

Министерство образования Республики Беларусь

Учреждение образования
«Гомельский государственный университет
имени Франциска Скорины»

Е.А. РУЖИЦКАЯ, И.В. САФОНОВ

СПЕЦИАЛИЗИРОВАННЫЕ ЯЗЫКИ РАЗМЕТКИ ДОКУМЕНТОВ (HTML)

*Тексты лекций по спецкурсу для студентов 3 курса
специальности 1–40 01 01 01 «Программное обеспечение
информационных технологий» специализации
«Компьютерные системы и Internet-технологии»*

Гомель 2006

УДК 004.4:004.7+681.3.06(075.8)

ББК 32.973.26–018.1я73

Р 837

Рецензенты:

В.Д.Левчук, доцент, кандидат технических наук;
кафедра вычислительной математики и программирования учре-
ждения образования «Гомельский государственный университет имени
Франциска Скорины»

Рекомендовано к изданию научно-методическим советом учре-
ждения образования «Гомельский государственный университет имени
Франциска Скорины».

Ружицкая, Е.А.

Р 837 Специализированные языки разметки документов (HTML):
[Текст] : [тексты лекций по спецкурсу «Специализированные
языки разметки документов (HTML)» для студентов 3 курса
специальности 1–40 01 01 01 «Программное обеспечение ин-
формационных технологий» специализации «Компьютерные
системы и Internet-технологии»] / Е.А.Ружицкая, И.В. Сафонов;
М-во обр. РБ, Гомельский государственный университет
им. Ф. Скорины. – Гомель: УО «ГГУ им. Ф.Скорины», 2006. –
87 с.

Тексты лекций содержит основные понятия теории сетей, основы
web-технологий и описание языка разметки гипертекста HTML, яв-
ляющегося основой создания web-сайтов. Тексты лекций составлены в
соответствии с учебной программой спецкурса «Специализированные
языки разметки документов» для студентов 3 курса специальности 1-40
01 01 01 – программное обеспечение информационных технологий спе-
циализации «Компьютерные системы и Internet-технологии».

УДК 004.4:004.7+681.3.06(075.8)

ББК 32.973.26–018.1я73

© Е.А. Ружицкая, И.В. Сафонов, 2006

© УО «ГГУ им. Ф. Скорины», 2006

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	5	7.4 Карты ссылок.....	58
1 ОСНОВЫ ТЕОРИИ СЕТЕЙ	6	8 МУЛЬТИМЕДИЯ НА WEB-СТРАНИЦЕ.....	62
1.1 История развития Интернета.....	6	8.1 Озвучивание web-страницы.....	61
1.2 Принципы работы Интернета.....	7	8.2 Внедрение видео.....	64
1.3 Сквозные протоколы и шлюзы.....	12	8.3 Использование элемента IMG по воспроизведению видео..	65
2 ТЕХНОЛОГИЯ WEB-ДИЗАЙНА.....	16	9 ФРЕЙМЫ.....	66
2.1 Основы web-технологий.....	16	9.1 Формат задания фреймовой структуры.....	66
2.2 Логическая и физическая структура сайта.....	20	9.2 Целевые фреймы и элементы привязки.....	69
3 ОСНОВЫ HTML И ФОРМАТИРОВАНИЕ ТЕКСТА.....	24	9.3 Базовый фрейм.....	69
3.1 Структура HTML документа.....	24	9.4 Плавающие фреймы.....	69
3.2 Цветовые спецификации.....	26	10 ФОРМЫ.....	74
3.3 Форматирование текста.....	26	10.1 Формат задания формы.....	74
3.4 Разделы, заголовки и горизонтальные линии.....	34	10.2 Элементы формы.....	76
4 СПИСКИ.....	36	11 МЕТА-ОПРЕДЕЛИТЕЛИ.....	85
4.1 Маркированный список.....	36	11.1 МЕТА-определители, использующие атрибут name.....	85
4.2 Нумерованный список.....	36	11.2 МЕТА-определители, использующие директиву	89
4.3 Вложенные списки.....	37	http-equiv.....	
4.4 Списки определений.....	38	ЛИТЕРАТУРА.....	91
5 ТАБЛИЦЫ.....	39		
5.1 Формат задания и общие параметры таблицы.....	39		
5.2 Параметры заголовка, строк и ячеек таблицы.....	42		
5.3 Группирование строк и столбцов таблицы.....	43		
5.4 Наследование свойств выравнивания.....	44		
6 ГИПЕРССЫЛКИ.....	49		
6.1 Ссылки на документы.....	49		
6.2 Ссылки на разделы.....	50		
6.3 Ссылки на адрес электронной почты.....	51		
6.4 Ссылки на файловые объекты.....	52		
7 ГРАФИКА В WEB-ДИЗАЙНЕ.....	53		
7.1 Сетевые графические форматы.....	53		
7.2 Включение графики в web-страницу.....	56		
7.3 Изображения ссылки.....	57		

ВВЕДЕНИЕ

Современный **Интернет** — сложная и высокотехнологичная система, позволяющая пользователю общаться с людьми, находящимися в любой точке земного шара, быстро и комфортно отыскивать любую необходимую информацию, публиковать для всеобщего сведения данные, которые он хотел бы сообщить всему миру.

Тексты лекций содержит основные понятия теории сетей, основы web-технологий и описание языка разметки гипертекста HTML, являющегося основой создания web-сайтов.

Тексты лекций составлены в соответствии с учебной программой спецкурса «Специализированные языки разметки документов» для студентов 3 курса специальности 1-40 01 01 01 «Программное обеспечение информационных технологий» специализации «Компьютерные системы и Internet-технологии». Тексты лекций ориентированы на самостоятельное изучение и предназначены для студентов математических специальностей университета, а также могут быть использованы студентами экономических и физических специальностей университета, изучающий Internet-технологии.

1 ОСНОВЫ ТЕОРИИ СЕТЕЙ

1.1 История развития Интернета.

1.2 Принципы работы Интернета.

1.3 Сквозные протоколы и шлюзы.

1.1 История развития Интернета. В конце 70-х годов XX века Министерство обороны США задалось идеей создать электронную сеть, с помощью которой компьютеры военных штабов и командных пунктов могли бы успешно обмениваться информацией в случае бомбардировки указанных объектов. Коммуникации, проложенные между бункерами, должны были выдержать наводнения, цунами, землетрясения, ураганы, прямые попадания метеоритов и другие неприятные погодные явления. Сеть была спроектирована, исходя из принципа малой надежности, то есть таким образом, что продолжала бы исправно перекачивать информацию между компьютерами, даже когда отдельные ее участки могли неожиданно исчезнуть. Исследования, проведенные в рамках этого проекта, были профинансированы Управлением перспективных научных разработок США (Advanced Research Project Agency, ARPA), и в 1968 году такая система была создана. В честь «спонсоров» ей дали название **ARPAnet**.

В ее проект были заложены несколько особенностей. Во-первых, все компьютеры, входящие в сеть, общались между собой «на равных», то есть в ARPAnet не существовало структуры «главный компьютер – подчиненный компьютер». Во-вторых, в качестве основного сетевого протокола ARPAnet был принят **межсетевой протокол IP**.

Сетевым протоколом называется согласованный и утвержденный стандарт, содержащий описание правил приема и передачи между двумя компьютерами команд, текста, графики, иных данных и служащий для синхронизации работы нескольких вычислительных машин в сети. Т.е. межсетевой протокол – это «свод законов для компьютеров», набор правил, позволяющий нескольким машинам обмениваться данными посредством сетевых коммуникаций.

Межсетевой протокол IP (Internet Protocol) является универсальным кросс-платформенным стандартом, позволяющим объединять в сеть разнородные вычислительные машины, работающие под управлением различных операционных систем. Именно протокол IP стал главным протоколом Всемирной сети Интернет.

В начале восьмидесятых годов Национальный научный фонд США (National Science Foundation) создал пять локальных сетей, соединив в единый комплекс их центральные компьютеры – **сетевые ра-**

бочие станции. Эти системы, как и ARPAnet, использовали протокол обмена данными IP. Согласно заложенной в данный проект идее планировалось объединить большинство американских исследовательских центров в глобальную информационную систему, создав своеобразную «сеть сетей» (Internetwork, сокращенно – Internet). Эта система должна была содержать самую свежую, постоянно обновляющуюся информацию о научных исследованиях изыскательских учреждений США. На этом протоколе Интернет работает до сих пор. К концу восьмидесятих годов совершенствование настольных персональных компьютеров и их удешевление привело к тому, что частные пользователи получили возможность осуществлять связь с Интернетом по коммутируемым телефонным каналам посредством **модемов** — устройств, преобразующих цифровой поток информации от компьютера в аналоговый звуковой сигнал и выдающих его в обычную телефонную линию. На другом конце модем принимающего компьютера трансформирует звуковой сигнал снова в цифровой. Каждый модем является как приемником, так и передатчиком информации.

Единица измерения скорости связи между двумя компьютерами **bps** (bit per second) определяется количеством бит передаваемой информации в секунду.

Трафиком называется общий суммарный поток информации через один сетевой компьютер.

Сетевой узел – это включенная в Интернет машина, которая объединяет несколько локальных сетей, использующих один сетевой протокол.

1.2 Принципы работы Интернета. Протокол IP позволяет только транслировать данные. Для того чтобы управлять этим процессом, служит протокол **TCP** (Transmission Control Protocol), опирающийся на возможности протокола IP. Он дробит информацию на несколько частей, присваивает каждой части номер, по которому данные впоследствии можно будет соединить воедино, добавляет к ней «служебную» информацию и укладывает все это в отдельный «IP-конверт». Далее этот «конверт» отправляется по Сети – ведь Интернет умеет обрабатывать IP-информацию. Поскольку в такой схеме протоколы TCP и IP тесно связаны, их часто объединяют в одно понятие: **TCP/IP**.

Одним из механизмов контроля за правильностью пересылки информации является метод, согласно которому в заголовок каждого передаваемого пакета записывается некая контрольная сумма, вычисленная компьютером-отправителем. Компьютер-получатель по аналогичной системе вычисляет контрольную сумму и сравнивает ее с числом,

имеющимся в заголовке пакета. Если цифры не совпадают, TCP пытается повторить передачу.

При отправке информационных пакетов протокол TCP требует от компьютера-получателя подтверждения приема информации. Это организуется путем создания временных задержек при приеме-передаче – тайм-аутов, или ожиданий. Тем временем отправитель продолжает пересылать данные. Образуется некий объем уже переданных, но еще не подтвержденных данных. То есть, TCP организует двунаправленный обмен информацией, что обеспечивает более высокую скорость ее трансляции.

При соединении двух компьютеров их модули TCP следят за состоянием связи. При этом само соединение, посредством которого осуществляется обмен данными, носит название **виртуального канала**.

Современная схема передачи данных в Интернете имеет многослойную структуру, включающую несколько уровней. Такая структура называется эталонной моделью **ISO OSI** (Open Systems Interconnection).

Передача данных с локальной машины на удаленный компьютер. Данные, попадая на компьютер, соединяющий со Всемирной сетью, передаются на другую машину, с которой соединен этот компьютер, и так – до первого узла. Далее определяется направление, приблизительно соответствующее тому, в котором находится конечный получатель, то есть маршрут информационного пакета. Пакет отправляется дальше до следующего узла, где снова определяется его дальнейший маршрут. Этот процесс называется **маршрутизацией**.

Для того чтобы пакет с информацией не «заблудился» по дороге, узлы Интернета, через которые он движется, имеют в своем распоряжении так называемые **таблицы маршрутизации** – электронные базы данных, в которых содержатся указания, куда именно отсылать тот или иной пакет информации, если он следует на такой-то адрес. Таблицы маршрутизации рассылаются на узлы централизованно, периодически меняются и дополняются. Серверы узлов, осуществляющие маршрутизацию, называются **маршрутизаторами**, или **роутерами** («router» – «маршрутизатор»). Правила маршрутизации описаны в протоколах ICMP (Internet Control Message Protocol), RIP (Routing Internet Protocol) и OSPF (Open Shortest Path First).

Система адресации в Интернете. IP-адрес состоит из четырех идентификаторов, или октетов, по одному байту каждый, разделенных точкой. Левый октет указывает тип локальной интрасети, в которой находится искомый компьютер. В рамках данного стандарта различается несколько подвидов интрасетей, определяемых значением первого октета. Это значение характеризует максимально возможное количест-

во подсетей и узлов, которые может включать такая сеть. В табл. 1.1 приведено соответствие классов сетей значению первого октета IP-адреса.

Таблица 1.1

Соответствие классов сетей значению первого октета IP-адреса

Класс сети	Диапазон значений первого октета	Возможное количество подсетей	Возможное количество узлов
A	1-126	126	16777214
B	128-191	16382	65534
C	192-223	2097150	254
D	224-239	–	2-28
E	240-247	–	2-27

Адреса класса А используются в крупных сетях общего пользования, поскольку позволяют создавать системы с большим количеством узлов. Адреса класса В применяют в корпоративных сетях средних размеров, адреса класса С – в локальных сетях небольших предприятий. Для обращения к группам машин предназначены адреса класса D, адреса класса Е пока не используются. Значение первого октета 127 зарезервировано для служебных целей, в основном для тестирования сетевого оборудования, поскольку IP-пакеты, направленные на такой адрес, не передаются в сеть, а ретранслируются обратно управляющей надстройке сетевого программного обеспечения как только что принятые. Кроме того, существует набор так называемых «выделенных» IP-адресов, имеющих особое значение. Эти адреса приведены в табл. 1.2.

Таблица 1.2

Значение выделенных IP-адресов

IP-адрес	Значение
0.0.0.0	Данный хост
Номер сети .0.0.0	Данная IP-сеть
0.0.0.номер хоста	Конкретный компьютер в данной локальной сети
1.1.1.1	Все компьютеры в данной локальной сети
Номер сети.1.1.1	Все компьютеры в указанной IP-сети

Хостом принято называть любой подключенный к Интернету компьютер независимо от его назначения, имеющий уникальный IP-адрес. **Виртуальный хост** – это хост, не имеющий уникального IP-адреса.

