32.97
С 64

А. СОПУЕВ
Т.Д. АСЫЛБЕКОВ
У.Д. МОЛДОЯРОВ

WEB - ПРОГРАММАЛОО

1

- *HTML*
- *DHTML*
- *JavaScript*

Ош 2008

УДК 004.43

ББК 32.973-01
С 64

ст. ч/з

Рецензенттер: физика-мат. илим. доктору, профессор Алымкулов К.А.
физика-мат. илим. доктору, профессор Сатыбаев А.Ж.

Сопуев А., ж.б.
С 64 Web-программалоо. HTML, DHTML, JavaScript: 1-бөлүк: Окуу колдонмо / А.Сопуев, Т.Д. Асылбеков, У.Д. Молдояров - Ош: ОшМУ, «Билим» редакциялык-басма бөлүмү, 2008. - 67 б.

ISBN 978-9967-03-451-8

Окуу колдонмодо HTML, DHTML, JavaScript программалоо тилдеринин каражаттары менен web-беттерди түзүү технологиясы баяндалып, негизги көңүл динамикалык жана интерактивдүү Web-беттерди программалоого бөлүнгөн. Теориялык материалдарды практикада колдонуу түшүнүктүү болсун үчүн мисалдар келтирилген.

Колдонмо студенттер, окутуучулар жана Web-сайттарды өз алдынча түзүүнү каалагандар үчүн арналат.

Ош мамлекеттик университетинин жана Ош технологиялык университетинин Окумуштуулар Кеңештеринин чечими менен сунуш кылынды.

: 004.43

: 32.973-01

итети, 2008
рситети, 2008

С 64

Мазмуну

Кириш сөз 5

1. Интернет жана Web–сайт жөнүндө түшүнүк. 6

1.1. Интернет жөнүндө түшүнүк. 6
1.2. Компьютердик тармактар. 6
1.3. WWW деген эмне? 7
1.4. Интернеттеги даректер жана протоколдор. 8
1.5. Издөө системалары 9
1.6. Web браузерлер 9
1.7. "Клиент-сервер" технологиясы. 10
1.8. WWW – массалык маалымат каражаты катарында. 10

2. Web–бетти түзүү инструменттери. 11

2.1. Web–беттердин редакторлору жөнүндө жалпы маалымат. 11
2.2. Web–беттерди Word тексттик редакторунда даярдоо. 11
2.3. Web–беттерди Excel таблицалык процессорунда даярдоо. 13
2.4. Web–беттерди MS Access колдонмосунда даярдоо. 14
2.5. Web–беттерди PowerPoint колдонмосунда даярдоо. 15
2.6. BOOM.RU Web–беттердин мастери. 15
2.7. Microsoft FrontPage программасы. 16

3. HTML тили жана анын тегдери. 17

3.1. HTML тилинин базалык элементтери 17
3.2. HTML-документтердин структурасы. 18
3.3. HTML-документтеги түшүндүрмөлөр. 19
3.4. Бөрктөр деңгээли. 19
3.5. Параграф. 20
3.6. Номерленген жана маркирленген тизмелер. 20
3.7. Тексттин элементтерин калыптоо. 21

4. Таблицаларды түзүү. Web-беттин интерактивдүүлүгү. 22

4.1. Таблицаларды түзүү. 22
4.2. Рамкаларды колдонуу жана аларды берилгендердин булагы менен байланыштыруу. 22
4.3. Web-беттин интерактивдүүлүгү жана гипершилтеме. 24

5. Форма. Формага башкаруучу элементтерди жайгаштыруу. 26

5.1. Форма. 26
5.2. Киргизүү талаасы. 26
5.3. Переключатель (Радиобаскыч). 27
5.4. Желекче (Check Box, Флажок). 28
5.5. Баскыч (Кнопка). 29
5.6. Көлөмдүү тексттерди киргизүү талаасы. 29
5.7. Сүрөттөлүштөрдү Web-бетке жайгаштыруу. 30

Библиотека Ошского государственного университета
ИНВ № 932048

6. Web-беттер үчүн программаларды түзүү. JavaScript тили....31
6.1. JavaScript программалоо тили жөнүндө түшүнүк....31
6.2. JavaScript программалоо тилинде сценарийлерди түзүү....32
6.3. Маалыматтарды киргизүү жана чыгаруу....33
7. Берилгендердин тиби. Операторлор....34
7.1. Берилгендердин тиби....34
7.2. Операторлор....35
8. Сценарийлерди аткаруу тартибин башкаруу....38
8.1. Логикалык туюнтмаларды түзүү. Маанилерди салыштыруу....38
8.2. Цикл операторлору....38
9. Программалык коддогу тармактануу....41
9.1. if-else конструкциясы....41
9.2. switch-case конструкциясы....43
10. Функциялар. Колдонуучунун функциясы....45
10.1. Тиркелген объекттер жана функциялар....45
10.2. Кокустук сандардын генератору....46
10.3. Колдонуучунун функциясы....46
10.4. Тиркелген функциялар....47
11. Массивдер....48
11.1. Массивдерди түзүү жана колдонуу....48
11.2. Массивдер методу....49
11.3. Сандык массивдердин статистикалык мүнөздөмөлөрүн аныктоо....50
12. Динамикалык Web-беттерди түзүү үчүн окуяларды колдонуу....52
12.1. Окуя жөнүндө түшүнүк....52
12.2. Окуянын касиеттери....52
12.3. Баскычтардын окуясы....52
13. Датаны жана убакытты аныктоо....53
14. Window объекти....55
15. DHTML. Стилдер жана катмарлар....58
16. Сырткы файлдар менен иштөө....60
16.1. Сырткы файлдардан берилгендерди киргизүү....60
16.2. Берилгендерди файлда сактоо....62
16.3. Конoктор китебинин модели....63
16.4. Web-бетти пароль менен коргоо....65
Колдонулган адабияттар....67

Кириш сөз

Бүгүнкү күндө компьютердик жана информациялык технологиялардын дүркүрөп өсүшү менен адам баласынын ишмердүүлүгүн Интернет тармагысыз элестетүү кыйын болуп калды. Күн сайын миллиондогон адамдар электрондук почтаны колдонуп, Web-сайттардан керектүү маалыматтарды издеп, эң акыркы жаңылыктарды окуп, интернет аркылуу маалымат алмашып жатышат.

Дүйнө жүзүндөгү миңдеген программисттер динамикалык жана интерактивдүү Web-сайттарды түзүүдө JavaScript программалоо тилинин көп деңгээлдүү менюларын, катышуу эсептегичтерин, коноктор китебин, маалыматтар базасын жана башка интерактивдүү элементтерин колдонууда.

Окуу колдонмонун 1-бөлүгүндө HTML, DHTML, JavaScript программалоо тилдеринин каражаттары менен Web-беттерди түзүү технологиясы баяндалып, негизги көңүл динамикалык жана интерактивдүү Web-беттерди программалоого бөлүнгөн.

Теориялык материалдарды практикада колдонуу түшүнүктүү болсун үчүн мисалдар келтирилген. Бул мисалдарды аткаруу үчүн атайын программаларды орнотуунун зарылчылыгы жок. Аларды «Блокнот» тексттик редакторунда даярдап, Internet Explorer маалымдагычында аткарып көрүү жетиштүү болот. Даярдалган Web-беттердин башка программалык каражаттарда иштешин текшерүү үчүн Microsoft FrontPage редакторунда сынап көрсө болот.

Окуу колдонмону өз алдынча иштерди аткарууда, Web-беттерди жана Web-сайттарды программалоодо, электрондук окуулуктарды түзүүдө, курстук, конкурстук жана дипломдук иштерди (проекттерди) даярдоодо кеңири колдонууга болот. Теориялык жана практикалык билимдерди (көндүмдөрдү) бышыктоо үчүн окуу колдонмодо келтирилген мисалдарды татаалдаштырып, кеңейтип, ар түрдүү варианттарда чыгарууга аракет кылып көрүү сунушталат. Эгерде колдонмодо берилген мисалдарды бириктирип жыйынтыктаса, анда студенттердин өздүк сайты негизинен даяр болуп калат, андан кийин аны өркүндөтүү ар бир колдонуучунун өзүнүн интеллектуалдык эрудициясына байланыштуу болот. Web-сайттарды түзүү боюнча кеңири маалымат алууну каалоочулар үчүн адабияттар да келтирилди.

Окуу колдонмонун 2-бөлүгүндө PHP программалоо тилинин каражаттары менен Web-сайттарды түзүү технологиясы баяндалат.

Окуу колдонмо боюнча пикириңиздерди a_sopuev@rambler.ru, ular_osh@rambler.ru электрондук даректерине жиберсеңиздер болот.

Колдонмо студенттер, окутуучулар жана Web-сайттарды өз алдынча түзүүнү каалагандар үчүн арналат.

1. Интернет жана Web–сайт жөнүндө түшүнүк.

1.1. Интернет жөнүндө түшүнүк.

Азыркы мезгилде Интернет түшүнүгү адам баласынын ишмердүүлүгүнүн бардык чөйрөлөрүнө терең сүңгүп кирип, маалыматтар менен болгон иш аракеттерди аткаруунун негизги каражаттарына айланды.

Интернеттин негиздери болуп компьютердик тармак жана маалыматтар ресурсу болуп эсептелет.

Интернет (Internet, International – эл аралык жана net - тармак) деп маалыматтык ресурстарды биргеликте пайдаланууга мүмкүн болгон компьютерлердин глобалдык тармагын айтабыз.

Интернет 1969-жылы 2-январда АКШнын коргоо министрлигинин ARPANET долбоорунун негизинде түзүлүп, компьютердик тармакка эң оболу окуу мекемелери, коргоо министрлигинин изилдөөлөр жана долбоорлор борборлору жана аскердик коргонуу борборлору туташтырылган. Ошондуктан интернеттин түзүлгөн күнү деп 1969-жылдын 2-январын кабыл алса болот.

Эгерде 1980-жылдары интернетти аскердик, илимий жана коммерциялык уюмдар гана колдонушкан болсо, 1990-жылдардан баштап интернетке үйдөгү компьютерлерди да туташтыруу мүмкүн болуп калды.

1.2. Компьютердик тармактар.

1946-жылы компьютерлер пайда болгондон баштап эле аларды бири-бири менен байланыштыруучу тармак түзүп, маалыматтарды (информацияларды) биринен экинчисине жөнөтүү же кабыл алуу зарылчылыгы келип чыккан.

Маалыматтарды бир компьютерден экинчи компьютерге магниттик дисктер (дискета), компакт-дисктер же флешкалар менен көчүрүп алса болот. Бирок бул ыкма убакытты талап кылат жана ыңгайсыз да болот. Эгерде эки компьютерди кабель менен байланыштырып, маалыматтарды берүүнү камсыз кылган программаны жазсак, анда эң жөнөкөй компьютердик тармакты алган болот элек.

Компьютердик тармак деп эки же андан ашык бири-бири менен байланыш каражаттары аркылуу туташтырылган жана бирдей эрежеде иштеген компьютерлерлердин жыйындысын айтабыз.

Компьютерлердин тармагын төмөнкүдөй бөлүп кароого болот: ***локалдык*** (жергиликтүү); ***регионалдык*** (аймактык); ***корпоративдик*** жана ***глобалдык*** (дүйнөлүк) тармак.

1). Локалдык тармак - мекемеде же аудиторияда жайгашкан чоң эмес компьютердик тармак болуп эсептелет.

2). Регионалдык тармак - регион чегиндеги локалдык тармактардын биригүүсү.

3). Корпоративдик тармак - кандайдыр бир мекемелердин же фирмалардын сырткы колдонуучулардан корголгон локалдык же регионалдык тармагы.

4). Глобалдык тармак – ар түрдүү тармактарды байланыш каналдарынын, шлюздардын жана протоколдордун жардамында бириктирүүчү жана бирдей режимде иштөөчү бүткүл дүйнөлүк тармак.

Компьютердик тармактарда төмөнкү байланыш каражаттары колдонулат:

- ***кабель,***
- ***телефон,***
- ***радио,***
- ***спутник***
- ***оптика-волокондук*** (маалымат жарыктын ылдамдыгына барабар ылдамдык менен берилет).

Маалыматтарды берүүдө тармактык плата, адаптер жана модем (модулятор/демодулятор) колдонулат. Бул электрондук түзүлүштөрдүн милдети болуп компьютерден берилүүчү маалыматтарды (жыштыгы, амплитудасы жана фазасы боюнча) электрдик, радио же жарык сигналдарына айландыруу менен жөнөтүү жана кабыл алынган сигналдарды кайрадан маалыматка айландыруу болуп саналат. Модем компьютерден берилген команда боюнча керектүү телефон номерин өзү терип алат да башка компьютердеги модем менен байланыша алат.

Модемдин жардамы менен телефон аркылуу маалыматтарды берүү эң кеңири таралганы менен, маалыматтарды берүү ылдамдыгы анча жогору болбойт.

Маалыматтарды берүү ылдамдыгынын бирдиги ***бит/секунда*** болуп эсептелет.

Интернеттин жаңы мүмкүнчүлүктөрүнөн пайдалануу үчүн ылдамдыгы 33600 бит/секундадан чоң болгон модемдердеги колдонуу зарыл. Бүгүнкү күндө интернетке туташтыруу үчүн сунушталган ылдамдык 56 Кбит/с. түзөт. Ал эми оптика-волокондук каналдар боюнча маалыматтарды берүү ылдамдыгы 1 Гбайт/с. чейин жетет. Компьютерлерди интернетке туташтыруу үчүн интернет кызматын камсыз кылуучу уюмдарга – провайдерлерге (поставщик) кайрылуу керек.

1.3. WWW деген эмне?

Интернетти колдонуудагы негизги объекттердин бири болуп маалыматтык ресурстар, б.а. WWW (Вэ Вэ Вэ же үч дабл-ю деп окулат) ресурсу (сервиси) эсептелет. Аны жөнөкөй болсун үчүн Web ресурсу деп аташат. Интернеттеги маалыматтардын жыйындысы - WWW (World Wide Web – бүткүл дүйнөлүк желе) деп аталган бүткүл дүйнөлүк маалыматтар ресурсун (сервисин) түзөт.

Web менен компьютердик тармактардын байланышын төмөнкүдөй түшүндүрүүгө болот: Web – маалыматтар ресурсу, ал эми компьютердик тармактар - маалыматтарды берүү каражаты болуп эсептелет.

WWW ресурсу интернетти колдонгондордун бардыгы үчүн ачык жана жеткиликтүү. WWW ресурсу менен дүйнөдөгү эң чоң китепканалардын бири да берилгендердин булагынын көлөмү боюнча, керектүү маалыматты табуу жана колдонуу боюнча теңтайлаша албайт. WWW ресурсунун документтеринде тек-

стти жана сүрөттү гана эмес, музыканы да, фильмдерди да, программаларды да жана берилгендердин башка булактарына шилтемелерди да табууга болот.

WWW ресурсундагы маалыматтар ал тармакка бириктирилген компьютерде Web-бет түрүндө жайгаштырылат. ***Web-бет*** деп интернет тармагы үчүн даярдалган документти айтабыз. Web-беттердин жыйындысын ***Web–сайт*** деп, ал эми сайт жайгаштырылган компьютерлерди ***сервер*** (Web-сервер) деп айтабыз.

Web-беттерди түзүү үчүн ***гипертекст*** (hypertext) технологиясы, б.а. берилген маалыматтарды, объекттерди, элементтерди жана беттерди бири-бири менен байланыштыруу методдорунун технологиясы колдонулат.

Гипертексттик документтер HTML [Эйч Ти Эм Эл] (Hyper Text Markup Language - гипертекстти түзүү тили) – тилинин каражаты менен даярдалат.

Гипертексттик документтерди тармак боюнча жөнөтүү үчүн HTTP [Эйч Ти Ти Пи] (Hyper Text Transfer Protokol) – технологиясы колдонулат.

1.4. Интернеттеги даректер жана протоколдор.

Интернетте даректи аныктоо үчүн эки түшүнүк колдонулат: ***дарек*** жана ***протокол***.

Интернетке туташтырылган ар бир компьютер уникалдык IP (Ай Пи) дарекке ээ. IP дарек ар бири 0 дөн 255 ке чейинки маанилерден турган жана бири биринен чекиттер менен ажыратылган төрт сандардын жыйындысынан турат, мисалы: 192.168.0.1.

Колдонуучулар үчүн IP дареги ыңгайсыз болгондуктан эсте сактоого жана колдонууга ыңгайлуу болгон Web-серверлердин DNS (Domain Name System) дареги, б.а. аталыштардык домендик системасы колдонулат. DNS дарек боюнча уюмдун жайгашкан дарегин жана профилин аныктоого болот. Мисалы, microsoft.com домени интернетте Microsoft корпорациясын түшүндүрөт.

Бул системада колдонуучулардын группасына карата аталыштар дайындалат жана алардын ар бир деңгээли **домендер** деп аталат. Домендин түзүүчү бөлүктөрү точка менен ажыратылат. Жогорку деңгээлдеги домен оң жакта, төмөнкү деңгээлдеги домен сол жакта жайгашкан болот. Төмөнкү деңгээлдеги домендер үчүн каалагандай аталыштар берилиши мүмкүн, бирок эң жогорку деңгээлдеги домен үчүн, б.а. домендин эң акыркы бөлүгү атайын келишимдер бар: эреже катары ал мамлекеттин же ар түрдүү уюмдардын кодун билдирет. Мисалы: kg - Кыргызстан; ru – Россия; kz – Казакстан; uz – Өзбекстан; au – Австралия; br – Бразилия; ca – Канада; cn – Кытай; de – Германия; jp – Япония; ua – Украина; uk – Великобритания; us - АКШ; com – коммерциялык уюмдар; edu – билим берүү мекемелери; net – тармактык ресурстар сервери; org – ар түрдүү коомдук уюмдар; int – эл аралык домен.

Интернетке туташтырылган компьютерлерде маалыматтарды берүүнүн жана компьютерлердин дарегин аныктоонун TCP/IP (Ти Си Пи / Ай Пи) протоколу деп аталган өзгөчө эрежелери колдонулат. TCP (Transmission Control Protocol) - берүүнү башкаруу протоколу, ал эми IP (Internet Protocol) – Интернет протоколу болуп саналат.

Интернет менен иштөөдө жөнөкөй домендик адрес эмес, ресурстардын көрсөткүчү болгон универсалдык URL [Ю Эр Эль] (Uniform Resource Localitor) адреси колдонулат. URL – деп кайсы протокол менен кайрылууга боло тургандыгы жазылган каалагандай ресурстун дарегин айтабыз. Мисалы, http:/www.mail.ru дарегинде http протоколу жана www.mail.ru Web-сайтынын аталышы көрсөтүлгөн. http:// - Web-сайтка болгон шилтемени (ссылка) http форматында жөнөтүлгөндүгүн түшүндүрсө, htp:// - FTP-түйүнгө (файлдар сакталуучу серверге) болгон шилтеме (ссылка) htp форматы боюнча, б.а. файлдарды жөнөтүү протоколу боюнча издөө жөнөтүлгөндүгүн түшүндүрөт.

FTP [Эф Ти Пи] (File Transfer Protocol) сервиси файлдар сакталуучу серверди билдирет.

