасын анықтау келесі синтаксис формасында болады:

```
<MAPNAME = " map name">
```

```
<AREA [SHAPE = "shape"] COORDS = "x, y, ..."
```

```
[HREF = "reference"] [NOHREF]>
```

```
</MAP>
```

### **Практикалық тапсырмалар:**

#### **Тапсырма 1.**

1. Өзіңіздің META - ақпаратыңызды түзетіңіз немесе толықтырыңыз, міндетті түрде келесі мәліметтерді қолданыңыз:

```
<META HTTP – EQUIV = "content – Language" Content = "">
```

```
<META NAME = "Description" CONTENT = "">
```

```
<META NAME = "Keywords" CONTENT = "">
```

```
<META NAME = "Author" CONTENT = "">
```

```
<TITLE> </TITLE>
```

2. Әрбір минут сайын бір бетке автоматты түрде жаңартулар жүргізіңіз.

3. Жаңа бет құрыңыз және [www.kstu.ru](http://www.kstu.ru). мекен-жайына 10 минут күту уақытымен автоматты түрде қайта бағыттау жасаңыз, бетте «10 минуттан соң автоматты түрде [www.kstu.ru](http://www.kstu.ru). сайтына ауыстырыласыз» деп жазыңыз.

4. <LINK> тегінің көмегімен барлық беттерден өз бетіңізге сілтеме құрыңыз.

5. Негізгі бетте, комментарий түрінде, мета-мәліметтердің қолдану мақсатын көрсетіңіз.

### **Тапсырма 2.**

1. Жаңа бет құрыңыз. Беттің ішінде негізгі бөлімдеріне белгілер орнатыңыз.

- Кеңейтілген іздеу. Индекс.
- Аудандар бойынша зерттеу. Индекс.
- Цитирлеу дегеніміз не? Индекс.
- “Нини - индекс” дегеніміз не? Индекс.
- “Саяси нини – индекс” дегеніміз не? Индекс.

Текстегі белгілі бір мақсаттағы еренсілтеелер түрінде беттің басында орнатылуы керек.

2. Беттің соңында, комментарий түрінде, белгілер мен сілтемелерге қолданылатын тегтер мен олардың атрибуттарының қолдану мақсатын көрсетіңіз.

### **Тапсырма №3.**

Дайын сурет көмегімен сурет партасын құру; картографиялық қала белгісінің үстін шерткенде тиісті www – сервермен байланысуы керек.

Серверлерге арналған сілтемелер.

[www.moswa.ru](http://www.moswa.ru)- Мәскеу

[www.kazan.ru](http://www.kazan.ru)- Казань

[www.spb.ru](http://www.spb.ru)- Санкт-Петербург

[www.novosibirsk.ru](http://www.novosibirsk.ru)- Новосибирск

[www.norilsk.ru](http://www.norilsk.ru)- Норилск

[www.omsk.ru](http://www.omsk.ru)- Омск

[www.magodin.ru](http://www.magodin.ru)- Магадан.

Беттің шығу жерінде комментарий түрінде сурет картасында қолданылып отырған тегтердің қолдану мақсатын көрсетіңіз.

### **Практикалық жұмыс № 3.**

**Тақырыбы:** Құжат «шапкасы», МЕТА-мәліметтер, НТМҒ-құжаттардағы суреттер белгісі мен картасы.

**Жұмыс мақсаты:** HTML - құжаттарды белгілер (метки) көмегімен құру, сурет картасын

құру және де МЕТА – мәліметтер мен құжат «шапкасын» қолдану.

#### **Теориялық мағлұматтар.**

Құжат «шапкасы» мен МЕТА – мәліметтер </head> тегтерімен шектеледі. МЕТА – ақпараттарды құрайды, яғни бет жөнінде қосымша мәліметтер, беттер атауы, браузер терезесіне шығарылатан тақырыптар және сілтемелер.

Тақырыпша </title> және </title> тегтерімен анықталады, бұл міндетті емес, бірақ бағаны жақсы <head> және <title> тегтерінде параметрлер болмайды.

Егер еренсілтемелік құжаттар авторы өз шығармасын оқитын қауымға жеткізгісі келсе, өзінің жұмысын, каталогтар мен іздеу жүйелерінде әрі ерекше, әрі тиімді етіп тіркеуі қажет.

Бұл үшін HTML тілінің командалар жинағын арнайы META инструкциялар тобы қондырылған, ал негізінен іздеу машиналары арқылы құжатты сипаттауға және индекстеуге арналған.

Өзіңіздің сайтыңызды іздеу жүйелері мен каталогтарда тіркеуден алдын дайындау қажет. Болмаса қолданылған күш пен уақыт нәтижесі төмен болады. Берілген мысалдарда біз META-тегтерді қарастырамыз. Олардың рөлі беттерді көрсетуде білінбейді. Бұл командалар Web-серверлер мен браузерге арналған, бірақ маңызды командалар, өйткені META-тегтерді дұрыс қолдану сізге өз артықшылығын береді.

Іздеу жүйелері беттерді тіркеу кезінде META дискрипторларда сақталған ақпараттарда қолданады, олар сіздің сайтыңыздағы <head> немесе </head> тегтерінің арасында орындалады. META барлық іздеу жүйелерінде қолданылмағанымен, оларсыз жұмыс істеу мүмкін емес. META элементтері құжаттардың кейбір қасиеттерін сипаттайтын ауыспалы мағынаны анықтайды: автор жөнінде ақпарат, кілттік өздер тізімі т.б. Name атрибут – ауыспалы, ал content – оның мағыналықұғымы. Есте сақтаңыз, барлық META – тегтер визуалды ақпаратты тасымайды және тек <head> </head> құжаттар акциясында орналасуы қажет.

META – жинағының барлығын қолдану міндет емес, бірақ базаның болуы қажетті. Оларға Title, Keywords, Description, Charset, Author- ді жатқызуға болады.

#### **Title – құжат тақырыбы.**

Дұрыс таңдалған тақырып сіздің сілтемеңізге көптеген қолданушыларды еліктіреді. Мысалы: <TITLE> мұнда құжат тақырыбын жазсаңыз </TITLE>.

#### **Description – сипаттау.**

Сіздің сайтыңыздың артықшылығын нақты сипаттаудан артық жарнама жоқ. Description сұранысқа жауап беру кезінде сіздің сайтыңызды сипаттауға жауап береді.

Қысқаша сипаттауға мына нұсқаулар қолданылады:

```
<meta name =” description “ content = “ бет сипаттамасы “>.
```

Егер сипаттауды орыс тілінде жүргізгіңіз келсе, «lang» атрибутын көрсету керек.

```
<meta name =” description” lang = “ru” content =” бет сипаттамасы “ >.
```

“Description” тегіне сіздің сайтыңызда көп қолданылатын барлық терминдерді, өздерді, олардың синонимдерін енгізу керек.

#### **Keywords – іздеу өздер мен өйлемдері.**

Өзіңіздің кілттік өзіңізді пайдаланыңыз, олар маңызды. Өртүрлі өздерді және олардың өзгертулерін, төменгі регистрді (әдетте, орфографиялық қателермен жазылады) қолданыңыз, оларды үтірлер мен пробелдер арқылы

айырып жазады. Өздердің мағынасы нақты, әрі қатесіз жазылуы маңызды. Сонда сіздің сайтыңызға өз «қажеттіліктері қанағаттандыратын» адамдар табылады.

«Keywords» тегінің құрамындағы таңбалар саны - 1000 таңбадан аспауы керек, бірақ кейбір іздеу құралдары тек 255 таңбаны ғана қабылдайды. (Іздеуіш машиналар нормадан асып кеткен таңбаларды қабылдамайды)

Кілттік өздер мен терминдерді көрсету үшін мыналар қолданылады:

<meta name = “keywords” content = “кілттік өздер“>.

Егер өздерді орыс тілінде көрсеткіңіз келсе, «lang» атрибутын көрсетіңіз:

<meta name = “keywords” lang = “ru” content = “кілттік өз“ >.

Көптеген іздеу машиналары кілттік өздер мағынасы беттің негізгі мәтініне сай келуін тексереді. Бір кілттік өзді бірнеше рет қайталауға болмайды, себебі сізді мәліметтер базасына тіркемеуі мүмкін. Бүкіл тексттік іздеу жүйелері беттің құрамын толықтай қорытындылайды., ал кілттік өздер ақпаратты жоғары сапада іздеуге мүмкіндік береді.

### **Content-Language**

META-тегіті пайдаланудың тағы да бір түрі – парақ жасақталған тілді көрсету. Мысалы, браузерге оның парағында орыс тілінде мәтін бар екенін білдірі үшін мынадай код жазуыңыз қажет:

<META HTTP-EQUIV="Content-Language" CONTENT="ru">

### **Charset – кодтау.**

Бұл жазылған бетте кодтауға арналған нұсқаулар, мысалы: сіз браузерге сіздің бетіңіз Windows-1251 кодтауына сәйкес текстті құрайтындығын айтқыңыз келсе, мына кодтарды теруіңіз қажет:

<meta http – equiv = “ content type “ content = “ text / html; charset = windows – 1251”>.

### **Robots – робот.**

Іздеу серверлеріне арналған индекстеуді басқару. Негізгі мағыналары:

ALL - барлығын индекстеу, NONE – индекстемей. Егер тег берілмесе, өзіндік мағынасы барлығын индекстеу.

< meta name = “robots” content = “ALL”>.

### **Revisit – ревизия.**

Жүйе құжатты индекстейтін уақыт аралығы көрсетіледі, берілген мысалда 7 күн аралығы:

<meta name = “revisit – after” content =”7 days”>.

### **Refresh – автоматты түрде қайта бағыттау.**

(URL – қайта бағыттау).

Кейде хостингті өзгертуге тура келеді. Сондықтан бұл жағдайда оған ескі сервер бойынша қосылуға болмайды. Біз өзіміздің қолданушыларымызды жоғалтқымыз келмейді ғой? Сондықтан бұрынғы жерде жаңа сайттың мекен-жайы көрсетілген бір беті орналастырылады. Мұнан гөрі ақпараттарға қосымша браузердің өзі жаңа мекен-жайға көшкені тиімді. Мұны «refresh» параметрлік META – тег көмегімен орындау қиын емес. Мысалы:

<meta http-equiv=”refresh” content = “10; url = http: [www.kstu.ru](http://www.kstu.ru) /”>.

Бұл мысалда біз браузерге 10 секундтан соң <http://www.kstu.ru/> мекен-жайына автоматты түрде көшірілу командасын бердік. 10 секунд уақыт қолданушының ақпаратты оқып үлгеруі үшін қажет.

Сонымен қатар, бұл ақпараттың көмегімен беттердің автоматты түрде қайта қосылуын орындауға болады.

### **Author – автор.**

«author» параметрлері META-тег бет авторын нұсқау үшін қолданылады және кейбір іздеу жүйелерінде есепке алынады. Негізгі әрбір беттің авторы болу керек, сондықтан бұл тегті қолданудың мәзі бар. Сонымен қатар сіздің сайтыңыздың мекен-жайы мен e-mail-ді енгізсеңіз болады, бұл құжатыңыз сайт шекарасынан шығып кеткен кезде оның қай жақтан екенін тез анықтауға мүмкіндік береді. Owner-ді де қолдануға болады, онда автордың e-mail-ы көрсетілсе болғаны:

```
<meta name =“ author” lang =”ru” content =”Вася Пупкин“>.
```

Осыған ұқсас «generator» параметрлік META-тег әртүрлі HTML – редакторлар мен автоматтарды сақтайды. Бұл тегтен пайда жоқ, бірақ сапалы жарнама. Кейбір авторлар бұл тегтің негізінде «author» META–тегі туралы ақпаратты қайталайды.

### **Copyright – авторлық құқықтыр.**

Авторлық құқықтарды көрсету үшін:

```
<meta name =” copyright” conter =”ұйым _атауы “>.
```

Expres – қолдану мерзімі немесе беттерді кәштеуге тыйым салу.

Кейбір тапсырмалар тобы беттің Web-браузер кәшінен емес, серверден құйылуын талап етеді.

Біріншіден бұл динамикалық түрде өзгертін мәліметтерді қоятын сайттар үшін қажет. Мысалы: сайт арқылы іздеу нәтижесі, қазіргі уақыттағы доллар курсы, электронды дүкен немесе қарапайым чат. Бұл кезде кәштегі бет шын өмірдегі ақпаратпен сәйкес екендігіне сенімді болмау керек.

Бұл мәселені шешу үшін сіз браузерге сіздің бетіңізді кәштеуге тыйым салсаңыз болады. Әрине мұны META-тегінің «expires» қасиетін пайдалану арқылы орындауға болады. Ол беттің құрамындағы мәліметтердің ескеру мерзімін көрсетеді. Беттің кәштелмеуі үшін өткен шақ мерзімін қондырсақ болғаны. Мысалы былай:

```
<meta http – equiv = “expires” content = “Mon, 01 Jan 1990 00:00:00 GMT”>.
```

Құжаттың кәштелген көшірмесі қай мезгілге дейін пайдалы екендігін анықтау былай көрсетіледі:

```
<meta http – equiv = “expires” content = “Tue, 22 Aug 2001 12:12:00 GMT”>.
```

«0» таңбасымен белгіленген күн жадысы (дата) «дәл қазір» деген ұғымда қолданылады. Браузер әрбір сұраныс жасаған сайын бұл құжаттың өзгергенін немесе өзгермегенін тексеріп отырады. Бұл құжатты қарағанда кәштегі емес, жана ақпаратты ұсынуы үшін қажет.

Tue, 22 Aug 2001 12:12:00 GMT-нің орнына never – ұғымын ешқашан қолдануға болмайды.

Бетіңізде орналасқан ақпаратты айқынырақ көрсету үшін әр бетте жеке мега-сипаттамаларды құруға тырысыңыз.

Бұл жалпылама түрде былай көрсетіледі:

```
<html>
<head>
<META NAME = “ Robots” CONTENT = “ALL”>.
<META HTTP – EQUIV = “content – Language” CONTENT = “ru”>
<META NAME = “Description” CONTENT = “мұнда бет сипаттамасын
жазасыз”>
<META NAME = “Keywords” CONTENT =“мұнда кілттік өздерді жазасыз”>
<META NAME = “Author” CONTENT = “сіздің аты-жөніңіз”>
<META HTTP – EQUIV = “Reply – to” CONTENT = “сіздің E-mail-ңіз”>
<META NAME = “Generator” CONTENT = “документ құрылған бағдарламалар
атауы”>
<META NAME = “Creation _Date” CONTENT = “күні/айы/жылы”>
<META NAME = “revisit – after” CONTENT = “қанша күн немесе апта
екендігі”>
<TITLE> мұнда құжат тақырыбын жазасыз
</TITLE> <head> <link> - берілген құжаттың басқа құжаттармен қарым
қатынасы туралы ақпаратты көрсетеді.
```

Авторлар LINK элементін іздеу машиналарына әртүрлі ақпараттарды ұсынғанда қолданса болады, соның ішінде:

- басқа тілдерге арналған альтернативті түрдегі құжат сілтемелеріне;
- басқа тасымалдаушыларға арналған альтернативтік түрдегі құжат сілтемелеріне;
- құжаттар жинағының алғашқы бетіндегі сілтемелерге;

Төмендегі мысал іздеу машинасы арқылы құжатты өңдеуді жеңілдету үшін тіл туралы, тасымалдаушы типтері және сілтемелер типтерінің ақпараттарын біріктірілген болуы мүмкін екендігін көрсетеді.

Мұнда біз құжаттың даттық, португалдық және арабтық түрін іздеу машинасына хабарлау үшін hreslang атрибутын қолданамыз. Оқулықтың араб тіліндегі нұсқасына charset атрибутының қолданылуына көңіл аударыңыз.

Французша оқулыққа Link элементінің title атрибутының lang атрибуты ұғымымен қатар қолданылуына көңіл бөліңіз:

1. <LINK title = “дат тіліндегі оқулық” type = “text /html” rel = “alternate” hreslang = “nl”.

href = “http: // someplace.com / manual / dutch html”>

2. <LINK title – “португал тіліндегі оқулық” type = “text / html” rel” alternate” hreflang = “pt”. href = “http: // someplace.com / manual / portuguese.html”>.

3. <LINK title = “араб тіліндегі оқулық” type = “text / html” rel = “alternate” charset = “ISO – 8859-6 hreslang = “ar”

Келесі мысалды біз іздеу машинасына басып шығаруға арналған құжат түрін қай жерден іздеу керектігін хабарлаймыз:

```
<LINK media = "print" title = "postscript-тегі анықтама"  
type = "application / postscript" rel = "alternate" href = "http: // someplace.com /  
manual / postscript.ps">
```

Келесі мысалда құжаттар тобының алғашқы бетін қай жерде іздеу керектігін іздеу машинасына хабарлаймыз:

```
<LINK rel = "start" title = "бірінші бет" type = "text / html"  
href = "http: // someplace.com / manual / start.html">
```

### **Белгілер.**

Жүріп жатқан беттің ішінде бір беттен, екінші бетке көшу керек болса, белгілер қойылады:

```
<A name = "метка" </A>
```

Мәтіндегі белгілерге мысал:

```
<A name = "метка 1"> </A>
```

```
<A name = "метка 2"> </A>
```

Белгіге өту үшін сілтеме келесі шаблондарды қолданады:

```
<A href = "# белгі " > шертуге арналған мәтін </A>
```

Белгілерге арналған сілтемелер мысал:

```
<A href = "# метка 1"> бірінші белгіге сілтеме </A>
```

```
<A href = # метка 2"> екінші белгіге арналған сілтеме </A>
```

### **Суреттер картасы.**

HTML спецификацмясы суретті гипертекстік сілтеме түрінде ұсынуға мүмкіндік береді. Берілген суреттің үстінен шертсеңіз сіз көрсетілген құжат бетіне ауысасыз. Егер сіз сілтемелерді бірнеше құжаттар үшін ұйымдастырғыңыз келсе, онда оны 3 немесе 4 ұсақ суреттер (пиксеграмма) көмегімен орындауға болады.

Бірақ бұл процесті орындаудың басқа әдісі де бар – суреттер картасы (Image Map). Бұл жағдайда сілтемелерді ауыстыру қай суретті шерткеніңізге байланысты болады. Осылайша бір сурет басқа суреттерге бағыттайтын векторға айналады.

Суреттер картасын қолдану Web-браузер Web-серверден процеспен әрекеттеседі деген (көзқарасқа) ойға әкеледі, ал шертту кезінде курсордың орналасу координатасын жіберу керек.

Image Map технологиясы әртүрлі облыстарда қолданылады. Бірақ оны көбінесе графикалық мәзірді құру кезінде кездестіруге болады, яғни (бір үлкен сурет) құрамында мәзір элементтері бар үлкен сурет құрылады және суреттің әрбір бөлігіне қандай да бір іс-әрекет бөлініп беріледі. Сонымен қатар бұл технологияны картографиялық мүмкіндіктері бар қарапайым ГИС- типтес жүйелерді құру кезінде кездестіруге болады.



Сурет картасын іске қосу үшін IMG тегіне қосымша параметрлер енгізу керек:

```
<IMG SRC = “url” USEMAP = “url # map name”>
```

USEMAP параметрі айтылып отырған сурет картасының қай жерде орналасқандығын көрсетеді.

Сурет картасы қайсы сурет бөлігіне қандай URL сай келетіндігін көрсетіп отырады. Сурет картасына сурет орналасқан құжат ішінде де, басқа құжатта да орналасуы мүмкін.

Сурет картасын анықтау келесі синтаксис формасында болады:

```
<MAPNAME = “ map name”>  
<AREA [SHAPE = “shape”] COORDS = “x, y, ...”  
[HREF = “reference”] [NOHREF]>  
</MAP>
```

### **Практикалық тапсырмалар:**

#### **Тапсырма 1.**

1. Өзіңіздің META - ақпаратыңызды түзетіңіз немесе толықтырыңыз, міндетті түрде келесі мәліметтерді қолданыңыз:

```
<META HTTP – EQUIV = “content – Language” Content = “”>
```

```
<META NAME = “Description” CONTENT = “”>
```

```
<META NAME = “Keywords” CONTENT = “”>
```

```
<META NAME = “Author” CONTENT = “”>
```

```
<TITLE> </TITLE>
```

2. Әрбір минут сайын бір бетке автоматты түрде жаңартулар жүргізіңіз.