Последний (правый) идентификатор IP-адреса обозначает номер хоста в данной локальной сети. Все, что расположено между правым и левым октетами в такой записи, – номера подсетей более низкого уровня. Положим, мы имеем некий адрес в Интернете, на который хотим

отправить пакет. В качестве примера возьмем IP-адрес – 195.85.102.14. Итак, мы отправляем пакет в 195-ю подсеть сети Интернет, которая, как видно из значения первого октета, относится к классу С. Допустим, 195-я сеть включает в себя еще 902 подсети, но наш пакет высылается в 85-ю. Она содержит 250 более мелких сетей, но нам нужна 102-я. И, наконец, к 102-й сети подключено 40 компьютеров. Исходя из рассматриваемого нами адреса, пакет получит машина, имеющая в этой сетевой системе номер 14.

DNS — доменная система имен. Domain Name System (DNS) «доменная система имен» позволяет значительно облегчить пользователям процесс работы в Интернете тем, что им уже не нужно запоминать цифровые адреса хостов, с которыми общаются их компьютеры. Адрес того или иного ресурса Всемирной сети, записанный в стандарте DNS, дробится на несколько составляющих, отделенных друг от друга точкой. Эти элементы носят название «**доменов**»: myhost.mydomain.gsu.by Такое обозначение принято называть **URL (Uniform Resource Locator)** «универсальный определитель местонахождения ресурса». Соответственно, строка вызова размещенной по этому адресу web-страницы будет выглядеть как <http://www.myhost.mydomain.gsu.by>, где http обозначает протокол передачи гипертекстового документа (Hyper Text Transfer Protocol), а www (World Wide Web – Всемирная паутина) – есть указание на то, что передаваемые данные являются стандартным сервисом Интернета, то есть web-страницами. В большинстве случаев элемент www в записи URL можно опустить.

Домен – это логический уровень Интернета, то есть группа сетевых ресурсов, имеющая собственное имя и управляемая своей сетевой станцией.

Основная составляющая адреса DNS – это так называемый «домен первого уровня», охватывающий, как правило, некую глобальную географическую зону, например территорию отдельного государства.

Доменам второго уровня (локальная сеть банка, университета, городская муниципальная служба или отдельный сервер, предоставляющий пользователям доступ к какому-либо ресурсу) назначается произвольное имя. Домены третьего уровня являются составляющей частью домена второго уровня.

Рассмотрим передачу информационного пакета по адресу <http://www.myhost.mydomain.gsu.by>. Электронный пакет информации, поступающий в домен первого уровня .by, белорусскую зону Интернета, управляемую своей сетевой станцией, перебрасывается в одну из входящих в неё зон – домен второго уровня .gsu. С этого момента домену

.bu становится глубоко безразлична дальнейшая судьба пакета, его последующей маршрутизацией занимается теперь домен .gsu. Домен .gsu, избавившись от пакета данных путем передачи его во входящую в него локальную сеть mydomain получает на свою кремневую голову следующую проблему: отыскать в подотчетной ему сети машину, на которой расположен web-сервер с именем myhost, чтобы передать этот пакет ей и больше никогда о нем не вспоминать. Получив информацию, компьютер, управляющий доменом четвертого уровня myhost, просто отправляет его соответствующей серверной программе.

Искать нужный компьютер в Интернете пользовательским машинам помогают *DNS-серверы* – программы, которые при обращении к ним выискивают нужный IP-адрес по введенному URL.

Электронная почта. Электронная почта (Electronic Mail, или E-Mail) – одно из самых распространенных средств общения между людьми. Протоколы, используемые при работе с электронной почтой, называются **SMTP** и **POP3**, первый «отвечает» за отправку исходящей корреспонденции, второй – за доставку входящей.

Входящая почта записывается в «электронный почтовый ящик» – как правило, это просто каталог на винчестере почтового сервера, доступ к которому имеет только абонент этого ящика. Обратившись к серверу и забрав почту, абонент может прочитать ее, уже разорвав соединение с Интернетом. Однако следует помнить, что после того, как почта была извлечена из почтового ящика, на сервере она не сохраняется.

Адрес электронного почтового ящика имеет следующий вид: Имя_абонента@адрес_почтового_сервера. Знак @ («ат», или «собака») служит для того, чтобы компьютер мог отделить адрес сервера, на который следует отправлять письмо, от имени абонента, вернее, его личного каталога, куда это письмо запишет по получению сам сервер.

В настоящее время получил широкое распространение еще один почтовый протокол, называется **IMAP**. Он предоставляет пользователю богатые возможности для работы с почтовыми ящиками, находящимися на центральном сервере. Почтовая программа, использующая этот протокол, получает доступ к хранилищу корреспонденции на сервере так, как будто эта корреспонденция расположена на компьютере получателя. Электронными письмами можно манипулировать с компьютера пользователя (клиента) без необходимости постоянной пересылки с сервера и обратно файлов с полным содержанием писем.

IMAP был разработан для замены более простого протокола POP и имеет следующие преимущества по сравнению с последним:

- Письма хранятся на сервере, а не на клиенте. Возможен доступ к одному и тому же почтовому ящику с разных клиентов. Поддер-

живается также *одновременный* доступ нескольких клиентов. В протоколе есть механизмы с помощью которых клиент может быть проинформирован об изменениях, сделанных другими клиентами.

- Поддержка нескольких почтовых ящиков (или папок). Клиент может создавать, удалять и переименовывать почтовые ящики на сервере, перемещать письма из одного почтового ящика в другой.
- Возможно создание общих папок, к которым могут иметь доступ несколько пользователей.
- Информация о состоянии писем хранится на сервере и доступна всем клиентам. Письма могут быть помечены как прочитанные, важные и т. п.
- Поддержка поиска на сервере. Нет необходимости скачивать с сервера множество сообщений для того чтобы найти одно нужное.
- Поддержка Online-работы. Клиент может поддерживать с сервером постоянное соединение, при этом сервер в реальном времени информирует клиента об изменениях в почтовых ящиках, в том числе о новых письмах.
- Предусмотрен механизм расширения возможностей протокола.

1.3 Сквозные протоколы и шлюзы. Основным протоколом сети Интернет является протокол IP. Однако пользователи локальных систем, не использующих IP, тоже могут подключиться к Интернету. Для этого существуют *шлюзы*.

Шлюз – это программа, при помощи которой можно передавать информацию между двумя сетевыми системами, использующими различные протоколы обмена данными.

Посредством шлюзов можно передавать любую информацию – и графику, и гипертекст, и музыку, и даже видео. Информация, пересылаемая через такие сети другим сетевым системам, транслируется с помощью *сквозного протокола*, обеспечивающего беспрепятственное прохождение IP-пакетов через не IP-сеть.

Протокол передачи данных UDP. Протокол передачи данных **UDP** (User Datagram Protocol) используется на медленных линиях для трансляции данных как *дейтограмм*.

Дейтограмма – это пакет, передаваемый через сеть независимо от других пакетов, без образования виртуального канала и подтверждения приема. Дейтограмма содержит полный комплекс данных, необходимых для ее отсылки и получения. При передаче дейтограмм компьютеры не занимаются обеспечением стабильности связи, поэтому следует принимать особые меры для обеспечения надежности.

Схема обработки информации протоколом UDP в принципе такая

же, как и в случае TCP, но с одним отличием: UDP всегда дробит информацию по одному и тому же алгоритму, строго определенным образом. Для осуществления связи с использованием протокола UDP применяется система отклика: получив UDP-пакет, компьютер отправляет отправителю заранее обусловленный сигнал. Если отправитель ожидает сигнала слишком долго, он просто повторяет передачу. Интернет-программы работают с UDP в два раза быстрее, чем с TCP.

Протокол передачи файлов FTP. Протокол передачи файлов **FTP** (File Transfer Protocol) предназначен для передачи файлов через Интернет. Именно на базе этого протокола реализованы процедуры **даунлоада** и **аплоада** данных во Всемирной сети.

Даунлоад (Download) – это процесс копирования файлов с удаленного сетевого компьютера (сервера) на вызывающий его компьютер (клиент). Но общепринятым техническим жаргоном эту процедуру называют иногда «скачиванием». **Аплоад** (Upload) – обратный процесс, загрузка файлов на удаленный компьютер.

FTP позволяет переносить с машины на машину не только файлы, но и целые директории, включающие поддиректории на любую глубину вложений. Осуществляется это путем обращения к системе команд FTP, описывающих ряд встроенных функций данного протокола.

Доступ к Интернету. Непосредственный (прямой) **доступ** к Интернету подразумевает использование самого полного спектра услуг Глобальной сети. Непосредственный доступ – это и есть фактически прямое включение локальной сети вашего предприятия в состав Интернета в качестве домена. Непосредственное соединение осуществляется методом подключения локальной сети или рабочей станции к оптоволоконной магистрали или выделенному каналу связи при помощи соответствующего сетевого оборудования.

Доступ по протоколам SLIP и PPP. Для использования такого вида доступа необходимо приобрести специальное программное обеспечение, поддерживающее протоколы **SLIP** (Serial Line Internet Protocol) и **PPP** (Point to Point Protocol). Договорившись с поставщиком услуг о том, что вы будете работать в Интернете именно по этим протоколам, вы подключаетесь к Сети. После этого, как и при непосредственном доступе, в вашем распоряжении абсолютно все ресурсы Глобальной сети, более того, вы входите в нее составной частью сами, а не через сервер провайдера Интернета, позволившего вам пользоваться его ресурсами (как это происходит в случае доступа по вызову).

Провайдер – это организация, обладающая специальной лицензией и предоставляющая за определенную плату частным лицам и организациям доступ в Интернет с помощью ресурсов собственного сервера

ра или корпоративной локальной сети.

При работе со SLIP и PPP вам не надо прокладывать в офис оптоволоконную магистраль, можно использовать выделенную коммутируемую телефонную линию, однако на компьютере должен быть установлен высокоскоростной модем.

SLIP – это сетевой протокол, позволяющий использовать для прямого выхода в Интернет обыкновенную телефонную линию и модем. Он требует специального программного обеспечения и работает совместно с протоколом TCP/IP как протокол более низкого уровня.

Принимаемую через модем кодированную информацию SLIP обрабатывает и преобразует в IP-пакет, который передает программной оболочке более высокого уровня, например приложению класса клиент/сервер, которое, получив его, преобразует этот пакет соответственно директивам протокола TCP. В обратную сторону данная система работает так: SLIP получает от программы клиент/сервер генерированный ею IP-пакет, извлекает из него всю ценную информацию, трансформирует ее в символьный код и отправляет в Интернет.

PPP – более поздняя и совершенная версия протокола, аналогичного SLIP. Он значительно эффективнее, быстрее и обладает большим количеством полезных функций. Различий в алгоритмах обработки информации между ними практически нет. SLIP и PPP подходят для прямого подключения к Интернету как локальных сетей, так и домашних компьютеров, используемых в качестве сервера.

Фирмой Lawrence Berkley Labs (а точнее ее программистом Ван Якобсоном) был создан протокол **CSLIP** (Compressed SLIP), который является модификацией SLIP и выигрывает в скорости благодаря универсальному алгоритму сжатия заголовков IP-пакетов. На низкоскоростных линиях этот протокол позволяет увеличить эффективность связи почти в 6 раз и немного компенсировать ее слабые возможности. То есть CSLIP – это вариант SLIP для медленных телефонных линий. Однако CSLIP гораздо менее надежен, чем его прототип. Он сжимает заголовки IP-пакетов, начиная со второго и используя при этом информацию о заголовке из первого, несжатого пакета. Образуется логическая последовательность IP-информации, и если одна из ее составляющих теряется, то есть один из пакетов последовательности был передан некорректно, CSLIP прерывает передачу и добросовестно начинает все заново. При плохой связи можно потерять на использовании CSLIP больше, чем при применении SLIP и PPP, поскольку этот протокол способен гробить на повторные пересылки данных уйму времени.

Доступ UUCP (Unix-to-Unix CoPy). Этот протокол создан для компьютеров, использующих UNIX-подобные операционные системы.

UUCP предназначен для обмена файлами между двумя компьютерами. С помощью данного вида доступа можно получать и отправлять электронную почту и, как разновидность последней, – новости с соответствующих серверов. Больше UUCP не способен ни на что. Суть подключения UUCP сводится к тому, что на самом деле никакого Интернета на вашей машине не будет. Интернет будет на той машине, владелец которой предоставил вам UUCP, а ваш компьютер станет просто соединяться в условленное время с его персоналкой, отправлять и принимать электронную почту. Дальше уже сетевые программы удаленного ПК будут пересылать вашу корреспонденцию во Всемирную сеть.

Доступ по вызову (Dial-up Access) – связь с поставщиком Интернет-услуг по коммутируемым телефонным каналам при помощи модема. Этот вид доступа предоставляется частным лицам и организациям провайдером услуг Интернета. Среди наиболее очевидных «минусов» соединения dial-up можно перечислить низкое качество и скорость канала передачи данных, высокую вероятность разрыва связи в самый неподходящий момент, длительное время, затрачиваемое пользователем на дозвон до провайдерского пула и авторизацию, и занятость телефонной линии на протяжении всего сеанса работы в Интернете.

Доступ по технологии «соax at a home». Эта технология подразумевает получение доступа к Интернету с использованием каналов кабельной телевизионной сети. Такая структура выглядит следующим образом: стандартное оборудование кабельного телевизионного центра, используемое для вещания, через специальное устройство передачи данных, называемое головным модемом, а затем через маршрутизатор подключается к высокоскоростному каналу Интернета. После этого абоненту достаточно установить на своем компьютере любую сетевую карту, поддерживающую протокол 10 Base-T, подключить ее к кабельному модему, а модем – к расположенному в квартире антенному выходу. Кабельный модем – устройство, которое представляет собой двунаправленный аналогово-цифровой преобразователь данных, использующий в процессе передачи информации принцип наложения на несущую частоту модулированного аналогового сигнала. Кабельный модем не требует установки каких-либо драйверов, он подключается к компьютеру посредством сетевой карты и является абсолютно прозрачным для системы: машина считает, что она работает в локальной сети. Преимуществом доступа к Интернету по методу «соax at a home» является высокая стабильность соединения, отсутствие непредвиденных разрывов связи, а также то, что на протяжении всего сеанса работы во Всемирной сети телефонная линия остается свободной. Однако данный метод связи пока не получил широкого распространения.