1.5. Издөө системалары

Көпчүлүк Web-сайттар издөө системасы катарында да колдонулат. Мисалы, ru домененинен: rambler.ru, mail.ru, yandex.ru, lycos.ru, aport.ru, km.ru, ал эми дүйнөлүк издөө системалары катары yahoo.com, infoseek.com, hotbot.com, altavista.digitial.com, hotmail.com, files.com, whowhere.com жана башкаларды алууга болот.

Кыргызстандын интернеттеги коду болгон kg домененинде төмөнкүдөй сайттар бар: www.kg, www.kyrgyzstan.kg, www.president.kg (Кыргыз Республикасынын Президентинин официалдык сайты), www.kyrgyzpatent.kg (Кыргызпатент), www.akipress.kg (АКИpress Информациялык Агенствосу) www.elcat.kg (Интернет Сервис Провайдер), www.vb.kg (Вечерний Бишкек газетасы), www.kabar.kg (Кыргыз Улуттук Информациялык Агентствосу), www.novosti.kg (Новости Кыргызстана), www.toktom.kg («Токтом» Информациялык борбору), www.airport.kg (Эл аралык Манас Аэропорту), www.osh.kg (Ош шаары боюнча маалыматтар бар) жана башка сайттар бар.

1.6. Web браузерлер

Web-сайттарды интернеттен көрүү үчүн Web браузерлер деп аталган программалар – ***маалымдагычтар*** (обозреватель) колдонулат. Алардын ичинен эң көп колдонулгандары төмөнкүлөр: Microsoft Internet Explorer (http: //www.microsoft.com/ie) жана Netscape Navigator (http://www.netscape.com).

Web-бетти окуу мүмкүнчүлүгүнө ээ болуу үчүн документти маалымдагычтар окуй ала турган форматта сактап, Web-серверге жайгаштыруу жетиштүү болот. Интернетке туташтырылган жана компьютердик тармактын иштөөсүн камсыз кылуучу атайын программа (мисалы, http://www.apache.org) орнотулган каалагандай компьютер Web-сервер болушу мүмкүн.

Microsoft Internet Explorer Web браузерин (маалымдагычты) экранга жүктөө үчүн

Пуск \ Программы \ Internet Explorer командасын колдонобуз.

Web-сайтты окуу үчүн анын URL дарегин маалымдагычтын (обозреватель) даректи жазуучу жолчосуна жазып, Enter баскычын басуу керек.

1.7. "Клиент-сервер" технологиясы.

WWW ресурсунда "клиент-сервер" технологиясы колдонулат, б.а. каалаган колдонуучунун – клиенттин издөөсүнө (запрос) серверден ар дайым жооп алууга болот, анткени Web-серверде документтердин, программалардын жана башка маалыматтардын каерде жайланышкандыгы, издөө салынган протоколдун түрү жана түйүнгө тиешелүү болгон бардык маалыматтар сакталат.

Мисалы, http://www.microsoft.com/Misc/Shotcuts.htm издөөсү боюнча Web-сервер эң оболу маалымдагычтан келген издөөнү интерпретациялап, Misc директориясына өтөт жана ал жерден Shotcuts.htm документин табат жана аны маалымдагычтын терезесине чыгарып берет.

1.8. WWW – массалык маалымат каражаты катарында.

Азыркы мезгилде WWW ресурсу эбегейсиз кеңейип, интернеттин жардамы менен керектүү маалыматтарды (тексттик, графикалык жана мультимедиалык файлдарды) алуу жана жиберүү, электрондук почта кызматынан пайдалануу, конференцияларды уюштуруп, бири-бири менен пикир алмашуу, таанышуу, электрондук соода-сатык иштерин жүргүзүү жөнөкөй эле маселелерге айланды.

Бүгүнкү күндө күнүнө 275 миллиондон көп адамдар интернет тармагынан пайдаланууда. WWW ресурсу эч кимге таандык эмес, ошондуктан аны массалык маалымат каражаты пайдаланууга болот.

WWW ресурсун төмөнкүдөй мүнөздөөгө болот:

- ***Мультимедиалуулугу*** – көп түрдүү документтерди (тексттерди, сүрөттөрдү, музыкаларды, аудио жана видеоклиптерди, фильмдерди, ж.б.) алууга болот;
- ***Жалпыга жеткиликтүүлүгү*** - Web-серверге жарыяланган маалыматтын дароо эле бүткүл дүйнөгө айкын болушу;
- ***Динамикалык интеграциялоо мүмкүнчүлүгүнө ээ болушу*** – китеп жана макалалардан айырмаланып, керектүү булактарга Web-серверлерге дароо шилтеме жасоо мүмкүнчүлүгүнө ээ болушу;
- ***Жаңыланып туруу мүмкүнчүлүгүнө ээ болушу*** - Web-беттеги каталарды же маалыматты жаңы берилгендер менен толуктап коюунун эч качан кеч болбогондугу;
- ***Интерактивдүүлүгү*** - Web-бетти колдонуучулар менен болгон байланышты орнотуунун эң жөнөкөйлүүлүгү – ал үчүн электрондук почтанын дарегин көрсөтүп коюу жетиштүү.

2. Web–бетти түзүү инструменттери.

2.1. Web–беттердин редакторлору жөнүндө жалпы маалымат.

Web–беттерди даярдоо үчүн документти htm же html форматта сактоо мүмкүнчүлүгүнө ээ болгон редакторлорду колдонуу керек. Web–беттердин редакторлорун тандоо маселенин коюлушуна жана татаалдыгына жараша болот.

Мисалы, эң жөнөкөй Web–беттерди "Блокнот" стандарттык программасында же WordPad тексттик редакторунда даярдоого болот.

Web–бетте интерактивдүү элементтерди камсыз кылуучу кичинекей программаларды – сценарийлерди кошуу үчүн JavaScript программалоо каражатын колдонуу керек.

Web–беттин функционалдык мүмкүнчүлүктөрүн жогорулатуу жана башкаруу элементтерин кошуу үчүн ActiveX же Java-аплет программалык объекттерин колдонуу керек.

Көп колдонулуучу Web-беттердин редакторлоруна төмөнкүлөрдү кошууга болот:

- MS Office колдонмолору (Word, Excel, PowerPoint);
- Web–беттердин мастерлери;
- GeoCities PageBuilder;
- Macromedia Dreamweawer;
- Microsoft FrontPage;
- Adobe GoLive.

2.2. Web–беттерди Word тексттик редакторунда даярдоо.

MS Office колдонмолорунун бардыгын HTML документти түзүү үчүн колдонууга болот.

Word тексттик редакторунда Web-беттерди жана ошондой эле рамкалары бар беттерди да даярдоого болот. Мындан тышкары каалаган Word документин Web форматта көрүү үчүн сактоого болот.

2.2.1. Word документти HTML форматында сактоо.

Word документти HTML форматында сактоо үчүн төмөнкүлөрдү аткарабыз:

1). ***Файл / Сохранить как веб-страницу*** командасын колдонобуз;

2). ***Сохранение документа*** деген диалогдук терезе ачылат жана анда учурдагы заголовканын аты чыгат жана ***Изменить*** деген баскыч пайда болот.

3). ***Изменить*** баскычын тандап заголовканын жаңы атын жазабыз жана ***ОК*** баскычын басып, Web-бетти HTML форматында сактайбыз.

2.2.2. Word документе Web-бетти даярдоо.

Word тексттик редакторунда жаңы Web-бетти даярдоо үчүн төмөнкү командаларды аткарабыз:

1. ***Файл*** менюсундагы ***Создать*** командасын тандайбыз.

2. ***Создание документа*** панелиндеги ***Web-страница*** опциясын алабыз.

3. Web-беттин жаңы терезеси ачылат. Керектүү маалыматтарды жайгаштырабыз.

4. Документти Web-бет катары сактоо үчүн ***Файл / Сохранить как ...*** командасын колдонобуз;

5. ***Сохранение документа*** деген диалогдук терезе ачылат жана анда учурдагы заголовканын аты чыгат жана ***Изменить*** деген баскыч пайда болот.

6. ***Изменить*** баскычын тандап заголовканын жаңы атын жазабыз жана ***ОК*** баскычын басып, Web-бетти HTML форматында сактайбыз.

2.2.3. Темаларды колдонуу

Тема деп документтин тиешелүү элементтерин автоматтык түрдө форматтоо үчүн колдонулуучу форматтардын жыйындысын айтабыз.

Темаларды колдонуу документти түзүү иштерин гана ылдамдатпастан, документке профессионалдык түр берүүгө да мүмкүнчүлүк берет. Темада беттин төмөнкү элементтеринин форматтары берилет:

- Шилтемелер панели;
- Фондун түсү жана сүрөтү;
- Ажыратуучу сызыктар;
- Бөрктөрдүн (заголовка) жана негизги тексттин шрифттеринин стили;
- Таблицалар;
- Номерленген жана маркирленген тизмелер;
- Гипершилтемелер.

Документке теманы колдонуу үчүн төмөнкүлөрдү аткарабыз:

1. Word тексттик редакторунда жаңы Web-бетти даярдайбыз;
2. ***Формат*** менюсунан ***Тема*** командасын тандайбыз;
3. ***Тема*** диалогдук терезесинен өзүбүзгө жаккан теманы тандап, ОК баскычын басабыз.

Натыйжада биздин документибиз тандалган теманын форматы боюнча даярдалган болот.

Теманы алмаштыруу үчүн 1-3 пункттагы командаларды кайталап, ***Выберите тему*** деген тизмеден жаңы теманы тандайбыз.

Теманы колдонуудан баш тартуу үчүн ***Нет темы*** деген опцияны тандайбыз.

2.2.4. Рамкасы бар Web-бетти даярдоо.

Word тексттик редакторунда рамкасы бар жаңы Web-бетти даярдоо үчүн төмөнкү командаларды аткарабыз:

1. ***Файл*** менюсундагы ***Создать*** командасын тандайбыз;
2. ***Создание документа*** панелиндеги ***Web-страница*** опциясын алабыз;
3. Рамкасы бар жаңы Web-бетти даярдоо үчүн ***Формат*** менюсунан ***Рамки / Новая страница рамок*** командасын алабыз;
4. Пайда болгон Рамка инструменттер панелинен төмөнкү параметрлер-

дин ичинен керектүүсүн тандайбыз:

- ***Новая рамка слева;***
- ***Новая рамка справа;***
- ***Новая рамка сверху;***
- ***Новая рамка снизу.***

5. Рамканын четин чычкандын жардамы менен керектүү позицияга жылдырабыз;

6. Эгерде беттин кандайдыр бир бөлүгүн кошумча рамкага бөлгүбүз келсе, анда курсор менен ошол бетти тандап, 4- жана 5-пункттардагы аракеттерди кайталайбыз.

Натыйжада рамкасы бар Web-бетти алабыз. Рамкалардын жардамы менен жогорку жана төмөнкү колонтитулдарды даярдоого болот.

2.2.5. Гипершилтемелерди кошуу.

Гипершилтемени кошуу үчүн беттеги текстти же сүрөттү бөлүп алып, ***Стандартная*** панелиндеги ***Вставить гиперссылку*** баскычын басабыз же ***Вставка / Гиперссылка*** командасын колдонобуз. Экранда пайда болгон **Добавление гиперссылки** терезесинин **Адрес** деген талаасына төмөнкүлөрдү тандоого болот:

- Интернет ресурсунун URL-адресин жазабыз;
- Локалдык тармактагы же Web-сервердеги компьютердин дискасындагы документтин файлын тандайбыз.

Алыстагы (удаленный) Web-сервердин файлдарына шилтеме жасаш үчүн төмөнкүлөрдү аткарабыз:

1). ***Сетевое окружение*** папкасын ачабыз. Ал үчүн Жумушчу столдон ***Мой компьютер*** опциясын тандайбыз. ***Другие места*** деген группадан ***Сетевое окружение*** деген значокту тандайбыз.

2). ***Сетевые задачи*** деген группадан ***Добавить новый элемент*** в ***сетевое окружение*** опциясын тандайбыз.

3). ***Добавление в сетевое окружение*** деген мастердин көрсөтмөсүн удаалаш аткарып олтуруп Web-түйүндүн ярлыгын түзөбүз.

2.3. Web–беттерди Excel таблицалык процессорунда даярдоо.

Excel таблицалык процессорунда web-бетти даярдоо боюнча опция жок болгону менен электрондук таблицаны жана китептерди HTML форматында сактоого болот.

Электрондук таблицаны Web-бет катары сактоо үчүн төмөнкү команданы аткарабыз:

Файл / Сохранить как Web-страницу

Сохранение документа деген терезе ачылат. Андан ары төмөнкү аракеттерди жасайбыз:

1). ***Заголовок*** бөлүмүндө ***Изменить*** баскычын чыкылдатып Web-беттин бөркүн киргизебиз.

2). Маалымдагычтын терезесинде электрондук таблица Excel-таблица катары көрүнүшү үчүн ***Добавить интерактивность пунктуна*** желекче (флажок) коебуз. Эгер желекче жок болсо, анда берилгендер кадимки эле таблицага айланат.

3). ***Сохранить*** бөлүмүндө ажыраткычтарды тандайбыз:

- ***Всю книгу*** – китептин бардык барактары сакталат;
- ***Выделенное*** – учурдагы барак же бөлүнгөн ячейкалардын блогу сакталат.

4). ***Опубликовать*** баскычын чыкылдатып, пайда болгон ***Публикация Web-страницы*** деген диалогдук терезеде документтин Web-сервердеги жарыяланышынын кошумча опцияларын тандайбыз.

- ***Публикуемые элементы*** деген бөлүмүндө сактала турган элементтерди сактайбыз: бардык китепти, баракты, ячейкалардын диапазонун, ж.б.
- ***Параметры просмотра*** деген бөлүмдө ***Добавить*** опциясына желекче коёбуз жана маалымдагычтын терезесинде таблицалар электрондук таблица катарында көрүнүшү үчүн интерактивдүүлүк режимин тандап алабыз;
- ***Имя файла*** талаасына тармактагы файлдын маршрутун же Web-сервердеги URL-даректи жазабыз;
- ***Автопереиздание при каждом сохранении книги*** деген опцияга желекче коёбуз. Бул учурда Web-беттин жаңы версиясы дайыма автоматтык түрдө жарыяланып турат;
- ***Открыть страницу в обозревателе*** деген опцияга желекче коебуз.

5). Жумушту аяктоо үчүн ***Публикация Web-страницы*** терезесиндеги ***Опубликовать*** баскычын басабыз же ***Сохранение документа*** терезесиндеги ***Сохранить*** баскычын басабыз.

2.4. Web–беттерди MS Access колдонмосунда даярдоо.

MS Access колдонмосунда Web–беттерди даярдоо үчүн Access колдонмосун жүктөйбүз жана төмөнкүлөрдү аткарабыз:

1. ***Файл / Создать*** командасын аткарабыз;

1. ***Создание файла*** панелинен ***Новая база данных*** опциясын тандайбыз;

2. ***Файл / Внешние данные / Импорт*** командасын чакырабыз. Мастердин көрсөтмөсү боюнча Excel-файлды Access-таблицага айлантабыз.

3. ***Файл / Сохранить как*** командасын чакырабыз. ***Сохранение в поле Как*** деген талаада ***Страница доступа*** к данным опциясын тандап ***ОК*** баскычын чекитебиз.

4. ***Новая страница доступа к данным*** терезесинде файлдын атын беребиз жана ***ОК*** баскычын басабыз.

Даярдалган бетти маалымдагычтын жардамы менен көрүп алууга болот.

2.5. Web–беттерди PowerPoint колдонмосунда даярдоо.

PowerPoint колдонмосунда презентацияларды Web те көрүү үчүн сактоого болот. Ал үчүн төмөнкү аракеттерди аткарабыз:

1. Керектүү презентацияны ачабыз.
2. ***Файл*** менюсунда ***Сохранить как Web-страницу*** командасын тандайбыз. ***Сохранение документа*** деген терезе ачылат.
3. ***Имя файла*** талаасына Web-беттин атын киргизебиз.
4. ***Тип файла*** талаасына төмөнкү опциялардын бирин тандайбыз.

- Ачылган тизмедеги ***Web-страница*** деген пунктту тандайбыз.
- Бардык кошумча берилгендер, сүрөттөр жана презентациянын башка элементтери бир файлда жайгашышы үчүн **Web-страница в одном файле** деген параметрди тандайбыз.

5. Web-беттин бөркүнүн текстин жазабыз. Ал үчүн ***Изменить*** баскычын тандайбыз жана ***Заголовок*** талаасына текстти жазып ***OK*** баскычын басабыз.
6. Презентацияны HTML форматында сактоо үчүн ***Сохранение документа*** терезесиндеги ***OK*** баскычын басабыз.

2.6. BOOM.RU Web–беттердин мастери.

BOOM.RU мастеринде Web–беттерди түзүү үчүн серверде каттодон өтүү керек. Ал үчүн баштапкы Web–бетке төмөнкү дарек боюнча өтүү керек: http://www.boom.ru. Бул бетте каттоодон өтүү жана жеке түйүндүн атын киргизүү сунушталат. Бул түйүндүн бардык Web–беттери төмөнкүдөй URL-дарекке ээ болот: http://түйүндүн_аты.boom.ru/файлдын_аты . Бул жерде жеке Web–түйүндү түзүү үчүн кеңири маалыматты табууга болот.

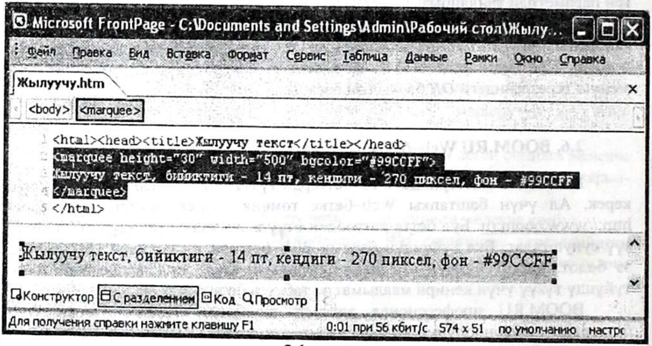
BOOM.RU профессионал адистер тарабынан түзүлгөн жеке Web–беттердин шаблондорунун коллекциясын сунуштайт. Шаблондор төмөнкү категориялар боюнча топтолгон.

- ***Главная страница*** – жогорку деңгээлдеги беттердин шаблондору, ал баштапкы бет деп аталат.
- ***Персональная страница*** – виртуалдык визиттик карточка.
- ***Друзья*** – достор жөнүндөгү электрондук маалыматтар.
- ***Фотоальбом*** – цифралык сүрөттөрдүн коллекциялары.
- ***Коммерческая страница*** – бизнес-сунуштар.
- ***Лучшие сайты*** – Интернеттин башка ресурстарына жасалган гипершилтемелердин жыйындысы.
- ***Мои увлечения*** – эмнеге кызыга тургандыгы жөнүндө маалымат.
- ***Резюме*** - өзүңөр жөнүндө кыскача маалымат.
- ***Познакомлюсь*** – досторду издегендер үчүн.
- ***Внимание*** – Интернеттеги жарыялар доскасы.