3. Жаңа бет құрыңыз және [www.kstu.ru](http://www.kstu.ru) мекен-жайына 10 минут күту уақытымен автоматты түрде қайта бағыттау жасаңыз, бетте «10 минуттан соң автоматты түрде [www.kstu.ru](http://www.kstu.ru) сайтына ауыстырыласыз» деп жазыңыз.

4. <LINK> тегінің көмегімен барлық беттерден өз бетіңізге сілтеме құрыңыз.

5. Негізгі бетте, комментарий түрінде, мета-мәліметтердің қолдану мақсатын көрсетіңіз.

#### **Тапсырма 2.**

1. Жаңа бет құрыңыз. Беттің ішінде негізгі бөлімдеріне белгілер орнатыңыз.

- Кеңейтілген іздеу. Индекс.
- Аудандар бойынша зерттеу. Индекс.
- Цитирлеу дегеніміз не? Индекс.
- “Нини - индекс” дегеніміз не? Индекс.
- “Саяси нини – индекс” дегеніміз не? Индекс.

Текстегі белгілі бір мақсаттағы еренсілтеелер түрінде беттің басында орнатылуы керек.

2. Беттің соңында, комментарий түрінде, белгілер мен сілтемелерге қолданылатын тегтер мен олардың атрибуттарының қолдану мақсатын көрсетіңіз.

### **Тапсырма №3.**

Дайын сурет көмегімен сурет партасын құру; картографиялық қала белгісінің үстін шерткенде тиісті www – сервермен байланысуы керек.

Беттің шығу жерінде комментарий түрінде сурет картасында қолданылып отырған тегтердің қолдану мақсатын көрсетіңіз.

### **Практикалық жұмыс №4**

**Тақырып: Кестелер көмегімен беттерді орналастыру (макеттеу)**

**Мақсаты:** Кестемелер көмегімен беттерді орналастыру қабілетін дамыту.

**Теориялық мәліметтер:**

- Барынша аз қате жіберу үшін өзгелердің тәжірибесін қолдану керек. Бетті орналастыру тәжірибесі ең көбіне басып шығару орындарында жақсы дамыған. Егер газетті, журналды, кітапты және тағы басқаларды анықтап қарасақ, мына негізгі элементтерді көреміз.

- Ақ фон мен қара әріптер қолданылады. Фоны қара, ақ әріппен жазылған қалың кітапты елестетіп қараңыз, оқып болған соң көзіңіздің көру қабілетін нашарлатасыз.

- Оң жақ және сол жақ шегімен, жоғары және төмен шегімен бет шектелген. Бір бетке ешбір шектеусіз жазылған бетті оқу ыңғайсыз.

- Абзацтар екі беттен де тегістелінген. Барлық жолдар бір түзу бойымен аяқталады. “Көз” бұған үйреніп, оқуға ыңғайлы болады.

- Жаңа абзац 1-ші жолды жазардан алдын бос орын қалдырылып басталады, ал абзацтар арасында жолдар арасымен салыстырғанда бос орын көптеу қалдырылады. Бұл визуалды түрде абзацтарды бөліп тұрады.

“Бағанды құрылымды” қолдану, бұл әсіресе газеттер мен журналдарда кеңінен таралған. Осылайша үлкен бір бетке кіші шрифті мәтіндер мен графикалық элементтерді тиімді орналастыруға болады. Сонымен қатар, жіңішке бағаналардағы мәтінді жоғарыдан төмен арқылы оқу көзді шаршатпайды.

Егер 1-ші пункттармен жұмыс істеу оңай болса, бағаналық құрылысты құру қиынырақ соғады.

HTML-де беттерді орналастыру қабілеттілігі, ескерілмеген соң, бұл процесті кесте құру арқылы орындайды.

Көңл бөлсек, көптеген сайттарда, беттерде кестелерді орналастыру арқылы құрылған.

Осылайша беттерді макеттеу беттің шегін құруға, мәтінді бағаналарға бөлуге, жекелеген мәтін бөліктерін әртүрлі фонға бояуға және фондық суреттерді орналастыруға мүмкіндік береді.

Ақпаратты бағана түрінде орналастыру үшін мәтін мен суретті кесте ұяшықтарының ішінде орналастырады. Ұяшықтар ішіне қосымша кестелер енгізуге болады.

Сервердің, баннердің т.б. атауы		
Бастапқы бетке оралу, көп таралған ресурстарды іздеу.		
Сервердің құрамы мекен-жайы, телефоны	Беттің негізгі мәтіні.	Маңызды ресурстар, белгілеуге арналған ресурстар.
Бастапқы бетке оралу, көп таралған ресурстарды іздеу.		
Авторлық құқықтар, адресі, телефоны.		

Қазіргі рұқсат етілген орналастыру көлемі- 800 x 600, яғни негізгі кестенің көлемін 750-ден асырмау керек, сонда бет экранда толықтай көрсетіледі.

Кесте көлемін пайыздық қатынас түрінде көрсетуге болады, мысалы: 80% (100% кестенің жартысы көрінбей қалады).

Беттің құйылуын жеделдету үшін мына үш ережені сақтау керек: ені бірдей болатын кестелерді – жоғары, төменгі және негізгі бөлігін көрсету керек. Сонымен қатар, бұл сайттың жалпы стилін сақтап отырады.

Мысал ретінде мына сайттарды қарастыруға болады.

<http://www.aport.ru>; [http://www.gis\\_meteo.ru](http://www.gis_meteo.ru); <http://www.lenta.ru>; <http://www.rbe.ru>; жасырынды кесте құрылысын көру үшін, Front Page бетін көшіріп қарауға болады.

Кестелерді қолдаудың кемшіліктері:

- “Жай құйылу”. Барлық кесте құйылғанша, ақыры экран бетіне көрсетілмейді.
- Артық кодтау. Қажет емес жолдар мен ұяшықтарды құруға тура келеді.
- Экранда элементті нақты орналастыру мүмкіндігінің жоқтығы.
- Бір элементті екіншісіне “ығыстыру” мүмкіндігінің жоқтығы.
- Сурет элементтерін басқару мүмкіндігінің жоқтығы (ақыры шектен асып кеткен кезде, кесте аз көлемін өзгертеді).

### **Практикалық тапсырмалар:**

#### **Тапсырма №1.**

1. Алынған біліміңізге сәйкес бірінші бетті құрыңыз. Бет кемінде 4 бағанадан тұруы керек. Әрбір бағана көлемі 3-30 пиксельден тұратын бос ұяшықтарға бөлінуі қажет, шектемелер көрінбеуі тиіс. Жоғарғы және төменгі бөлігі қара кесте түрінде істелуі керек, басқа беттерге де енгізу үшін.

2. Өзіңіздің қалауыңызша ұяшық фонының түсін немесе фондық сурет енгізіңіз.

3. Бірінші бетке сілтеме құруды ұмытпаңыз.

4. Әрбір бет соңында, комментарий ретінде, әрбір бағананың бастамасын нұсқаңыз.

#### **Тапсырма №2.**

1. Барлық беттерді бір жалпы түрге әкеліңіз, яғни жоғарғы және төменгі бөліктері бірдей болу қажет. ( фреймдік беттерден басқа).

2. Барлық беттерде мета- мәліметтерді орналастырыңыз.

## Практикалық жұмыс №5.

**Тақырып: HTML- құжаттардың формасы.**

**Мақсаты:** HTML- көмегімен формаларды құрып үйрену.

**Теориялық мәліметтер:**

Формалар ақпаратты жұп қылып бағдарлама - өңдеушіге жібереді.

[ауыспалының атауы] = [ауыспалының мағынасы].

Ауыспалылар атауын латын әріптерімен енгізу керек.

Ауыспалылар ақпараттар сан түрінде берілген болса да, жол түрінде қабылданады.

Форма <FORM> тегімен ашылып, </FORM> белгісімен аяқталады.

<FORM> тегі 3 атрибутты құрайды, олардың ішінде бір атрибут міндетті:

ACTION – міндетті атрибут. Форма өңдеушісінің қай жерде орналасқананын анықтайды.

**METHOD** – формадағы мәліметтердің өңдеушіге қандай жолмен жіберілетіндігін анықтайды. Қолданылатын өздер: METHOD = POST және METHOD = GET. Егер атрибуты көрсетілмесе, METHOD = GET қолданылады.

**GET** «дел» әдісімен HTTP браузер «action» элементін алады, оған «?» таңбасын қосып, сонан соң «application/x-www-form-urlencoded» типтес құралды қолданумен кодталған мәліметтер формасының жинағы қосылады.

Сонан соң барлық ақпаратты еренсілтеме арқылы осы URL- ға қайта бағыттайды. Бұл сценарийде формалар мәліметтері ASCII кодымен шектелген және енгізіліп жатқан ақпараттар көлеміне қатаң шектеулер қойылады.

**POST:** «post» әдісі арқылы HTTP браузер HTTP «post» транзакциясын жүргізеді. Ол үшін action атрибутының мағынасын және enctype атрибутымен анықталу негізінде мәліметтер типіне сәйкес хатты қолданады.

Формада әртүрлі атаулы немесе мағыналы submit типтес бірнеше батырмалары орналасуы мүмкін.

Өңдеуші осылайша, қолданушы қандай submit батырмасын шерткеніне қарай әртүрлі бағытта жұмыс істейді.

<INPUT> элементінің басқа да типтері бар. Әрбір <INPUT> элементі өңдеушіге жіберілетін, ауыспалы атауын анықтайтын NAME = [имя] атрибутын өзіне енгізуі керек. Атау тек ғана латын әріптерімен жазылуы қажет. <INPUT> элементтерінің көпшілігінің құрамында VALUE = “[ұғым атауы] атрибуты болуы тиіс, олар өндірушіге осы атаумен жіберіледі.

<INPUT> элементтерінің негізгі түрлері:

TYPE = text.

Мәтін жолдарын енгізетін терезені анықтайды. Қосымша атрибуттарды сақтай алады:

SIZE [саны] – енгізуге арналған шектердің ені, таңбалар түрінде;

MAXLENGTH = [сан] – енгізілетін жолдардың таңба түріндегі максималды ұзындығы.

Мысалы:

<INPUT TYPE = text size = 30 NAME = student VALUE = “Вася Пупкин”> ==  
> Вася Пупкин.

Текстке енгізілетін поля енін 30 таңба түрінде анықтайды. Өздігінен терезеде Вася Пупкин мәтіні орналасқан. Мұны қолданушы өзгерте алады. Өзгертілген мәтін өңдеуші «student» ауыспалысына жіберіледі. (student = мәтін – құрамы). Осыны өзгертіп қараңыз:

TYPE – PASSWORD

Парольді енгізуге арналған терезені анықтайды. Текст типі секілді өте қарапайым, тек енгізіліп жатырған тексттегі таңбалар орнына «\*» жұлдызшалар қолданылады. Мысалы:

<INPUT TYPE = password NAME = pswd size = 20 MAXLENGTH = 10 > == >

Парольді енгізуге, көлемі 20 таңбалық терезе қолданылады. Парольдің максималды саны – 10 таңбадан тұрады. Енгізілген пароль өңдеушіге ауыспалы pswd (pswd = мәтіннің – құрылысы). Ақпаратты поляға енгізіп көріңіз:

TYPE = radio

Радио батырманы анықтайды. Қосымша checked атрибутын құруы мүмкін.

Радио батырмалар тобында сәйкес атпен тек бір белгіленген радиобатырма болуы мүмкін. Мысалы:

<INPUT TYPE = radio NAME = modem VALUE = “ 9600 ” checked > 9600  
бит/с == > о 9600 бит/с;

<INPUT TYPE = radio NAME = modem VALUE = “14400”> 14400 бит/с == >  
о 14400 бит/с;

<INPUT TYPE = radio NAME = modem VALUE = “28800” > 28800 бит/с == >  
о 28800 бит/с;

9600 бит/с, 14400 бит/с және 28800 бит/с жазылған 3 радиобатырма тобынан тұрады. Алдымен бірінші батырма белгіленеді. Егер қолданушы басқа батырманы белгілемесе ауыспалы modem 9600 ұғымымен жіберіледі (modem = 9600). Егер қолданушы екінші батырманы белгілесе, өңдеушіге ауыспалы modem 14400 ұғымымен жіберіледі (modem = 14400).

TYPE = checkbox

Белгі істеуге болатын квадратты анықтайды. Құрамында қосымша checked атрибуты болуы мүмкін. Радиобатырмалармен салыстырғанда, квадраттар тобында бірнеше белгіленген квадраттар болуы мүмкін. Мысалы:

<INPUT TYPE = checkbox NAME = comp VALUE = “PC”> Дербес компьютерлер == > ДК.

<INPUT TYPE = checkbox NAME = comp VALUE = “WS” checked>  
қызметтегі станциялар == > қызметтегі станциялар.

Төртінші квадрат тобын анықтайды. Алдымен екінші және төртінші квадраттар белгіленеді. Егер қолданушы өзгерістер енгізбесе, өңдеушіге бір comp ауыспалы екі ұғыммен жіберіледі.

TYPE – hidden.

Формаларды толтыру кезінде қолданушыға көрінбейтін мәліметтер элементтерін анықтайды және өңдеушіге атаусыз жіберіледі. Мұндай

элементтерді формада сақтаған жөн, ол бөтен адамдардан мәліметтерді қорғауға мүмкіндік береді. Мысалы:

```
<INTER TYPE = hidden NAME = id VALUE = "1">
```

Өңдеушіге 1 таңбасымен жіберілетін ауыспалы Id индексін анықтайды.

TYPE – reset батырмасын басқан кезде, форма бастапқы қалпына келетін батырманы анықтайды. Бұл батырманы қолданған кезде өңдеушіге мәліметтерді жібермей-ақ қоюға болады, reset типті батырманың name атрибуты болмауы да мүмкін. Мысалы:

```
<INPUT TYPE = reset VALUE = "поляны тазарту"> == > поляны тазарту.
```

<SELECT> элементі:

n элементінен құрылған <SELECT> мәзірі былайша сипатталады:

```
<SELECT NAME = "[атауы]">
```

```
<OPTION VALUE = "[ұғым 1]" > [мәтін 1]
```

```
<OPTION VALUE = "[ұғым 2]" > [мәтін 2]
```

```
<OPTION VALUE = "[ұғым n]" > [мәтін n ] </select>
```

<select> белгісі ауыспалы атауын анықтайтын name атрибутын міндетті түрде құрайды. <select> белгісі MULTIPLE атрибутында құрайды. Оның болуы мәзірден бірнеше элементті таңдауға болатындығын көрсетеді. Көптеген браузерлер <select MULTIPLE> терезе түрінде көрсетеді, онда мәзірдің барлық элементтері орналасқан. Бірнеше элементтерді таңдау үшін «SHIFT» батырмасын басып тұрып, қажет элементтер үстінен тышқанмен шертесіз.

```
<select Multiple size = 3 NAME = "атауы">
```

```
<OPTION VALUE = "[ұғым 1]" > [мәтін 1]
```

```
<OPTION VALUE = "[ұғым 2]" > [мәтін 2]
```

```
<OPTION VALUE = "[ұғым n]" > [мәтін n ] </select>
```

<OPTION> белгісі мәзір элементін анықтауға, VALUE атрибуты мәзір элементі таңдалған кезде өңдеушіге жіберілетін ұғымды айқындайды,

<OPTION> белгісінің құрамында берілген элемент өздігінен белгіленгендігін көрсететін selected атрибуты болады.

Мұндай фрагмент мәзірді 3 элементтен құрайды. 1 нұсқа, 2 нұсқа және 3 нұсқа өздігінен 2 нұсқа таңдалған.

Өңдеушіге selection ауыспалысы жіберіледі, сәйкестігіне қарай option 1, option 2, option 3 көрсетіледі.

<INPUT> элементтерінің басқада түрлері бар. Әрбір <INPUT> элементі атрибут NAME=[имя] өзіне қосып алу керек, ауыспалының атын анықтайтын қабылдап алушыға жіберілгенде. Аты латын тілімен берілу керек. Көптеген <INPUT> элементтері атрибут VALUE="[мағына]" қосып алу керек, мағынаны анықтайтын қабылдап алушыға осы атпен берілетін.

<INPUT> элементтерінің негізгі түрлері:

TYPE=text

Мәтінді қосу үшін терезені анықтайды. Қосымша атрибуттар болуы мүмкін SIZE=[сан] (даланың жол ені символдар салу үшін) и MAXLENGTH=[сан] (максималды жол сызықшасы символдарда берілетін).

Мысалы:

```
<INPUT TYPE=text SIZE=30 NAME=student VALUE="Вася Пупкин"> ==>
```

Текстке енгізілетін поля енін 30 таңба түрінде анықтайды. Өздігінен терезеде Вася Пупкин мәтіні орналасқан. Мұны қолданушы өзгерте алады. Өзгертілген мәтін өңдеуші «student» ауыспалысына жіберіледі. (student = мәтін – құрамы).

Осыны өзгертіп қараңыз:

```
TYPE – PASSWORD
```

Парольді енгізуге арналған терезені анықтайды. Текст типі секілді өте қарапайым, тек енгізіліп жатырған тексттегі таңбалар орнына «\*» жұлдызшалар қолданылады. Мысалы:

```
<INPUT TYPE = password NAME = pswd size = 20 MAXLENGTH = 10 > == >
```

Парольді енгізуге, көлемі 20 таңбалық терезе қолданылады. Парольдің максималды саны – 10 таңбадан тұрады. Енгізілген пароль өңдеушіге ауыспалы pswd (pswd = мәтіннің – құрылысы). Ақпаратты поляға енгізіп көріңіз:

```
TYPE = radio
```

Радио батырманы анықтайды. Қосымша checked атрибутын құруы мүмкін. Радио батырмалар тобында сәйкес атпен тек бір белгіленген радиобатырма болуы мүмкін. Мысалы:

```
<INPUT TYPE = radio NAME = modem VALUE = “ 9600 ” checked > 9600 бит/с  
== > о 9600 бит/с;
```

```
<INPUT TYPE = radio NAME = modem VALUE = “14400”> 14400 бит/с == > о  
14400 бит/с;
```

```
<INPUT TYPE = radio NAME = modem VALUE = “28800” > 28800 бит/с == > о  
28800 бит/с;
```

9600 бит/с, 14400 бит/с және 28800 бит/с жазылған 3 радиобатырма тобынан тұрады. Алдымен бірінші батырма белгіленеді. Егер қолданушы басқа батырманы белгілемесе ауыспалы modem 9600 ұғымымен жіберіледі (modem = 9600). Егер қолданушы екінші батырманы белгілесе, өңдеушіге ауыспалы modem 14400 ұғымымен жіберіледі (modem = 14400).

```
TYPE=checkbox
```

Белгі істеуге болатын квадратты анықтайды. Құрамында қосымша checked атрибуты болуы мүмкін. Радиобатырмалармен салыстырғанда, квадраттар тобында бірнеше белгіленген квадраттар болуы мүмкін. Мысалы:

```
<INPUT TYPE = checkbox NAME = comp VALUE = “PC”> Дербес  
компьютерлер == > ДК.
```

```
<INPUT TYPE = checkbox NAME = comp VALUE = “WS” checked> қызметтегі  
станциялар == > қызметтегі станциялар.
```

Төртінші квадрат тобын анықтайды. Алдымен екінші және төртінші квадраттар белгіленеді. Егер қолданушы өзгерістер енгізбесе, өңдеушіге бір comp ауыспалы екі ұғыммен жіберіледі.

```
TYPE=hidden
```

Формаларды толтыру кезінде қолданушыға көрінбейтін мәліметтер элементтерін анықтайды және өңдеушіге атаусыз жіберіледі. Мұндай

элементтерді формада сақтаған жөн, ол бөтен адамдардан мәліметтерді қорғауға мүмкіндік береді. Мысалы:

```
<INTER TYPE = hidden NAME = id VALUE = "1">
```

Өңдеушіге 1 таңбасымен жіберілетін ауыспалы Id индексін анықтайды. TYPE – reset батырмасын басқан кезде, форма бастапқы қалпына келетін батырманы анықтайды. Бұл батырманы қолданған кезде өңдеушіге мәліметтерді жібермей-ақ қоюға болады, reset типті батырманың name атрибуты болмауы да мүмкін. Мысалы:

```
<INPUT TYPE = reset VALUE = "поляны тазарту"> = = > поляны тазарту.
```

<SELECT> элементі:

<SELECT> менюі n элементтерінен алынған осылай болады:

```
<SELECT NAME="[аты]">
```

```
<OPTION VALUE="[мағынасы 1]">[мәтін 1]
```

```
<OPTION VALUE="[мағынасы 2]">[мәтін 2]
```

...

```
<OPTION VALUE="[мағынасы n]">[мәтін n]
```

```
</SELECT>
```

<select> белгісі ауыспалы атауын анықтайтын name атрибутын міндетті түрде құрайды. <select> белгісі MULTIPLE атрибутында құрайды. Оның болуы мәзірден бірнеше элементті таңдауға болатындығын көрсетеді. Көптеген браузерлер <select MULTIPLE> терезе түрінде көрсетеді, онда мәзірдің барлық элементтері орналасқан. Бірнеше элементтерді таңдау үшін «SHIFT» батырмасын басып тұрып, қажет элементтер үстінен тышқанмен шертесіз.

```
<SELECT MULTIPLE SIZE=3 NAME="[аты]">
```

```
<OPTION VALUE="[мағына 1]">[мәтін 1]
```

```
<OPTION VALUE="[мағына 2]">[мәтін 2]
```

```
<OPTION VALUE="...">[...]
```

```
<OPTION VALUE="[мағына n]">[мәтін n]
```

```
</SELECT>
```

```
<OPTION>
```

<OPTION> белгісі мәзір элементін анықтауға, VALUE атрибуты мәзір элементі таңдалған кезде өңдеушіге жіберілетін ұғымды айқындайды, <OPTION> белгісінің құрамында берілген элемент өздігінен белгіленгендігін көрсететін selected атрибуты болады.