2 ТЕХНОЛОГИЯ WEB-ДИЗАЙНА

2.1 Основы web-технологий.

2.2 Логическая и физическая структура сайта.

2.1 Основы web-технологий.

HTML, Hyper Text Markup Language, «язык разметки гипертекста», язык, используемый для создания документов в Интернете. Он позволяет формировать на странице сайта текстовые блоки, включать в них изображения, организовывать таблицы, управлять отображением цвета документа и текста, добавлять в дизайн сайта звуковое сопровождение, организовывать гиперссылки с контекстным переходом в другие разделы сервера или обращаться к иным ресурсам Сети и компоновать все эти элементы между собой. Файлы, содержащие гипертекстовый код, имеют расширение .htm или .html.

HTML-документом называется файл, содержащий код HTML.

Одной из основных функциональных особенностей, характерных именно для данного языка, благодаря которой он и получил свое название, является гиперссылка.

Гиперссылка (Hyperlink) – базовый функциональный элемент html-документа, представляющий собой реализацию динамической связи какого-либо объекта данной web-страницы с контекстным содержимым другого документа.

HTML – язык не транслируемый, но интерпретируемый. Это означает, что для исполнения готового кода его не нужно компилировать, встроенный в специальное программное обеспечение, предназначенное для просмотра web-страниц, интерпретатор «компилирует» код непосредственно в процессе открытия документа. При этом, обнаружив в тексте ошибку, такая программа не выдает соответствующего предупреждения (если страница не содержит встроенных скриптов Java), а попросту игнорирует всю «ошибочную» строку.

Броузер – специальная клиентская программа, предназначенная для просмотра содержимого web-узлов и отображения документов HTML. Броузеры содержат встроенный транслятор языка разметки гипертекста, компилирующий html-код в процессе открытия web-страницы.

Существует несколько классов браузеров, различающихся в основном спектром реализуемых ими возможностей. Два основных класса – это браузеры, поддерживающие и не поддерживающие отображение графических элементов web-страниц. Большинство современных браузеров относятся к первому классу. Наиболее распространены сре-

ди пользователей Интернета браузеры Mozilla Firefox, Opera, Internet Explorer, Maxthon – по сути программная оболочка для Internet Explorer, Safari – основан на коде Konqueror.

Сервер – это компьютер с установленным на нем специальным программным обеспечением (специальная программа тоже называется сервером, web-сервером или http-сервером), которое отображает web-страницы по запросу клиентской машины, а также выполняет множество других полезных функций и имеющий собственное доменное имя, то есть адрес DNS, отвечающий стандартам Domain Name System. Когда компьютер связывается сервером и получает от него все необходимые данные, например код web-страницы, он выступает в роли «клиента», а всю систему в этом случае принято называть связкой «клиент—сервер».

Системой «клиент-сервер» называют механизм передачи информации между удаленным компьютером, предоставляющим свои ресурсы в распоряжение пользователей, и пользовательским компьютером, эксплуатирующим эти ресурсы.

Сервер должен представлять собой «информационный портал», то есть он является достаточно большим виртуальным пространством, состоящим из множества различных тематических разделов меньшего размера, либо некоторого количества самостоятельных проектов.

Порт – это число, которое определяет программу, желающую принимать данные из Интернета. Любая программа, стремящаяся передать данные другой, должна знать номер порта, который закреплен за ней. Web-серверу выделяется порт с номером 80.

Сайт (site, участок) – это участок сервера, то есть раздел, полностью посвященный какой-либо теме.

Анатомия сервера. Для загрузки web-страницы в клиентский браузер, он посылает установленной на серверном компьютере специальной программе, называемой http-сервером, соответствующий запрос и обрабатывает полученные от него данные. Сервер принимает запрос, ищет запрошенный документ и выдает клиенту либо содержимое найденного файла, либо сообщение об ошибке, если такой файл не был найден или доступ к нему почему-либо запрещен. http-сервер не анализирует содержимое передаваемого документа, он только передает его браузеру, а всю работу по структурированию и отображению полученной информации тот уже берет на себя. Поиск запрашиваемой страницы осуществляется в определенной директории, которая отведена на серверном компьютере под данный сайт – ссылка на эту директорию присутствует во введенном пользователем адресе. В случае когда обращение осуществляется не к конкретному документу, а к сайту в це-

лом, http-сервер автоматически подставляет вместо названия передаваемого файла так называемую «стартовую страницу», которая носит имя index.htm или index.html (в некоторых случаях — default.htm или default.html). Этот документ обязательно должен располагаться в корневом каталоге, в директории с названием WWW. Все остальные файлы можно размещать либо в этом же каталоге, либо во вложенных директориях.

Серверная директория содержит обычно еще несколько каталогов. Во-первых, это папка CGI-BIN, где размещаются CGI-скрипты и другие запускаемые с сайта интерактивные приложения, а также несколько служебных директорий, необходимых для нормальной работы сервера. Иногда в том же каталоге, где хранится index.html, присутствует ряд дополнительных файлов: документ, который отображается в случае, если http-сервер не смог найти запрашиваемый пользователем файл; документ, который отображается в качестве сообщения об ошибке, если доступ к запрашиваемому документу запрещен; и файл, в котором специальным образом описываются правила индексации сайта поисковыми машинами.

Кодировки кириллицы. **Кодировкой кириллицы** называется общепринятый стандарт, включающий набор символов, который позволяет отображать в документах HTML знаки русского алфавита. Существует пять основных кодировок кириллицы: ISO, KOI8-R, DOS, Windows 1251 и MAC и ряд менее распространенных кодировок, способных воспроизводить символы кириллицы, они называются Unicode (UTF-8) и транслит.

Кодировка KOI8 (код обмена информации восьмибитный), соответствующая стандарту ГОСТ 19768-74. KOI8 является основным стандартом для серверов, работающих на базе платформы UNIX (например, для http-сервера Apache), а также используется в качестве «формата по умолчанию» при пересылке сообщений электронной почты на русском языке.

Компания Microsoft предложила стандарт **Microsoft code page 1251 (Windows 1251)**, получивший широкое распространение благодаря популярности операционной системы Microsoft Windows и http-сервера Internet Information Server, входящего в комплект поставки Windows NT. Именно поэтому и Windows 1251, и KOI8-R входят в тот минимально допустимый набор кодировок, которые должна обязательно поддерживать любая web-страница.

Стандарт **Microsoft/IBM code page 866 (альтернативная кодировка DOS)** служит базовой кодировкой в операционных системах MS-

DOS и OS/2, данный стандарт принят по умолчанию в некоммерческой сети FidoNet.

Кодировка *ISO-8859-5* была разработана Комитетом по международным стандартам (International Standards Organization, ISO) и применяется в основном в UNIX-совместимых операционных системах. ISO не получила широкого распространения.

Macintosh CP (MAC) предназначена для использования на компьютерах Apple Macintosh, оснащенных операционной системой MacOS.

Универсальный международный стандарт *Unicode (UTF-8)* был создан с целью объединить все существующие кодировки в одну. Для отображения одного символа в Unicode отводится не один, а 2 байта, поэтому данная кодировка включает в себя 65 536 знаков вместо 256. В это число входят не только буквы всех алфавитов мира как существующих ныне или созданных искусственно, так и уже практически вышедших, но и множество специальных символов – математических, музыкальных, физических.

Для операционных систем, вообще не поддерживающих кириллицу, существует *транслит* который подразумевает запись русских слов символами латинского алфавита. Серверные программы обладают встроенной функцией автоматического определения кодировки.

Web-хостинг. Web-хостингом, или просто *хостингом*, называется размещение web-страниц в сети Интернет на заранее арендованном дисковом пространстве какого-либо сервера. При этом под хостингом подразумевается не публикация web-страницы, а именно аренда такого дискового пространства.

Баннером называют рекламное изображение фиксированного размера, как правило, содержащее анимацию, которое выполняет роль гиперссылки на тот или иной ресурс Интернета.

WWW-псевдонимом называют короткий адрес URL, назначаемый специальными серверами странице или сайту с более длинным адресом того же стандарта.

Основные «постулаты» web-дизайна. Web-сайт подразумевает наличие набора определенных «правил хорошего тона», которых следует придерживаться, чтобы сайт выглядел профессионально:

1. Сайт должен корректно отображаться при различных экранных разрешениях. Графические элементы должны быть в формате GIF или PNG и лишь самые необходимые изображения – в формате JPEG.

2. Web-страница должна идентично отображаться в различных браузерах.

3. Все страницы web-сайта, а также все интегрированные в них графические и интерактивные элементы должны быть минимальными по объему.

4. Web-страница должна обязательно включать навигационные элементы, охватывающие все разделы сайта, причем эти элементы должны всегда быть на виду.

5. Весь проект должен быть выдержан в одном дизайнерском стиле.

6. На одной web-странице не должно быть более трех различных шрифтов.

7. Необходимо использовать только корректные цветовые схемы и не применяйте при оформлении документов более трех различных цветов. Исключение здесь можно сделать только для полутонов одного и того же цвета. Корректность сочетания друг с другом цветов и оттенков проверяется с помощью следующего правила: необходимо перевести изображение в формат «grayscale» (256 оттенков серого). При удачном подборе цвета, все элемента должны выглядеть контрастно. В любом случае для текста рекомендуется выбирать темный цвет, в качестве фона лучше всего использовать тусклую, едва различимую заливку произвольного оттенка.

2.2 Логическая и физическая структура сайта. Каждый ресурс Интернета содержит несколько тематических рубрик, соединенных между собой гиперсвязями. Как правило, ссылки на все разделы сайта с краткими анонсами их содержимого приводятся на первой, так называемой стартовой странице, которой присваивается имя index.htm (.html). Если тематические рубрики содержат собственные подразделы, каждая из них также имеет свою стартовую страницу, называющуюся index.html.

Подобный набор тематических рубрик с распределенными по соответствующим разделам документами и заранее спроектированными гиперсвязями между всеми страницами ресурса и называется *логической структурой сайта. Физическая структура* подразумевает алгоритм размещения физических файлов по поддиректориям папки, в которой опубликован ваш сайт.

Рекомендуется размещать все графические изображения, являющиеся элементами проекта, в отдельной папке с названием «Images», расположенной в корневой директории сайта. Такой подход позволит обновлять хранящиеся в других тематических разделах документы HTML без переноса графики, использовать одни и те же графические файлы во всех разделах сайта и при необходимости удалять целые директории. Все имена файлов должны быть смысловыми.

Заглавная страница. Заглавная страница представляет собой html-документ, который не включает в себя какую-либо содержательную информацию и элементы навигации. Файлу заглавной страницы присваивается имя index.html, при этом стартовая страница вызывается посредством организации гиперссылки с заглавной страницы, загружающейся при обращении к сайту первой. Заглавная страница содержит, как правило:

- логотип компании-владельца данного ресурса,
- счетчик посещений,
- предложение выбора кодировки кириллицы,
- выбор между английской и русской версиями сайта.

При обращении к сайтам, не оснащенным заглавной страницей, первой отображается стартовый документ, включающий какое-либо информационное наполнение, элементы навигации и иногда анонсы составляющих данный ресурс тематических рубрик.

Динамическая и статическая компоновка сайта. Современные видеокарты поддерживают несколько произвольно устанавливаемых видеорежимов, характеризующихся экранным разрешением и количеством цветов, используемых при отображении информации на экране компьютера. При открытии в одном и том же браузере какой-либо web-страницы она может отображаться не одинаковым образом в зависимости от используемого экранного разрешения.

Для того чтобы избежать «съезжания» элементов html-документа друг относительно друга и, как следствие, деформации web-страницы в целом при изменении параметров экрана, все компоненты web-страницы заключаются в соответствующие ячейки невидимой таблицы, при этом каждому объекту назначается одно, строго определенное положение. Данной таблице можно назначить строго определенную ширину в пикселах, например, 640 точек, после чего жестко позиционировать ее по центру экрана или «прижать» к левому его краю. Такой вариант компоновки сайта называется **статическим**, поскольку ширина таблицы не меняется в зависимости от экранного разрешения. При изменении параметров экрана не происходит ни малейшего смещения элементов дизайна страницы.

Другой подход – когда ширину невидимой таблицы, содержащей фрагменты web-страницы, задают в процентах от текущей ширины экрана. При увеличении экранного разрешения таблица «растягивается» по горизонтали, и все размещенные в ее ячейках элементы, позиционированные либо по центру, либо по краям столбцов, смещаются согласно установленному алгоритму. В силу того, что параметры таблицы изменяются в зависимости от настроек экрана, такой принцип компо-

новки html-документа можно называется динамическим. И тот и другой подход обладает как достоинствами, так и недостатками.

Статическая компоновка страницы.

Достоинства. Простота алгоритма верстки документа. Оптимизировав страницу для отображения с экраным разрешением 640x480 точек, можно быть уверенным, что при изменении пользовательских экранных настроек элементы дизайна не «поплывут».

Недостатки. При отображении документа на компьютере с высоким экраным разрешением по краям экрана или с правой его стороны остается заметное пустое поле.

Динамическая компоновка страницы.

Достоинства. Документ растягивается по всей ширине экрана, не остается неиспользованных пустых полей.

Недостатки. Сложность верстки и отладки страницы

Элементы web-страницы. Любая web-страница содержит определенный набор стандартных элементов, являющихся обязательными компонентами каждого ресурса Интернета:

1. **Заголовок web-страницы.** Он может быть выполнен как в текстовом, так и в графическом варианте, и должен располагаться в верхней части документа. Иногда с заголовком совмещают меню выбора кодировки кириллицы и кнопки для перехода с русскоязычной на англоязычную версию сайта.

2. Непосредственно под заголовком документа, как правило, располагается **рекламный баннер**. Стандартный размер баннеров, публикуемых под заголовком документа, составляет обычно 468x60 точек.