2.7. Microsoft FrontPage программасы.

Web–бетти түзүү үчүн Microsoft FrontPage программасын колдонууга болот. Ал үчүн эң оболу бул программаны, мисалы, *Microsoft FrontPage 2003* версиясын орнотуп алуу керек. Аны жүктөө үчүн ***Пуск \ Программы \ Microsoft Office / Microsoft Office FrontPage 2003*** командасын колдонобуз.

Microsoft FrontPage 2003 ресурсу Web-беттерди түзүүнүн жаңы макеттери жана каражаттары менен камсыздалган, анын шаблондору жана темалары менен профессионалдык Web-сайттарды түзүүгө болот. Мисалы, Microsoft FrontPage шаблондору менен колдонуучуну регистрациялоочу, коноктор китебинин иштөөсүн жана издөөнү камсыз кылуучу формалары бар, Web-беттердин түзүүчүлөрү менен кайра байланышка чыгуу мүмкүнчүлүгү бар формаларды түзүүгө болот.



2.1-сүрөт.

Microsoft FrontPage программасында колдонуп, Web-беттерди түзүү үчүн 3 режимдин бирин пайдаланабыз: ***Конструктор, С разделением, Код*** (2.1-сүрөт). Ал эми ***Просмотр*** режиминде түзүлгөн Web-бетти сынап көрүүгө болот.

Web-беттерди түзүүдө, тексттерди киргизүүдө жана калыпка салууда, сүрөттөрдү, таблицаларды жана башка элементтерди кошууда HTML тилинин программалык коддорунун фондук режимде автоматтык түрдө жазылышы жана программалык коддордун жолчолорунун номерлеринин көрсөтүлүп турушу программанын дагы бир артыкчылыгы болуп эсептелет. Ошондуктан Web-беттерди конструктор режиминде эле редактирлөө ыңгайлуу болот.

HTML-кодду көрүү үчүн же аны редактирлөө үчүн ***Код*** режимин же ***С разделением*** режимин тандоо керек. ***Код*** режиминде Web-беттин HTML коду гана көрсөтүлөт, ал эми ***С разделением*** режиминде ***Код*** режими дагы, ***Конструктор*** режими дагы бир убакта экранда чагылдырылат.

3. HTML тили жана анын тегдери.

HTML тили деп Web-беттерди (HTML-документтерди) даярдоо үчүн дайындалган программалоо тилин айтабыз.

HTML тили - тэгдердин (дескрипторлордун, ярлыктардын), б.а. башкаруучу символдордун (белгилердин) коллекциясы болуп эсептелет.

Тэгдердин жардамы менен документке элементтерди кошууга жана аларды калыпка салууга (форматтоого) болот.

HTML тили 1992-жылы Тим Бернерс-Ли тарабынан иштелип чыгып, Mosaic браузеринде колдонулган. Бүгүнкү күндө анын өркүндөтүлгөн HTML 4.0 версиясы кеңири колдонулууда.

3.1. HTML тилинин базалык элементтери

HTML тилинин эки түрдүү базалык элементтери бар:

- **Тег** (тег – белги, ярлык, маркир, дескриптор).
- **Символдук константалар.**

1). ***Тег*** деп белгилердин жыйындысын айтабыз. Бардык тегдер «кичине» (<) белгиси менен башталып, «чоң» (>) белгиси менен бүтөт. Бул эки символдорду бурчтук кашаалар деп да айтабыз. HTML тилинде ар бир тегдин атайын мааниси бар. Тегдердин аталышында тамганын регистри мааниге ээ эмес, б.а. тегдерди жазууда кичине же чоң тамгалардын каалаганын пайдаланса болот. Бирок тегдерди документтин кадимки текстинен айырмалоо үчүн тегдерди чоң тамгалар менен жазуу сунушталат.

Тегдер ***түгөйлүү*** жана ***түгөйсүз*** болушу мүмкүн. Түгөйлүү тегдин ачуучу теги жана жабуучу теги болот. Жабуучу тегдин алдынан кыйгач бөлчөк белгиси (/) коюлат. Ал эми түгөйсүз тегдин жабуучу теги болбойт.

Ар бир тегдин өзүнүн аты - идентификатору (ID) бар. Тегдер белгилүү бир касиеттерге - атрибуттарга ээ болушу мүмкүн.

Жалпы учурда түгөйлүү тег төмөнкүдөй жазылат:

<Тег 1-атрибут='1-маани' 2-атрибут='2-маани' ...> Текст </Тег>

Мында: Тег – тегдин идентификатору, атрибут – атрибуттун аты, маани - атрибуттун мааниси, Текст – тегдин таасири аракет этүүчү текст. <Тег> - ачуучу тег, </Тег> - жабуучу тег.

Атрибуттун маанилери тексттик жолчо түрүндө кавычкага алынып жазылат. Ал үчүн жалгыз кавычканы (')же кош кавычканы (") колдонууга болот. Эң негизгиси ачуучу жана жабуучу кавычка бирдей болушу керек. Блокнот тексттик редакторунда коюлган кавычкаларды маалымдагыч эч кандай тосколдугу жок окуй алат, ал эми Word тексттик редакторунда коюлган кавычкаларды маалымдагыч туура эмес окуп коюшу мүмкүн, анткени Word туура кавычканы (") түгөйлүү кавычкалардын бирине (« ») же (“ ”) автоматтык түрдө которуп салышы мүмкүн.

...ТЕКА
Ошского государственного
университета
ИНВ № 932048

Түгөйлүү тегдин мисалдары:
<HTML>...</HTML>
<HEAD>...</HEAD>
<TITLE>...</TITLE>

Түгөйлүү эмес тегдин мисалдары:

<META>
<HR><INPUT>
<FRAME>

Атрибуту менен жазылган түгөйлүү тегдин мисалдары:
<P align='left' > Абзац солго түздөлгөн </P>
<P align='center' > Абзац ортого түздөлгөн </P>
<P align='right' > Абзац оңго түздөлгөн </P>

2). ***Символдук константалар*** амперсанд (&) белгиси менен башталып, андан кийин константанын аты жазылат же фунт белгиси менен башталган символдун ондук коду жазылат. Символдук константанын акыры үтүрлүү чекит (;) менен бүтөт (1-таблица).

1-таблица

Символдук константа	Код	Аталышы
Пробел		Пробел (боштук)
!	!	Илеп белгиси
"	" "	Кош кавычка
%	%	Процент
&	& &	Амперсанд (Ampersand)
<	< <	"Кичине" белгиси
>	> >	"Чоң" белгиси
§	§	Параграф
©	© ©	Copyright белгиси – автордук укук белгиси
«	«	Сол жакқы бурчтук кош кашаа
®	® ®	Регистрацияланган соода маркасы
±	±	Плюс-минус
»	»	Оң жакқы бурчтук кош кашаа
⁄	⁄	Кыйгач бөлчөк сызыгы
≠	≠	Барабар эмес
≡	≡	Тендеш барабар
“	“	Сол кош кавычка
”	”	Оң кош кавычка

3.2. HTML-документтердин структурасы.

HTML-документтер эки бөлүмдөн турат:

1). ***Документтин бөркү (заголовка)*** – Web-беттин параметрлерин орнотуучу бөлүгү.

2). ***Документтин негизги бөлүгү (телосу)*** – маалымдагычтын терезе-

синде көрсөтүлө турган тексттерди жана беттин элементтерин кармап турат.

```
<HTML>
        <HEAD>
              Беттин бөркү – беттин параметрлери
        </HEAD>
        <BODY>
              Беттин негизги бөлүгү (телосу) – беттин тексти
        </BODY>
</HTML>
```

HTML-документтер <HTML> теги менен башталып </HTML> теги менен бүтүшү керек. Мындай түгөйлүү тегдердин болушу браузерге бул документтин HTML-документ экендигин аныктап берет.

Документтин бөркү (заголовка) ачуучу <HEAD> жана жабуучу </HEAD> тегдеринин арасында болот жана документ жөнүндөгү маалыматты баяндайт. Мындан сырткары бул бөлүмдө документтин официалдуу бөркү <TITLE> жана </TITLE> тегдеринин арасына жазылат.

Документтин негизги бөлүгү (телосу) <BODY> теги менен башталып </BODY> теги менен аяктайт.

3.1-мисал.

```
<HTML>
    <HEAD>
        <TITLE>Резюме</TITLE>
    </HEAD>
    <BODY>
        Автобиография
    </BODY>
</HTML>
```

түрдөгү HTML-код бөркү "Резюме" деген, ал эми негизги бөлүгүндө "Автобиография" деген сөз жайгашкан Web-бетти аныктайт.

3.3. HTML-документтеги түшүндүрмөлөр.

HTML-документтеги түшүндүрмөлөр (комментариялар) документтин каалаган жеринде болушу мүмкүн жана алар браузерлер тарабынан каралбайт.

Түшүндүрмөлөр <!- жолчосунан баштап, каалагандай символдордун жыйындысынан турат жана ал -> символдору менен аякталат:

<!- *Түшүндүрмөнүн тексти* ->

Ар бир түшүндүрмөнү өзүнчө жолчого жазууга жана анын ичинде кызматчы символдорду (мисалы >, <, &, ж.б.) колдонбоого аракет кылуу керек.

3.4. Бөрктөр деңгээли.

HTML-документтин ичиндеги бөрктөрдү даярдоо үчүн 6 деңгээлдүү тегдер, б.а. **<Hn> Текст </Hn> (n** саны 1 ден 6 га чейинки маанилерди кабыл алат)

түрүндөгү тегдер колдонулат. Текстти мына ушул тегдердин арасына жайгаштырып, тиешелүү өлчөмдөгү бөрктү алабыз.

3.2-мисал.

```
<HTML>
    <HEAD>
        <TITLE> Заголовкалар </TITLE>
    </HEAD>
    <BODY>
        <H1> Бөрк 1 </H1>
        <H2> Бөрк 2 </H2>
        <H3> Бөрк 3 </H3>
        <H4> Бөрк 4 </H4>
        <H5> Бөрк 5 </H5>
        <H6> Бөрк 6 </H6>
    </BODY>
</HTML>
```

3.5. Параграф.

Жаңы жолчого өтүү (enter), табуляция (Tab), пробел (бoштук) жана башка калыпка салуу (форматтоо) белгилери HTML-браузерде эске алынбайт. Ошондуктан жаңы абзацты (параграфты) түзүү үчүн **<P>** тегин, ал эми абзацты түзбөй туруп жаңы жолчонун башталышына өткүбүз келсе **
** тегин колдонобуз.

3.3-мисал.

```
<HTML>
    <HEAD>
        <TITLE> Параграф </TITLE>
    </HEAD>
    <BODY>
        <P align='left'> Абзац солго түздөлгөн </P>
        <P align='center'> Абзац ортого түздөлгөн </P>
        <BR> <!- Жаңы жолчонун башталышына өтүү ->
        <P align='right'> Абзац оңго түздөлгөн </P>
    </BODY>
</HTML>
```

Эгерде жогорудагы символдорду HTML-браузери эске алсын десек, анда <PRE> жана </PRE> тегдерин колдонушубуз керек.

3.6. Номерленген жана маркирленген тизмелер.

Элементтери тартиптештирилген объектти мүнөздөө үчүн номерленген (тартиптештирилген) же маркирленген (тартиптештирилбеген) тизмелер (список) колдонулат. Мисалы, китептин мазмунун жазууда, алгоритмди аткаруунун

тартибин баяндоодо же объекттин касиетин мүнөздөп көрсөтүүдө номерленген же маркирленген тизмелерди колдонууга туура келет.

Номерленген тизмелерди түзүү үчүн <OL>…</OL> жана <LI>…</LI> тегдери колдонулат. <OL>…</OL> тегдери номерленген тизменин башталышын жана акырын билдирсе, <LI>…</LI> тегдери тизменин пункттарын билдирет.

3.4-мисал.

```
<HTML>
    <HEAD>
        <TITLE> Номерленген тизме </TITLE>
    </HEAD>
    <BODY>
        <H3> Компьютердик тармактардын түрлөрү </H3>
        <OL>
        <LI> Локалдык тармак</LI>
        <LI> Корпоративдик тармак</LI>
        <LI> Регионалдык тармак</LI>
        <LI> Глобалдык тармак</LI>
        </OL>
    </BODY>
</HTML>
```

Маркирленген (тартиптештирилбеген) тизмелер үчүн <UL>…</UL> тегдери колдонулат, ал эми тизменин ички пункттары үчүн жогорудагыдай эле <LI>…</LI> тегдерин колдонобуз.

3.7. Тексттин элементтерин калыптоо.

Тексттин элементтерин калыптоо (форматтоо) тегдерине төмөнкүлөр кирет:

<B> … </B>	- жоон шрифт;
<I> … </I>	- кыйгач шрифт;
<U> … </U>	- асты сызылган шрифт;
[…]	- жогорку индекс;
_{../}	- төмөнкү индекс;
<FONT атрибуттар> … </FONT>	- шрифттин атрибуттарын орнотуу.

3.5-мисал.

<FONT FACE="Times New Roman" SIZE="14" COLOR="#FF0000"> Асан </FONT>

Бул HTML-коддо Асан деген сөз Times New Roman шрифти менен жазылгандыгы, өлчөмү – 14 пт, түсү – кызыл экендиги баяндалат. Ал эми FACE - шрифттин тибин, SIZE – шрифттин өлчөмүн жана COLOR – шрифттин түсүн билдирет жана FONT тегинин атрибуттары болуп саналат.

4. Таблицаларды түзүү. Web-беттин интерактивдүүлүгү.

4.1. Таблицаларды түзүү.

Таблицаларды түзүү үчүн төрт түгөйлүү тегдер колдонулат:

- <TABLE> … </TABLE> Таблицанын башталышы жана аягы.
- <CAPTION> … </CAPTION > Таблицанын бөркү.
- <TR> … </TR> Таблицага жаңы жолчо кошуу.
- <TH> … </TH> Жолчого ячейкалардын бөркүн кошот.
- <TD> … </TD> Жолчого кадимки ячейканы кошот.

4.1-мисал. Таблицаларды түзүү.

```
<TABLE BORDER='1'> …
<CAPTION> Таблица </CAPTION >
<TR> <TH> Фамилиясы </TH> <TH> Аты </TH> <TH> Туулган жылы </TH>
</TR>
<TR> <TD> Асанов </TD> <TD> Акмат </TD> <TD> 1973 </TD></TR>
<TR> <TD> Туратова </TD> <TD> Гулиза </TD> <TD> 1974 </TD></TR>
<TR> <TD> Салиев </TD> <TD> Канат </TD> <TD> 1972</TD></TR>
</TABLE>
```

Бул мисалдагы код төмөнкү таблицаны аныктап берет:

Таблица

Фамилиясы	Аты	Туулган жылы
Асанов	Акмат	1973
Туратова	Гулиза	1974
Салиев	Канат	1972

4.2. Рамкаларды колдонуу жана аларды берилгендердин булагы менен байланыштыруу.

Рамкалардын жардамы менен маалымдагычтын терезесин бир нече секцияларга бөлүүгө болот жана алардын ар биринде башка секциядагы документтин мазмунуна көз каранды болбогон өзүнүн документин жайгаштырууга болот.

<FRAMESET> … </FRAMESET> тегдеринин жардамы менен маалымдагычтын терезесин горизонталдык же вертикалдык багыт боюнча эки рамкага (бөлүккө) бөлсөк болот.

Рамкалардын ориентациясы жана алардын өлчөмү ROWS жана COLS атрибуттары (касиеттери) аркылуу берилет. Бир <FRAMESET> … </FRAMESET> тегинин ичине ROWS же COLS атрибуттарынын бирөөсүн гана жайгаштырууга болот. ROWS атрибуту менен горизонталдык багыт боюнча, ал эми COLS атрибуту менен вертикалдык багыт боюнча эки терезеге бөлүнөт.

Рамкалардын бийиктигі же туурасы процент же пиксел менен берилет. Көпчүлүк учурда биринчи рамканын өлчөмү берилип, экинчисинин өлчөмү үчүн * гана берилет. * символунун берилиши экинчи рамка маалымдагычтын терезесинин калган бөлүгүнүн бардыгын ээлейт дегенди билдирет. Бөлүнгөн рамкалардын бирин дагы экиге бөлүш үчүн камтылган рамкалар эрежесинен пайдаланабыз.

Рамкалардын аттары FRAME тегинин NAME деген атрибуту менен, ал эми бул бетке жайгаштырыла турган берилгендердин файлы SRC атрибуту менен берилет.

4.2-мисал. Рамкалардын жардамы менен жогорку колонтитулга жана астынкы бөлүгү эки вертикалдык рамкага бөлүнгөн Web-бет түзгүлө.

Чыгаруу. Колонтитул.htm, Мазмуну.htm, Негизги.htm жана index.htm аттуу Web-беттерин Блокнот тексттик редакторунда даярдап, жумушчу столго түзүлгөн Рамка аттуу папкага жайгаштырабыз.

1). Колонтитул.htm

```
<HTML>
<BODY>
<H2> Асанов Акматтын сайты </H2>
<BODY>
</HTML>
```

2). Мазмуну.htm

```
<HTML>
<BODY><UL>
<LI>Башкы бет
<LI>Автобиография
<LI>Резюме
<LI>Окуучулук жылдар
<LI>Студенттик жылдар
<LI>Менин университетим
<LI>Менин үй-бүлөм
<LI>Менин досторум
<LI>Менин хоббим
<LI>Менин сүрөттөрүм
<LI>Менин максаттарым
<LI>Коноктор китеби
<UL><BODY>
</HTML>
```

3). Негизги.htm

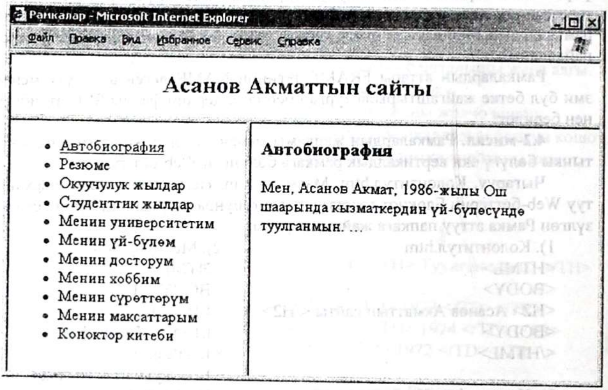
```
<HTML>
<BODY>
<H2> Негизиг бет </H2>
<BODY>
</HTML>
```

4). index.htm

```
<HTML>
   <HEAD>
        <TITLE> Рамкалар </TITLE>
   </HEAD>
   <FRAMESET ROWS='80,*'>
          <FRAME NAME = '1-рамка' SRC='Колонтитул.htm'>
          <FRAMESET COLS ='25%,*'>
          <FRAME NAME = '2-рамка' SRC='Мазмуну.htm'>
          <FRAME NAME = '3-рамка' SRC='Негизги.htm'>
          </FRAMESET>
   </FRAMESET>
</HTML>
```

Microsoft Internet Explorer браузеринде index.htm файлын жүктөгөндө 4.1-сүрөттө көрсөтүлгөн Web-бет пайда болот.

Эскертүү. Рамкаларды бөлүүнү <BODY> тегинин ичине жайгаштырбоо керек.