Практикалық тапсырмалар

Тапсырма №1

1. Жаңа бет құрыңыз. (POST әдісін пайдалану арқылы) анкета – формасын құрыңыз, өзінде келесідей жолдар болуы тиіс (барлық ауысатындар оқулыуы керек, мысалы; орыс тілі – langru, C1 емес немесе T2 емес). Онда үндемей қалған мағынада сіздің мәліметтер орналасуы тиіс .

Тегі

Аты

Әкесінің аты



E-mail

Елді таңдау (SELECT элементін қолдану арқылы, 10 ел көрсетілуі тиіс).

Қаланы таңдау (міндетті түрде radio элементі көмегімен, ауыспалылар бірдей болуы керек 5 – тен кем болмауы тиіс).

Тілді таңдау (checkbox көмегімен ауыспалылар әр түрлі және 5-тен кем болмау керек)

Мамандық таңдау (SELECT MULTIPLE көмегімен , ауыспалылар әр түрлі және 10-нан кем болмауы тиіс)

Пароль

Қосымша ақпарат (міндетті түрде TEXTAREA көмегімен)

Жасырылған далада (hidden), student ауыспалысын «Сіздің – есіміңіз» ұғымымен жіберіңіз (student = сіздің есіміңіз)

Серверге ақпаратты құю батырмасы

Форманы тазалауға арналған батырма

Формадағы мәліметтер өңдеуші серверіне URL арқылы жіберілуі тиіс /internet/test/form.php.

Форманың сервердегі жұмыс қабілеттілігін тексеріңіз.

1. Беттің шығу жерінде, коментарий түрінде, формаға қолданылған тестер мен оның қасиетінің қандай мақсатта қолднуынын көрсетіңіз.

2. Жаңа бет құрыңыз, алдындағы беттің көшірмесін осында орналастырыңыз, GET-ке жіберу әдісін өзгертіңіз

3. 1 беттен осы 2 бетке сілтеме құрыңыз, мета мәліметтер қойып кетіңіз. на GET.

Тапсырма №2

1. Жаңа бет құрыңыз. Келесі іздеу жүйелеріне қарапайым іздеу формаларын құрыңыз

[www.yandex.ru](http://www.yandex.ru)

[www.rambler.ru](http://www.rambler.ru)

[www.aport.ru](http://www.aport.ru)

[www.google.com](http://www.google.com)

[www.filesearch.ru](http://www.filesearch.ru)

Өңдеуші URL- ы мен ауыспалыларды беттің шығу жеріне сәйкес іздеу қабілеттілігін тексеріңіз.

2. Формалардың серверде жұмыс қабілеттілігін тексеріңіз.

3. Беттерді өзіңіздің серверіңіздің біріктірілген түріне әкеліңіз , мета – мәліметтерді қойып шығыңыз.

## 4.2 Клиенттік қосымшалар құру технологиялары.

### 4.2.1 Java Script тілі

Гипермәтіндік жүйе секілді, веб-ті де екі көзқараспен қарауға болады. Біріншіден, Гипермәтіндік өту арқылы байланысқан көрсетілу беттерін салыстыру . Екіншіден, көрсетілу беттерін құрайтын қарапайым ары/тық объектілер жинағы ретінде.

JavaScript –ті қолдану HTML-құжатының авторына динамикалық парақты құрып және тұтынушының толтырған формасын өңдеуде маманданған программист болмауына мүмкіндік береді. JavaScript –желілік тұтынушы және серверлік қосымшаларды өңдеу үшін ұтымды және объекті-бағытталған тіл. JavaScript –тің екі түрі бар: Navigator JavaScript тұтынушы JavaScript-ті және LiveWire JavaScript серверлер JavaScript-ті. LiveWire JavaScript CGI-бағдарламалары сияқты серверлік қосымшаларды құруға мүмкіндік береді

Екінші әдісте гипертекстік желі қарапайым ақпараттық объектілер жинағы, HTML-беттерінің дайын компоненттер арқылы «іс жүзінде» құруға тйімді. Беттерді генерациялау кезінде веб-те клиент –сервер архитектурасы арқылы байланысқан дилемма пайда болады. Беттерді клиент бетінде де, сервер-де де гинерациялауға сервер генерациялау кезінде веб-те клиент-сервер архитектурасы арқылы байланысқан дилемма пайда болады. Беттерді клиент бетіндегі, серверде генерациялауға механизм қондырғысын құру арқылы жүзеге асады (server site Includes ). 1995 жылы Netscape компаниясы

*Java Script* бағдарламалау тілін ойнап тапты, бұл клиентте беттерді басқару мех-змі еді.

*Java Script* – клиент бетіндегі гипертекстік веб – беттер сценаийлік қарауды қадағалайтын бағдарламалық тіл. Сонымен қатар, *Live ware*, *Java script* туындысы, сервер бетінде Netscape орнату құралы болып табылады.

*Java Script*-тің қызыметіне мыналарды жатқызуға болады: дәл қазіргі уақыт немесе күн секілді мәліметтердің ауысуын көрсету.

- Браузер қолданушыға, күніне және т.б жағдайлардың өзгеруін бағдарламалау.
- Егер қолданушы тышқанды шертсе немесе элемент үстіне тышқан курсорын тақаса бетінің сыртқы элементтерінің түрі өзгерту.

Жоғары дәрежелі *Java Script*тің бұданда мүмкіншіліктері бар. Ол машиналық кодтар қатарында жұмыс істеуге мүмкіншілік беріледі, бірақ сіз веб – беттердің браузердің кейде браузер жұмыс жасайтын бүкіл жүйеге қосылу рұқсатына ие бола аласыз Java™ немесе C бағдарламаларымен салыстырғанда, *Java Script* компияциясыз болады, ол браузерге бағдарламалық кодты орындайтын виртуалды машинаны құю қажет емес. Сонымен бірге, *Java Script* Java немесе C++ - ке ұқсайтын объектік бағытталған архитектурада жұмыс істейді. Конструкторлар немесе базада протапиттерді иелену сияқты тілдің мүмкіншіліктері даму схемасы немесе бағдарламалық кодтуды көбірет қолдануға мүмкіншілдік беретін абстракциянын жаңа дәрежеге қосады. Веб – дамытушылардың *Java Script* – ті қабылдауының басты себептерінің бірі, *Java Script* бұрын тек сервер бетінде орындайтын функцияларды клиент бетінде де

орындаға мүмкіндік береді. Ең танымал мысалдардың бірі – формаларды тексеру. Бірнеше жыл алдын программисттер HTML формасындағы қолданушы енгізген мәліметтерді тексеру үшін: ақпаратты веб – серверге одан CGI сценарийге жіберу керек еді. Сол жерде енгізілген ақпараттар тексерілетін еді.

Егер мәліметтерде қате болмаса, CGI өз жұмысын жалғастырады. Ал қате табылса сценарий қолданушыға қате бар білімнің сипаттамасымен хат жіберген. Бұл процесс жұмыс жасағанмен, мұнда көптеген артық, шамадан тыс әрекеттер қолдану керек болады. Формаларды жіберу үшін серверден HTTP- ге арнайы сұраныс жасайды. Желіге мәлімет жіберген соң, CGI сценарийге тағыда сұраныс жасалады. Бұл процесс қолданушы форманы енгізу кезінде әр қателескен сайын қайталанып отырады. Қолданушы қате келген соң ғана біледі *Java Script* пайда болды. Енді формалар элементтерін веб серверге жіберерден алдын тексеруге болады. Бұл HTTP транзакциясының санын кішіреуіне әкеледі, сонымен қатар формаларды қайта толтыру кезіндегі кателер соны төмендетеді. Бір жағынан, *Java Script cookie*-ді оқуға және жазуға мүмкіндік береді. Бір уақыттары бұл операция тек веб – серверде тақырыпшалармен жұмыс істеуде ғана мүмкін еді.

### ***JavaScript-ті HTML- параққа қосу.***

JavaScript-бағдарламаларын HTML құжатқа келесідей әдістер арқылы қосуға болады:

1. <SCRIPT> тегін қолдану.
2. бағдарлама мәтінін құрайтын JavaScript-файлды жариялау.
3. HTML-атрибуттары үшін мән ретіндегі JavaScript-өрнекті анықтау.
4. Оқиғаны өңдеуші ретіндегі JavaScript-бағдарламаны жариялау.

### ***SCRIPT тегін қолдану.***

Тег <SCRIPT> - JavaScript-операторларын өзіне қоса алатын HTML-дің кеңейтілуі.

<SCRIPT> JavaScript операторлар ... </SCRIPT>

Құжаттар SCRIPT тегінің кез-келген саны болуы мүмкін. SCRIPT тегінде JavaScript операторлары арналған браузердің нұсқасын анықтауға болады:

<SCRIPT LANGUAGE= "JavaScript"> Navigator 2.0 үшін JavaScript-ті анықтайды.

<SCRIPT LANGUAGE= "JavaScript1.1"> Navigator 3.0 үшін JavaScript-ті анықтайды.

LANGUAGE сынында көрсетілген, қолданушы браузері JavaScript деңгейінде ұстаныстары жоқ, <SCRIPT> тегі аумағында операторлар еленбейді

Navigator 3.0 ерекшеліктерінен тұратын бағдарлама жазу үшін LANGUAGE сынын қолдануға болады, егер де қолданушы Navigator 2.0 болса, ол бағдрлама қате болып саналады. Келесі мысал LANGUAGE сынын қолданудың кейбір тәсілдерін көрсетеді.

1 Мысал. Бұл мысал JavaScript 1.0 және JavaScript 1.1 үшін функцияны қалай екі рет анықтау керек екенін көрсетеді.

```
< SCRIPT LANGUAGE= "JavaScript" > // JavaScript 1.0 </SCRIPT> үшін функция
< SCRIPT LANGUAGE= "JavaScript1.1" > // JavaScript 1.1 үшін дәл сол функция
// тек қана 1.1 </SCRIPT> нұсқасы үшін функция
<FORM > <INPUT TYPE="button" onClick="doClick(this)"
...> ... </FORM>
```

2 Мысал. Бұл мысал JavaScript құжаттың әр-түрлі нұсқасы үшін қалай қолданылады соны көрсетеді, яғни JavaScript 1.0 және JavaScript1.1 үшін. Реті бойынша құжат JavaScript 1.0 үшін жүктеледі. Егер қолданушыда Navigator 3.0 болса, парақты replace-тәсіл ауыстырады.

```
< SCRIPT LANGUAGE= "JavaScript1.1" > // 1.1 нұсқасында жұмыс жасағанда ауыстыру
```

```
location.replace("js1.1/mypage.html" ) </SCRIPT>
```

[мұнда 1.0-мен сәйкес операторлармен жұмыс жалғасады ... ]

3 мысал. Бұл мысал нұсқаны анықтау үшін, navigator.userAgent көмегімен браузерді қалай тексереді.

```
<SCRIPT LANGUAGE="JavaScript"> if (navigator.userAgent.indexOf("3.0")!= -1)
jsVersion = "1.1" else jsVersion = "1.0" </SCRIPT>
```

[ Содан соң JavaScript 1.1 үшін кез-келген нәтиже қолданылмас бұрын jsVersion айнымалысы тексеріледі]

### ***Коментария аумағындағы бағдарламаны жасыру.***

Тек қана 2.0 Netscape Navigator нұсқасынан бастап JavaScript-ті оқиды. Басқа браузер JavaScript –ті елемейтінін кепілдеу үшін, HTML-каментариялар аумағына SCRIPT-операторларын енгізуге болады, төменде көрсетілгендей:

```
<SCRIPT>
```

```
<!-- Мұнда ескі браузерден жасурк керек SCRIPT-операторлары басталады.
```

```
JavaScript операторлар...
```

```
</SCRIPT>
```

```
// мұнда жасыру аяқталады. - >
```

Navigator керекті тәсілмен SCRIPT сынын түсіндіреді және қос слешпен басталатын жолды еленбейтіндей қылады (//). Қазіргі уақытта JavaScript-тегі бағдарламаны түсіндіретін браузерлер көптеген қолданушыларда бар, бірақ сіз қолданушылар мен ескі браузер арасында қарама-қайшылық тумау үшін келесі техниканы қолданамыз. Ескерту. Қарапайымдылық үшін кітаптағы кейбір мысалдардағы бағдарламалар жасырылмаған, яғни арнайы Navigator 2.0 үшін жазылған. Мысал:

```
<HEAD>
```

```
<SCRIPT LANGUAGE="JavaScript">
```

```
<!-- Мәтінді ескі браузерден жасырамыз.
```

```
document.write ("I am from JavaScript!")
```

```
// жасыруды аяқтаймыз -->
```

```
</SCRIPT>
<P>This is HTML-text.
</BODY>
```

Бұл SCRIPT Navigator-да келесіні көрсетеді:

I am from JavaScript!

This is HTML-text.

Бірінші беттің JavaScript –пен, ал екінші беттің HTML-мен жасалғандығында қолданушыға ешқандай айырмашылық жоқ екенін ескерейік. Оператордың соңында нүктелі үтір (;) тек қана ол оператор сол жолда жалғыз болмаған жағдайда қолданылады.

### ***JavaScript үшін файлды анықтау.***

<SCRIPT> тегасындағы SRC сыны файл JavaScript оператордарының көзі екенін анықтауға мүмкіндік береді. Мысалы:

```
<HEAD>
<TITLE>My Page</TITLE>
<SCRIPT SRC="myprog1.js">
```

...

```
</SCRIPT>
```

```
</HEAD>
```

```
<BODY>
```

...

Бұл сын парақтардың саны көп болған жағдайда функцияларды бөлу үшін ерекше тиімді. JavaScript операторлаы егер файлда қате жоқ болғанда, <SCRIPT> тегасы аумағында SRC сыны арқылы ескерілмейді. Мысалға, сіз келесі операторды <SCRIPT SRC="..." > және </ SCRIPT > арасына орналастыруға болады:

```
document.write("Included JS file not found")
```

Бұл оператор тек қана жүктеу кезінде қандай да бір қате кеткен жағдайда ғана орындалады. SRC сыны кез-келген URL-ді анықтай алады, салыстырмалы немесе абсолютті Мысалға:

```
< SCRIPT SRC= "http://home.netscape.com/functions/jsfuncs.js ">
```

JavaScript –тің сыртқы файлдарында тек қана JavaScript фунуция анықтамалары және операторлар болуы керек, бірақ ешқандай HTML-тегі болуы мүмкін емес. JavaScript сыртқы файлдарының кеңейтілуі .js болуы қажет,және олар орналасқан сервер арнайы күйге келтірілуі тиіс.

### ***JavaScript-өрнекті HTML-атрибуты мәні ретінде қолдану.***

JavaScript-ті қолдану кезінде HTML тегі үшін мәні ретінде JavaScript-өрнекті қолдануға болады. Бұл кішкене жұмсақрақ HTML-парақ құруға мүмкіндік береді, яғни бір HTML элементтің сыны парақтаға алдыңғы элементтерге қатысты болады. HTML тілінде арнайы мағынасы бар символдарды мәтінді құжатқа енгізу үшін, тілдің стандарты арнайы конструкцияны ұсынады. Олар келесідей объекттерден тұрады:

- символ & (амперсанд);
- сандық код немесе символдық есім;
- символ ; (нүктелі үтір).

Мысалға, \$GT ұсынушысының көмегімен "үлкен" ( > ); және \$LT ұсынушысының көмегімен "кіші" ( < ) символын енгізуге болады. Мұндай кезектесулердің қолданылуы, көрсетілген символдардың HTML тіліне бекітілгенімен байланысты. Сонымен қатар JavaScript-объектілері & символдан басталып, нүктелі үтірмен (;) аяқталады. Аты немесе номерінің орнына жүйелі жақшаға { } алынған JavaScript-өрнек қолданылады. JavaScript-объектілерді тек қана HTML тегін қайда қолдану керек сонда қолдануға болады. Мысалға, құжатта JavaScript құралдарымен анықталған мәні 50-ге тең barWidth айнымалысы анықталсын. Енді келесі тәсілмен терезенің жартысынан горизонтальды түзу жүргізейік:

```
<HR WIDTH="{barWidth}%" ALIGN="LEFT"
```

Басқада HTML-элементтер сияқты орналастыру өткеннен кейін, және оның қайта жүктелуінен кейін парақтың түрі өзгеруі мүмкін. Келесі мысал құжаттың HEAD бөліміндегі қарапайым функцияны анықтайды және оны құжаттың BODY бөлігінде шақырады:

```
<HEAD>
<SCRIPT LANGUAGE="JavaScript">
<!--Ескі браузерлерден жасыру
function myfunc(number) { return number + number }
// Жасырудың соңы -->
</SCRIPT>
</HEAD>
<BODY>
<SCRIPT>
    document.write("Two arguments is ", myfunc(7), ".")
</SCRIPT>
<P> OK !
</BODY>
```

myfunc функциясында number деп аталатын, жалғыз аргументі бар. Ол функция аргументінің екі еселенген мәнін алатын, return number+number жалғыз операторының тұрады. 7 аргументімен функция шақырылады. Экранға шығарылатын нәтиже келесідей болады:

Two arguments is 14.

OK !

Myfunc функциясында Navigator-ға нәтижені шығару үшін document.write(...) операторы қолданылады.

## 4.2.2 Visual Basic Script

*Visual Basic Script* – *Visual Basic* топтамаларының бірі. *VB Script Int* – пен байланыста есептерді шешуге мүмкіндік береді, соның ішінде web – беттегі

объектілерді басқару сценарийлер басқармасын құру. *VB Script* көмегімен жеке беттерді тез арада құруға және ойын жазуға болады, мұның барлығы HTML – құжаттың ішінде орналасады. *VB Script* кейбір ерекшеліктеріне тоқталып өтейік.

*VB Script* пен *Java Script* –тің айырмашылығы мынада:

- оператор соңында нүктелі үтір (;) қойылмайды
- теңдік (=) белгісі мағынаны құбылмалы етеді + және операцияларды салыстыруға қолданылады
- операторлар блоктарды тік жақшалармен ([ ]) кілттік өздер жұптармен көрсетіледі: if ... endif, do... loop және while ... wend

Және де *Java Script* - те, бұл таңбалар қажет жағдайда енгізіледі, ал олардың типі контекст арқылы анықталады.

*VB Script*-тің басты кемшілігі, тек Internet Explore арқылы жұмыс істейді. сондықтан *VB Script* бағдарламалау кезінде клиент бетінде қолданулуда шектелуі.