3. **Логотип** компании-владельца данного сайта или ссылку на сервер, осуществляющий web-хостинг. Логотип необходим не всегда, как правило, он включается в состав web-страницы лишь в случае, если сайт имеет коммерческую направленность.

4. Основную часть документа занимает так называемое **текстовое поле** – участок где размещается смысловое наполнение страницы: содержательный информационный текст и иллюстрации. Перечисленные элементы еще называют «контент» (content – содержание).

5. **Элементы навигации** – гиперссылки, связывающие данный документ с другими разделами сайта. Элементы навигации могут быть выполнены в виде текстовых строк, графических объектов, то есть кнопок, либо активных компонентов, т.е. кнопок, реагирующих на движение, например, мыши, выполняя при наведении на них курсора какие-либо несложные действия (включение подсветки, создание эффекта «нажатия», изменение формы и т. д.). Располагать элементы навигации следует таким образом, чтобы они всегда были «на виду». Наиболее

устоявшимся подходом является размещение элементов навигации у левой границы страницы.

6. В нижней части документа принято публиковать *информацию о разработчиках сайта* и адрес электронной почты, по которому посетители ресурса могут направить владельцам странички свои отклики, предложения и пожелания.

7. Если Web-страница является стартовым документом, в нижней ее части также размещают *счетчик посещений*.

3 ОСНОВЫ HTML И ФОРМАТИРОВАНИЕ ТЕКСТА

3.1 Структура HTML документа.

3.2 Цветовые спецификации.

3.3 Форматирование текста.

3.4 Разделы, заголовки и горизонтальные линии.

3.1 Структура HTML документа. Директивы HTML называются тегами. Теги заключаются в угловые скобки. Все теги за исключением некоторых – парные. Закрывающий тег отличается от открывающего наличием символа /. Синтаксис записи тега:

```
<ТЕГ> обрабатываемое значение </ТЕГ>
```

Теги допускают вложение друг в друга. Все объекты не заключенные в угловые скобки, воспринимаются как текстовые элементы.

HTML не чувствителен к регистру, однако рекомендуется применять при написании тегов заглавный регистр для улучшения читаемости кода. Исключением являются адреса URL и escape-последовательности.

Тег – это команда html, указывающая интерпретатору браузера, каким образом он должен обрабатывать соответствующее каждой конкретной директиве значение. Это значение называется *атрибутом тега*. Тег может иметь атрибут или не иметь его. Тег верхнего уровня <HTML> не имеет атрибутов.

Синтаксис записи тега в совокупности с его атрибутом:

```
<ТЕГ имя_атрибута_1="значение"
```

```
...
```

```
имя_атрибута_n="значение">
```

Значения атрибутов заключаются в прямые кавычки <">. Если внутри атрибута тега встречается другое значение, заключенное в кавычки, то есть имеет место вложение одних кавычек в другие, в качестве внутренних кавычек рекомендуется использовать одинарные. Такое выражение выглядит следующим образом:

```
<ТЕГ имя_атрибута_1="значение_1;  
'значение2';  
значение3">
```

Спецификация языка позволяет опускать кавычки в следующих видах атрибутов:

- атрибуты, записываемые только строчными или заглавными символами латинского алфавита и не включающие иных символов, например цифр.
- атрибуты, состоящие только из цифр от 0 до 9.

- атрибуты, обозначающие промежутки времени.
Например, запись атрибутов border="1" или align="center" допустимо представить в виде border=1 или align=center.

Структура HTML документа

```
<HTML>
  <HEAD> Заголовок документа
    <TITLE>Внешний заголовок </TITLE>
  </HEAD>
  <BODY>
    Тело документа, содержащее основной код
  </BODY>
</HTML>
```

Комментарии. Комментарии открываются символьной последовательностью, начинающейся с открывающей угловой скобки, восклицательного знака и двух знаков тире <!-- закрываются сочетанием -->.

Формат комментария:

```
<!-- текст комментария -->
```

Дефинитивный определить. Согласно спецификациям HTML тег DOCTYPE (что означает "объявление типа документа" или дефинитивный определить) сообщает валидатору, какая версия HTML используется на странице. Этот тег должен всегда находиться в первой строке каждой страницы. Тег DOCTYPE – ключевой компонент web-страниц, претендующих на соответствие стандартам: без него код и CSS не пройдет проверку валидатором.

Тег DOCTYPE также важен для правильного отображения и работы страницы в браузерах, соответствующих стандартам (Mozilla, IE5/Mac, и IE6/Win).

Тег DOCTYPE, в атрибутах которого указывается полный URI (полный web-адрес), сообщает браузерам, что страницу нужно вывести с соблюдением определенного стандарта или подвида этого стандарта.

Форматы тега DOCTYPE для стандартов HTML 4.01 Strict (строгий), Transitional (переходный), Frameset (с фреймами)

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
```

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Frameset//EN"
"http://www.w3.org/TR/html4/frameset.dtd">
```

3.2 Цветовые спецификации. Определить тот или иной цвет при создании web-страницы можно двумя методами: посредством задания специальных **цветовых меток**, обозначающий названия цветов словами, или с помощью шестнадцатеричного цифрового кода, заменяющего эти метки.

Для вывода цветных изображений на экран монитора применяется так называемая палитра RGB (Red, Green, Blue). Согласно данному стандарту, теоретические основы которого были разработаны ученым-физиком Г.Гельмгольцем, можно получить абсолютно любой желаемый оттенок путем смешивания всего лишь трех красок: красной, зеленой и синей.

Для обозначения цвета шестнадцатеричным цифровым кодом перед самим кодом ставится символ <#>, далее следует набор из шести знаков в шестнадцатеричной системе счисления.

Цвет	Символьная метка	Цифровой код
Белый	White	#FFFFFF
Черный	Black	#000000
Зеленый	Green	#008000
Светло-зеленый	Lime	#00FF00
Серый	Gray	#808080
Светло-серый	Silver	#C0C0C0
Желтый	Yellow	#FFFF00
Темно-бордовый	Maroon	#800000
Синий	Blue	#0000FF
Темно-синий	Navy	#000080
Голубой	Aqua	#00FFFF
Изумрудный	Teal	#008080
Красный	Red	#FF0000
Пурпурный	Purple	#800080
Розовый	Fuchsia	#FF00FF
Оливковый	Olive	#808000

Для передачи цветов в html предпочтительнее пользоваться шестнадцатеричным цифровым кодом.

3.3 Форматирование текста.

Escape-последовательности. Иногда возникает необходимость использования в тексте символы, зарезервированные для обозначения элементов кода HTML. Это могут быть угловые скобки, прямые ка-

вычки и т.д. Т.е. необходимо, чтобы символы отображались как текст, интерпретатор же воспринимает их как часть команд разметки гипертекста. Поэтому существуют escape-последовательности, или их еще называют «нотации символьных объектов». Они начинаются символом & и заканчиваются ; а между ними размещается сама команда, записываемая в строчном регистре.

Символ	Значение	Именной примитив	Числовой примитив
<	Левая угловая скобка	<	<
>	Правая угловая скобка	>	>
“	Прямые кавычки	"	"
&	Амперсант	&	&
	Неразрывный пробел	 	
©	Символ копирайта	©	©
®	Символ зарегистрированной торговой марки	®	®
=	Знак равенства	≡	=
←	Стрелка влево	←	←
→	Стрелка вправо	→	→
↑	Стрелка вверх	↑	↑
↓	Стрелка вниз	↓	↓

Пример:

На этой страничке изложены мои " мысли вслух " по данному вопросу

Перевод строки, пробелы, табуляция. Введенное пользователем с помощью клавиатуры форматирование текста, включающее в себя пробелы, отступы табуляции и переводы строки, игнорируется браузером при интерпретации html-документа.

Исключением из этого правила может служить только текст, помеченный специальным тегом <PRE> – командой предварительного форматирования. Формат команды:

```
<PRE>
предварительно отформатированный текст
</PRE>
```

В этом случае все, что расположено между открывающим и закрывающим тегами <PRE> отобразится в окне браузера с сохранением всех введенных дизайнером пробелов, переводов строк и отступов табуляции.

Тег стандартного абзаца.

```
<P align=аргумент>
```

```
Текст, заключенный в абзац
</P>
```

Аргументы ALIGN:

right – выравнивание текста по правому краю;

left – выравнивание текста по левому краю;

center – выравнивание текста по центру;

justify – выравнивание текста по ширине.

Если при использовании тега <P> не задано никаких атрибутов, по умолчанию используется левое выравнивание.

Для выравнивания текста по центру можно использовать тег:

```
<CENTER>
```

```
Текст
```

```
</CENTER>
```

Разрывы строк. Для того, чтобы внутри тестового блока браузер отобразил перевод строки, например, для отделения одного абзаца от другого, в конце строки следует поместить тег
. Следующее слово, встречающееся за данным тегом, будет перенесено на новую строку. Возможно использование нескольких тегов
 подряд для задания последовательности из нескольких пустых строк. Команда
 не имеет закрывающего тега.

Иногда тег
 содержит атрибут clear, который может принимать одно из трех значений: left, right, all. Указанные параметры позволяют переносить текст со смещением новой строки соответственно к левой, правой или обеим границам экрана одновременно. Синтаксис записи этого тега с атрибутом clear:

```
<BR clear="значение">
```

Некоторые браузеры переносят строки самостоятельно, причем разрывают строку в том месте, где она выходит за границы экрана. Если программисту необходимо отобразить текстовую строку фиксированной длины без переносов, используется тег <NOBR>:

```
<NOBR> строка без переносов </NOBR>
```

При использовании данного тега в случае, если длина строки превышает ширину экрана, в нижней части окна браузера появляется горизонтальная полоса прокрутки.

**Шрифты. Элемент ** определяет свойства шрифта.

Синтаксис записи тега :

```
<FONT face="значение1"
size="значение2"
color="значение3">
```

```
текст
```


Значением атрибута `face` является название шрифта, который используется для отображения текста, заключенного в теги `` ``. Если, например, вместо параметра "значение1" в приведенном примере вы подставите "Arial", помеченный тегом ``, текст будет отображаться именно этим шрифтом. В атрибуте `face` в качестве значений допустимо указывать несколько гарнитур шрифтов, разделяя их запятой. Для отображения текста браузер применит одну из перечисленных гарнитур, имя которой первой совпадает с имеющимся в его распоряжении набором гарнитур.

```
<FONT face="Helvetica, Arial">
```

Атрибут `size` указывает на размер шрифта. Параметр этого атрибута может быть описан либо абсолютной, либо относительной величиной. Абсолютная величина подразумевает использование в качестве параметра целого числа, указывающего на высоту шрифта в пунктах. Относительная же величина, обозначаемая целым числом со знаком плюс или минус (например, +2 или -1), — это количество пунктов, которые следует прибавить или отнять от размера шрифта, используемого браузером по умолчанию (12pt). Так, запись `` говорит о том, что размер помеченного таким тегом текста будет на один пункт больше, чем обычный текст документа. Кроме того, размер шрифта можно записывать целыми числами от 1 до 7. В этих единицах основному шрифту, используемому по умолчанию, присваивается размер 3. Самый мелкий шрифт имеет размер 1, самый крупный — 7. Каждый шрифт в шкале «1-7» на 20% больше или меньше размера основного шрифта.

Атрибут `color` определяет цвет шрифта и подставляется либо цифровым кодом, либо символьной меткой.

Очевидно, что при использовании тега `` любой из его атрибутов может быть опущен.

```
<FONT face="Arial"
      size="+2"
      color="#0000FF">
```

отрывок текста

```
</FONT>
```

```
<FONT face="Arial"
      size=5
      color="olive">
```

отрывок текста

```
</FONT>
```

```
<Font size=6> текст </Font>
```

```
<Font size=+2> текст </Font>
```

```
<Font size=-1> текст </Font>
```

Изменение основного шрифта. По умолчанию текстовое содержимое документа отображается основным шрифтом, заданным в начальных установках браузера. Обычно основным шрифтом имеет размер 3 и гарнитуру Times New Roman. Для изменения свойств основного шрифта предназначен одиночный тег `<BASEFONT>`, который имеет атрибуты `size`, `color` и `face`.

```
<BASEFONT size=5>
```

Задание цвета фона. Цвет фона всего документа определяется атрибутом `bgcolor` тега `<BODY>`.

```
<BODY bgcolor="blue">
```

Цвет текста. Для определения цвета шрифта можно пользоваться одним из 3 способов:

- назначить цвет символов всего документа с помощью атрибута `text` тега `<BODY>`

```
<BODY text="green">
```
- если в документе определен основной шрифт с помощью тега `<BASEFONT>`, можно дополнить этот элемент атрибутом `color`

```
<BASEFONT size=5 color="red">
```
- для текстовых фрагментов цвет задается с помощью атрибута `color` в тегах ``, при этом такое задание цвета отменяет действие атрибута `text` тега `<BODY>`

```
<FONT color="red">
```

Элементы стиля шрифтов.

```
<B> Выделение текста полужирным шрифтом</B>
```

```
<I> Выделение курсивным шрифтом</I>
```

```
<U> Выделение текста подчеркиванием </U>
```

```
<TT> Моноширинный текст </TT>
```

```
<SMALL> Уменьшение размера шрифта на 10-20% относительно текущего размера шрифта </SMALL>
```

```
<BIG> Увеличение размера шрифта на 10-20% относительно текущего размера шрифта </BIG>
```

```
<STRIKE> Зачеркнутый текст </STRIKE>
```

```
<S> Зачеркнутый текст </S>
```

```
<SUB> Нижний индекс </SUB>
```

```
<SUP> Верхний индекс </SUP>
```

Данные пары тегов можно вкладывать одна в другую. Например, выделить текст жирным курсивным шрифтом можно так:

```
<B><I>Текст, выделенный жирным курсивным шрифтом</I></B>
```

Фразеологические элементы. Фразеологический элемент не указывает точного способа отображения текста, а лишь определяет, что содержимое элемента несет определенную смысловую нагрузку и должно быть каким-то образом выделено относительно обычного текста.