4.1-сүрөт.

SRC атрибутунда берилгендер файлынын жайгашкан жолун көрсөтүү менен Web-беттин тиешелүү бөлүмдөрүнө керектүү маалыматтарды жайгаштыра алабыз. Эгерде файлдар башкы беттин файлы жаткан папкада болсо, анда SRC атрибутунда файлдын атын жазуу жетиштүү болот. Эгерде ал файл башка папкада болсо, анда ал файлдын жолунун маршрутун көрсөтүү керек. Мисалы, атрибут SRC ='C: / Рамка / MyPage' түрүндө жазылат.

4.3. Web-беттин интерактивдүүлүгү жана гипершилтеме.

Web-беттин эң негизги артыкчылыгы анын интерактивдүүлүгүндө, б.а. Web-беттин колдонуучу менен пикир алмашуу мүмкүнчүлүгүнө ээ болушунда турат: керектүү маалыматты гипершилтемелер менен башка булактардан алууга болот тургандыгы жана электрондук почта же коноктордун китеби менен кайра байланышка чыгуу мүмкүнчүлүгүнө ээ болушунда турат. Web-беттин интерактивдүүлүгү аны колдонуунун эффективдүүлүгүн жогорулатат.

Web-беттин интерактивдүүлүгүнүн негизги элементтеринин бири гипершилтеме болуп эсептелет.

Гипершилтеме <A>…</A> тегдери менен түзүлөт. HREF атрибутунун жардамы менен шилтеменин кайсыл бетке жасалганын аныктайбыз.

Гипершилтеме эки түрдүү болот:

- Беттин өзүнүн ички пункттарына болгон шилтеме;

- Башка документтерге (тышкы булактарга) болгон шилтеме.

1). Беттин өзүнүн ички пункттарына болгон шилтемени жасоо үчүн төмөнкүлөрдү аткаруу керек:

- Эң оболу шилтеме жасала турган абзацка белги (метка, якорь) орнотуу керек: <A NAME="1-белги">Текст</A>.
- Экинчиден бул пункттан мурда учурай турган Текст деген сөзгө төмөнкүдөй гипершилтеме орнотуу керек: <A HREF="#1-белги">Текст</A>. Белгинин аты фунт белгиси (#) менен башталышы керек.

2). Башка документтерге (тышкы булактарга) болгон шилтеме жасоо үчүн төмөнкү калыпты (форматты) колдонобуз:

<A HREF="Маршрут / Файлдын_аты" TARGET="Метод">Текст</A>.

Мында **Маршрут** – Текст аттуу Web-бетти (документти) чакыруу маршрутун (б.а. файл сакталган жерди) билдирет, ал эми **Метод** болсо Текст аттуу Web-бетти (документти) ачуу ыкмасын аныктап берет.

TARGET атрибуту төмөнкүдөй маанилерди кабыл алышы мүмкүн:

- "_blank" - документ маалымдагычтын жаңы терезесинде ачылат;
- "_self" - документ учурдагы рамкада ачылат;
- "_parent" - документ мурдагы деңгээлдеги рамкада ачылат;
- "_top" - документ жогорку деңгээлдеги рамкада ачылат;
- "рамканын_аты" - аты NAME атрибутунда көрсөтүлгөн рамкада ачылат;
- эгерде атрибут көрсөтүлбөсө, анда документ маалымдагычтын учурдагы терезесинде ачылат.

Эгерде гипершилтеме электрондук дарекке жасалса, анда HREF атрибутунун мааниси төмөнкүдөй жазылат:

<A HREF="mailto:a_sopuev" > Электрондук дарек </A>

4.3-мисал. 4.2-мисалдагы Web-беттин мазмунунда жайгашкан "Автобиография" сөзүн тандаганда автобиографиянын мазмуну негизги бетке чыгарыла тургандай кылып гипершилтеме жасагыла.

Чыгаруу. Ал үчүн эң оболу Авто.htm аттуу Web-бетти даярдайбыз.

1). Авто.htm

```
<HTML>
<BODY>
<H3> Автобиография </H3>
<P>Мен, Асанов Акмат, 1986-жылы Ош шаарында кызматкердин үй-бүлөсүндө
туулганмын. …</P>
</BODY>
</HTML>
```

Андан кийин 4.2-мисалда каралган Мазмуну.htm Web-бетин төмөнкүдөй өзгөртүп чыгабыз:

2). Мазмун.htm

```
<HTML>
```

```
<BODY>
<P><A HREF='Авто.htm' TARGET='3-рамка'> Автобиография </P>
<P> Резюме </P>
<P> Окуучулук жылдар </P>
...
</BODY>
</HTML>
```

Бул мисалдагы

```
<P><A HREF='Авто.htm' TARGET='3-рамка'> Автобиография </P>
```

коду Автобиография сөзүнө гипершилтеме орнотот: бул сөз башка түскө боёлуп, асты сызылып калат. Эгерде ушул сөздү тандасак, анда HREF='Авто.htm' атрибуту боюнча Авто.htm веб-бетинин мазмуну TARGET='Башкы' атрибутунун касиети боюнча Web-беттин Башкы деп аталган негизги бөлүгүндө пайда болот (4.1-сүрөт).

Web-беттин мазмунундагы калган пункттарга деле ушул эле сыяктуу гипершилтемелерди орнотууга болот.

Мисалы, Резюме деген пунктка гипершилтеме жасоо керек болсо, анда Резюме.htm аттуу Web-бет даярдап, төмөнкүдөй кодду колдонобуз:

```
<P><A HREF='Резюме.htm' TARGET='3-рамка'> Резюме </P>
```

5. Форма. Формага башкаруучу элементтерди жайгаштыруу.

5.1. Форма.

Web-беттерди түзүүдө формаларды колдонуу ыңгайлуу болот. **Форма** деп башкаруучу элементтерди (командалык баскычтарды, желекчелерди, ажыраткычтарды (переключателдерди), тексттик талааларды, тизмелерди, ж.б.) кармап туруучу объектти айтабыз.

Web-бетке форманы орнотуу үчүн <FORM>…</FORM> тегдери колдонулат. Ал эми формага жогоруда айтылган башкаруучу элементтерди жайгаштыруу үчүн <INPUT> түгөйсүз теги колдонулат.

5.2. Киргизүү талаасы.

1). TYPE="text".

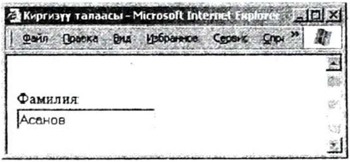
Киргизүү талаасы клавиатурадан тексттин бир жолчосун киргизүүгө мүмкүнчүлүк түзүп берет. Талаага киргизилген жолчо бул элементтин value деген атрибутунун маанисине автоматтык түрдө ыйгарылат.

Киргизүү талаасынын калыбы (форматы) төмөнкүдөй:

```
<INPUT type='text' value='Асанов'>
```

5.1-мисал. Киргизүү талаасын формага жайгаштыруу (5.1-сүрөт).

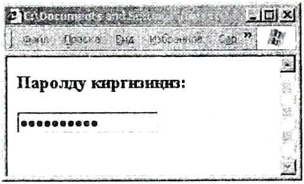
```
<HTML>
<HEAD>
<TITLE>Киргизүү талаасы
</TITLE></HEAD>
<BODY>
<FORM>
<BR>Фамилия:
<BR><INPUT TYPE="text"
NAME="1-талаа" SIZE="20" VALUE="Асанов">
</FORM>
</BODY>
</HTML>
```



5.1-сүрөт.

2). TYPE="password".

Киргизүү талаасынын бул тиби паролду киргизүү терезесин аныктайт. Иштөө принциби text тибине окшош болот, бирок киргизилген тексттин символдорунун ордуна экранда жылдызчалар (*) же (●) символдору көрсөтүлөт. Анын форматы төмөнкүдөй:



5.2-сүрөт.

```
<INPUT TYPE="password" NAME="PW" SIZE="20" MAXLENGTH="10">
```

Бул программалык коддо узундугу 20 символго чейин болгон паролду киргизүү үчүн терезе ачылат. Ал эми паролдун максималдуу узундугу үчүн 10 символ ажыратылат (5.2-сүрөт).

5.3. Переключатель (Радиобаскыч).

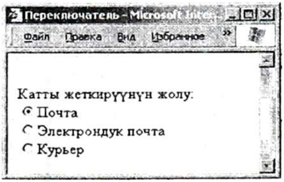
Эгерде бир нече маанилердин ичинен бирөөсүн гана тандоо талап кылынса, анда Переключатель деген элементти тандап алуу керек. Анын калыбы төмөнкүдөй: <INPUT TYPE="radio" NAME="Аты" VALUE="Мааниси">Почта.

5.2-мисал. Катты жеткирүүнүн жолун тандоого мүмкүнчүлүк берген форманы даярдагыла (5.3-сүрөт).

Чыгаруу. Төмөнкү программалык кодду түзөбүз:

Переключатель.htm

```
<HTML>
<HEAD>
<TITLE> Переключатель</TITLE>
</HEAD>
<FORM>
<BR>Катты жеткирүүнүн жолу:
```



5.3-сүрөт.

```
<BR><INPUT TYPE="radio" NAME="Почта" VALUE="1-ыкма">Почта
<BR><INPUT TYPE="radio" NAME="Электрондук почта" VALUE="2-ыкма">Электрондук почта
<BR><INPUT TYPE="radio" NAME="Курьер" VALUE="3-ыкма">Курьер
</FORM>
</HTML>
```

5.4. Желекче (Check Box, Флажок).

Эгерде маанилердин ичинен бир нечесин тандоо талап кылынса, анда желекче элементин тандап алуу керек. Анын форматы төмөнкүдөй:

<INPUT TYPE="checkbox" NAME="Аты">Желекченин аты.

5.3-мисал. Спорттун кайсы түрлөрү сизге жагат?

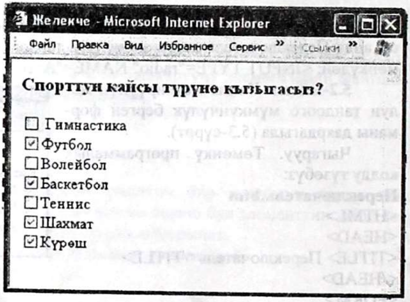
Чыгаруу. Төмөнкү программалык кодду түзөбүз:

Желекче.htm

```
<HTML>
<HEAD><TITLE> Желекче </TITLE></HEAD>
<FORM>
<BR><INPUT TYPE='checkbox' NAME='Гим'> Гимнастика
<BR><INPUT TYPE='checkbox' NAME='Фут'>Футбол
<BR><INPUT TYPE='checkbox' NAME='Вол'>Волейбол
<BR><INPUT TYPE='checkbox' NAME='Бас'>Баскетбол
<BR><INPUT TYPE='checkbox' NAME='Тен'>Теннис
<BR><INPUT TYPE='checkbox' NAME='Шах'>Шахмат
<BR><INPUT TYPE='checkbox' NAME='Күр'>Күрөш
</FORM>
</HTML>
```

Check Box элементи квадратты аныктайт. Анын ичине желекче белгисин коюу менен керектүү пунктту тандап алгандыгыбызды билдиребиз.

5.4-сүрөттө Желекче.htm Web-бети көрсөтүлгөн. Анда Переключатель элементинен айырмаланып спорттун бир нече түрлөрүн тандап алууга мүмкүн экендиги көрүнүп турат.



5.4-сүрөт.

5.5. Баскыч (Кнопка).

1). TYPE="submit".

Баскыч элементи команданы чакыруу (иштетүү) үчүн колдонулат.

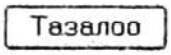
Баскычты басканда формада даярдалган маалымат анык бир тандалган дарекке жөнөтүлөт. Анын калыбы (форматы) төмөнкүдөй:

<INPUT TYPE=submit VALUE="Жөнөтүү">

2). TYPE="reset".

Баскычтын экинчи түрү болуп, басканда форманын талааларында даярдалган маалыматтар тазаланат. Анын калыбы (форматы) төмөнкүдөй:

<INPUT TYPE="reset" VALUE="Тазалоо">



5.6. Көлөмдүү тексттерди киргизүү талаасы.

Формалар көлөмдүү тексттерди киргизүүгө боло турган талааны да кармап туруўшу мүмкүн. Ал элемент <TEXTAREA> деп аталат. Анын калыбы (форматы) төмөнкүдөй:

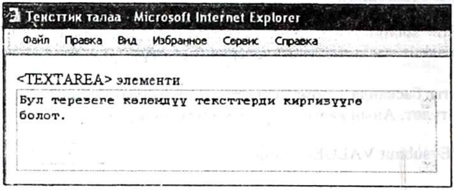
<TEXTAREA NAME="address" ROWS="5" COLS="50">Бул терезеге көлөмдүү тексттерди киргизүүгө болот.</TEXTAREA>

NAME атрибуту талаанын атын, ROWS атрибуту терезедеги жолчолордун санын, COLS атрибуту терезенин жолчосуна жайгаштырыла турган символдордун санын билдирет. <TEXTAREA>, </TEXTAREA> тегдеринин арасына жайгашкан текст түшүндүрмө текст катары пайдаланылат. Колдонуучу аны өчүрүп жиберсе да болот. Терезеге саны 5 жолчодон чоң болгон текстти деле киргизүүгө болот. Бирок терезенин өлчөмү өзгөрбөгөндүктөн, көрүнбөй калган тексттерди "жылдыруу" (прокрутка) баскычынын жардамы менен көрүүгө болот.

5.4-мисал. Көлөмдүү текст талаасын түзгүлө.

```
Текст.htm
<html>
<head> <title>Тексттик талаа</title> </head>
<body>
&ltTEXTAREA&gt элементи.
<BR> <TEXTAREA NAME="address" ROWS="5" COLS="50">Бул терезеге кө-
лөмдүү тексттерди киргизүүгө болот.</TEXTAREA>
</body> </html>
```

Текст.htm Web-бети жүктөлгөндө маалымдагычтын терезесинде төмөнкү терезе пайда болот (5.5-сүрөт).



5.5-сүрөт.

5.7. Сүрөттөлүштөрдү Web-бетке жайгаштыруу.

Сүрөттөлүштөрдү Web-бетке <IMG> теги менен жайгаштырууга болот. Анын калыбы (форматы) төмөнкүдөй:

<IMG SRC="Сүрөт" **BORDER="0" ALIGN=**"Түздөө" **WIDTH=**"Туурасы" **HEIGHT=**"Бийиктиги" **HSPACE=**"ГоризонталдыкЧеттөө" **VSPACE=**"ВертикалдыкЧеттөө" **NAME="**Аты**">.**

Атрибуттардын маанилери төмөнкү таблицада келтирилди.

5.1-таблица.

SRC	Графиктик файлдын кеңейтилиши менен алынган аты (*.jpg, *.gif, *.png). Эгерде файл башка каталогдо болсо, анда анын маршрутун көрсөтүү керек.
ALIGN	Документте сүрөттөлүштү түздөө: left – солго түздөө. right – оңго түздөө.
BORDER	Сүрөттүн сыртындагы рамка. Алгачкы маани - 0 (рамка жок).
WIDTH	Сүрөттөлүштүн туурасы (пикселдер менен).
HEIGHT	Сүрөттөлүштүн бийиктиги (пикселдер менен).
NAME	Сүрөттөлүштүн атын аныктайт. Сүрөттөлүш башка обьекттер менен байланышпаган учурда ат берүүнүн зарылчылыгы жок.
HSPACE	Графиктик сүрөттөлүштүн горизонталдык четтөөсү (пикселдер менен).
VSPACE	Графиктик сүрөттөлүштүн вертикалдык четтөөсү (пикселдер менен).

5.5-мисал. C:/ дискиндеги Рамка деген папкада жайгашкан Лилии.jpg сүрөтүн веб-бетке жайгаштыргыла (5.6-сүрөт).

Чыгаруу. Программаны түзөбүз.

```
Сүрөт.htm
<html>
<head> <title>Фото</title> </head>
<body bgcolor='yellow'>
&lt IMG &gt элементи.
<BR> <IMG>
SRC="C:/Рамка/Лилии.jpg"
BORDER="2" TOP='100'
WIDTH='150' HSPACE="10"
VSPACE="20">
</body>
</html>
```

5.6-сүрөт.

Программанын <body bgcolor='yellow'> жолчосунда маалымдагычтын терезесинин фону сары түстө боло тургандыгы белгиленген.

Төмөнкү таблицада түстөрдүн аттары жана RGB-маанилери көрсөтүлдү.

5.2-таблица.

Түстөрдүн аттары жана RGB-маанилери	
Black = "#000000"	Green = "#008000"
Silver = "#C0C0C0"	Lime = #00FF00"
Grey = "#808080"	Olive = "#808000"
White = "#FFFFFF"	Yellow = "#FFFF00"
Maroon = "#800000"	Navy = #000080"
Red = "#FF0000"	Blue = "#0000FF"
Purple = "#800080"	Teal = "#008080"
Fuchsia = "#FF00FF"	Aqua = "#00FFFF"

6. Web-беттер үчүн программаларды түзүү. JavaScript тили.

6.1. JavaScript программалоо тили жөнүндө түшүнүк.

Web-бетке жайгаштырылган ар түрдүү башкаруучу элементтер кандайдыр бир аракеттерди аткарышы үчүн атайын программаларды, б.а. сценарийлерди жазуу керек.

Сценарийлерди жазуу үчүн кенири таралган программалоо тилдеринин бири JavaScript тили болуп эсептелет. Сценарийлерди жазуу үчүн VBScript тили да кол-

донулат. Script деген сөз сценарий дегенди билдирет. JavaScript тилинин синтакси-си Java тилине сыртынан гана окшошуп кетет.

JavaScript тили Netscape Corporation кампаниясы тарабынан иштелип чыккан. Ал Web-беттердин HTML-коддоруна байланыштырылып коюлган. Ошондуктан ал маалымдагычтын курамында гана аткарылат.

JavaScript программалоо тили менен Web-беттер үчүн сценарийлерди жазып, аларды интерактивдүү HTML-документтерге айландырабыз.

JavaScript программалоо тилинин сценарийлери менен төмөнкү маселелерди чечүү мүмкүн:

- текстгер жана графиктер үчүн менен анимациялык эффекттерди түзүүгө болот;
- беттеги текстти үн менен коштоону камсыз кылууга болот;
- беттин түрүн жана маңызын динамикалык түрдө өзгөртүүгө болот;
- колдонуучу аркылуу форманын талааларына киргизилген маалыматтарды текшерүүгө жана жөнөтүүгө болот.

JavaScript программалоо тилинин сценарийлерин жазуу үчүн Блокнот же WordPad тексттик редакторлору сунушталат. Ал эми Word тексттик редакторлору документтин текстин автоматтык түрдө калыптап (форматтап) жибергендиктен маалымдагычтын терезесине жүктөгөндө каталар пайда болушу мүмкүн.

6.2. JavaScript программалоо тилинде сценарийлерди түзүү.

HTML-кодуна сценарийлерди кошуу үчүн <script>...</script> тегдери колдонулат. Бул эки тегдин ортосуна жазылган текстти маалымдагыч программалык код катары түшүнөт жана аларды экранга чыгарбайт.