(Visual Basic Scripting Edition ) VBScript - бұл Visual Basic-тің ықшамдалған нұсқасы. Әйтсе де, бастапқы Visual Basic кең мүмкіндіктері болмаса да, ол сіздің WEB парақтарыңызды "тірілту" үшін қолданыла алатын қуатты және ыңғайлы құрал болып табылады. VBScriptтерде жұмыс істеу өте оңай және керемет тиімді. VBScript зерттеуде өте қарапайым.

Web -бетшеге VBScript сценарийің орнатуы

VBScript және Java Script тілдерінің сценарийі HTML ге кеңейту сияқты жасалған. Броузер web -бетшенің қалған бөлігімен бірге сценарий алады және броузердің өзі сценарий талдап орындауы керек. HTML бетке сценарий енгізуі үшін қолданылатын SCRIPT тегі қосылды.

Тег <SCRIPT>

Бетте сценарий тегтер булардың іші орналасуы керек <SCRIPT> және </SCRIPT>. Мысалы:

<HTML>

<HEAD>

<TITLE> VBScript те жұмыс</TITLE>

<SCRIPT LANGUAGE="VBSCRIPT">

MsgBox "менің бетшеме қош келдіңіз!"

</SCRIPT>

Ашылатын тег LANGUAGE сценарийдің қолданылатын тілін көрсететін аргумент қосады. Ол бірнеше сценарий тілдері бар болатындықтан қажет. Бұл аргументсіз броузер сценарий қай тілде жазылғанын (бұл үндемеу бойыншаға Java Script) білмейді.

## Практикалық жұмыс №6

**Тақырып:** SSI көмегімен динамикалық беттерді құру.

**Мақсаты:** SSI қолдануды ұйымдастыру және қалыптастыру.

**Теориялық мәліметтер:**

**Server Side Include.**

SSI – Server Side Include болып шифрленеді, HTML бетіне енгізілген командалар жинағы, командалар элемент жасағын сұраныс жіберілерден алдында орындалады.

SSI – директивы бар құжатқа сұраныс алған соң, HTTP – сервер құжатты өңдейді, барлық SSI – директивтерде орындайды, сонан соң, нәтижесінде шыққан құжатты клиентке қайтарады.

HTTP – сервер SSI – директивте орналасқан құжаттардың барлығын автоматты түрде тексермейді, тек сервер настройкасында көрсетілген құжаттар ғана тексереді. (біздің жағдайымызда, shtml орамдарының барлығы).

SSI – директивтер html-кодқа белгілі бір форматтағы коментарилер түрінде енгізіледі:

```
<!--#SSI_директива=" параметрлер" - ->
```

### **echo var**

Ауыспалы орта ұғымын көрсету үшін қолданады:

```
<!--# echo var =" ауыспалы " - ->
```

Мысалы:

```
<!--# echo var ="REMOTE _ADDR" - >
```

Кейбір ауыспалылар

REMOTE \_ADDR = сіздің SP мекен – жайыңыз

REMOTE\_Port = сіздің портыңыздың номері

HTTP \_USER \_AGENT = сіздің браузеріңіз

SERVER \_NAME –сервер атауы

SERVER\_PORT – сервер портының атауы

SERVER\_SOFTWARE – серверге қондырылған HTTP түрі

SERVER\_PROTOCOL – сервер байланысатын протокол

SERVER\_ADMIN – сервер администраторы

HTTP\_REFERER – қайсы беттен келшеніңіз

DOCUMENT\_URI – URI құжаты

DOCUMENT\_NAME – құжаттағы файл атауы

QUERY\_STRING – GET әдісілі сұраныс жасалған құжат

LAST\_MODIFIED – файылдың соңғы мадификация уақыты

### **include file**

#### **include virtual**

Берілген команда орындалып жатқан html – файлдағы көрсетілген файлды сілтейді.

Include file мен Include virtual – дың айырмашылығы:

Include file – жүріп жатқан директориядағы файлға жол

Include virtual – құжатқа сервердегі виртуалды жол

Мысалы:

```
<!--# Include file="docext , html" - >
```

```
<!--# Include virtual="\counters\spylog.txt" - >
```

```
<!--# Include virtual="\cgi-bin\counter.pi" - >
```

Егер Include file немесе Include virtual көмегімен cgi – скрипт қосылса , онда беттің құрамына cgi- скритегі ақы-р емес, оның нәтижесі түседі. Бұл



дерективтер көмегімен бірдей беттерді автоматты түрде жинақтай аласыз. Егер сіздің бетіңіз бірнеше блоктан тұрып, ол кейбір блоктар барлық беттерде бірдей болса, онда бұл блоктарды бөлек файлдарға біркiтiруг, ол беттерге осы дерективтерді орналастырсаңыз б/ды. Осыған орай сіз бір файлға ғана өзгересіз, соның негізінде барлық беттердегі ақы өзгереді.

### **fsize file**

Көрсетілген файл көлемін енгізу?

Мысалы:

```
<!= #f size file =”komto shtml”- >
```

### **printenV**

Барлық ауыспалы ортаны бетке шығ\ды. Параметрлері жоқ . Өзіңізге қажет барлық ауыспалылар тізімін көре аласыз.

Ауыспалылар ортасының тізімі:

```
<!= # printenV- >
```

### **Setvar**

Ауыспалылар ортасының қыз – н орналастырады. Форматы:

Setv =” Ауыспалы атауы” value =” мағынасы“.

TITLE ауыспалының мағынасы:

```
<!= # Setvar =” TITLE” value =” Зертханалық жұмыс ”- >
```

SERVER\_NAME ауыспалысының мағынасы :

```
<!--# echo var =” TITLE”- >
```

Соңғы директива «зертханалық жұмыс 2» - ні шығарады. Егер сізде «шапочка» (<head></head>) бөлек файлда орналасса және онда сайттың негізгі <title> көрсетілсе бұл деректив кемегі сіз <title> - де толықтыра аласыз.

Мысалы:

```
<head></head> орн-н жеке файлда мынаны жазамыз:
```

```
<title> Вася Пупкиннің сервері : <!--# echo var =” TITLE”- ><title>
```

Алдындағы файл енгізілетін файлғы (директиваны орнындағанда дейін) жазамыз:

```
<!= # Setvar =” TITLE” value =” Зертханалық жұмыс ”- >
```

Нәтижесінде мынаны аламыз:

```
<title> Вася Пупкиннің сервері Зертханалық жұмыс <title>
```

### **Ifelse**

Бет нәтижесін шартты түрде басқаруға қолд\ды синтаксисі мынадай:

```
<!= #ifexpr = “ шарт ”- >
```

Егер шарт рас болса шығарылатын HTML – код:

```
<!= #elifexpr = “ шарт2”- >
```

Егер шарт1 жалған болып шарт 2 рас болса шығарылатын HTML – код:

```
<!= #else - >
```

Егер барлық шарттар жалған болса шығарылатын HTML – код: endif

```
<!= #else - >
```

Шарт – бұл рас болып табылатын жол, яғни жолдарды салыстырған операторлар жинағы. Операторлар =, !=, <, <=, >, >= болуы мүмкін . Егер екінші жол “\” алынған болса онда жол рас деген өз;

&& (u) және (или) операторларының көмегімен бірнеше операторларды салыстыру мақсатында біріктіруге болады. Шарттарды топтау үшін “( )” (жақша) қолданылады.

Егер `string 2 \string\` формасындай болса ол тұрақты тіркес ретінде интерпритерланды.

Мысалы:

```
<!= #ifexpr = “HTTP_COOKIE = \user\ = old / “- >
```

Берілген мысалмен COOKIE – та расында да `user = old` бар немесе жоқ екендігін тексереміз:

### **goto**

Goto командасы label белгісіне өту операторы:

Команда форматы: `<!= # goto= “<label >“- >`

`<label >` - белгілі атауы

мысалы

```
<!= # label = “testlabel” - >
```

Бұл жол көрініп тұрады.

Ескерту `<!= # goto` мен = “ таңбаларының ортасында міндетті түрде пробел болуы керек.

### **label**

label командасы goto және if.. goto командалары қолданылатын құжатта белгі қолданылады.

Команда формасы `<!= # label= <label >” - >`

`<label >` - дің орнында – құжатта орында идентификациялайтын s1 таңбадан аспайтын жол . Белгіні қою құжатты форматтауға әсерін тигізбейді.

Ескерту : `<!= # label` және = “ таңбаларының арасында міндетті түрде пробел болуы қажет.

### **Breaх**

Breaх командасы кездескен бойдан құжатты шығару тоқталады

Мысалы:

```
Бұл жолы көрініп тұрады: <!= # Breaх ” - >
```

### **Практикалық тапсырмалар.**

#### **Тапсырма 1**

1. Файлдарды сақтау үшін `\ssi\` каталогын құрыңыз.
2. Онда бос файлдарды құрыңыз (мысалы: `top,html,bottom,html` т.б.)
3. 1 бетті өзгертіңіз: жоғары ж\е төмен оң жақ ж\е сол жақ бөліктерін сәйкес өзіңіз құрған файлға орналастырыңыз `shtml` – де кеңейтуді ұмытпаңыз.
4. Серверде жұмыс жасау қабілеттілігін тексеріңіз
5. `<title>` динамикалық түрде форматталуын орындаңыз.
6. 1 бетте келесі ақпаратты шығарыңыз:

Файл көлемі

Файлдағы соңғы мадификациялық мезгілі

Сіздің IP адрес

Сіздің порттың номері

Сіздің браузер

Сервердің аты  
Сервер портының номері  
Серверде қандай HTTP қондырылған  
Сіз келген бет  
Құжат URI - i

Жүріп жатқан файл атауы

7. 1 бет пен енгізген файлдың шығу жерінде комментарий түрінде SSI команданын не үшін қажет екендігін көрсетіңіз.

### **Тапсырма 2**

1. Жаңа бетті құрыңыз. Барлық ауыспалылар орнын көрсетіңіз
2. Серверде жұмыс қабілеттілігін тексеріңіз.

### **Тапсырма 3**

1. SSI көмегімен барлық беттерде ортақ бір түрге келтіріңіз, яғни барлық беттер үшін жоғарғы және төмен бөлігі бірдей болуы тиіс.
2. Берілген құжатқа сәйкес әрбір бетте <title> - ға толықтыру жазыңыз.
3. Серверде жұмыс жасау қабілетін тексеріңіз.

## **4.2.3 Java тілі**

*Java* тілінің пайда болуы - соңғы жылдардағы бағдарламалау арасындағы басты жетістіктердің бірі HTML2 – тілі «Дүние жүзілік өрмекте» WWW ( World Wid Web) беттерді статистикалық орнатуда қажет.

*Java* тілі Int. Желісіне интерактивті продуктілерді құру үшін қажет болады.

*Java* тілі технологиясында ікілттік элемент біріктірілген және бүгінгі таңдаудағы барлық тілдерден ерекше етті. *Java* кеңінен қолдануда оплеттерін ұсынады, кішігірім, сенімді, динамикалық, веб-беттерге енгізілетін активті желілі қосымшаларға тәуелді емес платформалар *Java* кез - келген HTML2 – құжаттар секілді қолданушылар арасында жеңіл қолданылады және тез таралады.

*Java* шығарулар арасында сенімді және ыңғайлы қызыметпен қарақайым таныс синтактисті біріктіріп, приложениелерді шығаруда объектік – ориентирациялық күшті шығарады. Бұл бағдарламашылар тобына жаңа бағдарламалар мен жаңа апплеттерді тез арада құруға мүмкіндік береді.

*Java* бағдарламашыларға терезелермен, желілермен және енгізу-шығарумен жұмыс істегенде қолданатын көп жүйелі функцияларды абстрактар жинағын ұсынады.

*Java* тілі әртүрлі приборларға жіберу бағдарламасын құру проекті үшін шығарылған. Проекті құру C++ тілінде басталып, ол тиімсіз болған соң – бағдарламалау тіліне ауыстырылады.

Әртүрлі операциондық жүйеде әртүрлі процессор арқылы қолдануға болатын және әрбір арбір архитектура үшін жеке қолданылатын әмбебап - тәуелсіз бағдарламалау қажет.

## Java апплекттері

Апплет – сурет, дыбыстық файл, мультипликация элементті сияқты желі арқылы динамикалық түрде құйылатын кішігірім бағдарлама.

Апплеттің басты ерекшелігі, мультфильм немесе басқа ақпараттарды сақтайтын файл форматы емес, бағдарлама түрі болып табылады.

Апплет әрдайым сол бір сценарийді көрсете бермейді, қолданушының нұсқауы бойынша өзінің сценарийін өзгертіп отырады.

*Java* өзінің қолданушыларын арнайы анимацилар арқылы web – беттерге еліктіріп отырады. Қолданушылар *Java* аспектілерін бірден меңгермегендіктен, олар мұны басқа динамикалық суреттерді құятын технологиялармен және web – клиенттер мен әрекеттесуін салыстырып отырады.

*Java*-ны ойлап тапқан бағдарламашылар тілдің гиторогендік ортада мәселелерін шешу үшін мынадай болу керек деп көрсетті.

- қарпайым – оны барлық қолданушылар жеңіл қолдану керек
- анық – қолданушылар ешбір қиындықсыз *Java*-ны меңгеріп алулары қажет
- объекті – ориентирленген – ол қазіргі ПО әдістемелігінің барлық мүмкіншіліктерді қолдануы және жіктелген клиент – серверлік приложенияларды жазуы керек.
- көпбағытты – көпмезгілде әрекетті орнайтын болуы керек
- интерпретирленген – орын алмастыруға және динамик болуға.

*Java* тілінде сіз есептерді шешу үшін көрсетілген нұсқаулар санаулы ғана болады. Тілді тым қарапайым қылу интерпреттор командалары секілді ұсқынсыз және нәтижестікке әкеліп соқты. *Java* мұндай тілдер қатарына ілінбейді, ол сізге операциясының барлық күштілігін және кітапханалар тізімін ұсынады. Кәзіргі кезде интернет өзімен қатар, оның қауіпсіздік шаралары көп талқыланады. *Java*-ны құрудың тағы бір мақсаты заңсыз қосылудан сақтау болып табылады. *Java* шамадан тыс функцияларға және кез-келген жүйелерге өз ресурстарына қоснуға жол бермейдісондықтан *Java* тілінің қорғаныс қызыметі басқа тілдермен салыстырғанда жоғары дәрежеде болды *Java*-ның объект модельдері тез таралды,бірақ *Java* да сандар мен басқа қарапайым мәліметтер типі объектілер қатарына кірметді. *Java* бірнеше кілттік бөлімдермен шектеледі , осылайша жасалған қатенің бағдарламада дер кезінде көрсетіліп отыру мүмкіндігін береді. Сонымен қатар мұнда басқа тілдермен салыстырған көптеген қате көздері болмайды. Бүгінгі таңда қолданылып жүрген бағдарламалардың көптеген түрі 2 жағдайда жұмыс істелмейді: белгілі бір ережелерге сай келмесе.немесе жадты бөлген кезде. *Java* бұл екі мәселені де жаққа шығарады, яғни онда бос емес жадқа арналған арнайы қоқыс жинаушы енгізілген және ерекше ережелермен жұмыс жасайтын арнайы объектік – ориентациялық құрал қондырылған

*Java* ның тағы бір ерекшелігі – өзінің кодын, өзі бойына енгізген барлық платформаларда енгізу, өйткені онда байт-код деп аталатын арнайы бөлімше бар. Байт – *Java* құрылған барлық жүйеде интерпритерленді. Ерте құрылған платформалардың тәуелсіздікпен қамтамасыз етілген жүйелердің көбіне, өз қабілеттілігін төмендету кемшілігі болған. *Java*-да инспретатор

қолданылғанмен, байт-кодөзінің «туған» машиналық тіліне іс жүзінде жеңіл ауыса алады. Бұл кезде де жұмыс жасау қабілеті жоғары болады.

*Java* тілі командалық интерпретатор тілімен салыстырғанда күрделі болғанмен меңгеру кезінде оңай.

*Java* ортасы тек бағдарламалау тілі ғана емес. Оған сіздің бағдарламаңыз кездесетіншын дүние абстракциясын сақтайтын кілттер тізбегінің жиынтығы енгізілген. *Java* ның басты мақтанышы – платформаға тәуелсіз ететін абстракциялық - класстардың енгізілгені.

### Бақылау сұрақтары:

1. Клиент машиналары үшін интернет - қосымшаларыды құрудың артықшылығы неде?
2. *Java Script* технологиясының ерекшелігі неде?
3. *VB Script* технологиясының ерекшелігі қандай?
4. *Java* технологиясының ерекшелігі қандай?

### Зертханалық жұмыс №7

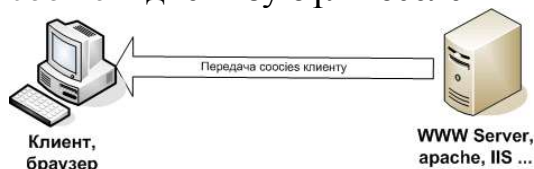
**Тақырып:** *Cookie*сті қолдану .

**Мақсаты:** *Cookie*сті қолдану қабілеттігін қалыптастыру.

**Теоретикалық мәліметтер**

#### **Cookie**s

*Cookie* HTTP спецификацисы мәселесінің басты шешімі болып табылады. FTP немесе Telnet сессиясындағындай клиент пен сервер арасындағы бастықтастықтың тұрақсыздығы, яғни HTTP хаттамасы арқылы ақпаратты жібергенде әрбір құжат үшін дара сұраныс жасалынады. HTTP- хаттамаға *cookie* – ді енгізу бұл мәселені шешуге мүмкіндік береді.



*cookie* арқылы сервермен клиентке жіберу



*cookie* арқылы клиентпен серверге жіберу

**Cookie** – бұл сервер клиентке жіберетін ақпараттың кішігерім бөлігі. Клиент (браузер) бұл ақпаратты сақтайды, сонан соң әрбір сұраныстан соң серверге http-нің бір бөлігі ретінде жіберіп отырады. Кейбір *cookie*-лер тек бір қолдануға ғана жарамды, олар браузер жабылған соң жайылып кетеді. Ал, басқалары, бірнеше периодқа қондырылғандар, файлға жазылып қалынады. Бұл файлдарды компьютердегі каталогта көре аласыз. `\Documents and Settings\Қолданушының_аты\Cookies\`.

Өздігімен cookies ештеңе істей алмайды, бұл тек кейбір ақпараттар жинағы ғана. Бірақ сервер cookies-те сақталған ақпаратқа әсер етуі мүмкін. Мысалы: WWW-дағы кез-келген ақпаратқа автоматты түрде қосылса, cookies-те сессия мезгілінде login және password сақталады, бұл әрбір сураныс жасағанда оларды енгізбеуге мүмкіндік береді, келесі бір мысал: cookies –ті жеке беттерді құру үшін қолдануға болады. Көптеген серверлердің «-ші бетінде кіргенде "Сәлем, Вася Пупкин!" секілді өздер жазады. Cookies-ті онлайн дүкендерінен тіркеу функциясы үшін де қолданылады.

Cookie http-нің бір бөлігі болып табылады. HTTP-нің Set-Cookie жолының толық сипаты:

```
Set-Cookie: NAME=value; EXPIRES=date; DOMAIN=domain_name; PATH=path; SECURE
```

Set-Cookie-дің HTTP-дағы қысқаша сипаты:

```
Set-Cookie: NAME=VALUE;
```

NAME=VALUE - NAME - ауспыялы (аты cookie), VALUE - мағынасы.

expires=DATE – сақтау мерзімі cookie (жазу форматы "expires=Monday, DD-Mon-YYYY HH:MM:SS GMT").

Мысалы:

```
expires=Friday,31-Dec-99 23:59:59 GMT;
```

Егер бұл атрибут көрсетілмесе, cookie браузерді жапқанға дейін сақталады.

domain=DOMAIN\_NAME – cookie мағынасының домені. Мысалы, "domain=ipm.kstu.ru".

Егер бұл атрибут көрсетілмесе, сервердің өздігінен енгізілген доменді атау қолданылады.

path=PATH – бұл атрибут құжатқа негізгі жолды орналастырады. Мысалы "path=/students/pupkin/" сілтемесі Васи Пупкиннің /students/pupkin/ директориясындағы барлық құжаттар үшін қолданылады (Сервердегі құжаттардың барлығы үшін cookie). Cookie сервердегі құжатқа жасалған сұраныстың барлығы жіберілуі үшін, сервердің негізгі каталогын көрсету керек ("path=/").