`` Элемент акцентирования. Большинство браузеров воспринимают его как выделение текста курсивом. ``

`` Элемент акцентирования. Этот элемент обычно интерпретируется браузерами как выделение полужирным шрифтом. ``

`<DFN>` Термин и его определение. Отображается курсивом. `</DFN>`

`<KBD>` “Текст, набранный с клавиатуры”. Отображается шрифтом Courier. `</KBD>`

`<VAR>` Задание переменной или ее значения. Содержимое выводится курсивом. `</VAR>`

`<CODE>` Воспроизведение специальных символов. Применяется в случаях, когда необходимо показать, что текст является исходным кодом какой-либо программы. `</CODE>`

`<SAMP>` Воспроизведение буквенных символов. Указывает, что текст является образцом. `</SAMP>`

`<ACRONYM title="значение">`

Элемент аббревиатуры. В содержимом располагается сокращенное название аббревиатуры. Расшифровка термина появляется в виде всплывающей подсказки, текст которой задается значением атрибута `title`. Содержимое `ACRONYM` появляется обычным текстом. Назначение `ACRONYM` состоит в подготовке информации для поисковых серверов по расшифровке аббревиатур.

`</ACRONYM>`

Оформление цитат.

`<BLOCKQUOTE cite="URL">`

Текст длинной цитаты

`</BLOCKQUOTE>`

Задает отображение длинных цитат отдельным текстовым блоком. Содержимое элемента выводится с небольшим отступом от левого и правого краев страницы (по умолчанию отступ равен 40px). Перед цитатой и после нее браузер оставляет по одной пустой строке. Атрибут `cite` предназначен для указания URL-адреса цитируемого документа. Этот атрибут не влияет на отображение цитаты и носит справочный характер.

`<Q>` Текст короткой цитаты `</Q>`

Тег применяется для форматирования коротких цитат. Содержимое элемента отображается без разрыва строки.

`<CITE>` Текст цитаты `</CITE>`

Цитаты, оформленные с помощью этого элемента, выделяются курсивным шрифтом.

Элемент `<ADDRESS>`. Используется для размещения в документе контактной информации. Элемент представляет собой блок, содержимое которого отображается курсивом и внутрь которого можно помещать другие элементы, например, разрывы строк, рисунки, гиперссылки и т.д., но нельзя вставлять блоки (разделы, списки, абзацы).

Формат элемента:

`<ADDRESS>`

Содержимое

`</ADDRESS>`

Пример:

`<ADDRESS>`

Екатерина Семенова `
`

Ведущий менеджер

Телефон 123-123-123

`realiz@kdk.com`

`</ADDRESS>`

Атрибуты тега `<BODY>`. Общий формат записи тега `<BODY>` со всеми допустимыми атрибутами имеет вид:

```
<BODY background="URL"
  bgcolor="значение1"
  text="значение2"
  link="значение3"
  vlink="значение4"
  alink="значение5">
```

тело документа HTML

`</BODY>`

Атрибут `background` позволяет поместить на web-страницу фоновый рисунок, записав в качестве параметра атрибута `URL` этого рисунка. `URL` можно задавать либо в виде полного адреса Интернета (например, "http://www.server.ru/images/имя_файла.gif"), либо в виде сокращенного адреса с указанием пути к директории на текущем сервере, в которой хранится данное изображение (например, "../images/имя_файла.gif"). Допускается просто указывать имя графического файла, если он хранится в той же директории, что и использующий его файл HTML. Данное изображение может иметь любой

размер, поскольку при интерпретации кода оно многократно повторяется, заполняя все доступное пространство в окне браузера.

Атрибут `text` позволяет задать цвет текста для всего документа в целом. Однако он может быть изменен в определенном участке текста путем использования команды `` с атрибутом `color`.

Атрибут `bgcolor` используется для задания фоновой цвета всему документу. Атрибуты `bgcolor` и `background` не исключают друг друга, однако у последнего имеется приоритет. Это означает, что в случае, когда заданы оба эти атрибута, сначала выполняется заливка web-страницы цветом, назначенным в атрибуте `bgcolor`, поверх которой размещается изображение, заданное атрибутом `background`.

Атрибут `link` задает цвет, которым отображается непосещенная гиперссылка, то есть ссылка, к которой посетитель данного web-сайта еще не обращался. По умолчанию ей присваивается значение "blue" (#0000FF). В свою очередь, атрибут `vlink` указывает на цвет посещенной ссылки, значение которой по умолчанию – "purple" (#800080).

Атрибут `alink` определяет цвет гиперссылки в момент нажатия. По умолчанию данный атрибут также имеет значение "purple".

Пример использования тега `<BODY>` со всеми допустимыми атрибутами:

```
<BODY
background=http://www.myserver.com/images/back.jpg
bgcolor="#000000"
text="#FFFFFF"
link="#008000"
vlink="#800080"
alink="#FF0000">
```

Тело документа HTML
</BODY>

Применение многоязычного текста. Указание языка, на котором составлен документ или его фрагмент выполняется атрибутом `lang`. В качестве значения этого атрибута представляется код языка (`en` – английский, `en-US` – английский США, `de` – немецкий, `ru` – русский, `uk` – украинский). Например,

`<HTML lang="de">` – документ составлен на немецком языке,

`<Q lang="uk">` – цитата на украинском.

Направление чтения текста. Задается атрибутом `dir`, принимающим значения: `ltr` – слева направо, `rtl` – справа налево. Атрибут задается для текстов на арабском языке или иврите.

3.4 Разделы, заголовки и горизонтальные линии.

Разделы. Формат задания раздела:

```
<DIV>
HTML-код или текст
</DIV>
или
<SPAN>
HTML-код или текст
</SPAN>
```

Разделы используются для задания параметров текста и его расположения. К разделу можно применять любые атрибуты стиля. Теги `<DIV>` и `` отличаются тем, что при использовании тега `<DIV>` браузер производит перенос раздела на новую строку и перед ним автоматически устанавливается пустая строка. При использовании тега `` переноса не возникает.

Заголовки. Заголовки позволяют разделить web-страницу на логически законченные блоки и помогают ориентироваться в содержимом документа.

Спецификация HTML позволяет программисту использовать шесть различных стилей, или, как их еще принято называть, логических уровней заголовков.

Для того чтобы отобразить заголовок на web-странице, необходимо использовать тег `<Hn>`, где `n` — целое число от 1 до 6, обозначающее номер уровня заголовка, причем самым верхним уровнем принято считать 1. Синтаксис записи этой команды:

```
<Hn align="параметр">
Текст заголовка
</Hn>
```

Атрибут `align` дает возможность web-мастеру определить расположение заголовка в окне браузера. Параметр этого атрибута может принимать одно из трех возможных значений:

`right` – позиционирование заголовка по правой границе документа;

`left` – позиционирование заголовка по левой границе документа;

`center` – позиционирование заголовка по центру документа.

По умолчанию заголовки выравниваются по левому краю страницы. Высоты заголовков относятся как 2:1,5:1,17:1,33:0,83:0,67. Заголовки как блочные элементы отделяются от текста пустыми строками.

Пример:

```
<H1 align="center">
```

ДОБРО ПОЖАЛОВАТЬ НА МОЮ СТРАНИЧКУ!

</h1>

Горизонтальные линии. Чтобы визуально отделить часть объектов документа от других, применяются горизонтальные линии. Горизонтальная линия – это графический элемент, имеющий длину, толщину и цвет. Для создания таких линий в HTML используется непарный тег <HR> (сокращение от Horizontal Ruler). В спецификации HTML для отображения вертикальных линий не предусмотрено специальных тегов.

Формат создание стандартной линии (по умолчанию стандартная линия занимает всю ширину окна броузера, а её толщина составляет 2px, верхняя часть линии несколько темнее нижней):

```
<HR>
```

Формат сплошной темной линии без тени:

```
<HR noshade>
```

Формат горизонтальной линии с указанием свойств:

```
<HR align="значение"  
width="значение"  
size="значение"  
color="значение">
```

Атрибут align задает выравнивание линии и может принимать значения center, left или right. По умолчанию линии выравниваются по центру страницы.

Атрибут width задает длину линии в пикселях или процентах. Например, тег <HR width=25> определяет горизонтальную линию шириной 25 пикселей. Если значение атрибута задается в процентах (ставится символ %), то ширина линии вычисляется относительно ширины окна, например <HR width="25%">. Задание длины линии в относительных единицах предпочтительнее.

Толщина линии задается атрибутом size. Значение этого атрибута задается в пикселях в диапазоне от 1 до 175. Присвоение больших значений игнорируется. По умолчанию size=2px.

Цвет линии определяется атрибутом color.

4 СПИСКИ

4.1 Маркированный список

4.2 Нумерованный список

4.3 Вложенные списки

4.4 Списки определений

Существуют 3 вида списков: маркированный (неупорядоченный), нумерованный (упорядоченный), список определений.

4.1 Маркированный список.

Формат задания списка:

```
<UL type="параметр">  
  <LH>Заголовок списка</LH>  
  <LI>Содержимое 1 строки списка</LI>  
  <LI>Содержимое 2 строки списка </LI>  
  ...  
  <LI>Содержимое n строки списка</LI>  
</UL>
```

Атрибут type позволяет программисту управлять так называемым «буллет-стилем», то есть внешним видом и формой обозначающих каждое вхождение меток. Параметр этого атрибута может принимать одно из трех значений:

disc — метки отображаются в виде заполненных черным цветом окружностей;
circle — метки отображаются в виде полых окружностей;
square — метки отображаются в виде заполненных черным цветом квадратов.

По умолчанию, то есть в случае, когда в маркированном списке тег записывается без атрибутов, type="disc".

4.2 Нумерованный список. Нумерованный список позволяет отображать упорядоченную информацию, которая выводится на экран в виде обозначенного цифрами перечисления.

Формат задания списка:

```
<OL type="параметр"  
start="значение">  
  <LI>Содержимое 1 строки списка</LI>  
  <LI>Содержимое 2 строки списка </LI>  
  ...
```

```
<LI> Содержимое n строки списка</LI>
</OL>
```

Атрибут `type` позволяет изменять буллет-стиль, то есть определять вид маркеров, которыми будут обозначаться составляющие список значения. В составе атрибута `type` можно использовать один из следующих параметров:

- "1" – обычные арабские числа 1, 2, 3 и т. д.
- "I" – римские цифры в заглавном регистре I, II, III, IV и т. д.
- "i" – римские цифры в строчном регистре i, ii, iii и т. д.
- "A" – символьная маркировка в заглавном регистре A, B, C и т. д.
- "a" – символьная маркировка в строчном регистре a, b, c и т. д.

В HTML возможно изменить порядок нумерации строк списка и порядок нумерации внутри списка.

Атрибут `start` позволяет задавать позицию, с которой следует начать маркировку. Например, если атрибут `type` задан как "I", а атрибут `start` – как "5", то нумерация начнется с цифры 5. Если же `type` имеет значение "A", а `start` – "D", то маркировка начнется с символа D.

Можно изменить нумерацию внутри списка, указав в теге `` атрибут `value`.

```
<LI value=значение> Содержимое l строки списка </LI>
```

По умолчанию значение атрибута `type` принимается как "I", нумерация начинается с первого элемента в каждом из типов маркировки.

4.3 Вложенные списки. Возможно создание так называемых вложенных списков, реализуемых посредством размещения одной пары тегов в другой, например, команд создания маркированного списка внутри нумерованного. Простая реализация вложенного списка имеет вид:

1. Вхождение 1
 - Вхождение 1-а
 - Вхождение 1-б
2. Вхождение 2
 - Вхождение 2-а
 - Вхождение 2-б

Такой список можно представить следующим набором команд:

```
<OL>
  <LI> Вхождение 1
    <UL>
      <LI>Вхождение 1-а</LI>
```

```
      <LI>Вхождение 1-б</LI>
    </UL>
  </LI>
  <LI> Вхождение 2
    <UL>
      <LI>Вхождение 2-а</LI>
      <LI>Вхождение 2-б</LI>
    </UL>
  </LI>
</OL>
```

Тег `` может включать те же атрибуты, что и теги самих списков `` и ``. Если, например, вхождения маркированного списка отображаются принятым по умолчанию маркером `disc`, но одно из наиболее значимых по контексту вхождений программисту необходимо пометить маркером `square`, можно записать тег данного элемента следующим образом: `<LI type="square">Вхождение`. Аналогично, один из пунктов нумерованного арабскими цифрами списка можно пометить символьным маркером, записав соответствующий тег следующим образом: `<LI type="A">Вхождение`.

4.4 Списки определений. Эти списки представляют собой списки терминов вместе с их определениями, то есть они напоминают толковый словарь.

Формат записи списка:

```
<DL>
  <DT> Тер, задающий определяемый термин </DT>
  <DD> Тер, задающий определение термина </DD>
  <DD> Тер, задающий определение термина </DD>
  <DT> Тер задающий определяемый термин </DT>
  <DD> Тер, задающий определение термина </DD>
  <DD> Тер, задающий определение термина </DD>
</DL>
```

Закрывающие теги для команд `<DT>` и `<DD>` списка определений можно опустить. Данные теги могут использоваться и самостоятельно: `<DT>` для отображения терминов, выделяемых в тексте шрифтом, а `<DD>` – для их расшифровки, помеченной отступом. В этом случае наличие соответствующих закрывающих тегов обязательно.

5 ТАБЛИЦЫ

5.1 Формат задания и общие параметры таблицы

5.2 Параметры заголовка, строк и ячеек таблицы

5.3 Группирование строк и столбцов таблицы

5.4 Наследование свойств выравнивания

5.1 Формат задания и общие параметры таблицы. Для создания таблиц применяется тег <TABLE>. Состав таблицы по строкам и ячейкам задается с помощью следующих элементов:

TR – элемент, формирующий отдельную строку;

TD – элемент, определяющий содержимое ячейки данных;

TH – элемент, задающий ячейку заголовка;

CAPTION – элемент названия таблицы.