Сценарийлерди Web-беттин бөркүнө жана негизги бөлүгүнө (телосуна) кошууга болот. Сценарийлерди жазууда кайсыл тил колдонулгандыгын даана аныктоо үчүн **language** атрибуту колдонулат:

```
<script language="JavaScript">...</script>
```

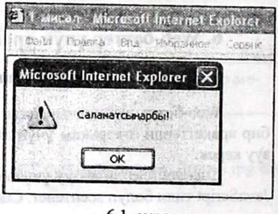
Бул коддо программалоо тили үчүн JavaScript тандалган. Эгерде программалоо тили үчүн VBScript тандалган болсо, анда кодду төмөнкүдөй жазабыз:

```
<script language="VBScript">...</script>
```

6.1-мисал. Экранга Саламатсыңарбы деген билдирүүнү чыгаруучу сценарийди жазгыла.

JavaScript тилин колдонуп, төмөнкүдөй программа жазабыз:

```
Салам.htm
<html>
<head><title>6.1-мисал</title></head>
<body>
<script language="javascript">
alert("Саламатсыңарбы!")
</script>
```



6.1-сүрөт.

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

6.3. Маалыматтарды киргизүү жана чыгаруу.

JavaScript тилинде маалыматтарды киргизүү жана чыгаруу үчүн төмөнкү 3 метод колдонулат:

- **Alert();**
- **Confirm();**
- **Prompt()/**

1). Alert методу. Alert методу берилген билдирүүнү жана ОК баскычын кармап турган диалогдук терезени экранга чыгарып берет (6.1-сүрөт). Программалык коддогу синтаксис төмөнкүдөй (1-мисал):

alert("билдирүү")

2). Confirm методу. Confirm методу берилген билдирүүнү жана эки баскычты (ОК жана Отмена (Cancel)) кармап турган диалогдук терезени экранга чыгарып берет (1-сүрөт). Программалык коддогу синтаксис төмөнкүдөй:

confirm("билдирүү")

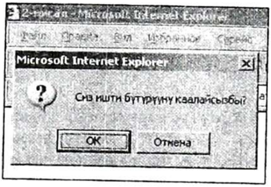
Alert методунан айырмаланып бул метод логикалык чоңдуктун маанисин аныктап берет. Эгерде ОК баскычы басылса true (чын, ооба) мааниси, ал эми Отмена баскычы басылса false (жалган, жок) мааниси алынат.

6.2-мисал. Сиз ишти бүтүрүүнү каалайсызбы? Деген суроого жооп берүүчү диалогдук терезени экранга чыгаргыла.

JavaScript тилинде төмөнкү программа түзөбүз:

Конфирм.htm

```
<html>
<head><title>2-мисал</title></head>
<body>
<script language="javascript">
confirm("Сиз ишти бүтүрүүнү каалайсызбы?")
</script>
<body>
</html>
```



Программанын аткарылышы 6.2-сүрөттө көрсөтүлдү.

6.2-сүрөт.

3). Prompt методу. Prompt методу колдонуучу тарабынан маалыматтарды киргизе ала турган тексттик терезени, берилген билдирүүнү жана эки баскычты (ОК жана Отмена (Cancel)) кармап турган диалогдук терезени экранга чыгарып берет (6.3-сүрөт). Программалык коддогу синтаксис төмөнкүдөй:

prompt("билдирүү", "маалымат")

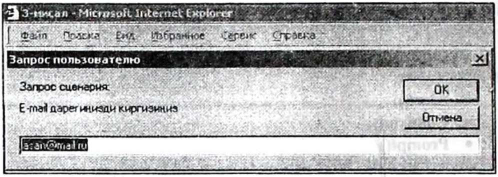
6.3-мисал. prompt методу менен түзүлгөн диалогдук терезени экранга чыгаргыла.

JavaScript тилинде төмөнкү программа түзөбүз:

Промпт.htm

```
<html>
<head><title>6.3-мисал</title></head>
<body>
```

```
<script language="javascript">
prompt("E-mail дарегиңизди киргизиңиз","asan@mail.ru")
</script>
<body>
</html>
```



6.3-сүрөт.

6.3-сүрөттө программанын аткарылышы көрсөтүлдү.

Диалогдук терезелерге чыгарылуучу маалыматтарды форматтоо үчүн төмөнкү кызматчы символдорду колдонууга болот (6.1-таблица).

6.1-таблица.

№	Символ	Мааниси
1	\n	Жаңы жолчо
2	\t	Табуляция
3	\f	Жаңы бет
4	\b	Забой
5	\r	Enter баскычы

6.4-мисал. Символ.htm.

```
<html>
<head><title>Форматтоо</title></head>
<body>
<script language="javascript">
alert("Фамилия – Асанов \nАты – Акмат
\nАтасынын аты - Эрмекович!")
</script>
<body>
</html>
```

1-таблицада көрсөтүлгөн символдорду пайдаланып, 6.4-сүрөттө көрсөтүлгөн маалыматтарды alert методу менен экранга чыгарабыз.

6.4-сүрөт.

7. Берилгендердин тиби. Операторлор.

7.1. Берилгендердин тиби.

Программалоо тилинде берилгендердин тибин аныктап алуу чоң мааниге ээ. Берилгендер компьютердин эсинде сакталат жана аларды кайра иштетүүгө болот, б.а. алардын үстүнөн кандайдыр бир операцияларды аткарабыз.

Мисалы, көбөйтүү операциясы сандарга колдонулат, б.а. сандык типтерге колдонулат. Бирок “Асан” деген сөздү 5 деген санга көбөйтүүгө болобу? Бул суроо-

го жооп берүүдөн мурда ойлонууга туура келет, анткени бул ар түрдүү типтеги берилгендерди көбөйтүүгө болобу деген суроо менен тең күчтүү.

JavaScript тилинде берилгендердин төмөнкүдөй тиби бар (7.1-таблица):

7.1-таблица

№	Типтин аталышы	Жазылышы	Маанилеринин мисалдары
1	Сандык	Number	11, 3.14, 325
2	Символдук	String	«Салам», «12»
3	Логикалык	Boolean	True, False
4	Объект	Object	Программалык объект
5	Функция	Function	Маанини кайтарып берүүчү программа

JavaScript тилинде өзгөрмөлөрдү жарыялоо үчүн var кызматчы сөзү колдонулат. Мисалы:

Var a, t, name

Башка тилдерден айырмаланып, өзгөрмөлөргө биринчи маани берилгенде эле өзгөрмөлөрдүн тиби автоматтык түрдө жарыяланат:

var n=1

var a=3.14

var name="Асан"

«=» белгиси ыйгаруу операторун билдирет.

7.2. Операторлор.

Өзгөрмөлөр менен болгон операцияларды аткаруу үчүн JavaScript тилинде операторлордун төмөнкүдөй типтери колдонулат:

- Ыйгаруу оператору;
- Комментария (түшүндүрмө) оператору;
- Арифметикалык операторлор;
- Салыштыруу операторлору;
- Логикалык операторлор;
- Символдук операторлор.

1). Ыйгаруу оператору. Ыйгаруу оператору «=» белгиси менен аныкталат. Калыбы төмөнкүдөй:

өзгөрмөнүн_аты = маани

Барабардыктын оң жагындагы маани сол жактагы өзгөрмөнүн атына ыйгарылат.

Мисал:

myName="Акматов"

2). Комментария оператору. Комментария операторунун эки формасы бар:

- // - ушул белгиден кийинки жазылган символдордун жолчосу комментария (түшүндүрмө) деп аталат;

- /* … */ - ушул белгилердин арасына жайгашкан символдордун жолчосу да комментария (түшүндүрмө) деп аталат.

3). Арифметикалык операторлор. Арифметикалык операторлордун төмөнкүдөй түрлөрү бар (7.2-таблица).

7.2-таблица.

№	Оператор	Аталышы	Мисалдар
1	+	Кошуу	x+y
2	-	Кемитүү	x-y
3	*	Көбөйтүү	x*y
4	/	Бөлүү	x/y
5	%	Модулу боюнча бөлүү	x%y
6	++	1 ге чоңойтуу	x++
7	--	1 ге кичирейтүү	y--

4). Салыштыруу операторлору. Эки өзгөрмөнүн маанилеринин айырмасын же барабардыгын аныктоо үчүн салыштыруу операторлорунун төмөнкүдөй түрлөрү колдонулат (7.3-таблица):

7.3-таблица.

№	Оператор	Аталышы	Мисалдар
1	= =	Барабар	x= =y
2	!=	Барабар эмес	x!=y
3	>	Чоң	x>y
4	>=	Чоң же барабар	x>=y
5	<	Кичине	x<y
6	<=	Кичине же барабар	x<=y

Бардык салыштыруу операторлорунун маанилери true (чын) же false (жалган) болот. Мисалы,

var bVal = 5= =3;

туюнтмасында bVal өзгөрмөсүнө false мааниси ыйгарылат.

5). Логикалык операторлор. Логикалык операторлордун төмөнкүдөй түрлөрү бар (7.4-таблица).

7.4-таблица.

№	Оператор	Аталышы	Мисалдар
1	!	Тануу	!x
2	&&	Жана	x&&y
3	||	Же	x||y

6). Символдук операторлор. Web-беттеги тексттик жолчолорду жана өзгөрмөлөрдү анализдөө жана кайра иштетүү үчүн конкатенация (суммалоо) деп аталган символдук оператор колдонулат. Конкатенация деп жолчолорду бириктирүү процессин айтабыз.

Көпчүлүк учурда ар түрдүү өзгөрмөлөрдө сакталган тексттердин фрагмент-

терин бир текст катары бириктирип, маалымдагычтын терезесине чыгаруу үчүн сценарий жазууга туура келет. Бул маселелени жолчолорду конкатенациялоо (суммалоо) операторунун жардамы менен аткарууга болот. Конкатенациялоо (суммалоо) операторунун белгиси катары + символу колдонулат.

7.1-мисал. Суммалоо операторунун жардамы менен жолчолорду бириктирүү.

```
var str1
str2="Урматтуу";
str3="Акмат Асанович"
str1=str1+" "+str2+"!"
```

Бул мисалда str2, " ", str2, "!" тексттик фрагменттеринен пайдаланып жана суммалоо операторун 3 жолу колдонуу менен бир сүйлөм түзүлдү жана анын мааниси str1 өзгөрмөсүнө ыйгарылды. Натыйжада str1="Урматтуу Акмат Асанович!" болот.

7.3. Write() методу. Write() методу документтин терезесине (Web-бетке) бир же бир нече тексттик жолчолорду чыгаруу үчүн колдонулат.

Анын калыбы төмөнкүдөй:

document.write(1-жолчо, [, 2-жолчо … [, n-жолчо]])

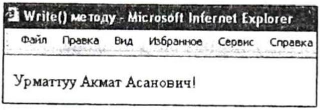
Бул жерде квадраттык кашаалардагы параметрлер шарттуу параметрлер эмес, б.а. алар эске алынбашы да мүмкүн. Бирок Write() методу параметр катары каалаган сандагы жолчону кабыл алышы мүмкүн.

Параметр катары алынган жолчолордо тексттер же HTML-коддор кош кавычкага же жалгыз кавычкага алынып жазылышы керек. Эгерде өзгөрмөлөр кавычкага алынбай жазылса, анда экранга алардын маанилери чыгарылат.

7.2-мисал. 7.1-мисалда алынган натыйжаны маалымдагычтын терезесине (Web-бетке) чыгаргыла.

Программалык кодду төмөнкүдөй түзөбүз:

```
<html><head>
<title>Write() методу</title></head>
<script>
    var str1
    str2='Урматтуу'
    str3='Акмат Асанович'
    str1=str2+' '+str3+'!'
    document.write(str1)
</script>
</html>
```



7.1-сүрөт.

Мисалда эки тексттик фрагменттен, бир пробелден (боштуктан) жана ! белгисинен str1 тексттик жолчосу түзүлдү жана анын мааниси 7.1-сүрөттө көрсөтүлдү.

8. Сценарийлерди аткаруу тартибин башкаруу.

Эң жөнөкөй учурда сценарийдин коддору биринчи жолчодон акыркы жолчого чейин ырааты (тартиби) менен өзгөртүлбөй аткарылат.

Тартиби өзгөртүлбөй аткарылган программалык кодду сызыктуу программа деп айтабыз.

Көпчүлүк учурларда белгилүү бир шарттар орун алганда альтернативдүү түрдө аткарыла турган программаларды түзүүгө же программалык кодду бир нече жолу кайталап колдонууга туура келет. Мындай программаларды сызыктуу эмес программа деп айтабыз.

8.1. Логикалык туюнтмаларды түзүү. Маанилерди салыштыруу.

Салыштыруу операторлорунун жардамы менен өзгөрмөлөрдүн маанилерин салыштырууга болот. Кээде өзгөрмөнүн маанилеринин кандайдыр бир диапазондо жатышын текшерүүгө туура келет. Айрым учурларда программаны бир нече шарттар орун алганда гана аткарууга туура келет. Маселелерди чечүү үчүн мындай учурларда логикалык туюнтмаларды түзүү максатка ылайыктуу болот.

Логикалык туюнтмаларды түзүү үчүн && (логикалык ***жана***) же || (логикалык ***же***) операторлорун колдонобуз.

Логикалык && (***жана***) операторунун жардамы менен эки шарттын ар бири чын болгондо чындык болот турган туюнтманы түзүүгө болот.

Мисалы,

bval = (*x*>=5) && (*x*<=12)

туюнтмасы *x* өзгөрмөсү [5, 12] сегментине таандык болгондо bval өзгөрмөсүнө true мааниси ыйгарыла тургандыгын билдирет.

Логикалык || (*же*) операторунун жардамы менен эки шарттын жок дегенде бирөө чын болгондо чындык болот турган туюнтманы түзүүгө болот.

Мисалы,

bval = (*x*>10) || (*y*= ="Асан")

туюнтмасы *x* өзгөрмөсү 10 дон чоң болгондо же *y* тин мааниси Асан деген текст менен дал келгенде bval өзгөрмөсүнө true мааниси ыйгарыла тургандыгын билдирет.

8.2. Цикл операторлору.

Эгерде бир эле маселени өзгөрмөнүн ар түрдүү маанилеринде же ар түрдүү объекттер үчүн чечүү зарыл болсо, анда программаны цикл боюнча түзүү максатка ылайыктуу.

JavaScript тилинде циклдик процесстер үчүн үч оператор колдонулат:

- for;
- while;
- do-while.

1). for цикли. Циклди уюштуруу үчүн for оператору колдонулат жана анын синтаксиси төмөнкүдөй болот:

```
for (эсептегич = баштапкы_маани; бүтүрүү_шарты; эсептегичтин_өсүндүсү)
        {циклдин телосу}
```

for операторунан кийин тегерек кашаанын ичине циклди аныктоо коду жазылат, андан кийин фигуралык кашаанын ичине цикл боюнча кайталануучу программа – циклдин телосу жазылат. Циклдин телосу бир эле жолчодон турган учурда фигуралык кашааны жазбаса деле болот.

Циклди аныктоо коду үтүрлүү чекит менен ажыратылган үч туюнтмадан турат:

- Биринчи туюнтма бүтүн санды аныктоочу өзгөрмө-эсептегич болот жана анын баштапкы мааниси берилиши керек;
- Экинчи туюнтма – циклди бүтүрүүчү логикалык шартты аныктайт: шарт туура (чын) болсо цикл уланат, шарт аткарылбаса (жалган болсо) цикл токтотулат;
- Үчүнчү туюнтма эсептегичтин маанисинин өзгөрүү өсүндүсүн аныктайт.

8.1-мисал. 1 ден 100 го чейинки натуралдык сандардын суммасын тапкыла.

Маселени чыгаруу үчүн төмөнкү программаны түзөбүз.

```
<html><head><title>for цикли</title></head>
<script>
var S=0, n=100
for (i=1; i<=n; i=i+1)
{ S=S+i }
document.write("S=1+...+"+n+"<BR>")
document.write("for циклин колдонгон-
до:<BR>").
document.write("S=", S+"  болот.")
</script></html>
```

8.1-сүрөт.

Мисалда і өзгөрмөсү эсептегичти аныктайт. Цикл токтотуучу шарт катары i<=100 логикалык шарты алынды. Бул шарт аткарылбаганда цикл токтотулат жана жыйынтык экранга чыгарылат. Эсептегичтин өсүндүсү үчүн i=i+1 туюнтмасы тандалган, б.а. ар бир цикл аяктаганда эсептегичтин мааниси улам бирге өсүп олтурат. Программалык код аткарылганда экранга S=5050 деген маалымат чыгат (8.1-сүрөт).

2). while цикли. while циклинде программа берилген логикалык шарт аткарылбай калганга чейин улана берет. while циклинде for циклинен айырмаланып эсептегич жок.

Циклдин иштөө принциби төмөнкүдөй: берилген логикалык шарт циклге кирүүдөн мурда жана циклдин телосунун эң акыркы жолчосу аткарылгандан кийин текшерилет. Эгерде шарт чын (true) болсо, цикл уланат, жалган (false) болсо цикл токтотулат.

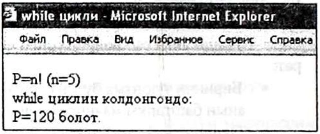
Циклдин синтаксиси төмөнкүдөй:

```
while (логикалык_туюнтма)
        { циклдин телосу }
```

8.2-мисал. n! санын эсептөө программасын түзгүлө.

Мисалды чыгаруу үчүн while циклин колдонобуз.

```
<html><head>
<title>while цикли</title></head>
<script>
var P=1, n=5, i=1
while (i<=n)
{ P=P*i
i=i+1 }
document.write("P=n! (n=5) <BR>")
document.write("while циклин колдонгондо:<BR>")
document.write("P=", P+"  болот.")
</script>
</html>
```



8.2-сүрөт.

Бул мисалда логикалык шарт үчүн i<=n туюнтмасы алынды. Цикл ушул шарт орундалбай калганга чейин улантылат. Биздин мисалда цикл 5 жолу кайталанат. Программанын жыйынтыгы 8.2-сүрөттө көрсөтүлдү.

3). do-while цикли. do-while цикли деле берилген логикалык шарт аткарылбай калганга чейин улана берет. Бирок while циклинен айырмаланып, шарт эң акырында текшерилет. Эгерде шарт орун алса цикл улантылат, тескери учурда цикл токтотулат.

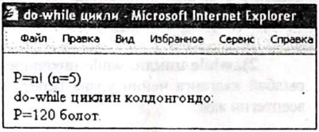
Циклдин синтаксиси төмөнкүдөй:

```
do
{ циклдин телосу }
while (логикалык_туюнтма)
```

8.3-мисал. n! санын do-while циклин колдонуп эсептегиле.

Маселени чыгаруу үчүн төмөнкүдөй программа түзөбүз.

```
<html><head>
<title> do-while цикли</title></head>
<script>
var P=1, n=5, i=1
do
{ P=P*i
i=i+1 }
while (i<=n)
document.write("P=n! (n=5)<BR>")
document.write("do-while циклин колдонгондо:<BR>")
```



8.3-сүрөт.

```
document.write("P=", P+"  болот.")
</script> </html>
```

Мисалда циклдин телосу i<=n шарты орундалбай калганга чейин улантылат. Программанын аткарылышы 8.3-сүрөттө көрсөтүлдү.