Егер атрибут көрсетілмесе, онда cookie тек нұсқалған директорияға сәйкес құжаттарға сәйкес жіберіледі.

secure – егер бұл маркер орнатылса, онда cookie ақпараты тек HTTPS (HTTP түрімен қолданып SSL - Secure Socket Level) арқылы жасырынды түрде жіберіледі. Егер бұл маркер көрсетілмесе, ақпарат қарапайым түрде жіберіледі.

### **HTML көмегімен cookie-ді қондыру.**

Cookie-ді орнатудың қарапайым әдісі- HTML құжаттың <HEAD> </HEAD>

элементіне сәйкес кез-келген META-тегді қолдану. Көрінісі былайша:

```
<META HTTP-EQUIV="Set-Cookie" CONTENT="NAME=value; EXPIRES=date; DOMAIN=domain_name; PATH=path; SECURE">
```

HTML көмегімен бірнеше cookie-ді бірден орнатуға болады:

```
<META HTTP-EQUIV="Set-Cookie" CONTENT="NAME1=value1;
```

```
EXPIRES=date1; DOMAIN=domain_name1; PATH=path1; SECURE">
```

```
<META HTTP-EQUIV="Set-Cookie" CONTENT="NAME2=value2;
```

EXPIRES=date2; DOMAIN=domain\_name2; PATH=path2; SECURE">

HTTP – серверден құжатқа сұраныс жасағанда, браузер Cookie сервер доменіне сәйкестілігін тексереді, егер сақталу мерзімі өтіп кетсе, ол жайылады. Егер cookie-тің мағыналығыбарлық ережелергесай болса, браузер оларды серверге жұп түрінде жібереді аты/мағынасы:

Cookie: NAME1=STRING1; NAME2=STRING2 ...

Қосымша мәлімет: Егер cookie браузерде бұрынна ораналасатын NAME, domain и path, параметрлеріне сәйкес жаңа мағыналы қабылдаса, ескісін алмастырады. Ал, басқа жағдайларда cookie-дің жаңа мағынасы ескісіне қосылады.

Expires-ті қолдану cookie-дің берілген уақытта сақталуына кепілдік бермеуі, браузер жазбаны бос орын болмағандықтан немесе өзге себептерін өшіріп тастауы мүмкін.

Клиент (браузер) cookies-тің жұмыс жасағанда кейбір шектеуліктерге ие, мысалы:

Cookies-дің 300 мағыналысы ғана сақталады

әрбір cookie 4Кбайттан асып түспеу керек

бір серверден немесе доменде 20 cookie мағыналысы сақталу керек

Егер 300 бен 20 шектеулігі асып кетсе, алдыңғы жазбалар өшіріледі. 4Кбайт лимиты шектеуден артылса, жазбаның кейбір бөліктері үзіліп кетеді.

SSI көмегімен cookie-ді қабылдау немесе қолдану.

Серверде SSI бар болса <!--#echo var="..."--> көмегімен кез-келген ауыспалылар ортасын санауға болады, сонымен қатар cookie-мен берілген мағыналарды да: (ауыспалы HTTP\_COOKIE).

Мысалдар:

```
<!--#echo var="HTTP_COOKIE"-->
```

Барлық алынған cookies-ді құжатқа енгіземіз.

```
<!--#if expr="$HTTP_COOKIE=/user\=old/" -->
```

Бар-жоқтығын тексереміз cookies "user=old".

```
<META HTTP-EQUIV="Set-Cookie" CONTENT="<!--#echo  
var="QUERY_STRING"-->";">
```

GET әдісімен cookie-ді орнатуға арналған осы жолды қоямыз.

## **Практикалық тапсырмалар**

### **Тапсырма 1**

1. Сіздің «-бетіңіз қолданушыны танитындай етіп, сәлемдеме құрыңыз (сәлемдесу) if SSI-дағы директиваны қолданыңыз.

Толығырақ айтқанда:

Бетті бірінші рет ашқанда, клиентке cookie-ті жіберу керек.

Келесіде бетті ашқан-сайын клиент серверге cookie-ті жіберу қажет.

Сервер cookie-ді қабылдаған соң сәлемдеме жолдайды, әйтпесе сервер сізді «танымайтындығын» көрсетеді.

2. Серверде жұмыс жасау қабілетін тексеріңіз.

3. Комментарий түрінде, әрбір SSI командамының қолдану мақсатын көрсетіңіз.

### **Тапсырма 2**

1. Өзіңізді былай тіркейтіндей етіп бет құрыңыз:

- жаңадан келген
- орта білімді қолданушы
- сапалы қолданушы

(radio немесе select) формаларының көмегімен тіркеліңіз. Мәліметтер 2-ші бетке GET әдісімен жіберілуі тиіс.

2. 2-ші бетте ауыспалыны енгізіңіз, оның <META> тегіндегі мағынасы cookie қондыру үшін.

3. IF сайтқа кіргендігін қадағалайтын IF көмегімен cookie-ті өңдейтін бет құрыңыз.

4. Серверде жұмыс жасау қабілетін тексеріңіз.

5. Комментарий түрінде, әрбір SSI командамының қолдану мақсатын көрсетіңіз.

### **Тапсырма 3**

1. Сіздің cookie-ңізді жаңартатын бет құрыңыз, яғни бұл бетке сураныс жасағанда ммерзімі өтіп кеткен cookies-ңіз жіберілуі тиіс.

2. Бұл бетке сілтемені "submit" көмегімен тіркеңіз.

3. Серверде жұмыс жасау қабілетін тексеріңіз.

## **Практикалық жұмыс № 8**

**Тақырып: Каскадтық кестені GSS стилінде қолдану.**

**Мақсаты:** Каскадтық кестелер стилін пайдалана отырып HTML құжатын құра білу.

### **Теоретикалық мәліметтер**

Сіз алдыңғы беттерді каскадтық немесе GSS пайда болғанға дейін, осы сияқты етіп құрастырыңыз.

GSS-тің пайда болуына дейін сайт қызметшілері мынадай негізгі мәселелермен қақтығысып отырған:

- **Позиционирлеу мәселе (бетті орналастыру).** Дегенмен бұл үшін кесте қолданып отырды (олар ең бастапқы кезінен ол үшін арналмаған болатын), олар барлық мәселелерді шешпеді. Келесіні орындауға мүмкін емес:

- қандайда бір блоктың тіркелген көлемдерін көрсетуге болмайды.

Мысалы, мәтіндік блок толып кеткен жағдайда жүріп отырады (тіпті ол кесте ішінде болса да)

- Бетте блоктың тіркелген координаталар жағдайын көрсетуге болмайды.

- Бір блокты басқа блок үстіне орналастыруға болмайды. Мысалы суретті кестенің немесе басқа суреттің үстіне ығыстырып көріңіз.

- Екінші бір мәселе бетке әр кез көлем және шрифт түсін, ұяшықтар қызметін, кестелер т.б. енгізіп отыру болды. Бұл бет көлемін қатты үлкейтіп отырды, сондықтан ол жәй құйылады. Егер сіз түс немесе шрифт көлемін өзгерткіңіз келген жағдайда сізге барлық бетті өзгертуге тура келер еді.



- үшінші бір мәселе барлық браузерлер жеке өзгерістерге ие бола алады. Мысалы, барлық беттерді өзгерту кезінде браузерде шрифті үлкейтуге болады.

### **GSS-ті қолдану әдісі.**

Кесте стилін қолданудың үш әдісі болады.

- **Стильдердің ішкі кестесі ( inline style sheets)** – арнайы атрибут көмегімен HTML-

типтеріне тікелей орналастырады.

HTML-ге мысал:

```
<font color = `` blue `` size = ``3`` Arial``> Пример </font >
```

CSS-ге мысал:

```
< font style =`` color: blue; font-size: 12 pt; font-family : Arial``> мысал </font>
```

Ескере кететін жағдай HTML-ге қарғанда Inline Style Sheets коды үлкендеу болып шықты. Сондықтан CSS-ті белгілі бір элементті жеке стиль үшін қолданады.

Кез келген тегте бізге керекті стильдік параметрлерді біз стиль атрибутының қосымша көмегімен анықтай аламыз. Бұл ең жеңіл әдіс, және де ол тек бір ғана тег шегінде жүзеге асады. Бірақ елестетіп қараңыз егер біз әр бір тегтің стилін көрсетіп отырсақ, қаншалықты файл көлемі өседі және де қаншалықты оны өзгерту ыңғайсыз болады.

- **Стильдің глобальды кестесі (Global Style Sheets)**- стиль элементін барлық документте анықтайды.

Ол үшін тег қолданылады < STYLE type = `` text / css``>

Ол құжаттың тақырыбында орналасады < head>.

Мысал:

```
<html>
```

```
<head>
```

```
<Style type=*text\css*>
```

H1(color: red; font-style: italic; font-size 32px 3- стандартты теганы қайтадан анықтау.

Blue (color: blue)- жаңа классты анықтау.

```
<|STYLE>
```

```
</head>
```

```
<body>
```

Енді осы стилді html-кодының кез-келген орнында қолдануға болады. Ол үшін келесі конструкциялар қолданылады.

<h1> бұл тақырыпша ірі қызыл түсті курсивпен жазылған. </h1> міне =font class=\*blue\*> мынау </font> өз көк.

```
</body>
```

```
</html>
```

Берілген мысалда барлық H1 элементтері ірі қызыл түсті курсивпен, BLUE классымен көрсетілген барлық элементтер көк түспен жазылады.

**Стильдегі байланыстырылған кестелер (Linked Style Sheets)** - бірнеше құжаттарға бірден қолданылуы мүмкін, және сыртқы файлда сақталады.

## Сыртқы файлға мысал:

Файл main.css

```
body {background:black; font-size:9pt; color:red; font-family:Arial Black }
```

```
base{color:blue; font-style:italic }
```

```
h1 {color:white }
```

```
#bold {font-weight:bold }
```

HTML құжатының өзінде де сілтеме осы файл ішіне тега (LINK) көмегімен бөлінеді. Ол былай көрсетіледі:

```
<LINK rel="STYLESHEET" TYPE="text/css" HREF="путь до файла">
```

Мысал:

Файл Index.html

```
<html>
```

```
<head>
```

```
<LINK rel="stylesheet" type="text/css" href="main.css">
```

```
</head>
```

```
<body>
```

```
<LINK rel=STYLESHEET*TYPE=*text|css*> HREF=*файлға дейінгі жол*>
```

Мысал:

Файл INDEX.html

```
<Html>
```

```
<head>
```

```
<LINK rel=*type\css*href=*main.css*>
```

```
</head>
```

```
<body>
```

Құжат құрылысы:

```
<|body>
```

```
<|html>
```

Бұл ең қолайлы әдіс, және стильдің негізгі кестелерін дәл осы әдісте қолдану ұсынылады. Каскадтау стильдердің қайта анықталу мүмкіндігімен тұжырымдалады. Төмендегі көрсетілген стильдерді енгізу әдісінің тізімі қайта анықтау ережесіне сай келеді. Төмен орналастырылған әдіс жоғары орналастырылған әдісті қайта анықтауы мүмкін.

Стильдерді бетке енгізу әдісі:

Дара стильдік файлды қолдану және оны мега <LINK> көмегімен қою.

Стильді құжат тақырыпшасында сипаттау.

Стильді меге дағы параметр сияқты қолдану.

Мысалы, біз <p> жегіндегі мәтіні 10-пунктік бөліктіктегі шрифт көмегімен жазылуы керектігін сыртқы стильдік файлда анықтадық.

Егер біз <p> тегіндегі дәл сол мәтіннің 12 пункті шрифтке өзгертіледі, яғни бет тақырыпшасындағы стиль, сыртқы файл стиліне қайта анықталды.

Ол не үшін керек? Біздің бетіміздегібарлық сілтемелерге келесі стиль көрсетілген деп елестетейтік:

```
«style type = “text less”»
```

```
<!—
```

A. link ( text – decoration: none; түсі қызыл)

→

< / style >

< / head >

Яғни, тексттегі еренсілтеменің бәрі автоматты түрде қызыл түске ауысып, асты сызылып жазылуын тоқтатады. Біз сілтеменің соңында автор мекен-жайын көрсеткіміз келді, оны тым көңіл аударатындай емес қылып жасауы қажет. Мұны біз беттің бір шетінде, төменгі бұрышында көрсетуіміз тиіс. Локальды түрде түсті алмастырсақ ғана болғаны:

```
< a href = “ http: // www.pupkin.ru “ > < span style = “ түсі: # 000000,” > Вася Пупкин < / span > < / a > немесе былай:
```

```
< a href = “http: // www.pupkin.ru “ style = “ color : # 000000;” > Вася Пупкин < / a >
```

Біз мұны style атрибутының көмегімен орындадық, ал ол тек нұсқалған тек шегінде жұмыс істейді.

< span > < / span > тегі бұл стильді қолданатын шекті анықтайды.

## **Мұралану**

Мұралану

Кейбір мағыналар тегтер көмегімен (мұраланады)

H1 элементі бар, онда анықтайтын EM элементі орналасқан деп есептейік.

Егер EM элементіне түстің түрі берілмесе, онда “ тым ” өзі автоматты түрде “ аталық ” түсті қабылдайды. Осылайша, мысалы H1- ге көк түс нұсқалса, EM элементі де көк түсті қабылдайды.

Құжатта үнсіздік бойынша қолданылатын стиль қасиетін анықтау, берілген құжат бұтақтарының түпкі элементінде іске асыруға болады. Мысалы, HTML тілінде бұл мүмкіндікті HTML немесе BODY элементтерінің көмегімен жүзеге асыруға болады.

### **Мысал (дар):**

'color' түстік қасиеті тұрақсыз, сондықтан BODY элементінің «мұрагерлері»

'black' түсін қабылдайды:

```
BODY { color: black; }
```

пайыз түрінде берілген мағына қабылданбайды да, ал теңдік нәтижесі қабылданады.

### **Мысал (дар):**

Келесі кестелік стиль берілген болсын:

```
BODY { font-size: 10pt }
```

```
H1 { font-size: 120% }
```

және құжат фрагменті:

```
<BODY>
```

```
<H1> біреулер<EM>дардай</EM> тақырыпша </H1>
```

```
</BODY>
```

H1 элементінің 'font-size' қасиеті '12pt' теңдік мағынасын иеленеді (10 pt-дан 120%, «аталық» қасиетті иеленеді). Яғни 'font-size'-тың есептік қасиеті

«мұрагерлік» берілетіндіктен, ЕН элементі де '12pt' теңдік қасиетіне ие.

### **CSS синтаксисі**

CSS-дің барлық хабарландырулары (занды түрде "селекторлар" д.а.) жүйелік жақшамен жазылады:

ТЕГ: псевдокласс, класс {мінездеме 1: мағынасы 1; мінездеме 2: мағынасы 2}

Әрбір класстың сипаты төмендегідей конструкция көмегімен жүзеге асады:

```
.small { font-size: 9pt; }
```

Әуелі класс атауы көрсетіледі – ол еркін түрде болуы мүмкін, бірақ белгілі бір нақты атауды көрсеткен жөн. Содан соң жүйелік жақшада {} берілген классқа қажетті барлық параметрлер тізіліп жазылады. Параметрлер бір-бірінен нүктелі үтір арқылы ажыратылады Мысалы:

```
.small { font-size: 9pt; color: #eeeeee; text-align: center; }
```

Әмбебап класс – кез-келген тег-ке қолданылады, класс атауы нүктеден басталады.

Мысалы:

```
<p class=small> Бұл мәтінге сай стиль енгіземіз</p>
```

```
<td class=small> Бұл мәтінге сай стиль енгіземіз </td>
```

Тегтік класстар – нақты бір тегке қолданылады класс атауы тиісті тегті көрсетуден басталады, сонан соң нүкте қойылады да класс атауы жазылады.

Мысалы:

```
p.small { font-size: 9pt; }
```

Осылайша анықталған класс, көрсетілген тегте ғана іске қосылып, қалғанында жұмыс істеледі:

```
<p class=small>Бұл мәтін small</p> стиліне жұмыс істейді
```

```
<td class=small>Ал бұл мәтін өзгермейді</td>
```

Біз тек бір тегке ғана емес, бірнеше тегке де параметрлерді анықтай аламыз. Ол үшін стильді нұсқау кезінде бірнеше – уін үтір арқылы тізіп жазсақ болады:

```
P, TD, LI { font-size: 9pt; color:green; }
```

Тегтерге өзгеріс енгізу үшін стиль көрсеткішінде қажетті тег параметрін нұсқасақ болғаны. Ал қалғандары, өзіндік мағынасына ие болады.

### **Псевдокласстар**

CSS-те псевдокласс түсінігі бар. Қарапайым класспен салыстырғанда, псевдокласс әрекеті барлық мәтінде емес, тек оның бір бөлігіне ғана жер етеді.

Түсінікті болуы үшін толығырақ тоқталып өтейік. Сілтемелер оған курсорды нұсқағанда белгіленеді. Бұл эффект кеңінен таралған, ол осы сайтта да бар.

Мынау жоғарыдағы іс-әрекеттерді сипаттайтын стиль кестесінің фрагменті:

```
a { text-decoration: none; }
```

```
a:hover { text-decoration: underline; }
```

Жоғарғы жол – сілтемелерді белгілеуге тыйым салатын стандартты <a> тегінің қайта анықталуы, ал төмендегі жол – hover псевдоклассына арналған стиль анықтамасы, ол сілтеме стилін оның үстіне курсорды тақағанда өзгеріске ұшыратады.

Ал, мынау псевдоклассқа арналған келесі мысал – абзац басында әріпті анықтау

```
p:first-letter { font-size: 200%; font-weight: bold; }
```

Ескеріңіз, екі жағдайда да стиль әрекеті белгілі бір жағдайға ғана әсер етеді немесе мәтін фрагментіне ғана әсер етеді. Псевдокласстың түіндісі осылда.

### **Комментарийлер**

Кез-келген күрделі тілдердегідей стильдер кестесін құрғанда да комментарийлер қолануға болады.

Мысалы:

/\* Бұл мәтін комментарий болып табылады \*/

Қарапайым сайттар үшін бұл сізге қажет емес, ал стильдердің күрделі, көпдәріжелі кестелерін құрғанда комментарийлер керек болады. Ескере кететін маңызды ереже – ауыспалының атауы нақтырақ болған сайын (бұл жағдайда класс атауы), азырақ комментарийлер қажет етеді.

### **CSS кейбір параметрлері**

#### **Шрифттің кейбір параметрлері**

font-weight: [ bold | normal | number ] – шрифттің қалыңдығы

font-style: [ normal | italic | oblique ] – шрифтті бағыты

font-size: number – шрифттің көлемі

font-family: name – шрифттің гарнитурасы

color: number – шрифттің түсі

background-color: number – негізгі фон түсі

background: url – мәтінді фон

Абзацтың кейбір параметрлері

text-align: [ left | right | center | justify ] - тегістеу

text-indent: number – азат жолға шегіну

line-height: number - интерлиньяж

letter-spacing: number - трекинг

padding-left: number – мәтіннен солға шегіну

padding-right: number - мәтіннен оңға шегіну

padding-top: number - мәтіннен жоғарыдан шегіну

padding-bottom: number – мәтіннен төменнен шегіну

margin-left: number – шектеуден сол жақтан шегіну

margin-right: number - шектеуден оң жақтан шегіну

margin-top: number - шектеуден жоғарысынан шегіну

margin-bottom: number - шектеуден төменнен шегіну

#### **CSS өлшеу бірлігі**

Көлемін көрсетуді талап ететін кейбір мағыналар үшін оларды енгізудің келесі амалын қолдануға болады:

(%) пайыздық қатынас көлемі

Өздік сипаттау бойынша қатынас көлемі (larger, smaller, xx-small, x-small, small, medium, large, x-large, xx-large)

Типографиялық бірліктегі абсолюттық көлем – көлемдер пунктар (pt), пиктер (pc), "m"(em), әрпінің орташа ені, "x" (ex) әрпінің орташа ені арқылы берілуі мүмкін.