Структура записи тега <TABLE> в общем виде следующая:

```
<TABLE align="значение"
  width="значение"
  height="значение"
  background="URL"
  bgcolor="значение"
  border="целое число"
  bordercolor="значение"
  bordercolordark="значение"
  bordercolorlight="значение"
  cellpadding="целое число"
  cellspacing="целое число"
  cols="значение"
  frame="значение"
  rules="значение">
  <CAPTION align="значение"
    valign="значение">
    Название таблицы
  </CAPTION>
  <TR align="значение"
    valign="значение"
    width="значение"
    height="целое число"
    bgcolor="значение"
    bordercolor="значение"
```

```
bordercolordark="значение"
bordercolorlight="значение">
<TH align="значение"
  valign="значение"
  width="значение"
  height="целое число"
  background="URL"
  bgcolor="значение"
  bordercolor="значение"
  bordercolordark="значение"
  bordercolorlight="значение"
  colspan="целое число"
  rowspan="целое число"
  nowrap>
```

Ячейка заголовка таблицы

```
</TH>
```

```
</TR>
```

```
<TR>
```

```
<TD align="значение"
  valign="значение"
  width="значение"
  height="целое число"
  background="URL"
  bgcolor="значение"
  bordercolor="значение"
  bordercolordark="значение"
  bordercolorlight="значение"
  colspan="целое число"
  rowspan="целое число"
  nowrap>
```

Содержимое ячеек

```
</TD>
```

```
</TR>
```

```
</TABLE>
```

Общие параметры таблицы определяются значениями атрибутов тега <TABLE>:

- align – задает горизонтальное позиционирование всей таблицы в целом, может принимать значения left, right или center.

- `width` определяет ширину *всей таблицы* и может принимать значение целого числа, если ширина таблицы указывается в пикселах, либо в процентах от ширины экрана пользователя. Если этот атрибут не задан, то он определяется на основе содержимого таблицы.
- `height` устанавливает высоту таблицы.
- `background` определяет изображение, которое будет использоваться в качестве фонового рисунка таблицы.
- `bgcolor` устанавливает цвет фона таблицы.
- `border` указывает на толщину рамки в пикселах. Если `border=0`, таблица становится «невидимой».
- `bordercolor` устанавливает цвет рамки таблицы.
- `bordercolordark`, `bordercolorlight` – используются для задания объемной границы таблицы. Такой эффект достигается за счет разных цветов верхней и нижней линий, образующих рамку. Атрибут `bordercolordark` определяет цвет затененной части рамки, `bordercolorlight` задает цвет освещенной рамки.
- `cellpadding` определяет расстояние в пикселах между границей ячейки и её содержимым.
- `cellspacing` задает расстояние между внешними границами ячеек в пикселах.
- `cols` указывает количество столбцов в таблице.
- `frame` определяют параметры внешних границ таблицы. Чтобы этот атрибут влиял на отображение рамок, должен быть установлен атрибут `border` с ненулевым значением. Атрибут `frame` может принимать следующие значения:
 - `border` (или `box`) – устанавливает все линии внешней рамки;
 - `void` – внешняя рамка не отображается;
 - `above` (или `below`) – выводит только верхнюю (или нижнюю) границу таблицы;
 - `hsides` – выводит только верхнюю и нижнюю линии рамки;
 - `vsides` – выводит только левую и правую линии рамки;
 - `lhs` – устанавливает только левую линию рамки;
 - `rhs` – устанавливает только правую линию рамки.
- `rules` определяют параметры внутренних границ таблицы. Чтобы этот атрибут влиял на отображение рамок, должен быть установлен атрибут `border` с ненулевым значением. Атрибут `rules` может принимать следующие значения:
 - `none` – все внутренние линии не отображаются;

- `groups` – отображает горизонтальные границы между группами строк и столбцов;
- `rows` – определяет отображение только горизонтальных линий между строками;
- `cols` – устанавливает отображение только вертикальных линий между столбцами;
- `all` – все внутренние рамки отображаются.

5.2 Параметры заголовка, строк и ячеек таблицы. Параметры заголовка определяются следующими значениями атрибутов тега `<CAPTION>`:

- `align` – определяет выравнивание названия относительно границ таблицы и может принимать значения: `left` (выравнивание по левой границе, название располагается над таблицей), `right` (выравнивание по правой границе, название располагается над таблицей), `center` или `top` (выравнивание по центру, название располагается над таблицей), `bottom` (выравнивание по центру, название располагается под таблицей).
- `valign` – используется для выравнивания названия при размещении его над или под таблицей. Атрибут `align` не выполняет выравнивания названия вправо или влево при значениях `top` и `bottom`. Для такого выравнивания нужно применять оба атрибута `valign` и `align`: атрибут `valign` будет задавать расположение названия над или под таблицей, а атрибут `align` – определять его выравнивание по левой или правой границе таблицы.

Параметры строк таблицы задаются значениями атрибутов тега `<TR>`, а параметры ячеек – значениями атрибутов тегов `<TD>`, `<TH>`:

- `align` – задает выравнивание содержимого ячейки по горизонтали и определяется значениями `left`, `right`, `center`.
- `valign` – задает выравнивание содержимого ячейки по вертикали и определяется значениями `top` (по верхнему краю), `bottom` (по нижнему краю), `middle` (посередине), `baseline` (по базовой линии).
- `width` задает ширину ячейки.
- `height` устанавливает высоту ячеек таблицы. Если значение `height` меньше содержимого ячейки, этот параметр игнорируется.
- `background` определяет изображение, которое будет использоваться в качестве фонового рисунка ячейки таблицы.

- bgcolor устанавливает цвет фона ячейки таблицы.
- bordercolor устанавливает цвет рамки ячеек таблицы.
- bordercolordark, bordercolorlight – используются для задания объемной границы ячеек таблицы. Атрибут bordercolordark определяет цвет затененной части рамки, bordercolorlight задает цвет освещенной рамки.
- colspan – устанавливает число ячеек, которые должны быть объединены в одну по горизонтали.
- rowspan – устанавливает число ячеек, которые должны быть объединены в одну по вертикали.
- nowrap – запрет переноса слов внутри ячейки. Текст отображается одной сплошной строкой.

5.3 Группирование строк и столбцов таблицы. Для удобства форматирования содержимого таблицы её удобно разбить на заголовочную, основную и нижнюю части. Группирование позволяет присваивать каждой группе элементов общие свойства (выравнивание, гарнитуру, размер шрифта и т.д.).

Группирование строк задается с помощью трёх тегов:

```
<THEAD>
    Заголовочная часть таблицы
</THEAD>
<TBODY>
    Основная часть таблицы
</TBODY>
<TFOOT>
    Нижняя часть таблицы
</TFOOT>
```

В таблице может быть только один элемент TBODY и TFOOT, но несколько элементов THEAD. Допустимыми атрибутами этих тегов являются:

- align и valign – для горизонтального и вертикального выравнивания;
- title – для формирования всплывающей подсказки;
- lang и dir – для задания языка и направления чтения документа.

Для группирования элементов столбцов используется тег COLGROUP. Формат записи тега:

```
<COLGROUP span=значение
            width="значение"
            align="значение">
```

```
valign="значение">
```

Строки, для которых сгруппированы столбцы
</COLGROUP>

Значения атрибутов:

- span – задает количество столбцов в группе;
- width – задает ширину каждого столбца группы, её значение может задаваться в пикселах;
- align и valign – для горизонтального и вертикального выравнивания.

5.4 Наследование свойств выравнивания. По отношению к элементу TABLE дочерними являются: TH, TD, THEAD, TFOOT, COL, COLGROUP. В каждом из них можно задать параметры выравнивания. Наследование свойств горизонтального выравнивания align наследуется с учетом приоритета (от высшего к низшему):

- наиболее высоким приоритетом обладают атрибуты, установленные внутри символьных данных ячейки;
- атрибуты тегов <TH> и <TD>;
- атрибуты тегов группирования столбцов (COL, COLGROUP);
- атрибуты элементов строки или групп строк (элементы TR, THEAD, TFOOT);
- атрибуты тега <TABLE>;
- атрибуты, установленные по умолчанию.

Свойства вертикального выравнивания (значения атрибута valign) для содержимого ячеек наследуется в том же порядке, что и свойства горизонтального выравнивания. Отличие состоит лишь в том, что значения этого атрибута в элементах строк или групп строк (элементы TR, THEAD, TFOOT) имеют более высокий приоритет, чем значения в элементах группирования столбцов (элементы COL, COLGROUP).

Примеры задания таблиц

1. Простые таблицы

Самые знаменитые алмазы

Название	Страна	Год	Масса в каратах
Куллинан	Южная Африка	1905	3106
Экстадальсюр	Южная Африка	1893	971.5
Звезда Сьерра-Леоне	Западная Африка	1972	968.9

```

<TABLE align=center border=3>
  <CAPTION align=center>
    Самые знаменитые алмазы
  </CAPTION>
  <TR align=center>
    <TH>Название</TH>
    <TH>Страна</TH>
    <TH>Год</TH>
    <TH>Масса в каратах</TH>
  </TR>
  <TR>
    <TD>Куллинан</TD>
    <TD>Южная Африка</TD>
    <TD>1905</TD>
    <TD>3106</TD>
  </TR>
  <TR>
    <TD>Эксельсиор</TD>
    <TD>Южная Африка</TD>
    <TD>1893</TD>
    <TD>971.5</TD>
  </TR>
  <TR>
    <TD>Звезда Сьерра-Леоне</TD>
    <TD>Западная Африка</TD>
    <TD>1972</TD>
    <TD>968.9</TD>
  </TR>
</TABLE>

```

2. Объединение строк и столбцов таблицы.

Магазин предлагает учебники

Предмет	Раздел	Классы
Математика	Арифметика	1 - 4
	Алгебра	5 - 11
	Геометрия	7 - 11
	Количество наименований: 1090	
Физика	Механика	5 - 7
	Оптика	10 - 11
	Количество наименований: 457	

```

<TABLE align=center border=3>
  <CAPTION align=center>
    Магазин предлагает учебники
  </CAPTION>
  <TR align=center>
    <TH>Предмет</TH>
    <TH>Раздел</TH>
    <TH>Классы</TH>
  </TR>
  <TR>
    <TD rowspan=4>Математика</TD>
    <TD>Арифметика</TD>
    <TD>1 - 4</TD>
  </TR>
  <TR>
    <TD>Алгебра</TD>
    <TD>5 - 11</TD>
  </TR>
  <TR>
    <TD>Геометрия</TD>
    <TD>7 - 11</TD>
  </TR>
  <TR>
    <TD colspan=2>Количество наименований: 1090</TD>
  </TR>
  <TR>
    <TD rowspan=3>Физика</TD>
    <TD>Механика</TD>
    <TD>5 - 7</TD>
  </TR>
  <TR>
    <TD>Оптика</TD>
    <TD>10 - 11</TD>
  </TR>
  <TR>
    <TD colspan=2>Количество наименований: 457</TD>
  </TR>
</TABLE>

```

3. **Вложенные таблицы.** Команды, управляющие созданием и отображением таблиц, допускают вложение одного тега <TABLE> внутрь другого. Эта особенность языка используется для представления «вложенных» таблиц.

Магазин предлагает учебники

Предмет	Раздел	Классы
Математика	Арифметика	1 - 4
	Алгебра	5 - 11
	Геометрия	7 - 11
	Количество наименований: 1090	
Физика	Механика	5 - 7
	Оптика	10 - 11
	Количество наименований: 457	

```
<TABLE align=center border=3>
  <CAPTION align=center>
    Магазин предлагает учебники
  </CAPTION>
  <TR align=center>
    <TH>Предмет</TH>
    <TH>Раздел</TH>
    <TH>Классы</TH>
  </TR>
  <TR>
    <TD> Математика</TD>
    <TD colspan=2>
      <TABLE border=4>
        <TR>
          <TD>Арифметика</TD>
          <TD>1 - 4</TD>
        </TR>
        <TR>
          <TD>Алгебра</TD>
          <TD>5 - 11</TD>
        </TR>
      </TABLE>
    </TD>
  </TR>
  <TR>
    <TD> Физика</TD>
    <TD colspan=2>
      <TABLE border=4>
        <TR>
          <TD>Механика</TD>
          <TD>5 - 7</TD>
        </TR>
        <TR>
          <TD>Оптика</TD>
          <TD>10 - 11</TD>
        </TR>
        <TR>
          <TD colspan=2>Количество наименований: 457</TD>
        </TR>
      </TABLE>
    </TD>
  </TR>
</TABLE>
```

```
<TR>
  <TD>Геометрия</TD>
  <TD>7 - 11</TD>
</TR>
<TR>
  <TD colspan=2>
    Количество наименований: 1090
  </TD>
</TR>
</TABLE>
<TR>
  <TD>Физика</TD>
  <TD colspan=2>
    <TABLE border=4>
      <TR>
        <TD>Механика</TD>
        <TD>5 - 7</TD>
      </TR>
      <TR>
        <TD>Оптика</TD>
        <TD>10 - 11</TD>
      </TR>
      <TR>
        <TD colspan=2>Количество наименований: 457</TD>
      </TR>
    </TABLE>
  </TD>
</TR>
</TABLE>
```


6 ГИПЕРССЫЛКИ

6.1 Ссылки на документы

6.2 Ссылки на разделы

6.3 Ссылки на адрес электронной почты

6.4 Ссылки на файловые объекты

Ссылка представляет собой логическую связь одного фрагмента (элемента) документа с другим фрагментом в том же самом или в другом документе. Ссылка характеризуется двумя точками, называемыми закладками. Различают начальную и конечную закладки.

Начальная закладка устанавливается на конкретном HTML-элементе и определяет точку, из которой задается ссылка. Начальные закладки можно размещать в тексте, таблицах, списках изображений, фреймах. Начальная закладка выделяется на странице подчеркиванием и цветом.

Конечная закладка определяет точку назначения ссылки и может относиться не только к конкретному HTML-элементу, но и к программе или документу в целом.

В HTML различают внутренние и внешние гиперссылки. **Внутренние ссылки** осуществляют переход в пределах одного и того же документа. **Внешние ссылки** обеспечивают переходы к другим документам. Внешние ссылки и называют **гиперссылками**.

Текстовые гиперссылки можно разделить на четыре условных категории: ссылки на документы, ссылки на разделы, ссылки на адрес электронной почты, ссылки на файловые объекты.