9. Программалык коддогу тармактануу.

Циклдер программалык коддун белгилүү бир фрагментин көп жолу кайталоо менен берилгендердин маанилеринин көптүгүн кайра иштетүү үчүн ыңгайлуу шарт түзөт жана программанын кубаттуулугун жогорулатат.

Көпчүлүк учурларда логикалык шарттардын чын же жалган болушуна карата программалык коддун бул же тигил тармагы аткарылат.

JavaScript тилинде программалык коддогу тармактануу процессин камсыз кылуу үчүн төмөнкү конструкциялар колдонулат:

- if-else;
- switch-case;

9.1. if-else конструкциясы.

Көпчүлүк учурларда логикалык шарттын аткарылышына же аткарылбашына карап, программалык коддун эки альтернативдүү блогунун бирин тандоого туура келет. Мындай ситуацияларды моделдөө үчүн if-else конструкциясы колдонулат:

```
if (логикалык_туюнтма)
      { логикалык_туюнтманын мааниси true болгон учурдагы
        аткарыла турган программалык коддордун блогу }
else
      { логикалык_туюнтманын мааниси false болгон учурдагы
        аткарыла турган программалык коддордун блогу }
```

Эгерде маселелерди чечүүдө экиден көп альтернативдүү блоктордун бирин тандоого туура келсе, анда бири бирине камтылуучу if-else конструкциясын колдонууга болот:

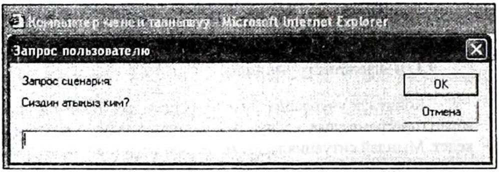
```
var A=логикалык_туюнтма
var B=логикалык_туюнтма
      if (A)
            {1-программа}
      else
         {if (B)
            {2-программа}
         else
            {3-программа}
         }
```

9.1-мисал. Компьютердин колдонуучу менен таанышуу диалогун демонстрациялоочу программа түзгүлө.

Маселени чечүү үчүн if-else конструкциясын колдонобуз.

```
<html><head><title>Компьютер менен таанышуу</title></head>
<body><script>
    var a
    a=prompt("Сиздин атыңыз ким?","")
    if (a=="")
    {alert("Сиз атыңызды айтпадыңыз!")}
    else
    {if (a==null)
    {alert("Сиз программадан чыгууну каалардыңыз. Жакшы калыңыз!")}
    else
    {alert("Саламатсызбы "+a+"!")}
    }
</script>
</body></html>
```

Программа аткарылганда 9.1-сүрөттү көрсөтүлгөн терезе пайда болот.



9.1-сүрөт.

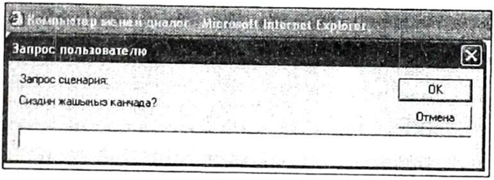
9.2-мисал. Компьютер менен жаштын курагын аныктай турган диалогду түзгүлө.

Маселени чечүү үчүн if-else конструкциясы менен төмөнкүдөй программа түзөбүз.

```
<html>
<head><title>Компьютер менен диалог</title></head>
var a
a=prompt("Сиздин жашыңыз канчада?","")
if (a=="")
{alert("Сиз жашыңызды айтпадыңыз!")}
else
   {if (a==null)
   alert("Сиз программадан чыгууну каалардыңыз. Жакшы калыңыз!")}
   else
      {if (a<12)
      {alert("Сиз жаш баласыз!")}
      else
         {if (a<18)
         {alert("Сиз өспүрүмсүз!")}
```

```
                else
                    {if (a<35)
                    {alert("Сиз жаш адамсыз!")}
                    else
                        {if (a<63)
                        {alert("Сиз орто курактагы адамсыз!")}
                        else
                            {if (a<100)
                            {alert("Сиз пенсиядагы адамсыз!")}
                            else
                            {alert("Сиз бир кылым жашаган адамсыз!")}
}}}}}}
</script></html>
```

Программанын аткарылышы 9.2-сүрөттө көрсөтүлдү.



9.2-сүрөт.

9.2. switch-case конструкциясы.

Эгерде өзгөрмөнүн же туюнтманын маанисине жараша экиден көп альтернативдүү программалык коддордун блогунун бирин аткарууга (тандоого) туура келсе, анда switch-case конструкциясы колдонулат:

```
switch (өзгөрмө же туюнтма)
    { case 1-маани:  1-программа;  break;
      case 2-маани:  2-программа;  break;
    ...............................................
      case n-маани:  n-программа;  break;
      default:  (n+1)-программа
    }
```

switch операторунун шарты бир нече маанилерди аныктай турган өзгөрмө же туюнтма болот. Оператордун телосу фигуралык кашаага алынат.

Программанын иштөө принциби төмөнкүдөй:

- эң оболу switch операторунун шартындагы маани аныкталып, ал case операторундагы 1-маани менен салыштырылат. Эгерде маанилер дал келсе, анда 1-мааниден кийинки программалык коддор аткарылат;
- эгерде маанилер дал келбесе, анда 2-маани салыштырылат жана 2-

мааниден кийинки программалык коддор аткарылат, б.а. switch операторунун шартындагы маани case операторлорундагы кайсы маани менен дал келсе, ошол мааниден кийинки жайгашкан программалык коддор аткарылат;

- эгерде бир да маани туура келбесе, анда башкаруу эң акырында турган default операторундагы программалык коддорго берилет.

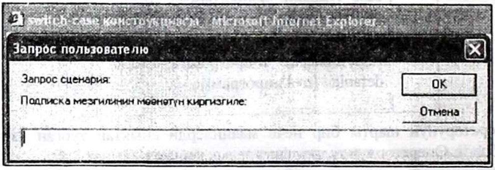
Эгерде switch операторунун шартындагы бардык маанилери case операторлорунда каралган болсо, анда default операторунун программада катышышы шарт эмес.

case операторундагы программалык коддордо чексиз цикл пайда болуп калбашы үчүн break командасы жазылган. break командасы циклди токтотуу үчүн колдонулат.

9.3-мисал. Подпискага жазылуу баасын эсептөөнүн программасын түзгүлө.

```
<html><head><title> switch-case конструкциясы </title></head>
<b>Подпискага жазылуу баасын эсептөө</b> <br>
<script>
var n=Number(prompt("Подписка мезгилинин мөөнөтүн киргизгиле:",""))
var s
switch(n)
{case 1: s=60; break;
 case 3: s=168; break;
 case 6: s=312; break;
 case 12: s=540; break;
 default: s="<br>Сиз жаңылыштыңыз!"}
 document.write("Подписканын баасы:<br>",s)
</script></html>
```

Программа аткарылганда экранда 9.3-сүрөттө көрсөтүлгөн диалогдук терезе пайда болот. Клавиатурадан маалыматтарды киргизүү менен жазылуунун баасын аныктап алабыз. Эгерде жазылуу мезгилинин мөөнөтүн туура эмес киргизсек, анда "Сиз жаңылыштыңыз! деген маалыматты алабыз.



9.3-сүрөт.

Программада колдонулган Number функциясы prompt методунун тексттик талаасына киргизилген маанини санга айландырып берет.

10. Функциялар. Колдонуучунун функциясы.

Функция деп аты боюнча чакырылып, аткарууга мүмкүн болгон программаны айтабыз. Функциянын аргументтерине маанилерди берип жыйынтык алууга болот. Эгерде функциянын аргументтери бир канча болсо, анда алар үтүрлөр менен ажыратылат:

```
function Функциянын_аты (x1, x2,..., xn)
{функциянын телосу}
```

Кашаанын ичиндеги өзгөрмөлөр функциянын аргументи болуп эсептелет. Функциянын аргументи болбошу да мүмкүн. Бул учурда кашаанын ичине эч нерсе жазылбайт. Ал эми функциянын кандайдыр бир маанини кайтарып берүүсү үчүн анын телосуна return операторун жазуу керек.

10.1. Тиркелген объекттер жана функциялар.

JavaScript тилинде тиркелген (встроенный) стандарттык объекттер жана функциялар бар. Алардын кодун биз көрө албайбыз жана оңдой (редакциялай) да албайбыз. Тиркелген функциялардын параметрлери, алардын аткарган кызматы жана кайтарган маанилери жөнүндө гана маалымат алышыбыз мүмкүн.

Стандарттык объекттердин бири болуп Math объекти болуп эсептелет. Бул объект математикалык константаларды сактоо үчүн дайындалган. Math объектинин методдору менен негизги математикалык функцияларды чакырууга болот.

Math объектинин методдору.

№	Методдор	Мааниси
	Тригонометриялык методдор	
1	abs(x)	x санынын абсолюттук маанисин аныктайт
2	acos(x)	x санынын арккосинусун аныктайт ($-1 \le x \le 1$)
3	asin(x)	x санынын арксинусун аныктайт ($-1 \le x \le 1$)
4	atan(x)	x санынын арктангенсин аныктайт
5	cos(x)	x санынын косинусун аныктайт
6	sin(x)	x санынын синусун аныктайт
7	tan(x)	x санынын тангенсин аныктайт
	Тегеректөө методдору	
1	ceil(x)	x тен кичине же ага барабар болгон бүтүн санды аныктайт
2	floor(x)	x тен чоң же ага барабар болгон бүтүн санды аныктайт
3	round(x)	x ке жакын болгон бүтүн санды аныктайт
	Салыштыруу методу	
1	max(x, y)	x жана y сандарынын чоңун аныктайт
2	min(x, y)	x жана y сандарынын кичинесин аныктайт
	Эсептөө методдору	
1	exp(x)	e^x функциясынын маанисин аныктайт
2	log(x)	ln(x) функциясынын маанисин аныктайт ($x>0$)
3	pow(x, y)	x^y функциясынын маанисин аныктайт ($x>0$)
4	sqrt(x)	$\sqrt{x}$ функциясынын маанисин аныктайт ($x \ge 0$)

Math объектинин методдорун төмөнкүдөй форматта колдонобуз:

Math.метод (параметрлер)

Мисалы, Math.sqrt (2) - $\sqrt{2}$ ни эсептейт. Math.max(3, 7) – 7 ни аныктайт.

Аргументи градус менен берилген тригонометриялык функциянын маанисин эсептөө үчүн радиандык ченге өтүү керек. Мисалы, sin(x) (x – градус менен берилген) маанисин эсептөө үчүн

Math.sin(Math.PI*x/180)

туюнтмасын пайдаланабыз. Мында Math.PI туюнтмасы π санын аныктайт.

10.2. Кокустук сандардын генератору.

Math объекти (0, 1) интервалында жатуучу кокустук сандарды аныктоочу функцияны – *рандомайзер* функциясын кармап турат. Кокустук сандардын генератору төмөнкүдөй чакырылат:

var rand=Math.random();

- (0, a) интервалында жатуучу кокустук сандарды алуу үчүн
  var rand=a*Math.random();
  конструкциясын пайдаланабыз.
- (a, b) интервалында жатуучу кокустук сандарды алуу үчүн
  var rand=a+(b-a)*Math.random();
  конструкциясын пайдаланабыз.

10.3. Колдонуучунун функциясы.

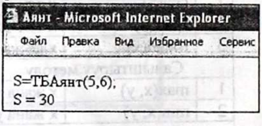
Колдонуучу конкреттүү маселенин шартына карап өзүнүн функциясын түзүп алышы да мүмкүн.

Колдонуучунун функциясы function кызматчы сөзүнүн жардамы менен төмөнкүдөй түзүлөт:

```
<script language='javascript'>
   function функциянын_аты (параметрлердин_тизмеси)
   {программалык коддор}
</script>
```

10.1-мисал. Тик бурчтуктун аянтын эсептөөчү функцияны түзгүлө.
Мисалды чыгаруу үчүн төмөнкү программа түзөбүз.

```
<html><head><title>Аянт</title></head>
<body><script language="javascript">
function ТБАянт(узуну, туурасы)
{ var узуну, туурасы
  Аянт=узуну*туурасы
  return Аянт }
S=ТБАянт(5, 6)
document.writeln("S=ТБАянт(5, 6); <BR> S = ", S)
</script>
</body></html>
```



10.1-сүрөт.

Бул мисалда *ТБАянт* тексти функциянын атын, *узуну* жана *туурасы* деген сөздөр функциянын параметрлери болуп эсептелет, ал эми *Аянт* сөзү функциянын маанисин аныктайт. Программалык коддогу S=*ТБАянт*(5, 6) жолчосу *ТБАянт* функциясынын параметрлерине 5 жана 6 деген маанилер ыйгарылып, return оператору менен функциянын мааниси *Аянт* өзгөрмөсү аркылуу кайтарылды жана ал маани S өзгөрмөсүнө ыйгарылгандыгын билдирет. Программа аткарылгандагы жыйынтык 10.1-сүрөттө көрсөтүлдү.

Бул мисал функциянын аталышынын жана анын параметрлеринин маңызын терең түшүнүү үчүн ушундай формада жазылды. Аны кыскалык үчүн символдордун жардамы менен төмөнкүдөй жазсак деле болот эле:

```
<html><head><title>Аянт</title></head>
<body><script>
      function f(x, y)
      { var x, y
        z=x*y
        return z }
      S=f(5, 6)
   document.writeln("S=f(5, 6); <BR> S = ", S)
   </script>
   </body></html>
```

10.2-мисал. a, b жана c аттуу продукцияларды сатуудан келип чыккан акча каражатын эсептөө функциясын түзгүлө.

```
<html><script>
    var a=12,  b=34, c=13, S
    function f1(a, b, c)
          {S=a*79 + b*129 + c*699
           return S}
    y=f1(10,22,3)
    document.write("Кассага түшкөн акча: <BR> y=", y)
</script></html>
```

Программа аткарылганда экранга төмөнкү маалымат чыгат:

```
Кассага түшкөн акча:
y=5725
```

10.4. Тиркелген функциялар.

JavaScript тилинде төмөнкүдөй тиркелген функциялар бар.

№	Функциянын аталышы	Функциянын аткарган кызматы
1	Number(жолчо)	Көрсөтүлгөн жолчону санга айландырат.
2	String(n)	n санын жолчого айландырат.
3	parseInt(жолчо, негизи)	Көрсөтүлгөн жолчону берилген негиз (8, 10 же 16) боюнча бүтүн санга айландырып берет.
4	parseFloat(жолчо, негизи)	Көрсөтүлгөн жолчону берилген негиз (8, 10 же

		16) боюнча чыныгы санга айландырып берет.
5	isNaN(маани)	Эгерде параметрде көрсөтүлгөн маани сан болбосо, анда функциянын мааниси true болот, тескери учурда false болот.
6	eval(жолчо)	Көрсөтүлгөн жолчодогу туюнтманын маанисин эсептейт.
7	escape(жолчо)	Көрсөтүлгөн жолчодогу символдордун ASCII коддорун аныктап берет.
8	unescape(жолчо)	ASCII коддору боюнча символдордун жолчосун аныктап берет.
9	typeOf(объект)	Көрсөтүлгөн объекттин тибин символдук жолчо түрүндө көрсөтүп берет, мисалы “boolean”, “function” ж.б.

11. Массивдер.

Массив деп бир типтүү берилгендердин же өз ара байланышкан берилгендердин жыйындысын айтабыз.

11.1. Массивдерди түзүү жана колдонуу.

Массив өзгөрмөлөрдүн өзгөчө тиби болуп, ал элементтери деп аталуучу маанилерди кармап турат. Массивдин элементтери киргизилген тартиби боюнча сакталат. Ошондуктан массив тартиптештирилген маалыматтардын жыйындысы болот. Анын ар бир элементине массивдин аты жана индекси деп аталган катар номери боюнча кайрылууга болот. Кайрылуунун синтаксиси төмөнкүдөй:

элемент=массивдин_аты[индекс]

Массивдин элементтери 0 (нөлдөн) баштап номерленген болот. Демек,

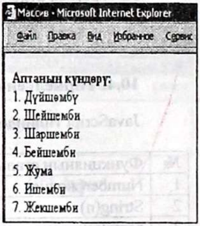
биринчи-элемент= *массивдин_аты[0]*
акыркы элемент= *массивдин_аты[массивдин_узундугу-1]*

Массивдин узундугу деп анын элементтеринин санын түшүнөбүз. Индекс катары бүтүн санды же мааниси бүтүн боло турган өзгөрмөнү же туюнтманы алууга болот.

Массив new кызматчы сөзүнүн жардамы менен Array системалык функциясынын (объектинин) жардамында төмөнкүдөй түзүлөт:

массивдин_аты=new Array([массивдин_узундугу])

мында массивдин_узундугу массивдин шарттуу эмес сандык параметри болуп эсептелет жана массивдин өлчөмүн аныктайт. Эгерде массивдин узундугу көрсөтүлбөсө, анда бир да элементи жок болгон бош массивди түшүнөбүз.



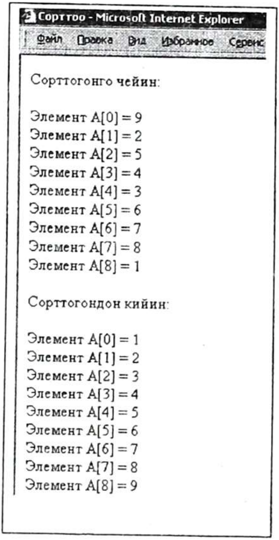
11.1-сүрөт.

Array объектинин lehgth аттуу касиети бар жана анын мааниси катары массивдин узундугу алынат. Бул касиеттин маанисин алуу үчүн массивдин_аты.lehgth

туюнтмасын алуу керек.

11.1-мисал. Аптанын күндөрүнөн турган массивди түзгүлө.

```
<html><head><title>Массив</title></head>
<script>
document.writeln("<b>Аптанын
күндөрү:</b>")
var апта=new Array(7)
апта[0]="Дүйшөмбү"
апта[1]="Шейшемби"
апта[2]="Шаршемби"
апта[3]="Бейшемби"
апта[4]="Жума"
апта[5]="Ишемби"
апта[6]="Жекшемби"
for(i=0;i<=6;i=i+1)
{document.writeln("<br>"+(i+1)+".  ",
апта[i])}
</script></html>
```



11.2-сүрөт.

Мисалда 7 элементтен турган апта массиви түзүлдү жана анын элементтери 11.1-сүрөттө чагылдырылды.

11.2. Массивдер методу.

Array (массивинин) объектисинин бир топ методдору бар. Метод деп объектин бир бөлүгү болгон функцияны айтабыз.

Ага төмөнкүдөй кайрылууга болот: *объектин_аты.метод()*. Мисалы document объектинин write() методуна document.write() түрүндө кайрылабыз. Төмөнкү таблицада массивдин методдорунун түрлөрү келтирилди.