Ұзындықтың стандартты бірліктегі абсолюттық көлем – көлемдер сантиметрлер (cm), миллиметрлер (mm), дюймдер (in) арқылы беріле алады. абсолютті пиксельдер арқылы (px)

### **CSS түсті енгізу**

Қажетті жерде түсті енгізу қасиеті келесі үш әдістер арқылы орындалуы мүмкін:

- Түстің атауы арқылы yellow, red, green, grey,..
- Форматты он алты таңбалы арқылы #RRGGBB: #ff0000, #883490, #ffffff,..
- Форматындағы ондық таңбалар арқылы rgb(red, green, blue): rgb(255,0,0), rgb(100,23,78),..

### **CSS қолданудың мысалдары**

Барлық сілтемелерге түсті енгізейік. Немесе олардың барлығы қай жағдайда да асты сызылмай жазылуы тиіс, ал үстіне курсорды әкелгенде сілтеменің асты сызылуы керек.

```
A:link {COLOR: #0000CC; text-decoration: none}
```

```
A:visited {COLOR: black; text-decoration: none}
```

```
A:active {COLOR: red; text-decoration: none}
```

```
A:hover {COLOR: #6666FF; text-decoration: underline}
```

- сілтеме (link) – көк түс (#0000CC);
  - араны сілтеме (visited) – кара түс (black);
  - белсенді сілтеме (active) – қызыл түс (red);
  - үстіне курсорды әкелу кезіндегі сілтемек (hover) - көгілдір түс (#6666FF);
- CSS қолдану арқылы біз сілтемені қайта енгізу немесе жойып жібере аламыз. Ол үшін text-decoration қасиетін қолданамыз, ол келесі мағыналарды құрау мүмкін:

none - безендірусіз

underline – сызып кету

overline – сызықша үстінен

line-through – сызып тастау

blink - жыпықтау

### **Практикалық тапсырмалар**

#### **Тапсырма 1**

1. /css/ каталогын құрыңыз
2. main.css файлды ішінде құрыңыз
3. тега <LINK> көмегімен сайттың барлық беттеріне қосылуды құрыңыз.
4. Барлық беттерге жай келесідей параметрлер ұсыныңыз (тег <body> -ні қайта анықтап алып):
  - фонның түсі
  - шрифтін өлшемі
  - шрифтін түсі
  - шрифтін жан ұйялары (мысалы, Arial)
5. Серверде жұмыс жасау қабілеттігін тексеріңіз.
6. Түсініктемелерде (main.css файлда) бұл параметрлерді түсіндіріп кетіңіз.

## Тапсырма 2

1. Барлық абзацтарға жай келесідей параметрлер ұсыныңыз (тег `<p>` и псевдосынып тега `<p>` қайта анықтап алып):

- абзацты теңестіру
- азат жолының басталуы
- бірінші әріптің түсімен және өлшемі

2. Серверде жұмыс жасау қабілеттігін тексеріңіз.

3. Түсініктемелерде (main.css файлда) бұл параметрлерді түсіндіріп кетіңіз.

## Тапсырма 3

1. Задайте по умолчанию следующие свойства ссылок для всех страниц:

- түс және ссылқаның безендіруі
- түс және хабарласқан ссылқаны безендіру
- түс және активті ссылқаны безендіру

2. – түс және ссылқаны безендіру, тышқан курсорының оны үстінде табылу моментінде

3. Серверде жұмыс жасау қабілеттігін тексеріңіз.

4. Түсініктемелерде (main.css файлда) бұл параметрлерді түсіндіріп кетіңіз.

## Практикалық жұмыс №9

**Тақырып: (DIV) және CSS блоктардың көмегімен беттерді жазып кету.**

**Позиционирлеу.**

**Мақсаты:** (DIV) и CSS блоктардың көмегімен бетті сызып кету үйренуді қалыптастыру.

**Теоретикалық мәліметтер:**

Кестелердің қолданғада кемшіліктерін еске алайық:

- "Жай қабылдау". Барлық кесте толмай, экранда ақпарат шықпайды.
- Жай толтыру. Көптеген жәшіктер мен жолдарды құру керек, қажеттілігі шамалы.
- Элементті экранда ораналстырудың дәл пикселдерге дейін мүмкіншіліктің болмауы.
- Элементтерді бір біріне "мінгізуінің" мүмкіншілігінің жоқтығы.
- Элементтердің көрсетілуімен басшылық етуінің мүмкіншілігін жоқтығы (толған кезде, кестенің өлшемедері үлкееді).

Бұл мәселелерді CSS және `<DIV>` көмегімен шешуге болады.

Тег `<DIV>` - элементтерді топтастырылып блокқа қызмет ету үшін арналған.

Топталған элементтерге стильдерді қолдануға болады.

Қалай жасалатын мысалдарда көрейік:

**Мысалдар.** Блокты бордюрмен айырып көрсету

Шекараны бастамасына, тізімге, абзацқа немесе олардың топтарын жеңіл айналасына `div` элементімен салуға болады.

Бұл блокқа арналға стильдерді жазып кетейік

Келесідей түрде сызықшамен қолдануға болады:

```
<div style="border-color:#FF00FF; border-style:dotted; ">
```

Бұл `DIV` элементінің мазмұны арасы ашық рамкаға салынады.

</div>

<div style="border:15px double #008000; ">

Бұл DIV элементінің мазмұны ашық рамкаға ені 15 пиксельге салынады.

</div>

<div style="border-style: outset; border-color: #FF0000">

Бұл DIV элементінің мазмұны осындай рамкаға салынады.

</div>

<div style="border-right: 10px solid #00FF00; border-bottom: 10px solid #00FF00">

Бұл DIV элементінің мазмұны осындай рамкаға салынады.

</div>

<div style="border-left: 20px solid #FFFF00; border-right: 20px solid #FFFF00">

Бұл DIV элементінің мазмұны осындай рамкаға салынады.

</div>

**Мысал. Блокты фон түсімен айырып көрсету**

<div style="background-color: #00FFFF">

DIV элементінің құрамы мынадай түспен көрсетіледі.

</div>

**Мысал. Позиционирлеу және өлшемді тапсырма**

<div style="background-color: #AAAAFF">

Бұл DIV элементінің мазмұны осы жерде позиционирленеді.

</div>

<div style="float: left; width: 200; height: 200; background-color: #00FFFF">

Бұл DIV элементінің мазмұны осы жерде позиционирленеді.

</div>

<div style="float: right; width: 200; height: 200; background-color: #FFFF00">

Бұл DIV элементінің мазмұны осы жерде позиционирленеді.

</div>

<div style="width: 200; height: 200; position: relative; left: 20; background-color: #0FFD02">

Бұл DIV элементінің мазмұны осы жерде позиционирленеді.

</div>

<div style="background-color: #FCC403">

Бұл DIV элементінің мазмұны осы жерде позиционирленеді.

</div>

**Мысал. 2,5 мерді өлшемдегі позиционирлеу**

Блоктар қойылу (жауып тастау) тәртібі z-index атрибутымен анықталады.

Блок - 1

Блок - 2

Блок - 3

Блок - 4

3D-ге жақындау

Бұл мысалдың коды:



```
<div style="position: relative; width: 200; height: 200; z-index: 0; background-color: #FFFF00">
```

**Блок - 1**

```
</div>
```

```
<div style="position: relative; width: 200; height: 200; left: 100; z-index: 1; top: -100; background-color: #00FFFF">
```

**Блок - 2**

```
</div>
```

```
<div style="position: relative; left: 220; top: -400; width: 200; height: 200; z-index: 2; background-color: #00FF00">
```

**Блок - 3**

```
</div>
```

```
<div style="position: relative; width: 750; height: 20; z-index: 1; top: -580; background-color: #FE76AF">
```

**Блок - 4**

```
</div>
```

```
<div style="position: relative; top: -600; z-index: 3; left: 100">
```

3D жақындау

```
</div>
```

## Практикалық тапсырмалар

### Тасырма1

1. Жаңа бет ашыңыз
2. Бірінші беттің көшірмесін нақты етіп жасаңыз, бірақ кесте емес, блок пен стильді қолданыңыз. Во FrontPage-ді қолдануға болады( дегенмен онда барлығы анық емес) менюFormat менюдің асты ВоD. Және [CSS2 \(RUS\) каскадты стильдер кестесінің спецификациясымен қолданыңыз.](#)
3. Блоктар үшін стильдерді main.css. шығарыңыз
4. Серверде жұмыс жасау қабілеттілігін тексеріңіз.
5. Өз ойыңызды блокта көрсетіңіз.

### Тапсырма 2

1. Жаңа бет ашыңыз.
2. Суреттен немесе фотосуреттен ( кемінде 5-тей) 2,5 көлемін қолдана отырып коллаж жасаңыз.
3. Серверде жұмыс жасау қабілеттілігін тексеріңіз.
4. Өз ойыңызды блокта көрсетіңіз.

### Тапсырма 3

1. Белгіленген беттерге (лабораториялық жұмыс 3) overflow терезесін қолданыңыз:

- visible – 1-ші бөлім үшін

- auto – 2-ші бөлім үшін

- hidden – 3-ші бөлім үшін

- scroll – 4-ші бөлім үшін

- auto – 5-ші бөлім үшін  
Міндетті түрде өлшемдерді фон түсін немесе блоктардың бордюрін көрсетіңіз.  
2. Серверде жұмыс жасау қабілеттілігін тексеріңіз

### **4.3 Интернеттің серверлік - қосымшаларын құру технологиялары**

#### **4.3.1 Серверлерді бағдарламалау тілдері.**

Серверлерге орын сценарийлер көмегімен серверде сценарийлер көмегімен сервер де сақталатын мәліметтер базасы және т.б файлдарға және электрондық почта немесе факс – қызмет секілді сервердің орталықтандырылған ресурстарына қосылуына болады.

Сервер сценарийінің тағы бір ерекшелігі – басқару ортасын функциялау. Сіздің кодыңыз

көптеген браузердің бірнеше нұсқасында емес, бір сервердің жалғыз нұсқасында болады.

Бірақ сервер бетінде сценарийді қолданудың 3 кедергісі бар:

1. Серверде скрипттерді іске қосу web – маманнан және жүйелік администратордан арнайы құқыққа ие болуды талап етеді.
2. Серверде орындалатын сценарилармен жұмыс істеу үшін қолданушы сілтемеледе немесе беттегі батырмада тышқанды шерту керек. сонан соң сервер сценарийді орындап, жауабын қайтаруды күту керек. Динамикалық HTML-ді қолдану арқылы байланысу жедел орындалады.
3. Сценарийлерді тесттеу үшін серверге жеке www-сервер талап етіледі. Осы уақытта сервер бетіне бағдарламалау тапсырмаларды шешу мүмкіндігін арттыру үшін маңызды. Ол мына процесстерді жүргізуге мүмкіндік береді.:
  - Қолданушының форма көмегімен серверге негізген мәліметтерді қабылдау және өзгерту.
  - Платаформаға да, клиент браузерлеріне де тәуелді емес web – құжаттарды динамикалық түрде құру.
  - Серверде орналасқан мәліметтерге динамикалық түрде қосылуды қамтамасыз ету.
  - Типтік есептерді шешуге арналған компоненттерін қолдану жарнамалық банккерлерді периодты түрде ауысуы
  - Клиент браузерлі түрде ақпарат алу
  - Клиент бетінде кілттерді Клиент браузерлі түрде ақпарат алу
  - Клиент бетінде кілттерді құру және оқу

### **ASP, JSP, PHP**

Интернет - қосымшалардың жасаудың технологияларының дамытуындағы кезекті адым web – дизайн мен қосымшалардың функционалдығын іске асыруға қатысты міндеттерді бөлуге мүмкіндік беретін құралдарының пайда болуы. Мұндай технологиялардың біріншісі ISAPI сүзгінің негізде салынған (ASP ) Active Server Pagesі болды. ASPның негізгі идеясы - скриптік тілдердегі кодтар енгізілген web-беттер жасауында болады.

Алайда, жоғарыда қарастырылған браузер функционалдығын кеңейтуге арналған скрипт тілдерінің қолдану құралдарынан айырмашылығы көрсетілген кодтың бөліктерін браузер емес сервер талдау жасайды (дәлірек айтсақ ол үшін қолайлы ISAPI кітапхана арқылы), және кодтың бұл бөліктерінің орындалу нәтижесін қолданбалы браузерге берілген беттің нұсқасында коддың фрагменті атқарады. ASP дан кешікпей Web сервердің орындайтын кодтың Web бетшенің ішіне орналастыруды идеянысын іске асыратын тағы басқа технологиялары пайда болды. Қазіргі таңда олардың ішіндегі өте белгілісі JSP (Java Server Pages). Оның негізгі идеясы оған бірінші рет жолдағанда Java кодын (сервлет) бір рет компиляциялануы, осы сервелат әдістерін орындап оларды браузерге жеткізілетін жинаққа орналастыру. Тағы бір әйгілі технология PHP (Personal Home Pages ) скрипт тілі. Ол HTML – бетке скриптік тілде енгізілген кодты интерпретациялайтын CGI - қосымшалар пайдаланады.

**PHP** (ағылшынша: *PHP: Hypertext Preprocessor* — «PHP: гипермәтінді алдын-ала үдірістегіш») — Веб-серверде HTML беттерін құру және дерекқормен жұмыс істеуге арналған скрипті программалау тілі. Қазіргі күндерде хостинг қызметін ұсынатындардың көпшілігімен қолданады. LAMP веб-торап жасауға арналған стандартты жинақталымына енеді. Тілдің оңайлығы, орындалу жылдамдығы, функциялық байлығы және PHP негізіндегі әуелгі кодтардың кеңге таралуының арқасында PHP тілі Торда программалау саласында ең белгілі тілдердің бірі боп есептеледі (JSP мен ASP-де қолданылатын тілдермен қатар). Бұл тілдің айрықшылығы ядро мен соған қосылатын модульдарында. Соңғылары дерекқор, сокет, динамикалық графика, криптографиялық кітапханалар, PDF форматты құжаттармен және т.б. жұмыс істеуге арналған. Бұндай модульды қалаған адам дамытып қоса алады. Модульдардың саны бір неше жүз болғанымен, стандарттық жабдықтауға тек жақсы нәтиже көрсеткен бір неше оны ғана кіреді. PHP интерпретаторы веб-серверге не арнайы сол сервер үшін жасалған модуль арқылы (мысалы, Apache әлде IIS), не CGI-қосымшасы ретінде қосылады. Бұған қоса, UNIX, Linux, Windows және Mac OS X амалдық жүйелерінде әкімшілік тапсырыстарын атқаруға пайдаланылуы мүмкін.

Тілдің синтаксисі Си тәріздес. Ассоциативтік массивтар мен foreach циклі секілді кейбір элементтері Perl тілінен алынған.

Қазіргі заманда PHP жүздеген мың дамытушылармен пайдалануда. Интернет желісі сайттарының 5-тен бірі осы тілде жазылған.

PHP дамытушылар тобы тілдің ядросы, қосымшаларымен жұмыс істейтін, соған қатар PEAR не тілдің құжаттамасы сияқты байланысты жобалармен айналысатын көптеген адамдардан тұрады.

### **Зертханалық жұмыс №10**

**Тақырыбы: PHP көмегімен динамикалық беттерді құру.**

**Жұмыстың мақсаты: PHP-ді қолдану мүмкіндіктерін жүзеге асыру.**

### **Теориялық мағлұматтар:**

PHP – бұл HTML енүші тіл сервері жағында жұмыс істейтін, синтаксисі бар, Си тіліне жақын. PHP тілі HTML-ге PHP тілі ережелерінің динамикалық құрамын құру үшін мүмкіндік береді. Бұл кодты шығаратын бұл инструкциялар PHP-дің препроцессорымен-интерпретаторымен өңдейді және оларды сол құрамымен ауытырады. рҒр бағдарламасымен PHP файлының стандартты үлкеюі.

Принципі SSI-кі сияқты.

### **PHP-дің синтаксисі:**

PHP-коды, html-кодына келесідей түрде қосылады:

```
<?PHP тексттің_коды ?>
```

немесе

```
<?PHP
```

```
тексттің_коды;
```

```
?>
```

Мысалы, файлдаң құрамын салу былай көрінеді: <?PHP include "ssi/top.php"; ?>

### **Ойлар:**

<?php PHP 'C', 'C++' ойларын қолдайды және Unix қабығын. Мысалы:

```
echo "This is a test"; // Бұл с++ стиліндегі бір жолдық ой.
```

```
/*Бұл көп жолдық ойлар, бұл оның тағы бір жолы*/
```

```
echo "This is yet another test";
```

```
echo "One Final Test"; # Бұл shell-стиліндегі ойлары
```

```
?>
```

### **echo**

```
<?php echo "Бұл ақпарат HTML-ға шығады";?>
```

### **Ауыспалыларға мағыналар беру:**

PHP ауыспалы бағдарламаларындағы ауыспалылар \$ белгісімен бөлінеді.

```
$city = "Kazan";
```

city - ауыспалы

Kazan - мағына

### **Кейбір операциялар:**

инкременті/декременті;

++\$a Pre-increment \$a-ны 1-ге көбейтеді, сосын \$a-ны қайтарады.

\$a++ Post-increment \$a-ны қайтарады, сосын \$a –ны 1-ге көбейтеді..

--\$a Pre-decrement \$a –ны 1-ге, сосын \$a-ны қайтарады.

\$a-- Post-decrement \$a-ны қайтарады, сосын \$a 1-ге азайтады.

арифметикалықтар:

FALSE немесе TRUE –ді білдіріп, салыстыру ұғымы 0 немесе 1 мен

табылады. PHP > (көп), >= (көп немесе тең), == (тең), != (тең емес), < (аз) және

<= (аз немес тең) қолдайды. if сияқты ішкі операторлармен қолданылады..

салыстыру:

\$a == \$b тең TRUE, егер \$a тең \$b.

\$a != \$b тең емес TRUE, егер \$a тең емес \$b.

\$a <> \$b тең емес TRUE, егер \$a тең емес \$b.

\$a < \$b аз TRUE, егер \$a жолдар аз болса \$b.

`$a > $b` көп TRUE, егер `$a` жолдар көп болса `$b`.  
`$a <= $b` аз немесе тең TRUE, егер `$a` аз немесе тең `$b`.  
`$a >= $b` көп немесе тең TRUE, егер `$a` көп немесе тең `$b`.  
жолдық:

Жолдық екі операция бар. Бірінші – оң және сол аргументтердің қосылуын қайтаратын ('.') операциясы. Екінші – оң аргументті сол аргументке қосатын ('.=') операциясы.

```
$a = "Hello ";  
$b = $a . "World!"; // енді $b "Hello World!"-тан тұрады.
```

### Басқада операторлар:

#### **if/else**

```
if ($a > $b) {  
    print " b-дан көп";  
} else {  
    print " b-дан көп емес";  
}
```

#### **while**

```
$i = 1;  
while ($i <= 10) {  
    print $i++; /* $i 10-нан көп болмағанға дейін $i++ мағынасы басылады */  
}
```

#### **For**

```
for ($i = 1; $i <= 10; $i++) {  
    print $i; /* $i=1 $i <= 10 қадамымен $i++ цикл орындалады*/  
}
```

#### PHP көмегімен **cookie** –ді орнату.

`setcookie` – кукиді жолдайды.

Бұл функцияларды шақыру `<html>` или `<head>!!!` Тэгтері алдында болуы керек.

Синтаксис:

```
int setcookie (string name [, string value [, int expire [, string path [, string domain [, int secure]]]])
```

Мысал:

```
setcookie ("student", "pupkin");
```

#### PHP көмегімен **cookie** серверін алу және қолдану.

Куки орнатылған соң, оларды бетті массивті `$_COOKIE` арқылы енгізгенде қолдануға болады.

`Cookie` автоматты түрде өзгереді және ауыспалыға дәл сол куки атымен беріледі. Скрипте біздің тесттік кукиіміздің құрамын қарау үшін жай ғана келесі мысалдардың біреуін қолданыңыз:

```
echo $student;  
echo $_COOKIE["student"];
```

#### **Әрі бір беттің есептегішін құруға керекті компоненттер:**

Әр бір бетке есептеу жүргізілуі қажет, сондықтан әр беттің мағынасын бөлек файлда сақтау керек. Файлдар көп болатындықтан,

/www/students/group/login/counter локальді дискінде оларға бөлек каталог құрамыз.

Оны бағдарламада қолдану үшін осы каталогқа ауыспалы жол береміз.

#### **Мысалы:**

```
$counter_path = " сіздің каталогыңызға жол";
```

Файлдың атын автоматты түрде өзгерту үшін осы беттің URI-ын пайдаланамыз.