6.1 Ссылки на документы. Для реализации простой гиперссылки на другую web-страницу можно использовать в html-документе тег <A>. Формат записи тега:

```
<A href="URL"
  target="параметр"
  title="альтернативный текст">
```

Текст гиперссылки

```
</A>
```

Атрибут href указывает на адрес страницы, с которой нужно установить гиперсвязь. Этот адрес может быть представлен как в виде полного URL (например, "http://www.server.ru/page.htm1"), так и в виде сокращенного URL с указанием пути к искомой странице в пределах локального сервера (например, "/folder/page.html"). Если искомый до-

кумент расположен в одной директории с текущим, в качестве параметра атрибута href достаточно записать его название.

Атрибут target содержит директивы, описывающие правила открытия целевого документа в браузере, эти директивы выделяются в коде символом подчеркивания.

Значения атрибута target:

_self отображает открываемый по ссылке документ HTML в текущем окне браузера;

_blank отображает открываемый документ в новом окне;

_parent загружает страницу во фрейм-родитель, если фреймов нет – в текущее окно;

_top отменяет все фреймы и загружает страницу в полном окне, если фреймов нет – в текущем окне.

Атрибут title предназначен для создания так называемого альтернативного текста – всплывающей подсказки, появляющейся в небольшом прямоугольнике, если пользователь задержит курсор мыши над гиперссылкой на несколько секунд.

6.2 Ссылки на разделы. Иногда возникает необходимость разместить на web-странице гиперссылку не на другой документ, а на расположенный в пределах этой же страницы раздел или участок текста. При активизации подобной гиперссылки браузер «отматывает» экран таким образом, что искомый раздел или фраза оказываются в верхней его части. Процесс создания ссылки на раздел можно условно разделить на два этапа.

Первый заключается в подготовке так называемой «закладки» – специальной метки с уникальным в пределах данного документа именем, присваиваемым «закладке» посредством атрибута name. Для этого в том участке текста, где необходимо разместить метку, применяется следующая команда:

```
<A name="имя_закладки">
```

Ключевое слово или заголовок раздела

```
</A>
```

Второй этап – создание самой ссылки при помощи команды:

```
<A href="#имя_закладки">Текст ссылки</A>
```

Для задания имени «закладки» лучше использовать латинские символы. При вызове раздела посредством гиперссылки имя «закладки» предваряется символом «#».

Если нужно разместить ссылку на раздел какого-либо другого документа, описанный выше алгоритм в целом сохраняется, за исключением того, что в вызывающем теге перед именем «закладки» указыва-

ется URL документа. В этом случае рекомендуется также использовать атрибут `target` с параметром `_blank`. Подобная команда в обобщенном виде будет выглядеть следующим образом:

```
<A href=document.htm#имя_закладки
    target="_blank">
```

Текст ссылки

```
</A>
```

6.3 Ссылки на адрес электронной почты. При активизации такой ссылки автоматически запускается почтовая программа и на экране формируется уже готовый к отправке бланк электронного письма с заполненным адресным полем и иногда полем `Subject`. Ссылка на адрес электронной почты имеет формат:

```
<A href="mailto:user-1@server.ru?cc=user-
2@server.ru,user-3@server.ru&subject=Тема_письма">
```

Текст ссылки

```
</A>
```

Директива `mailto:` указывает на основной адрес электронной почты, куда следует отправлять письмо. В минимальной записи атрибута `href` при реализации гиперссылки на адрес электронной почты можно указать только эту директиву и какой-либо адрес `e-mail`.

Функция `?cc` позволяет определить адреса пользователей, которым будет отправлена копия сообщения. Если таких адресов больше одного, они записываются подряд через запятую без пробелов.

С помощью функции `&subject` можно задать тему отсылаемого сообщения. Пример:

```
<A
```

```
href="mailto:ivanov@gsu.unibel.by&subject=Привет">
```

Жду писем!

```
</A>
```

6.4 Ссылки на файловые объекты. Гиперссылка на файловый объект подразумевает организацию гиперсвязи документа HTML с каким-либо файлом, хранящимся на сервере, например архивом. При нажатии мышью на такой гиперссылке автоматически открывается сессия загрузки данного файла с удаленного сайта на локальный пользовательский компьютер. Формат ссылки:

```
<A
```

```
href="http://www.myserver.com/files/archive.zip">
```

Текст гиперссылки

```
</A>
```

Иногда для реализации гиперсвязи с файловым объектом в записи URL этого объекта используется префикс `file://`, однако такой подход применяется в основном для организации ссылок на файловый объект, расположенный на локальном компьютере. Если загрузку файла планируется осуществлять по протоколу FTP, используйте префикс `ftp://`. Если искомый файл расположен в той же директории, что и вызывающий его документ, в качестве значения атрибута `href` можно просто указать его имя.

7 ГРАФИКА В WEB-ДИЗАЙНЕ

- 7.1 Сетевые графические форматы
- 7.2 Включение графики в web-страницу
- 7.3 Изображения ссылки
- 7.4 Карты ссылок

Графические компоненты web-страниц по назначению можно условно разделить на три категории: **иллюстративная графика**, включающая дополняющие текст фотографии, пояснительные рисунки, чертежи и схемы, **функциональная графика**, представляющая собой элементы управления сайтом (кнопки навигации, счетчики и элементы интерактивных форм) и, **декоративная графика** – элементы дизайна страницы, включенные в нее «для красоты» и не несущие информационной нагрузки (фоновые рисунки, линии-разделители, выполненные в виде графических файлов заголовки и т.д.).

7.1 Сетевые графические форматы. Наиболее популярными форматами графических изображений являются GIF и JPEG. В этих форматах хранится 90% всей графики, представленной в Интернете. В них заложены различные алгоритмы сжатия изображения с потерей качества, благодаря использованию которых удастся значительно уменьшить размер целевого файла.

Стандарт **GIF** (Graphic Interchange Format, формат обмена графическими данными) создан в 1987 году. Он позволяет значительно сокращать объем итогового графического файла по сравнению с исходным изображением за счет смешения сходных оттенков в один. Если, например, в составе рисунка имеется участок, состоящий из нескольких сходных полутонов, к примеру, голубого, светло-голубого и темно-голубого цвета, они будут кодированы одним оттенком – голубым. Информация об изображении в файле стандарта GIF записывается построчно, то есть представляет собой массив описаний строк высотой в один пиксел. Данный формат оперирует **фиксированной**, так называемой **индексированной палитрой**, число цветов в которой не превышает 256. Поэтому стандарт GIF используется в документах HTML только для отображения так называемой бизнес-графики: диаграмм, логотипов, кнопок, разделительных линий, других элементов оформления страницы (т.е. для рисованных изображений). Для размещения на веб-сайте фотографий, репродукций картин и изображений с большим количеством цветов и цветовых переходов используется стандарт JPEG.

Одним из свойств стандарта GIF является его особенность, на-

званная разработчиками «*interlace*», или «черезстрочность». Она позволяет загружать картинку с сервера в клиентский браузер не целиком, а частями следующим образом: сначала на экране отображаются первая, пятая и десятая строки, составляющие изображение, затем – вторая, шестая и одиннадцатая и т. д. Таким образом, для пользователя создается иллюзия постепенной загрузки графического элемента: картинка как бы медленно «проявляется» на странице.

В 1989 году компания CompuServe выпустила новую версию этого графического формата, получившую название GIF89a. Данная модификация включает в себя еще две особенности. Первая называется «*transparency*» и подразумевает создание для изображения прозрачного фона методом сохранения вместе с файлом так называемого **альфа-канала**, представляющего собой маску прозрачности рисунка. Цвета, помеченные в альфа-канале как прозрачные, станут невидимы в браузерах и большинстве других программ, предназначенных для просмотра изображений. Эта функция необходима, например, при размещении картинок неправильной геометрической формы на странице со сложным фоновым рисунком, когда корректно «подогнать» части изображений друг к другу не представляется возможным. Убрать ненужный фон из файла GIF можно встроенной процедурой графического редактора Adobe Photoshop.

Другая особенность стандарта GIF89a заключается в том, что этот формат позволяет сохранять в файле с одним физическим заголовком несколько разных изображений, демонстрируя их на экране последовательно одно за другим, причем с возможностью специфицировать порядок их чередования и временной интервал между сменами кадра. Именно на этом принципе построена GIF-анимация.

JPEG (Joint Photographic Experts Group, объединенная группа экспертов в области фотографии) – графический стандарт, созданный на основе одноименного алгоритма сжатия, изображений с потерей качества, кодирующего не идентичные элементы, а межпиксельные интервалы.

Механизм сжатия изображения в файл формата JPEG. Первой ступенью компрессии является преобразование изображения в цветовой образ **LAB**, раскладывающий картинку на три независимых канала, один из которых (Lightness) выделен для сохранения значений интенсивности цветов, а два других (A и B) — для запоминания непосредственно цветовой информации. Причем данные о цветах сохраняются в виде шкалы, организованной по принципу непрерывного спектра. Вторая ступень – собственно компрессия; из получившейся цветовой модели удаляются приблизительно три четверти информации о цвете, затем

образ дробится на участки размером 8x8 точек, и преобразуется в числовой массив данных. Заголовок каждого блока описывает доминирующий цвет участка, остальная информация – менее заметные оттенки. На третьей ступени сжатия из массива данных удаляется определенная часть информации, описывающей второстепенные оттенки, причем количество изымаемых данных зависит от выбранного пользователем качества результирующего изображения. И наконец, готовый файл сжимается согласно алгоритму Хаффмана, который предусматривает замену наиболее часто встречающихся в массиве данных знаков более компактной двухбитной кодировкой. Декомпрессия файла JPEG происходит в обратном порядке.

JPEG оптимален для передачи фотографических изображений, а также картинок с большим количеством полутонов и цветовых переходов. Максимальное число цветов, которое может содержать изображение в формате JPEG, достигает 16 миллионов.

Существует несколько разновидностей стандарта JPEG. Формат JPEG Baseline Optimized основан на более совершенном алгоритме компрессии изображений. Недостатком данной реализации JPEG является то, что она не распознается целым рядом часто используемых приложений. Progressive JPEG был оптимизирован специально для представления графики во Всемирной сети, изображения в этом формате сжимаются чуть лучше, чем в стандартном JPEG, но хуже, чем в JPEG Baseline Optimized. Отличительная особенность Progressive JPEG – возможность сохранять графику в чересстрочном режиме, как это реализовано в стандарте GIF.

PNG (Portable Network Graphics, портативная сетевая графика), пользователи придумали этой аббревиатуре иную расшифровку: Picture is Not GIF. PNG поддерживает interlacing (черезстрочность) не только по горизонтали, но и по вертикали. Палитра PNG позволяет создавать изображения с глубиной цвета до 48 бит, а также поддерживает альфа-канал, но состоит он не из одного уровня, а из 254 уровней для каждого элемента изображения, с диапазоном прозрачности от 0 до 99%. Благодаря этому дизайнер получил возможность создавать картинки с переменной прозрачностью, например, логотипы, фон которых плавно сливается с фоновым цветом html-документа, каким бы тот ни был. Изображения PNG содержат так называемый блок описания *гамма-коррекции*. Под переменной «гамма» в данном случае подразумевается некая величина, выражающая зависимость яркости свечения каждой точки экрана пользователя от напряжения на электродах кинескопа, которое берется из файлов управления видеоподсистемой компьютера. Таким образом, при открытии изображения, созданного с по-

мощью IBM-совместимой машины, на компьютере другого производителя, например на Apple Macintosh, в яркость цветопередачи вводится соответствующая поправка, и картинка отображается совершенно идентично на разных типах оборудования. Этот формат поддерживают браузеры Microsoft Internet Explorer, начиная с версии 4.0, и Netscape Communicator версий 4.04 и выше.

Оптимизация графических изображений. При сохранении изображений GIF и JPEG на диск в файлы записывается определенное количество «лишних» данных, таких, например, как невидимые текстовые комментарии, ссылки на разработчиков стандарта и соответствующего программного обеспечения и т. д. Все эти дополнения увеличивают физический размер файла и не несут никакой полезной нагрузки. Чтобы сократить время загрузки иллюстраций в клиентский браузер, используют специальные методы оптимизации изображений с использованием специальных программ – *оптимизаторов графики*.

7.2 Включение графики в web-страницу. Включение графики в html-документ осуществляется с использованием команды , синтаксис которой в общем виде записывается следующим образом:

```
<IMG src="URL"
      align="значение"
      border="значение"
      width="значение"
      height="значение"
      alt="текст"
      hspace="значение"
      vspace="значение"
      ismap>
```

В качестве параметра атрибута `src` указывается путь к изображению в виде либо полного URL (например, `http://www.mysite.ru/images/picture.jpg`), либо сокращенного URL с указанием пути к изображению на локальном сервере (например, `../Images/picture.jpg`).

С помощью атрибута `align` картинка позиционируется в документе HTML. Этот атрибут может принимать значения:

`right` – изображение выравнивается по правому краю,

`left` – изображение выравнивается по левому краю,

`center` – изображение выравнивается по центру,

`texttop` или `top` – изображение совмещается верхним краем с верхней границей ячейки таблицы или с верхней границей текстовой строки, в которой размещена картинка,

absbottom или bottom – изображение совмещается нижним краем с нижней границей ячейки таблицы или текстовой строки,

absmiddle или middle – центр изображения выравнивается по основной строке, либо оно размещается в середине табличной ячейки,

baseline – изображение выравнивается по условной «базовой линии». Применение этого параметра рекомендуется в случае, когда web-мастер размещает несколько рисунков или несколько фрагментов одного рисунка в разных ячейках строки таблицы.

Атрибут border управляет толщиной синей рамки, в которую заключается изображение, включенное в тег <A>, при превращении его в гиперссылку. В большинстве случаев эта рамка портит дизайн web-страницы, и чтобы избавиться от нее, достаточно установить значение border равным нулю.

В качестве параметров атрибутов width и height указывают ширину и высоту интегрируемого в документ HTML графического изображения в пикселах.

Атрибут alt содержит *альтернативный текст*, который отображается вместо картинки в браузерах, не поддерживающих графику, а также в случае, если поддержка графики отключена пользователем. Этот же текст выводится в виде всплывающей «подсказки», если пользователь задержит на изображении курсор мыши.