№	Методдор	Аткарган кызматы
1	sort()	Массивдин элементтерин сорттойт
2	reverse()	Массивдин элементтерин тескери тартипте жазат
3	push(маани)	Массивдин акырына бир же бир нече элементти кошот
4	pop()	Массивдин акыркы элементин өчүрөт
5	unshift(маани)	Массивдин башына бир же бир нече элементти кошот
6	shift()	Массивдин биринчи элементин өчүрөт
7	join()	Массивдин элементтерин бир жолчого бириктирет
8	concat()	Эки массивди бир массивге бириктирет

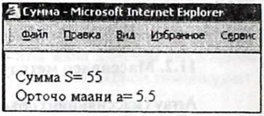
11.2-мисал. sort() функциясы.

```
<html>
<head><title>Сорттоо</title></head>
<script>
var A = new Array (9,2,5,4,3,6,7,8,1)
document.write("Сорттогонго чейин:<br><br>")
for (i=0; i<=A.length-1;i=i+1)
{document.write("Элементы A["+i+"] = ", A[i]+"<br>")}
A.sort()
document.write("<br>Сорттогондон кийин:<br><br>")
for (i=0; i<=A.length-1;i=i+1)
{document.write("Элементы A["+i+"] = ", A[i]+"<br>")}
</script></html>
```

Программанын аткарылышы 11.2-сүрөттө көрсөтүлдү.

11.3. Сандык массивдердин статистикалык мүнөздөмөлөрүн аныктоо.

Көпчүлүк учурларда массивдерде сакталган сандык берилгендердин статистикалык мүнөздөмөлөрүн, б.а. массивдин бардык элементтеринин суммасын, алардын орточосун, минималдык жана максималдык элементтерин аныктоого туура келет.



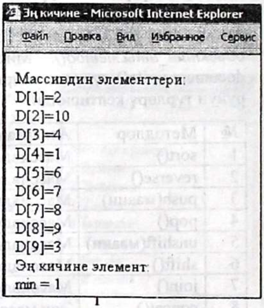
11.3-мисал. Массивдин бардык элементтеринин суммасын жана алардын арифметикалык орточо маанисин тапкыла.

```
<html><head><title>Сумма</title></head>
```

11.3-сүрөт.

```
<script>
function f(V)
{ var s=V[0]
for (var i=1; i<=V.length-1; i=i+1)
{ s=s+V[i]}
return s}
var B = new Array (1,2,3,4,5,6,7,8,9,10)
C=f(B);       a=C/B.length
document.write("Сумма= ", C)
document.write("<br>Орточо маани a= ", a)
</script></html>   <html><head><title>
```

Программанын натыйжасы 11.3-сүрөттө көрсөтүлдү.



1.4-сүрөт.

11.4-мисал. Массивдин элементтеринин эң кичинесин тапкыла.

Мисалды чыгаруу программасы төмөнкүдөй:

```
<html><head><title>Эң кичине</title></head>
<script>
function мин(V)
{ var min=V[0]
for (var i=1; i<=V.length-1; i=i+1)
{if (V[i]<min)
min=V[i]}
return min}
var D =new Array(5,2,10,4,1,6,7,8,9,3)
document.write("Массивдин элементтери:<br>")
for (var i=1; i<=D.length-1; i=i+1)
{document.write("D["+i+"]=", D[i]+"<br>")}
m=мин(D)
document.write("Эң кичине элемент: <br> min = ", m)
</script></html>
```

Программанын натыйжасы 11.4-сүрөттө көрсөтүлдү.

11.5-мисал. Массивдин элементтеринин эң чоңун тапкыла.

```
<html><head><title>Эн чоң</title></head>
<script>
function мак(V)
{ var max=V[0]
for (var i=1; i<=V.length-1; i=i+1)
{if (V[i]>max)
max=V[i]}
return max}
var D =new Array(5,2,10,4,1,6,7,8,9,3)
document.write("Массивдин элементте-
ри:<br>")
for (var i=1; i<=D.length-1; i=i+1)
{document.write("D["+i+"]=",D[i]+"<br>")}
m=мак(D)
document.write("Эң чоң элемент:
<br> max = ", m)
</script></html>
```

Программанын натыйжасы 11.5-сүрөттө көрсөтүлдү.

11.5-сүрөт.

12. Динамикалык Web-беттерди түзүү үчүн окуяларды колдонуу.

12.1. Окуя жөнүндө түшүнүк.

Окуя деп колдонуучунун иш-аракетин айтабыз. Мисалы колдонуучунун клавиатуранын баскычын басышы, чычкандын сол же оң баскычы менен тигил же бул объектке чыкылдатышы, тексттин кандайдыр бир фрагментин бөлүп алуусу ж.б. аракеттери окуяны пайда кылат. Ар бир окуяны операциялык система анализдеп, эмне кылуу керектиги жөнүндө маалымдагычка (обозреватель) жөнөтөт. Окуяга жараша иш-аракеттерди аткаруу үчүн сценарий колдонулат.

12.2. Окуянын касиеттери.

Окуя жөнүндөгү билдирүү объект катарында түзүлөт. Окуя түзүлгөндө маалымдагыч анын касиеттерине маанилерди ыйгарат. Мисалы, чычкандын көрсөткүчүнүн чыкылдатылышына (щелчок) туура келүүчү окуя чычкандын көрсөткүчүнүн координатасы, кайсы баскычынын басылгандыгы жана окуя кайсы элемент менен байланышкандыгы жөнүндөгү маалыматтарды өзүнүн касиеттеринде сактап калат.

Окуяны кайра иштетүүчү параметр on сөзү менен башталат. Мисалы Click окуясын кайра иштетүүчү параметр onClick болуп эсептелет. Окуяны программалык коддо ишке ашыруу үчүн баскычтар (кнопки) колдонулат.

12.3. Баскычтардын окуясы.

Баскычтарды INPUT же BUTTON элементтери менен түзүүгө болот. Баскычтардын үч түрү бар: "submit", "reset", "button". Аларды type атрибуту менен жарыялайбыз.

- submit - жөнөтүү баскычы: Баскыч активдештирилгенде формадагы маалыматтар жөнөтүлөт. Бул маани унчукпаганда колдонулат.
- reset – тазалоо баскычы: Баскыч активдештирилгенде бардык башкаруучу элементтердин маанилери баштапкы маанини алат.
- button – башка баскычтарды билдирет. Бул баскыч үчүн баштапкы маани аныкталбайт. Мындай баскычтардын окуясы үчүн скриптерди колдонууга болот.

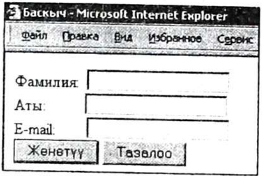
BUTTON элементи менен түзүлгөн баскычтар INPUT элементи менен түзүлгөн баскычтардай эле иштейт, бирок мүмкүнчүлүгү бир топ чоң.

12.1-мисал. Тексттик талаалардагы маалыматтарды жөнөтүү жана тазалоо үчүн BUTTON элементин колдонгула.

```
<html>
<form>
Фамилия: <input type="text" size="20"><BR>
Аты:         
```

```
<input type="text" size="20"><BR>
E-mail:       
<input type="text" size="20"> <BR>
<button name="submit" type="submit">Жөнөтүү</button> 
<button name="reset" type="reset">Тазалоо</button>
</form></html>
```

Жыйынтык 12.1-сүрөттө келтирилди.



12.1-сүрөт.

12.2-мисал. Тексттик талаалардагы маалыматтарды жөнөтүү жана тазалоо үчүн INPUT элементин колдонгула.

```
<html>
<head><title>Баскыч</title></head>
<form>
Фамилия:  <input type="text" size="20"><BR>
Аты:         
<input type="text" size="20"><BR>
E-mail:       
<input type="text" size="20"> <BR>
<input name="submit" type="submit" value="Жөнөтүү"> 
<input name="reset" type="reset" value="Тазалоо">
</form>
</html>
```

13. Датаны жана убакытты аныктоо.

Системалык датаны жана убакытты сценарийлерде колдонуу жана аларды Web-беттерде ар түрдүү форматтарда чагылдыруу үчүн Date() объектин пайдаланабыз. Анын төмөнкүдөй методдору бар:

• getYear()	- жыл	• getHours()	- саат
• getMonth()	- айдын катар номери	• getMinutes()	- минута
• getDay()	- аптанын күндөрү	• getSecondes()	- секунда
• getDate()	- айдын катар номери		

13.1-мисал. Date() объектинин методдорун колдонуп Web-бетке датаны түзүүчү элементтерди чыгаргыла.

```
<html><head><title>Date() объекти</title></head>
<script>
var D=new Date();
document.write("Жыл:", D.getYear(),"<br>")
document.write("Айдын катар номери:", D.getMonth(),"<br>")
document.write("Аптанын күндөрү:", D.getDay(),"<br>")
```

```
document.write("Айдын күндөрү:", D.getDate(),"<br>")
document.write("Саат:", D.getHours(),"<br>")
document.write("Минута:", D.getMinutes(),"<br>")
document.write("Секунда:", D.getSeconds(),"<br>")
</script></html>
```

13.1-сүрөт.

Программанын аткарылышы 13.1-сүрөттө көрсөтүлдү. Бул маалымат 2008-жылдын 16-мартында жекшемби күнү алынган.

Эскертүү. Компьютерде айдын катар номери 0 дөн баштап эсептелет. Ошондуктан 13.1-мисалда март айынын катар номери 2 деп чыгып калган.

Бул кемчилдикти оңдоо үчүн программалык кодго төмөнкүдөй өзгөртүү киргизүү керек:

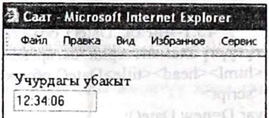
document.write("Айдын катар номери:", D.getMonth()+1,"<br>")

Мындан тышкары, англичандарда аптанын биринчи күнү деп жекшемби эсептелгендиктен, жекшембиге 0 саны тиешелештикке коюлган.

13.2-мисал. Date() объектинин методдорун колдонуп Web-бетке учурдагы убакытты чыгаргыла.

Саатты, минутаны жана секунданы чыгаруу үчүн getHours(); getMinutes(); getSeconds() методдорун пайдаланабыз. Сааттын, минутанын жана секунданын маанилери 10 санынан кичине (б.а. бир орундуу цифра) болгондо аны 0 саны менен толуктоо үчүн Толуктоо(Маани) деген функцияны түзөбүз. Төмөнкү программа аткарылган учурда Web-бетке учурдагы убакыт чыгарылат (13.2-сүрөт).

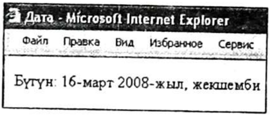
```
<html><title>Саат</title><head>
<script>
function Толуктоо(Маани)
{return (Маани > 9) ? "" + Маани: "0" + Маани }
function clock()
{D = new Date();
document.f1.cl.value = Толуктоо(D.getHours()) + ":" + Толуктоо(D.getMinutes()) +
":" + Толуктоо(D.getSeconds());
setTimeout("clock()",1000)}
</script></head>
<body onLoad="clock()">
<form name="f1">
Учурдагы убакыт<br>
<input type="text" name="cl"
size="10"></form>
</body></html>
```



13.2-сүрөт.

13.3-мисал. Date() объектинин методдорун колдонуп Web-бетке учурдагы датаны кыргыз тилинде чыгаргыла.

Бул маселени чечүү үчүн эң оболу аптанын күндөрүн жана айлардын аттарын массив катарында түзүп алабыз. Web-бетке чыгарылуучу маалыматты динамикалык түрдө өзгөртүү үчүн innerHTML атрибутун колдонобуз.



13.3-сүрөт.

Программаны төмөнкүдөй түзөбүз:

```
<html><head><title>Дата</title></head>
<body>
<p name="clock" id="clock">
<script>
var апта=new
Array("жекшемби","дүйшөмбү","шейшемби","шаршемби","бейшемби","жума",
"ишемби");
var ай=new
Array("январь","февраль","март","апрель","май","июнь","июль","август","сентябрь","октябрь","ноябрь","декабрь");
var D=new Date();
clock.innerHTML = "Бүгүн: " + D.getDate() + "-" + ай[D.getMonth()] + " " +
D.getYear() + "-жыл, " + апта[D.getDay()];
</script></body></html>
```

Программанын аткарылышы 13.3-сүрөттө көрсөтүлдү.

14. Window объекти.

Браузердин иерархиясынын эң башында window объекти турат. Ал бардык фреймдердин коллекциясы болгон frames жана document объектин кармап турат. window объектинин касиеттери, методдору жана окуялары бар.

Касиеттери:

№	Касиеттери	Мааниси
1	name	<frameset> тегиндеги фреймдин аты менен дал келет
2	top	Эң башкы браузердин терезесине шилтеме кылат
3	self	Учурдагы терезеге шилтеме жасайт
4	opener	Түзүлгөн терезе учурдагы терезеге айланат
5	closed	Терезенин жабык экендигин маалымдайт

Методдору:

№	Касиеттери	Мааниси
1	open()	Браузердин жаңы терезесин ачат
2	close()	Браузердин учурдагы терезесин жабат
3	alert()	ОК баскычы жана эскертүү берүүчү билдирүүсү бар те-

		резени көрсөтөт
4	confirm()	ОК жана Cancel (Отмена) баскычтары жана тастыктоону талап кылган билдирүүсү бар терезени көрсөтөт
5	prompt()	ОК жана Cancel баскычтары, чакырууну сунуштаган билдирүүсү, тексттик талаасы бар терезени көрсөтөт
6	scroll()	Туурасы жана бийиктиги берилген терезеге ылайыктайт
7	setInterval()	Берилген миллисекундадан кийин программанын мезгилдүү түрдө аткарыла беришин камсыздайт
8	setTimeout()	Бет жүктөлгөндөн кийин берилген миллисекунда өткөндө программаны ишке түшүрөт
9	clearInterval()	setInterval() методу менен берилген таймердин маанисин нөлгө барабарлайт
10	clearTimeout()	setTimeout()методу менен берилген таймердин маанисин нөлгө барабарлайт

Колдонуучу сценарийдин жардамы менен жаңы терезени түзө алат жана ага ар түрдүү документтерди жайгаштыра алат.

Жаңы терезе windows.open методу менен түзүлөт:

```
wint= windows.open("URL-дарек", "Терезенин аты","Параметрлер")
```

- wint өзгөрмөсү жаңы терезе жөнүндөгү маалыматтарды сактоо үчүн пайдаланылат. Анын жардамы менен жаңы терезенин касиеттерин жана методдорун колдонууга болот.
- Биринчи параметр жаңы терезеге жүктөлө турган документтин URL-дарегин аныктайт. Эгерде бул параметрдин мааниси көрсөтүлбөсө, анда жаңы терезе бош (эч кандай маалымат камтылбаган) болот.
- Экинчи параметр - ачылган терезенин атын, б.а. name касиетин аныктайт.
- Үчүнчү аргументте параметрлердин төмөнкүдөй маанилери тандалат: width, height, scrollbars, resizable, toolbar, status, menubar, location, directories.

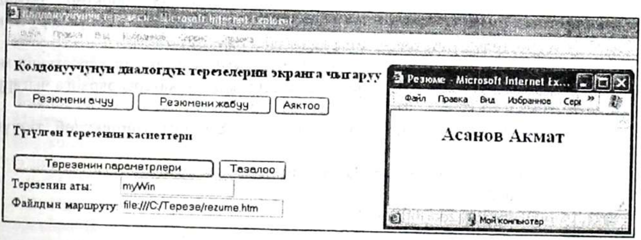
14.1-мисал. Жаңы терезени ачуу, параметрлерин көрүү, терезени жабуу жана ишти аяктоо программасын түзгүлө.

```
<html><head><title>Колдонуучунун терезеси</title>
<script language = "javascript" >
function opnewWin()
{wint = window.open("rezume.htm","myWin",
"width=300, height=100, scrollbars=yes, resizable=yes, toolbar=no, status=yes,
menubar=yes, location=no, directories=no")}
function propwin(obj)
{obj.n1.value = wint.name
obj.n2.value = wint.location.href}
</script></head>
<body>
```

```
<h3>Колдонуучунун диалогдук терезелерин экранга чыгаруу</h3>
<form name = "form1">
<input type = "button" value = "Резюмени ачуу" onClick = "opnewWin()">
<input type = "button" value = "Резюмени жабуу" onClick = "wint.close()">
<input type = "button" value = "Аяктоо" onClick = "window.close()">
</form>
<h4>Түзүлгөн терезенин касиеттери</h4>
<form name = "form2">
<input type = "button" value = "Терезенин параметрлери" onClick = "propwin
(form2)">
<input type = "submit" value = "Тазалоо"><br>
Терезенин аты:         <input
type =text name ="n1"><br>
Файлдын маршруту: <input type = text name = "n2" size = "30">
</form>
</body>
</html>
```

Маселени чечүү үчүн function opnewWin() – жаңы терезени ачуучу, function opnewWin() – ачылган терезенин параметрлерин аныктоочу функцияларды түзүп алабыз. Жаңы терезеге rezume.htm аттуу Web-бетин жайгаштырабыз. Программанын аткарылышы 14.1-сүрөттө көрсөтүлдү. Терезени ачуу, жабуу, параметрлерин аныктоо, тексттик талааларды тазалоо жана программадан чыгуу үчүн тиешелүү баскычтарды даярдап алабыз.

Терезенин параметрлерин төмөнкүдөй тандап алабыз: width=300, height=100, scrollbars=yes, resizable=yes, toolbar=no, status=yes, menubar=yes, location=no, directories=no.



14.1-сүрөт.

15. DHTML. Стилдер жана катмарлар.

HTML-документтердин динамикалык (кыймылдуу) жана интерактивдүү (колдонуучу менен өз ара аракет этүү мүмкүнчүлүгүнө ээ) болушу аларга коюлган талаптардын негизгилери болуп саналат. Бул максатка жетүү үчүн каскад стилдеринин таблицасы (**CSS** - Cascading Style Sheets) иштеп чыгылган.

Каскад стилдеринин таблицасы деп Web-беттин ар түрдүү элементерине колдонулуучу жасалгалоо жана калыптоо эрежелеринин жыйындысын айтабыз.

Стандарттык HTML тилинде элементке тиешелүү касиетти (түсүн, өлчөмүн, беттеги абалын ж.б.) ыйгаруу үчүн бул касиетти ар жолу жазууга туура келет.

CSS технологиясында элементтердин касиеттеринин баяндамасы программалык коддо алдын ала жазылат. Андан кийин кайсыл элементтер үчүн жогорудагы баяндама стил катары кабыл алына тургандыгын көрсөтүп коюу жетиштүү болот. Стилдин баяндамасын ар түрдүү Web-беттерде колдонуу үчүн өзүнчө файлда сактаса да болот.

Стилдин баяндамалары <STYLE>...</STYLE> тегдеринин арасына жазылып, <HEAD>...</HEAD> тегдеринин арасына жайгаштырылат. Стилдин синтаксиси төмөнкүдөй:

```
<head><style type = "text/css">
Тег {1-атрибут: 1-маани; 2-атрибут: 2-маани; …; n-атрибут: n-маани}
</style></head>
```

Тегдин (элементтин) атрибуттары жана алардын тиешелеш маанилери ";" (үтүрлүү чекит) менен ажыратылып жазылат, ал эми атрибут менен маанилер ":" (кош чекит "=" белгисин алмаштырат) менен ажыратылып жазылат.