Бірақ URI – да барлық «/», «\_» ауытырамыз, себебі файлдың атында / болуы мүмкін емес

```
str_replace("нені ауыстырамыз", " неге ауыстырамыз", алмастру жүретін жол) – алмастыру функциясы.
```

getenv("ауыспалы айналым") – функция ауыспалы айналым мағынасын алады.

REQUEST\_URI – URI –дың ауыспалы айналымы.

Түзелген URI-ды ауыспалы \$count\_page-ге тапсырамыз, яғни құрылып жатқан файлдың аты.

#### **Мысалы:**

```
$count_page = str_replace(бұл беттің URI-ында "нені ауыстырамыз", "неге ауыстырамыз"); getenv("REQUEST_URI") – бұл беттің URI-ы.
```

Егер файл (\$count\_page) бар болса, яғни бет бұрын ашылса, онда файлдағы санақ мағынасын санап оны бірге көбейту керек.

```
file("каталогқа жол/ файл аты") – Функция тексттік файлдың құрамын массивті түрде санайды, массивтің әрбір элементі 'n' белгісімен файл жолына келіседі.
```

Файлды ашайық, және \$contents-ке ауыспалы мағына береміз.

### **Практикалық тапсырмалар**

#### **Тапсырма 1**

1. SSI-дың орнына PHP-ді пайдаланып 1-ші бетті қайта жасаңыз (жаңасын жасауға болады).
2. Cookie META мағлұматтар арқылы берілмей, setcookie көмегімен берілетін етіңіз.
3. Серверде жұмыс істеу біліктілігін тексеріңіз.
4. Өз ойыңыз түрінде PHP командасы не үшін арналғанын көрсетіңіз.

#### **Тапсырма 2**

1. Бетке кіру санағын құрыңыз (count.inc файлы ). Санақ мағынасы бар файлдар жазу үшін, counter кталогын қолданыңыз (ол сізде құрылды!!!)
2. Оны include көмегімен botom.php-ға қойыңыз.
3. Серверде жұмыс істеу біліктілігін тексеріңіз.
4. Өз ойыңыз түрінде PHP командасы не үшін арналғанын көрсетіңіз.

### **Практикалық жұмыс №11**

**Тақырып:** СУБД MySQL PHP көмегімен мағлұматтар базаларымен жұмыс істеу.

**Максат:** Мағлұматтар базасына кіру үшін PHP-ді қолдануды үйрену, және MySQL жұмыс жасауды үйрену.

**Теоретикалық мәліметтер:**

## СУБД MySQL және phpMyAdmin бағдарламасы

MySQL - бұл реляциондық мәліметтер базасы. (арнайы сервер - [www.mysql.com](http://www.mysql.com) ).

Кейбір ерекшеліктері:

- Реляциондық
- Жоғары мәртебелі, яғни әрбір кіруші қосылу бөлек топпен басқарылады, бұл теориялық жағынан шексіз сұраныстың орындалынатынын білдіреді.
- “Клиент сервер” технологияларымен жұмыс істейді.
- Ресурстарға талап жоқ.
- Тегін (POSIX- системасында комерциялық емес қолдану үшін, қалғандарына ақылы)
- SQL тілін қолдайды (бәрін болмасада)
- Таратылған
- MySQL- де жұмыс жасайтын көптеген қосымшалар бар.

MySQL. **phpMyAdmin** – MySQL серверінде администрациялауға түгелдей PHP-де жазылған программа.

Кейбір мүмкіндіктер:

- Мағлұматтар базасын құру, өзгерту, кетіру.
- Мағлұматтар базасының кестелерін құру, өзгерту, кетіру.
- Қолданушыларды құру, өзгерту. кетіру.
- Мағлұматтар базасындағы жазбаларды құру, кетіру, өзгерту.

### PHP көмегімен мағлұматтар базасымен жұмыс істеу.

PHP тілінің бағалы сапасының бірі бұл интерфейстің үлкен мөлшердегі мағлұматтар базасын басқару системасы. Бұл Abadas D, dBase, Empress, FilePro, Informix, Interbase, mSQL, MySQL, Oracle, PostGreSQL, Solid, Sybase, Velocis, Unix dbm, және басқалары. Мысал ретінде MySQL интерфейсін қолданамыз. Басқа СУБД мағлұматтарына жету интерфейстері аналогиялық болып көрінеді.

### PHP-да бастапқы түрлерінің мысалдары:

Мағлұматтар жолындағы келесі жолдардың мағлұматтар іздеу, өзгерту, кетіру, құю скриптілерінің мысалдарын қарастырайық:

- есім - user\_name
- телефон – phone
- Сурет - img (foto/ катлогында тұрған файлдың аты)

### Енгізілген мағлұматтарды тазалау скрипті.

```
<?php
/*Ауыспалылардың мағынасын анықтаймыз*/
$hostname="localhost";
$username="Сіздің логиніңіз";
$password="Сіздің пароліңіз";
/* Мәліметтер базасының аты */
$dbName=" Сіздің мәліметтер базасының аты ";
/* Кесте аты */
$userTable="Кесте_аты";
/* MySql байланысты орнату */
```

```

MYSQL_CONNECT($hostname,$username,$password) OR DIE("Қосылу мүмкін емес");
/*БД-ны таңдау*/
MYSQL_SELECT_DB($dbName) or die("МБ таңдалмайды");
/* Ақпаратты МБ енгізіңіз */
$query="INSERT INTO $usertable VALUES ('$user_name','$phone','$img)";
$result=MYSQL_QUERY($query);
/* Байланысты жабу */
MYSQL_CLOSE();
print "Жазылым МБ енгізілді! <br>";
?>

```

### Іздеу сұранысын өңдеу скрипті.

```

<?php
$hostname="localhost";
$username="Сіздің_логиніңіз";
$password="Сіздің_құпия_өзіңіз";
$dbName="Сіздің_мәліметтер_базаңыздың_аты";
$usertable="кесте_аты";
/* MySQL байланысты орнату*/
mysql_connect($hostname,$username,$password) or die("No connect");
mysql_select_db("$dbName") or die("No select DB");
/* Аты $usr_name деп басталатын адамдар аттарын таңдау */
$query="select * from $usertable where user_name like '$user_name%'";
$result=mysql_query($query);
/* Мұндай қызметкерлер қанша? */
$number=mysql_numrows($result);
/* Таңдалған қызметкерлерді баспадан шығару */
$i=0;
if ($number == 0)
{
print "МБ $user_name бойынша мәліметтер жоқ <br>";
}
elseif ($number > 0)
{
while ($i < $number)
{
/* i жолындағы user_name, phone, img жлдарына ауыспалы атау беру */
$user_name=mysql_result($result,$i,"user_name");
$phone=mysql_result($result,$i,"phone");
$img=mysql_result($result,$i,"img");
/* Табылған мәндерді шығару */
print "<br><img src='foto/$img' height=130><br>";
print "<u> Имя:</u> <br>";
print "$user_name <br><br>";
}
}

```



```
print "<u> Телефон: </u><br>";
print "$phone <br><br>";
$i++;
print "<hr>";
}
}
?>
```

### **Практикалық тапсырмалар**

#### Тапсырма 1

1. <http://ipm.kstu.ru/mysql/> мекен жайындағы сервердің phpMyAdmin программасына кіріңіз. Мағлұматтар базасын құрыңыз (БД атауы сіздің login'іңізден басталуы керек, мысалы - vasy-catalog).
2. Келесі жолдағы мына БД кестесін құрыңыз. (тыныс белгілерін өз қарауыңызбен қойыңыз).
  - Аты (индекстеледі)
  - Фамилиясы (уникальное)
  - Әкесінің аты(индекстеледі)
  - e-mail (индекстеледі)
  - телефон (индекстеледі)
  - қала (индекстеледі)
  - ел (индекстеледі)
  - суреті
3. Жүктеп көріңіз, кетіріңіз және мәліметтерді редакциялаңыз

#### Тапсырма 2

1. Мәліметтер базасына жазу қосу үшін пішім құрау керек. Скрипт құрап қайтадан сұрауды өңдеңіз.
2. Пішімін құрап осы базаға ізденіс жіберің. Скрипт құрап қайтадан сұрауды өңдеңіз.
3. Пішімін құрап осы базасына сұрау жібереді, осыны іздеу үшін жазылғанда өзгерту керек (жолдың бәрі болу керек) Скрипті құрап қайтадан өңдеңіз.
4. Пішімін құрап осы базасына сұрау жібереді, осыны іздеу үшін жазылғанды кетіру керек (тек тегі болу керек). Скрипт құрап қайтадан сұрауды өңдеңіз.
5. Серверде жұмыс істейтінін тексеріңіз.
6. Ссылканы бірінші бетке жасаныз.

### **4.3.2 «Клиент-сервер» технологиясы**

World Wide Web (www) қосымшалардың көпшілігі Web-беттер басқаратын сыртқы бағдарламалар негізінде құрылған.

Бұл бағдарламаларды қолдану мәліметтер базасында сақталған динамикалық түрде жаңартып тұратын ақпараттар арқылы Web- беттері құруға мүмкіндік береді. Web – сервер мен қосылып жатқан бағдарлама арасындағы байланыс үшін CGI кеңінен қолданылады, ол Windows ориентирілген бағдарламаларда да, unix ортасында қызмет ететін қосымшаларды да қолданылады.

CGI- Web сервер HTTP типтегі ақық тұрушы құрал.

CGI - көмегімен шлюздер деп аталатын CGI-бағдарламаны құруға брлады.

Шлюздер деп қолданушының экран бетіне мәліметтер базасын басқару жүйесі электрондық кесте, жұмыс графикасы т.б., сияқты ақпараттар жүйесімен әрекеттеседі CGI-бағдарламаны айтады.

Шлюз- бағдарлама www сервер негізінде белгілі бір уақыт аралығында жұмысқа қосылады.

WWW сервер қолданушының сұраныс шлюзге жіберілуін қадағалайды, олшлюз қосымша жүйе құралдарын қолдану арқылы жасалған. Сұраныс нәтижесін қолданушының экранына шығарады.

Шлюз – бағдарлама c/c++, Fortran, Perl, Tch, Unix Shell, Visual Basic, Apple Script тілдерінде кодталуы мүмкін. Орындалатын модуль болғандықтан ол www- серверінде сді- ві деген атаумен жазылады. Интернетте, wwwда клиент – сервер технологиясы бойынша жұмыс жасайды, яғни барлық бағдарламалар клиенттік және серверлік болып 2ге бөлінеді. Сонымен қатар олардың арасында қызымет ету міндеттері де бөлінген. Тағы бір ескере кететін жағдай, сервердің жұмысын және ішкі алгоритмдердің орындалу принциптерін білмейді және ішкі алгоритмдердің орындалу принциптерін білмейді және білуге міндетті емес, ал сервер клиенттің жұмысына араласпайды. Бұл қарым-қатынас үшін арнайы хаттама болады клиентпен сервер арасындағы барлық іс әрекет осы протокол шегінен шығып кетуі керек. Броузерге серверде қандай бағдарлама орнатылғаны, қандай операциялық жүйе жұмыс жасап жатқандағы сұраныс жасалған құжаттардың қайда сақталғаны тиісті емес. Серверді де сіздің броузеріңіз алынған ақпаратты қалай қолданылатындығын да, оны қалай көрсетеді ма, онымен жұмысы болмайды. Әдетте серверге сұраныс жасағанда UR2 қолданылады. Сервермен байланысқа түскенде домендік атау символы аударылады. www.Siemens.com сонан соң IP иесімен TCP / IP байланысты құру орындалады. HTTP – сұраныс жасалған кезде, клиент жауап қайтарылуын күтіп отырады. Клиент сұраныс жасағанға дейін HTTP – сервер «ұйқыда» болады. Клиент байланысқа түскенде, сервер оянады. Сұраныс мәліметтерін қабыдап оны өңдеуге кіріседі. Келіп түскен сұранысты, сервер қалай өңдейтіндігін өзіне ғана мәлім. Сервер клиентке жауап қайтарған соң, қайтадан ұйқыға кетеді. Егер қате жасалған жағдайда HTTP- келген уақытында үзіліп кетеді. Егер құжат табылмаса немесе сіздің ол құжатты алуға құқыңыз болмаса сізге «қате» деген код жіберіледі. Ал егер бәрі жақсы болса құжатта сақталған ақпарат мәліметтер типімен қоса, ISAPI – жауап түрінде клиентке жіберіледі.

### 4.3.3 ISAPI бағдарламасы

ISAPI бағдарламасы Web – сервер мен кемшіліктерін шолып өтейік. CGI – дың басты артықшылығы, әмбебаптылығында. CGI- сценарий әртүрлі тілдерде жазыла алады және құрлысы әртүрлі компьютерде орындала алады. Егер сіз сценарийді жазу кезінде, барлық ережені ескерген боласыз, онда сіз құрған бағдарлама web-серверлерінебайланысқа түсе алады. CGI

қарпайымдылығы бұл бағдарламаның кеңінен таралуына да негіз бола алады. Егер сіз сценарийді жазу кезінде барлық ережені ескерген болсаңыз, онда сіз құрған бағдарлама Web – серверлеріне байланысқа түсе алады.

CGI-дың артықшылықтарымен қатар, өзіне тән кемшіліктері де бар. Солардың ішінде ең бастысы: ресурстарды тиімсіз пайдалану. Клиенттің жасаған әрбір сұранысы компьютер серверіне жаңа процесс тізбегін туғызады. Тәуелсіз процессті орындау кезінде процесс құрамында жаңа бағдарлама жұмысына қарағанда көптеген ресурстарды талап өтетіндігі белгілі.

Жұмыс көлемін жоғарлатып, ол ресурстар шығынын төмендету корпоративті бағдарламаларды құруға әкеледі. Олардың ішінде ең танымалы:

#### **Бақылау сұрақтары:**

1. Интернет қосымшалардың сервер жағында орындалатын құруының ерекшеліктері қандай?
2. CGI технологияларының ерекшеліктері неде?
3. ISAPI технологиясының ерекшеліктері қандай?
4. Сервердің жағында орындалатын технологиялардың жақсы және кемістік жақтары қандай?

### **Практикалық жұмыс №12**

**Тақырып: Форманың дұрыс толтырылуын JavaScript көмегімен тексеру. Сұхбат терезелері.**

**Мақсаты:** Форманың дұрыс толтырылуына және сұхбат терезесін құруда JavaScript пайдалану біліктілігін қалыптастыру.

#### **Теориялық ақпарат**

Браузердің объектілі моделі 12 объектіден тұрады :

1. Document - HTML құжатының компоненттеріне кіру үшін рұқсат беретін жағдай.
2. Event – соңғы оқиғалар болғаннан соң, оқиғаларының қасиеттеріне кіру үшін рұқсат беретін жағдай.
3. History – пайдаланушы қосылған соңғы адресстер бойынша ақпараттарды береді.
4. Location – ағымдағы құжат адресі бойынша ақпаратты береді.
5. MimeType - MIME типі туралы ақпаратты береді.
6. Navigator – браузер қасиеттеріне хабарласуға жағдай береді.
7. Selection – құжаттың ағымдағы белгіленуін көрсетеді.
8. Style - представляет конкретный элемент стиля в таблице стилей.
9. TextRange - HTML құжатты толықтыратын мәтіннің бөлімдерін көрсетеді.
10. Screen - предоставляет информацию о мониторе и системе вывода, информации клиента.
11. Window – браузер терезесімен байланысты қасиеттер, әдістер және оқиғаларды жеткізеді.
12. StyleSheet –JavaScript тілінің көмегімен беттің жартысына немесе браузердің жартысына хабарласуға мүмкіншілік беретін стильдер кестесінің ішіндегі барлық стильдердің элементтерін жеткізеді.

Объектілерден тыс, объектілердің моделіне келесідей түсініктер қосылады:

**Әдістер** – объектілермен жұмыс құралдары. Мысалы: терезені жабу. Маңызы бойынша бұл функция, объектімен ассоциацияланады.

object.methodname

**Оқиға** – объект бір нәрсе болғандығын бізге хабарлайды. Мысалға: элемент активті болады.

**Қасиетте** – объект қасиеттері. Мысалы: терезенің атауы мен өлшемі.

### **Сұхбат терезесі. Window Объектісі**

#### **Терезені ашу**

Синтаксис:

```
window.open ("URL немесе URI", "терезе атауы", "терезе қасиеттері")
```

Келесі оператор <http://www.kstu.ru> парағының мазмұнын көрсететін терезені құрады:дай

```
window.open("http://www.kstu.ru")
```

Терезені құрған кезде атау ұсынуға мүмкіндігіңіз бар, берілген жағдайда ол - kstuWindow, терезеге мақсат ретінде қараған уақытта/target пішінді жөнелткен уақытта немесе сілтеме бойынша өткен кезде

```
window.open("http://www.kstu.ru", "kstuWindow" )
```

терезе атауы терезені құрған уақытта қажет емес. Алайда терезеге басқа терезеден әрекет ететін болсаңыз, терезе атуы міндетті түрде болуы қажет.

Терезені ашқанда сіз атрибутты специфицировать ете аласыз,осындай ұзындығымен және ені, перне утилиті,адрестік жолы,немесе сызық айналдырымы.Келесі оператор терезені пернесіз,бірақ сызық айналдыруымен.

```
window.open ("http://www.kstu.ru", "wwwWindow", "toolbar=no,scrollbars=yes")
```

Терезенің кейбір қасиеттері:

directories - егер болса пернелі стандартты директориялық браузері пайда болады.

height - терзенің биіктігін пикселде спецификациялайды.

innerHeight - терезенің құрамдық жағын пикселде ұзындығында спецификациялайды. Бұл қасиет қарсы ұқсастыққа қамтамасыз етілуге қалдырылған height ті ауыстырды.

innerWidth - innerWidth – терезенің құрамдық жағын пикселде енін спецификациялайды. Бұл қасиет қарсы ұқсастыққа қамтамасыз етілуге қалдырылған width ті ауыстырды.

outerHeight – терезенің сыртқы шекарасының өлшемін пикселде тігінен спецификациялайды.

resizable - егер yes болса, қолданушыға терезенің өлшемін өзгертуге мүмкіндік береді.

screenX - қашықтықта спецификациялайды, сонда жаңа терезе экранның сол жақ шетінен сыяды.

screenY - қашықтықта спецификациялайды, сонда жаңа терезе экранның жоғарғы сыяды.

scrollbars - scrollbars - егер yes болса, сызықтық айналым көлденең және тігінен пайда болады, егер құжат өлшемі терезеден үлкен болып кетсе.

status – егер yes болса, терезенің астында беделді жол пайда болады.

titlebar - егер yes болса, терезеде такырыпты жол пайда болады.

toolbar - Егер yes болса, Back және Forward секілді пернелі браузер стандартты терезесі пайда болады.

width - терезенің енін пикселде спецификациялау.

Функция тышқанның пернесін құжаттағы элементке басқанда жұмыс жасауы үшін, Document объектісіндегі onClick құрылымын қолданамыз.

### Мысал:

Сілтемені жаңа терезеден ашайық

```
<a href=""  
onClick="window.open('http://www.kstu.ru','kstuWindow','left=300,top=300,width=  
200,height=400,toolbar=no,menubar=no,location=no,directories=no')">
```

Жаңа терезені ашу

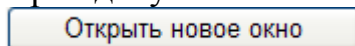
```
</a>
```

Орындалуы: [Жаңа терезе ашу](#)

Дәл соны батырма көмегіменде жүзеге асыруға болады:

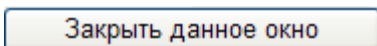
```
<input type="button" value="Жаңа терезені ашу"  
onClick="window.open('http://www.kstu.ru','kstuWindow','left=300,top=300,width=  
200,height=400,toolbar=no,menubar=no,location=no,directories=no')">
```

Орындалуы:



Терезенің жабылуы

Мысал:



JavaScript көмегімен форманың дұрыс толтырылғандығын тексеру.

JavaScript көмегімен форманың дұрыс толтырылғандығын бақылауға болады.

Төменде орналасқан формадағы "жіберу" батырмасын, оны толтырмастан шертіп көріңіз.

Атау

E-mail

Коментарийлер

</td></tr>
<tr><td height="26"><input type
</tbody></table>

## Практикалық жұмыс №13

**Тақырып:** JavaScript көмегімен Web-парақтарды қарау сценарийлерін басқару.

**Максат:** JavaScript-ті Web-парақтарды қарау сценарийлерін басқару үшін қолдануға үйрену

**Теоретикалық мәліметтер:**

**СУБД MySQL және phpMyAdmin бағдарламасы**

**MySQL** - бұл реляциондық мәліметтер базасы. (арнайы сервер - [www.mysql.com](http://www.mysql.com) ).

Кейбір ерекшеліктері:

- Реляциондық
- Жоғары мәртебелі, яғни әрбір кіруші қосылу бөлек топпен басқарылады, бұл теориялық жағынан шексіз сұраныстың орындалынатынын білдіреді.
- “Клиент сервер” технологияларымен жұмыс істейді.
- Ресурстарға талап жоқ.
- Тегін (POSIX- системасында комерциялық емес қолдану үшін, қалғандарына ақылы)
- SQL тілін қолдайды (бәрін болмасада)
- Таратылған
- MySQL- де жұмыс жасайтын көптеген қосымшалар бар.

MySQL. **phpMyAdmin** – MySQL серверінде администрациялауға түгелдей PHP-де жазылған программа.

Кейбір мүмкіндіктер:

- Мағлұматтар базасын құру, өзгерту, кетіру.
- Мағлұматтар базасының кестелерін құру, өзгерту, кетіру.
- Қолданушыларды құру, өзгерту, кетіру.
- Мағлұматтар базасындағы жазбаларды құру, кетіру, өзгерту.

**PHP комегімен мағлұматтар базасымен жұмыс істеу.**

PHP тілінің бағалы сапасының бірі бұл интерфейстің үлкен мөлшердегі мағлұматтар базасын басқару системасы. Бұл Abadas D, dBase, Empress, FilePro, Informix, Interbase, mSQL, MySQL, Oracle, PostGreSQL, Solid, Sybase, Velocis, Unix dbm, және басқалары. Мысал ретінде MySQL интерфейсін қолданамыз. Басқа СУБД мағлұматтарына жету интерфейстері аналогиялық болып көрінеді.

**PHP-да бастапқы түрлерінің мысалдары:**

Мағлұматтар жолындағы келесі жолдардың мағлұматтар іздеу, өзгерту, кетіру, құю скриптілерінің мысалдарын қарастырайық:

- есім - user\_name
- телефон – phone
- Сурет - img (foto/ катлогында тұрған файлдың аты)

**Енгізілген мағлұматтарды тазалау скрипті.**

```
<?php
/* Ауыспалылардың мағынасын анықтаймыз*/
$hostname="localhost";
$username="Сіздің логиніңіз";
$password="Сіздің пароліңіз";
/* Мәліметтер базасының аты */
$dbName=" Сіздің мәліметтер базасының аты ";
/* Кесте аты */
$userTable="Кесте_аты";
/* MySQL байланысты орнату */
MYSQL_CONNECT($hostname,$username,$password) OR DIE("Қосылу мүмкін емес");
/*БД-ны таңдау*/
```

```

MYSQL_SELECT_DB($dbName) or die("МБ таңдалмайды");
/* Ақпаратты МБ енгізіңіз */
$query="INSERT INTO $usertable VALUES ('$user_name','$phone','$img)";
$result=MYSQL_QUERY($query);
/* Байланысты жабу */
MYSQL_CLOSE();
print "Жазылым МБ енгізілді! <br>";
?>

```

### Іздеу сұранысын өңдеу скрипті.

```

<?php
$hostname="localhost";
$username="Сіздің_логиніңіз";
$password="Сіздің_құпия_өзіңіз";
$dbName="Сіздің_мәліметтер_базаңыздың_аты";
$usertable="кесте_аты";
/* MySQL байланысты орнату*/
mysql_connect($hostname,$username,$password) or die("No connect");
mysql_select_db("$dbName") or die("No select DB");
/* Аты $usr_name деп басталатын адамдар аттарын таңдау */
$query="select * from $usertable where user_name like '$user_name%'";
$result=mysql_query($query);
/* Мұндай қызметкерлер қанша? */
$number=mysql_numrows($result);
/* Таңдалған қызметкерлерді баспадан шығару */
$i=0;
if ($number == 0)
{
print "МБ $user_name бойынша мәліметтер жоқ <br>";
}
elseif ($number > 0)
{
while ($i < $number)
{
/* i жолындағы user_name, phone, img жлдарына ауыспалы атау беру */
$user_name=mysql_result($result,$i,"user_name");
$phone=mysql_result($result,$i,"phone");
$img=mysql_result($result,$i,"img");
/* Табылған мәндерді шығару */
print "<br><img src='foto/$img' height=130><br>";
print "<u> Имя:</u> <br>";
print "$user_name <br><br>";
print "<u> Телефон: </u><br>";
print "$phone <br><br>";
$i++;
}
}
}

```

```
print "<hr>";  
}  
}  
?>
```

## **Практикалық тапсырмалар**

### **Тапсырма 1**

1. <http://ipm.kstu.ru/mysql/> мекен жайындағы сервердің phpMyAdmin программасына кіріңіз. Мағлұматтар базасын құрыңыз (БД атауы сіздің login'іңізден басталуы керек, мысалы - vasy-catalog).
2. Келесі жолдағы мына БД кестесін құрыңыз. (тыныс белгілерін өз қарауыңызбен қойыңыз).
  - Аты (индекстеледі)
  - Фамилиясы (уникальное)
  - Әкесінің аты(индекстеледі)
  - e-mail (индекстеледі)
  - телефон (индекстеледі)
  - қала (индекстеледі)
  - ел (индекстеледі)
  - суреті
3. Жүктеп көріңіз, кетіріңіз және мәліметтерді редакциялаңыз

### **Тапсырма 2**

1. Мәліметтер базасына жазу қосу үшін пішім құрау керек. Скрипт құрап қайтадан сұрауды өңдеңіз.
2. Пішімін құрап осы базаға ізденіс жіберің. Скрипт құрап қайтадан сұрауды өңдеңіз.
3. Пішімін құрап осы базасына сұрау жібереді, осыны іздеу үшін жазылғанда өзгерту керек (жолдың бәрі болу керек) Скрипті құрап қайтадан өңдеңіз.
4. Пішімін құрап осы базасына сұрау жібереді, осыны іздеу үшін жазылғанды кетіру керек (тек тегі болу керек). Скрипт құрап қайтадан сұрауды өңдеңіз.
5. Серверде жұмыс істейтінін тексеріңіз.
6. Ссылканы бірінші бетке жасаныз.



## Қорытынды Интернет- технологиясының даму болашағы.

Интернет- технология жедел дамып келеді, сондықтан желі болашағын елестету қиынға соғады. Бірақ технологияның болашағы бүгіннен басталады деген өз бар. Мысалы: қазіргі оқушылардың құрастырып отырған пилоттық үлгілері болашақта жүзеге асады, ал бұл интернеттің жаңа құрылысына әкеледі

Тағы бір жетістік - қазіргі таңды Интернетке қосылу кеңінен дамуда. Барлық желеілерде жоғары жылдамдықты қосылу орнатылса, Интернетке қосылу мүмкіндігін арттырады, мысалы білім алу, ойын-сауық интерактивті формаларына қосылу. Видео бағытты болашақта қазіргі тексттер сияқты желіде жеңіл қол жеткізілімді болады. Кең тараған практикаға сіздің сүйекті фильміңізді сұраныс жасау немесе оны жіберу процессі болуы мүмкін процессор лекциясын көзіңе қызықты тәртіпте тыңдау мүмкіндігі.

Екінші бір бағыты – интернетке мобильді телефондар арқылы қосылу кез-келген жерде тәулігіне 24 сағат бойы қосылуға мүмкіндік береді.

3G типті желінің таралуы желіге жоғары жылдамдықты қосылуы кез-келген смартфондар немесе коммуникаторлар арқылы жүзеге асады. Мобильді құралдар арқылы интернетке қосылумен өз демалысын, өмір сүру әлеуметін біріктіретін жас ұрпақтар да пайда болады. *" always on generator" - " желіге кез-келген уақытта қосылған "*

Мобильдік құралдар желігі қосылудың жаңа қажеттіліктерді құрады. Желісіз мобильдік құралдардың кең таралуы Интернет жүйесіне кез-келген қосылу мүмкіндігін арттырады. Мобильдік құралдарды пайдаланатын адамдар тобы бұл қондырылғыларды тек интернетке қосылу үшін ғана емес, әртүрлі тауарлармен қызметтерде кең көлемдерде таңдау, сатып алу, емделу мүмкіндіктеріне ие болады. Бұл процесс әртүрлі адамдар тобының арасындағы әрекеттесулер бағытын өзгертеді: мысалы сатушы мен сатып алушы, дәрігер мен емделуші және т.б. Мысалы көптеген аурулар денсаулығын тексерту үшін мобильдік құралдарды пайдалана алады. Бұл құралдар қазіргі таңдарда-ақ санаулы түрде шығарылуда ақпаратты жіберу үшін пейджер секілді клиент беліне қондырылатын құрал арқылы жіберіледі. Келешекте бұл сияқты мобильдік құралдар азайып, терінің астына қондырылатын технология пайда болу мүмкін. Мобильдік қондырғыларды еңгізу Интернет – қызмет түріне қарағанда да қоршаған ортаға үлкен дағдарыс әкеледі. Тіпті, Интернеттің пайда болуы бұлай әсер еткен жоқ. Қолданушылар кез-келген уақытта желілік және желілік емес құралдар негізінде Дербес компьютер арқылы, интерактивтік теледидар, ұялы телефондар, РДА, электрондық ойыншықтар, сандық камералар және мобильдік құралдар көмегімен Интернетке қосыла алады.

Мобильдік құралдар мен мобильдік бизнестің дамуы абоненттік төлем мен ақпараттық платформалар бағасын төмендеуіне әсеретеді. Жаңа технология бұрын пайдаланатын мүмкіншілігі жоқ қызметтерді қолдануға мүмкіндік береді.

Қазіргі таңдағы құралдар мен мобильдік құралдар арасындағы байланыс уақыт өте келе бастайды. *(бизнес- жеке көмекші + телефон - жеке көмекші+)*

*ойыншық + жеке көмекші* ). интернет- техниканың даму тенденциясының тағы бір белгісі ұй электроникалары мен электірлік құралдарының желігі біріктіріліп, интернет жүйесі арқылы басқарылады. Мысалы офисте отырып үйдегі робот - шаңсорғыштың қызметін басқаруға, телебағдарламаны жазуға, асты әзірлеуге болады.

Даму тенденциясының тағы бір белгісі болып интернеттің бизнесті жүргізу құралына табылады.

Бүгінгі таңның өзінде өзінде НТМХ - беттерін де бизнесті жүргізу, корпоративтің ақпараттық жүйемен байланысу операцияларын жүргізіп отырады.

Қазіргі web- беттер компаниялардың қызметін жарнамалау құралдарының тауарлар мен қызметтер көрсету құралдарына ауыстырылуда. Мобильдік бизнес көз қарасы барша техника үш фаза арқылы құрылады:

1) *электронды коммерциу кезеңі (E- Business)* пен сипатталады, мәліметтер бөлінген компьютерлердің тапсырмасы, ал мәліметтерді жіберу тек проводтар арқылы жүзеге асады. Ал интернет тағы да бір клиенттер мен жұмысшылар мәліметтеріне қосылу каналына айналады.

Бұл периодтың ерекшелігі - бизнес- процессте ешбір өзгеріссіз " web-ке шығу"  
2) *MC- Business (Multi Chanel Business - көп каналды бизнес)*- әртүрлі формадағы, көлемдегі, ақпаратты өңдеуі құралдар тобы қондырылған ортада өтетін бизнеспен сипатталады. сонымен қатар, ақпараттарды әндеудің кейбіреулері PDA, ұялы телефондар және т.с.с., мобильдік құралдар арқылы жүзеге асады. йымдар бірнеше каналдар жүйесін қолданады, оған комбинациялық әдістерін коналдық коммерциялық меншіктерін жатқызуға болады.

3) *Supranet Garner Group* аналитикалық компанияның тұжырымдамасы бойынша бұл фаза физикалық және электрондық дүниелердің бірігіуі кезінде пайда болады.

Бұл бөлімде процессор күші мен интернетке қосылу желілік емес барлық құралдар арқылы жүзеге асырылып отырған сонымен қатар, болашақта интернет арқылы тек аудио мен тек табысты ғана емес жарнамаланған тауардың иесі мен дәттін де жіберуге болады.

### **Бақылау сұрақтары:**

1. Интернет дамуының негізгі бағыты қандай?
2. Болашақта қандай құралдар интернетке қосылуды қамтамасыз етеді?
3. Болашақтағы мобильдік бизнестің қандай фазасы бар?
4. " ALWAIR on generation " терминінің мағынасы қандай?

### Қолданылған әдебиеттер

1. Петров В.Н. Информационные системы. Учебник. СПб: Издательство "Питер", 2002.
2. Прохоров А.Н. Интернет: как это работает. - СПб: БХВ-Петербург, 2004.
3. Филимонов А.Ю. Протоколы Интернета. - СПб: БХВ-Петербург, 2003.
4. Найк Д. Стандарты и протоколы Интернета. Пер. с англ. - М.: 1999.
5. Ганеев Р.М. Проектирование интерактивных WEB-приложений. М.: 2001.
6. А. Павлов. CGI-программирование: учебный курс. - СПб: Питер, 2000.
7. А.В. Картузов, Д.В. Николенко. Программируем на языке Java: краткий курс. - СПб: Наука и техника, 2001.
8. Ливингстон Д., Белью К., Браун М. Perl 5. Web - профессионалам: Пер. с англ. - К.: Издательская группа ВHV, 2001.
9. Косентино К. PHP. Web - профессионалам: Пер. с англ. - К.: Издательская группа ВHV, 2001.
10. Э. Кингсли-Хью, К. Кингсли-Хью. JavaScript 1.5. Учебный курс. - СПб: Питер, 2001.
11. Бишоп Д., Java 2. Питер, 2002
12. Д. Вебер, Технология Java в подлиннике: Пер. с англ. - СПб.: БХВ-Петербург, 2000.
13. В.Г. Олифер, Н.А. Олифер. Компьютерные сети. Учебник. - СПб: Питер, 1999.
1. Кузнецов С.Д. PHP 4.0. Руководство пользователя. - М.: Майор, 2001.
2. Дунаев В.В. Сам себе Web-дизайнер. СПб.: Арлит. 2002.
3. Леонтьев Б. Web-дизайн: Хитрости и тонкости: - М.: Познавательная книга плюс, МиК, 2001.
4. Будилов В.А. PHP 5. Экспресс-курс. - СПб: БХВ-Петербург, 2005.
5. Андерсон Э. Macromedia Flash MX 2004. Пер с англ. - М.: ИТ Пресс, .
6. Николенко Д.В. Практические занятия по JavaScript. СПб.: 2002.
7. Гулятьев А.К., Машин В.А. Уроки WEB-мастера. СПб.: 2002.
8. Ахромов Я.В. Технологии Web-дизайна и Flash-технологии. Серия "Высшее образование", Ростов н/Д: Феникс, 2004.
9. Жумагалиев Б.И. Лабораторный практикум по интернет-технологиям. Учебное пособие. Алматы: ААЭИС, 2003.
10. Дейтел Х.М., Дейтел П.Д., Сантри С.И. «Технологии программирования на Java», 2003
11. Питер Вейнер. Языки программирования Java и JavaScript, ЛОРИ, 1998
12. Шкрыль А.А. PHP-это просто. Программируем для WEB-сайта. - СПб: БХВ-Петербург, 2006.
13. Успенский И.В. Интернет как инструмент маркетинга. - СПб: БХВ-Петербург, 1999

## Мазмұны

Интернет технологияларына кіріспе .....	3
Тарау 1. Интернеттің ұйымдастырылуы .....	8
1.1 Интернеттің негізгі түсініктері.....	8
1.1.1 Серверлерлік компьютерлер және клиенттік – компьютерлер.....	8
1.1.2 Серверлерлік - қосымшалар және клиенттік – қосымшалар.....	8
1.1.3 Провайдерлер және олардың желілері.....	9
1.1.4 Провайдер желілерінің біріктірілуі.....	10
1.1.5 Провайдер желісінің иерархиясы.....	10
1.2. Интернет коммуникациялардың модельдері .....	11
1.2.1 Pull- и Push-модели Интернет коммуникациясы.....	11
1.2.2 Интерактивтеу.....	12
1.2.3 Гипермәтін .....	12
1.2.4 Мультимедиа.....	13
1.2.5 Қатысу эффектісі.....	13
1.2.6 Желілік навигация.....	14
1.2.7 Интернет сервисіне коммуникациялық сипаттама.....	14
1.3 Қолданушылардың интернетке қосылу мүмкіндігі .....	15
1.3.1 Телефон линиясы арқылы қосылу .....	15
1.4 Интернет желісіне қосылу технологиялары .....	17
1.4.1 Ассиметриялық DSL (ADSL).....	17
1.4.2 ISDN – технологиясы.....	19
1.5 Интернеттегі web – серверлер .....	20
1.5.1 WEB-сервердің жұмыс жасау механизмі.....	20
1.5.2 Статикалық және динамикалық беттер.....	20
1.5.3 CGI – технологиясы.....	21
1.6 Web – қорларды алу технологиялары.....	22
1.6.1 Web – беттерді кәштеу.....	22
1.6.2 Прокси-сервердің жұмыс жасау механизмі.....	22
1.6.3 Құжаттардың сақталу мерзімі.....	23
1.6.4 Кәштеу процессін оптималдау негізінде жеделдету .....	23
1.7 Интернетте іздеу технологиясы.....	24
1.7.1 Web-те іздеу.....	24
1.7.2 Индекстендірілген каталогтар.....	25
1.7.3 Домендік атау арқылы іздеу.....	25
1.7.4 Іздеу машиналарының жұмыс жасау механизмі.....	26
1.7.5 Индексті құрастыру.....	27
1.7.6 Индекс бойынша іздеу.....	27
Тарау 2. Интернет сервистері.....	29
2.1 Электрондық пошта.....	29
2.1.1 Электрондық почта мекен-жайы.....	29
2.1.2 Екі сервис – екі хаттама.....	29

2.1.3	Пошталық клиенттердің негізгі қызметтері.....	30
2.1.4	Хатты жіберу және жеткізу .....	30
2.1.5	Хатқа енгізілген арнайы таңбалар жіберу.....	31
2.2	Файлдар мен алмасу .....	31
2.2.1	FTP файлдарды тасымалдау хаттамасы.....	31
2.2.2	FTP – ресурстар мекен-жайының форматы.....	32
2.2.3	FTP – клиенттер.....	32
2.2.4	ftp. exe көмегімен қосылу.....	32
	Тарау 3. Интернет желісінде ақпаратты қорғау.....	34
3.1	Қауіпсіздік мәселелері.....	34
3.1.1	Криптография және интернет.....	34
3.1.2	Симметриялық және ассиметриялық кілттер.....	34
3.1.3	Цифрлік қолтанба.....	35
3.2	Интернетте қолданушыларды идентификациялау.....	36
3.2.1	Cookies механизмі.....	36
	Тарау 4. Web-қосымшалар құру технологиясы.....	38
4.1.	Интернет-қосымшаларын құру.....	38
4.1.1	HTML тіліне жалпы түсінік.....	38
4.1.2	Динамикалық HTML.....	41
4.1.3	Macromedia Flash бағдарламасындағы Action script тілі.....	42
4.2	Клиенттік қосымшалар құру технологиялары.....	66
4.2.1	Java Script тілі.....	66
4.2.2	Visual Basic Script.....	70
4.2.3	Java тілі.....	75
4.3	Интернеттің серверлік - қосымшаларын құру технологиялары.....	90
4.3.1	Серверлерді бағдарламалау тілдері.....	90
4.3.2	«Клиент-сервер» технологиясы.....	97
4.3.3	ISAPI бағдарламасы.....	98
	Қорытынды. Интернет дамуының болашағы.....	105
	Қолданылған әдебиеттер .....	107



Формат 60x84 1\12  
Объем 111 стр., 9,25 печатных листа  
Тираж 20 экз.  
Отпечатано  
В Редакционно- издательском отделе  
КГУТиИ им.Ш.Есенова  
г.Актау, 32мкр.