Мисалы, экинчи деңгээлдеги бөрктөрдүн бардыгын кызыл түстө жазуу үчүн стилдин баяндамасы төмөнкүдөй жазуу керек: H2 {color : red}

<STYLE> тегинен тышкары STYLE атрибуту бөрктөр тегинде (<H1>,<H2>,…,<H6>), <P> (абзац), <BODY> (тело), <DIV> (бөлүмдөр) жана <IMG> (сүрөттөлүштөр) тегдеринде да колдонулат.

15.1-мисал. CSS-технологиясын пайдаланып чычкандын көрсөткүчү менен позициялоодо сабактардын расписаниесин катмар катары экранга чыгаруу программасын түзгүлө.

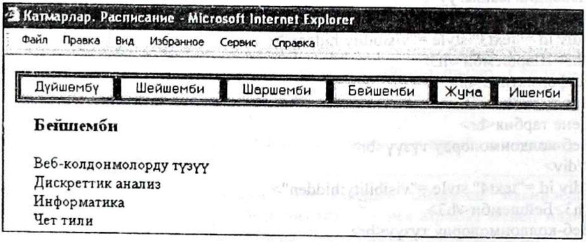
CSS-технологиясынын жардамында программаны төмөнкүдөй түзүүгө болот:

```
<html><head><title>Катмарлар. Расписание</title>
<style type ="text/css">
div{position:absolute;top=60;left:30}
</style>
<script language="javascript" >
var d = document
function rasp(n,m)
{if (m==1)
eval('d.all["text'+n+'"].style.visibility="visible"')
```

```
else
eval('d.all["text'+n+'"].style.visibility="hidden"')}
</script></head>
<body>
<div id ="text1" style ="visibility:hidden">
<h3>Дүйшөмбү</h3>
Алгебра<br>
Анализ<br>
Геометрия<br>
Веб-колдонмолорду түзүү
</div>
<div id ="text2" style ="visibility:hidden">
<h3>Шейшемби</h3>
Аскердик машыгуу
</div>
<div id ="text3" style ="visibility:hidden">
<h3>Шаршемби</h3>
Информатика<br>
Чет тили<br>
Дене тарбия<br>
Веб-колдонмолорду түзүү<br>
</div>
<div id ="text4" style ="visibility:hidden">
<h3>Бейшемби</h3>
Веб-колдонмолорду түзүү<br>
Дискреттик анализ<br>
Информатика<br>
Чет тили<br>
</div>
<div id ="text5" style ="visibility:hidden">
<h3>Жума</h3>
Дене тарбия<br>
Физика<br>
Математикалык логика<br>
Веб-колдонмолорду түзүү
</div>
<div id ="text6" style ="visibility:hidden">
<h3>Ишемби</h3>
Алгебра<br>
Анализ<br>
Геометрия<br>
Физика
</div>
<table border =1 bgcolor ="blue"><tr><td>
<input type =button value = "Дүйшөмбү" onmouseover ="rasp(1,1)" onmouse-
```

```
out="rasp(1,0)">
<input type =button value = "Шейшемби" onmouseover ="rasp(2,1)" onmouse-
out="rasp(2,0)">
<input type =button value = "Шаршемби" onmouseover ="rasp(3,1)" onmouse-
out="rasp(3,0)">
<input type =button value = "Бейшемби" onmouseover ="rasp(4,1)" onmouse-
out="rasp(4,0)">
<input type =button value = "Жума" onmouseover ="rasp(5,1)" onmouse-
out="rasp(5,0)">
<input type =button value = "Ишемби" onmouseover ="rasp(6,1)" onmouse-
out="rasp(6,0)">
</td></tr> </table>
</body></html>
```

Программанын аткарылышы 15.1-сүрөттө көрсөтүлдү.



15.1-сүрөт.

16. Сырткы файлдар менен иштөө.

Web-бетте башка файлдарда сакталган маалыматтарды колдонууга жана көрсетүүгө болот. HTML-документтердеги сценарийлерди компьютердин дисктерине сактоо үчүн да колдонсо болот.

16.1. Сырткы файлдардан берилгендерди киргизүү.

Тексттик файлды ачуу үчүн төмөнкү синтаксис колдонулат:

```
var fso = new ActiveXObject("Scripting.FileSystemObject");
var file = fso.OpenTextFile("жол", параметр);
```

Тексттик файл эки этап менен ачылат. Эң оболу fso өзгөрмөсү түзүлөт, андан кийин OpenTextFile методу менен дисктеги файл ачылып, жаңы file деген өзгөрмөгө ыйгарылат.

OpenTextFile методунда төмөнкүдөй аргументтер бар:

- жол – дисктеги файлга жетүүнүн жолу;
- параметр – файлды ачуу режимин көрсөтүүчү сандык маани:

* 0 – окуу үчүн гана (унчукпаганда);
* 2 – мурдагы маалыматты өчүрүү менен жазуу;
* 8 – файлдын акырына жаңы берилгендерди кошуу.

Файлдагы берилгендерди окуунун төмөндөгүдөй методдору бар:

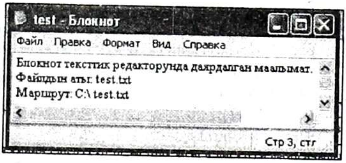
№	Метод	Аткарган кызматы
1	Read(n)	кийинки n символду окуйт
2	ReadAll(n)	файлдын акырына чейинки бардык текстти окуйт
3	ReadLine(n)	учурдагы жолчону окуйт
4	Skip()	кийинки n символду калтырып кетет
5	SkipLine()	учурдагы жолчону калтырып кетет

16.1–мисал. Блокнот тексттик редакторунда даярдалып, C:\ дискине test.txt аты менен сакталган файлдагы берилгендерди Web-бетке чыгаруучу программа түзгүлө.

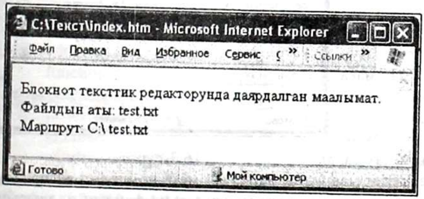
Мисалды чыгаруу үчүн файлды ачуу технологиясын пайдаланабыз. while(!f.AtEndOfStream) цикли файлдын акырына чыкканга чейин окууну камсыз кылат. ReadLine() методу ар бир жолчону окуп берет. Акырында Close() методу менен файлды жабабыз.

```
<html><script>
var fso = new ActiveXObject("Scripting.FileSystemObject");
var f = fso.OpenTextFile("c:\\test.txt");
while(!f.AtEndOfStream)
document.write(f.ReadLine()+"<br>");
f.Close();
</script></html>
```

16.1-сүрөттө Блокнот тексттик редакторунда даярдалган файлдын мазмуну көрсөтүлгөн.



16.1-сүрөт.



16.2-сүрөттө программа аткарылгандан кийинки Web-бетте алынган жыйынтык көрсөтүлгөн.

16.2-сүрөт.

16.2. Берилгендерди файлда сактоо.

Берилгендерди файлда сактоо үчүн эң оболу файлды ачуу керек. Бирок файлды ачканда ага жазууга мүмкүндүк берүүчү режимдердин бирин (мисалы 2 же 8-режимди) тандоо керек.

Берилгендерди файлга жазуу үчүн төмөнкү методдорду пайдаланабыз:

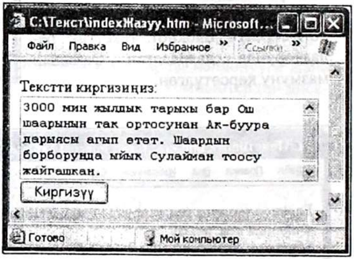
№	Метод	Аткарган кызматы
1	Write("текст")	Учурдагы позицияга киргизет
2	WriteBlankLines(n)	n бош жолчо киргизет
3	WriteLine("текст")	Текстти киргизет жана акырына жолчонун акырын билдирүүчү символду киргизет

Берилгендерди жазуу, берилгендерди окуу сыяктуу эле, бир багытта башынан акырына карай жүргүзүлөт.

16.2–мисал. Web-беттеги тексттик талаага киргизилген текстти C:\Текст\test.txt маршруту боюнча test.txt файлына сактагыла.

Мисалды чыгаруу программасы төмөнкүдөй болот.

```
<html>
<form name ="forma1" id ="forma1">Текстти киргизиңиз:<br>
<textarea name ="tta" id ="tta" rows ="5" cols ="25"></textarea><br>
<input type =button value ="Киргизүү" onclick ="text1()">
</form>
<script>
function text1()
{var fso = new ActiveXObject("Scripting.FileSystemObject");
var file1 = fso.OpenTextFile("C:\\Текст\\test.txt",8);
t=document.forma1.tta.innerText;
file1.WriteLine(t);
file1.Close();}
</script></html>
```



16.3-сүрөт.

Маселени чечүү үчүн text1() деген функция түзүп алабыз. Бул функция test.txt файлын ачып, анын акырына жаңы маалыматты жазуу үчүн ачып берет. Формага <textarea> элементин жана “Киргизүү” аттуу баскычты жайгаштырабыз. “Киргизүү” баскычын басканда text1() функциясы чакырылат да, ал 16.3-сүрөттө жазылган тексттин маанисин innerText методу менен t өзгөрмөсүнө ыйгарып, WriteLine() методунун жардамы менен test.txt файлынын акырына жазат.

16.3. Коноктор китебинин модели.

Web-беттеги маалыматтарды сырткы файлдарга жазуу жана алардан маалыматтарды окуу ыкмаларынан пайдаланып коноктор китебинин эң жөнөкөй моделин түзүүгө болот.

16.3-мисал. Коноктор китебине кирген колдонуучунун билдирүүсүн Web-беттин өзүнө чыгарууну камсыз кылуучу программа түзгүлө.

Маселени чечүү үчүн төмөнкүдөй киришебиз:

1). C:\ дискине MyWeb папкасын түзүп, анын ичине Конок.html Web-бетин түзөбүз. Бул бетке эң оболу форма, анын ичине колдонуучунун билдирүүсүн жазуу үчүн <textarea> элементин, билдирүүнү жөнөтүү, тазалоо жана өчүрүү баскычтарын орнотобуз.

2). Колдонуучунун билдирүүсүн файлга жазуу үчүн C:\ дискиндеги MyWeb папкасына алдын ала konok.txt файлын түзүп алабыз.

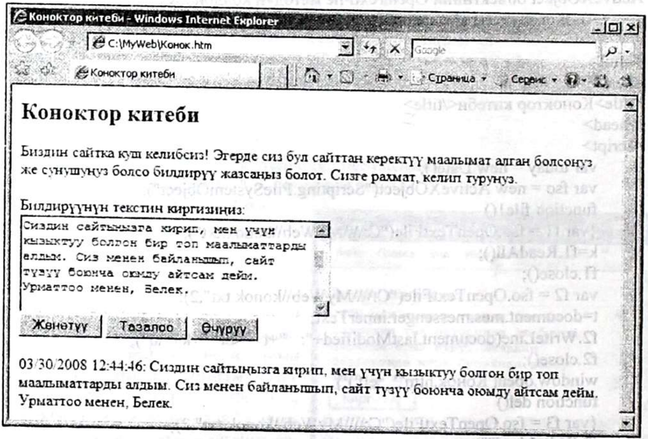
3). Маалыматтарды konok.txt файлынан окуу, ага жазуу жана өчүрүү үчүн ActiveXObject объектинин OpenTextFile методун колдонобуз.

4). Жогорудагы айтылгандардын негизинде программа түзөбүз.

```
<html>
<head>
<title>Коноктор китеби</title>
</head>
<script>
    var today = new Date();
    var fso = new ActiveXObject("Scripting.FileSystemObject");
    function file1()
    {var f1 = fso.OpenTextFile("C:\\\\MyWeb\\konok.txt");
    k=f1.ReadAll();
    f1.close();
    var f2 = fso.OpenTextFile("C:\\\\MyWeb\\konok.txt",2);
    t=document.mes.messenger.innerText;
    f2.WriteLine(document.lastModified+":   "+t+"<br>"+k+"\n");
    f2.close();
    window.open('Конок.htm','_self');}
    function del()
    {var f3 = fso.OpenTextFile("C:\\\\MyWeb\\konok.txt",2);
    f3.WriteLine("");
    f3.close();
    window.open('Конок.htm','_self');}
</script>
<body> <h2>Коноктор китеби</h2>
Биздин сайтка куш келибсиз! Эгерде сиз бул сайттан керектүү маалымат алган
болсоңуз <br> же сунушуңуз болсо билдирүү жазсаңыз болот. Сизге рахмат,
```

```
келип туруңуз.
<form name=mes method="POST">
        Билдирүүнүн текстин киргизиңиз:<br>
        <textarea rows="5" name="messenger" cols="35"></textarea><br>
        <input type="button" value="Жөнөтүү" name="B1" onclick="file1()">
        <input type="reset" value="Тазалоо" name="B2">
        <input type="button" value="Өчүрүү" name="B3" onclick="del()"></p>
</form>
<script>
        var f4 = fso.OpenTextFile("C:\\\\MyWeb\\konok.txt");
        while(!f4.AtEndOfStream)
        document.write(f4.ReadLine());
        f4.Close();
</script>
</body></html>
```

Программанын натыйжа 16.4-сүрөттө көрсөтүлдү.



16.4-сүрөт.

16.4. Web-бетти пароль менен коргоо

Web-бетти коргоо үчүн логин (ойлоп табылган колдонуучунун кыскача аты - псевдоним) жана пароль колдонулат.

16.4-мисал. Web-бетке кирүү үчүн логинди жана паролду катточу программа түзгүлө.

Маселени чечүү үчүн төмөнкүдөй киришебиз:

1). C:\ дискине MyWeb папкасын түзүп, анын ичине Пароль.html аттуу Web-бетин түзүп, жайгаштырабыз. Бул бетке эң оболу форма, анын ичине колдонуучунун билдирүүсүн жазуу үчүн <textarea> элементин, билдирүүнү жөнөтүү, тазалоо жана өчүрүү баскычтарын орнотобуз.

2). Колдонуучунун билдирүүсүн файлга жазуу үчүн C:\ дискиндеги MyWeb папкасына алдын ала konok.txt файлын түзүп алабыз.

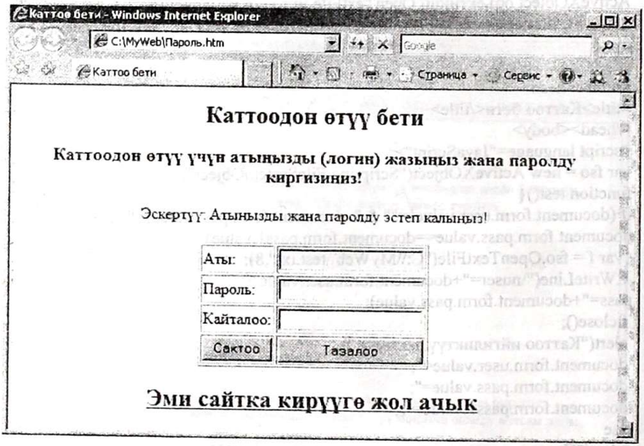
3). Маалыматтарды konok.txt файлынан окуу, ага жазуу жана өчүрүү үчүн ActiveXObject объектинин OpenTextFile методун колдонобуз.

4). Жогорудагы айтылгандардын негизинде төмөнкүдөй программа түзөбүз.

```
<html><head>
<title>Каттоо бети</title>
</head><body>
<script language="JavaScript">
var fso = new ActiveXObject("Scripting.FileSystemObject");
function test(){
if (document.form.user.value!="" && document.form.pass.value!="" &&
document.form.pass.value==document.form.pass1.value)
{var f = fso.OpenTextFile("C:\\MyWeb\\test.txt",8);
 f.WriteLine("\nuser="+document.form.user.value+"
pass="+document.form.pass.value);
 f.close();
 alert("Каттоо ийгиликтүү аяктады.")
 document.form.user.value=";
 document.form.pass.value=";
 document.form.pass1.value=";}
else
{alert("Пароль же колдонуучунун аты туура эмес!")}}
</script><br><br><br>
<h2 align ="center">Каттоодон өтүү бети</h2><br>
<h3 align =center >Каттоодон өтүү үчүн атыңызды (логин) жазыңыз жана паролду киргизиңиз!</h3>
<p align =center >Эскертүү: Атыңызды жана паролду эстеп калыңыз!</p>
<form name="form" method="POST" action="--WEBBOT-SELF--">
<table  align ="center" border="1" width="180">
```

```
<tr><td width="85">Аты:</td>
<td><input type="text" name="user" size="20"></td></tr>
<tr><td width="85">Пароль:</td>
<td><input type="password" name="pass" size="20"></td></tr>
<tr><td width="85">Кайталоо:</td>
<td><input type="password" name="pass1" size="20"></td></tr>
<tr><td width="85"><input type="button" value="Сактоо" onclick="test()"
name="B1"></td>
<td><input type="reset" value="Тазалоо" name="B2" style ="width:146;
height:25"></td></tr>
</table></form>
<a href ="index.htm" target ="_parent"><h2 align ="center">Эми сайтка кирүүгө
жол ачык</h2></a>
</body></html>
```

Программанын натыйжасы 16.5-сүрөттө көрсөтүлдү.



16.5-сүрөт.

50с

Колдонулган адабияттар

1. Коцюбинский А.О., Грошев С.В. Современный самоучитель работы в сети Интернет. Быстрый старт. – М.: ТРИУМФ, 2000. – 320 с.
2. Гончаров А. Самоучитель HTML. - СПб.: Питер, 2001. – 240 с.
3. Рева О.Н. Создание Web-страниц. Просто как дважды два. – М.: Эксмо, 2006. – 208 с.
4. Макфедрис П. Создание Web-страниц. - М.: АСТ: Астрель, 2005. – 387 с.
5. Microsoft FrontPage 2000. Шаг за шагом. Практ. пособ. – М.: ЭКОМ, 2001. – 328 с.
6. Рева О.Н. JavaScript. Просто как дважды два. – М.: Эксмо, 2007. – 256 с.
7. Дмитриева М.В. JavaScript. Экспресс-курс. – СПб.: БХВ-Петербург, 2004. – 336 с.
8. Дарнелл Р. JavaScript. Справочник. - СПб.: Питер, 1998. – 192 с.
9. Дунаев В. Самоучитель JavaScript. - СПб.: Питер, 2006. – 395 с.
10. Гудман Д. JavaScript. Библия пользователя. – М.: Вильямс, 2002. – 960 с
11. Днепров А. JavaScript на 100%. - СПб.: Питер, 2008. – 304 с.
12. Борисенко А.А. Dreamweaver 8. Просто как дважды два. – М.: Эксмо, 2007. – 224 с.

БИБЛИОТЕКА
Ошского государственного
университета
ИНВ № 50-007

Басууга берилди: 07.06.2008.

Формат: 60x84 1/16 Көлөмү: 4,25 б.т.
Буйрутма: №25 Нускасы: 400 даана.

ОшМУ, "Билим" редакциялык-басма бөлүмү
Ош шаары, Ленин к., 331, каб.135., тел.: 7.20.61

932048