Атрибуты hspace и vspace задают горизонтальные и вертикальные отступы от изображения в пикселах, которые необходимы, например, в случае, когда картинка помещается на web-странице рядом с текстом. hspace определяет величину незаполненного пространства справа и слева от картинки, vspace соответственно сверху и снизу.

Атрибут ismap говорит о том, что рисунок является серверной картой-изображением. Карты-изображения позволяют привязывать ссылки к разным областям одного изображения.

Пример применения тега :

```
<IMG src="/images/sunset.ipg"
      align="CENTER"
      width="325"
      height="215"
      alt="Красивый закат на берегу Карибского моря">
```

7.3 Изображения ссылки. Для того чтобы превратить интегрированное в html-документ изображение в гиперссылку, необходимо восполь-

зоваться комбинацией тегов <A> и , записав ее следующим образом:

```
<A href="URL документа, на который организуется ссылка">
  <IMG src="URL изображения"
        align="значение"
        width="значение"
        height="значение"
        alt="альтернативный текст"
        border="значение">
</A>
```

Пример кода HTML, позволяющего использовать изображение в качестве гиперссылки:

```
<A href="/hobbie/index.html">
  <IMG src="/mages/hobbie.jpg"
        align="BASELINE"
        width="315"
        height="226"
        alt="Тематический раздел 'Мои увлечения'"
        border="0">
</A>
```

7.4 Карты ссылок. Карта ссылок (image map) представляет собой графическое изображение, разбитое на фрагменты, причем каждый фрагмент является одновременно гиперссылкой. Для задания карты ссылок необходимо:

1. Задать изображение карты с указанием того, что это изображение представляет собой карту, а не простой рисунок.
2. Разбить изображение на отдельные фрагменты и составить их описание.
3. Описать каждый фрагмент.

Графические карты создаются с помощью тега <MAP>. Элементы <MAP> динамически связаны с изображениями, вызываемыми директивой , и в общем виде описываются следующим образом:

```
<MAP name="имя карты">
  <AREA href="URL"
        shape="параметр"
        coords="x1, y1, x2, y2"
        alt="альтернативный текст"
        target="параметр"
        nohref>
```

</MAP>

Атрибут name тега <MAP> определяет уникальное имя карты, записываемое латинскими символами. При задании имени карты необходимо соблюдать регистр.

Тег <AREA> определяет непосредственно активную область изображения. Атрибут href указывает на адрес документа, вызываемого при нажатии кнопки мыши над данным участком изображения.

Атрибут alt определяет всплывающую подсказку при наведении курсора мыши на данный элемент.

Атрибут shape управляет формой активной области и может принимать одно из трех значений: rect – прямоугольник, circle – круг или poly – многоугольник.

Атрибут coords позволяет определить относительные координаты вершин активной области. Возможные значения данного атрибута:

Значение shape	Синтаксис записи coords	Значения параметров coords
shape=rect	coords="x1,y1,x2,y2"	x1 и y1 – координаты левого верхнего угла фигуры, x2 и y2 – правого нижнего
shape=circle	coords="x,y,r"	x и y – координаты центра окружности, r – ее радиус в пикселах
shape=poly	coords="x1,y1,x2,y2,x3,y3,...xn,yn,x1,y1"	x1, y1 ... xn, yn – координаты вершин многоугольника

Значения координат активной области отсчитываются в пикселах по длине и ширине картинки от левого верхнего угла изображения, принимаемого за точку с координатами 0,0. То есть, значение x возрастает справа налево, а значение y – сверху вниз. Координаты x и y могут быть также заданы в процентах от реального размера изображения, например: shape="rect" coords="0,0,50%,50%". Если несколько активных областей одного изображения перекрывают друг друга, область, описанная тегом <AREA> первой, имеет приоритет.

Значения атрибута target такие же, как и у тега <A>.

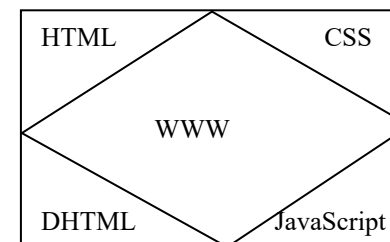
Необязательный атрибут nohref определяет неактивные участки карты и исключает переходы при щелчке мышью по данному фрагменту.

Пример 1. Пусть изображение состоит из четырех прямоугольников, а его полный размер составляет 340x60 пикселей. Каждый из этих прямоугольников может быть использован в качестве фрагмента карты ссылок.

HTML	CSS
DHTML	JavaScript

```
<IMG src="map1.gif"
      usemap="#refer">
<MAP name="refer">
  <AREA shape="rect"
        coords="0,0,169,29"
        href="html.html">
  <AREA shape="rect"
        coords="170,0,339,29"
        href="css.html">
  <AREA shape="rect"
        coords="0,30,169,59"
        href="dhtml.html">
  <AREA shape="rect"
        coords="170,30,339,59"
        href="javascript.html">
</MAP>
```

Пример 2. Пусть изображение состоит из многоугольников, а его полный размер составляет 370x120 пикселей. Координаты вершин многоугольников задаются последовательно, причем первая и последняя координаты должны совпадать, чтобы многоугольник был замкнутым.



```
<IMG src="map2.gif"
      usemap="#polygons">
<MAP name="polygons">
  <AREA shape="poly"
        coords="0,0,184,0,0,59,0,0"
        href="html.html">
  <AREA shape="poly"
        coords="185,0,369,0,369,60,184,0"
        href="css.html">
  <AREA shape="poly"
        coords="185,119,369,59,369,119,185,119">
```

```

    href="javascript.html">
<AREA shape="poly"
    coords="0,59,185,119,0,119,0,59"
    href="dhtml.html">
<AREA shape="poly"
    coords="185,0,369,59,185,119,0,59,185,0"
    href="web.html">
</MAP>

```

Карты ссылок можно создавать и другим способом. Изображение можно рассечь на отдельные фрагменты и «склеить» его по принципу мозаики в невидимой таблице с назначением отдельным фрагментам картинки свойств гиперссылки при помощи директивы <A>. Данный прием применяется не только при создании элементов, подобных image map, но и при подготовке рисованных заголовков или полей для размещения рекламного баннера. Однако аккуратно «разрезать» изображение при помощи традиционных графических редакторов, причем так, чтобы готовая «мозаика» правильно вписывалась в таблицу, весьма трудно. Для решения этой задачи web-мастеру помогают специальные утилиты – *фрагментаторы графики*.

8 МУЛЬТИМЕДИА НА WEB-СТРАНИЦЕ

8.1 Озвучивание web-страницы

8.2 Внедрение видео

8.3 Использование элемента IMG по воспроизведению видео

8.1 Озвучивание Web-страницы. Существует два способа обращения к мультимедиа-файлам: с помощью элемента привязки <A>, либо с применением специальных тегов.

Элемент привязки <A>. Этот элемент предназначен для задания ссылки на другой файл (в том числе и мультимедийный). Например,

Текст гиперссылки

После щелчка по тексту гиперссылки отобразится содержимое видеоклипа.

Специальные элементы позволяют вставлять (внедрять) мультимедиа файл в содержимое Web-страницы. Мультимедийный файл загружается одновременно с содержимым всей страницы. Это элементы , <APPLET>, <OBJECT>, <EMBED> и <BGSOUND>.

Форматы аудиофайлов. Звуковые эффекты создаются с применением звуковых (аудио) файлов. Форматы звуковых файлов различаются способами записи и областями применения. Наиболее распространены форматы:

AU (AUdio format, «звуковой формат», расширение .au) – стандартный формат для 8-битового звука, который является одним из старейших звуковых форматов в Internet (первоначально разработан для UNIX). Размер звуковых файлов AU относительно небольшой, однако качество звука невысокое.

WAV (Waveform Audio, «звук в форме волны», расширение .wav) – стандартный формат цифрового аудио в системе Windows.

MIDI (от Musical Instrument Digital Interface, «цифровой интерфейс музыкальных инструментов»), имеет расширение .midi или .mid) – формат, который не связан с реальным звуком. Этот формат содержит просто запись нот, коды музыкальных инструментов и прочую информацию, по которой синтезируется музыка.

MP3 (MPEG Audio Layer 3, расширение .mp3). Формат MP3 применительно к звуку отличается высокой степенью сжатия данных за счет некоторого ухудшения качества звука, незаметного на слух.

RealAudio («реальный звук», расширение .ra) – применяется для воспроизведения с повышенным качеством звучания. Файлы этого

формата поддерживают поток звуковых данных, то есть они могут воспроизводиться по мере поступления данных, не ожидая окончания загрузки файла.

Файлы перечисленных форматов являются загружаемыми, то есть для начала воспроизведения они должны быть полностью загружены.

Звуковой фон (элемент BGSOUND). Подключение звукозаписи в фоновом режиме обычно применяется для приветствий или музыкального сопровождения. Как только пользователь загрузит страницу, начнется автоматическое воспроизведение фонового звука. Для подключения звукового файла в фоновом режиме применяется элемент <BGSOUND>. Формат тега:

```
<BGSOUND src="URL"
  loop=значение
  volume=значение
  balance=значение>
```

Атрибут `src` задает URL-адрес звукового файла.

Атрибут `loop` определяет, сколько раз будет воспроизводиться файл. Например, тег <BGSOUND src="music.mp3" loop=5> означает, что звуковой файл `duet` будет воспроизведен 5 раз. Если `loop="infinite"`, звук будет повторяться до тех пор, пока пользователь находится на странице.

Атрибут `volume` назначает уровень воспроизведения звука и может принимать числовые значения от -10000 до 0. Можно только «приглушить» звучание, а увеличить уровень воспроизведения звука выше максимального уровня (значение 0) нельзя.

Атрибут `balance` – настройка стереобаланса. Значения этого атрибута выбираются из диапазона чисел от -10000 до 10000. Назначение любого крайнего значения (-10000 или 10000) приводит к звучанию только одного динамика. Введение тега

```
<BGSOUND src="duet.mid"
  loop=infinite
  volume=0
  balance=0>
```

приведет к звучанию файла `duet.mid` с максимальным уровнем звука и нормальным стереобалансом. Звучание будет продолжаться от момента загрузки до тех пор, пока пользователь находится на данной странице.

Внедрение аудио (элемент EMBED). Звуковое сопровождение в фоновом режиме лишает посетителя страницы возможности управления звуком. Управлять звуком можно, если на Web-страницу встроить

аудиоплеер с помощью тега <EMBED>, который размещается в любом месте внутри элемента BODY. Формат тега:

```
<EMBED src="URL"
  width="значение"
  height="значение"
  autostart="значение"
  hidden="значение"
  title="Всплывающая подсказка">
```

Атрибут `src` задает URL звукового файла.

Атрибуты `width` и `height` устанавливают размеры панели управления проигрывателя (ширину и высоту); значения этих атрибутов указываются в пикселах или в процентах.

Атрибут `autostart` назначает возможные варианты включения воспроизведения звука. Данный атрибут принимает значения `false` (включение звука пользователем) и `true` (автоматическое начало воспроизведения сразу после загрузки страницы).

Атрибут `hidden` используется для управления отображением панели проигрывателя. Значение `true` этого атрибута задает воспроизведение звука без отображения панели проигрывателя, а значение `false` – вывод панели на экран.

Атрибут `title` служит для задания всплывающей подсказки.

8.2 Внедрение видео. Форматы видеофайлов:

MPEG (Moving Picture Expert Group – «группа экспертов по движущимся изображениям», расширение `.mpg` или `.mpeg`) – предназначен для сжатия как аудио-, так и видеофайлов. Этот формат обеспечивает высокое качество и является своего рода стандартом в цифровом видео. Однако файлы этого формата имеют большой размер, и работать с ними трудно.

AVI (Audio Video Interleaved – «перемежающиеся аудио и видео»), имеет расширение `.avi` – применяется в системе Windows для хранения и воспроизведения аудио- и видеоданных.

QuickTime (имеет расширения `.mov` или `.qt`) – наиболее популярный видеоформат, разработанный компанией Apple. Этот формат не требует длительной загрузки по сети, обеспечен множеством модулей.

Внедрение видео (элемент EMBED). Простейшим способом размещения видео является использование тега <EMBED>. Форма его записи, атрибуты и допустимые значения атрибутов такие же, как и в случае вставки аудиофайлов. Например, <EMBED src="flyover.avi">

В результате загрузки видеофайла в окне браузера Internet Explorer появится панель проигрывателя Windows Media с набором управляющих кнопок и начнется воспроизведение видеоклипа.

8.3 Использование элемента *IMG* по воспроизведению видео. При просмотре страницы в Internet Explorer для вставки видеоклипа можно воспользоваться элементом . Для этого вместо атрибута src в тег нужно подставить атрибут dynsrc, значением которого является URL видеофайла.

```
<IMG dynsrc="flyover.avi">
```

При внедрении видео с помощью элемента панель управления проигрывателя на экран не выводится (по умолчанию).

При вставке видео с помощью элемента IMG можно использовать все допустимые атрибуты как и при вставке изображений. Однако элемент IMG, поддерживает для видео и специальные атрибуты:

start – определяет момент начала включения клипа. Если присвоить этому атрибуту значение fileopen, воспроизведение начнется сразу после загрузки видеофайла. Если присвоить значение start="mouseover", видео будет запущено при наведении на изображение указателя мыши;

loop – задает число воспроизведений видеофайла. Если значение этого атрибута равно infinite (или -1), видео будет циклически повторяться все время, пока пользователь находится на странице;

loopdelay – устанавливает время задержки (в миллисекундах) перед началом воспроизведения;

controls – управляет отображением панели проигрывателя (по умолчанию панель управления не отображается).

Гиперссылки на видео. Задание гиперссылки на видеофайл ничем не отличается от формирования любых других ссылок. Например,

```
<A href="speed.avi">
```

```
<IMG src="ski.gif"
```

```
title="Для просмотра щелкните по пиктограмме">
```

```
</A>
```

9 ФРЕЙМЫ

9.1 Формат задания фреймовой структуры

9.2 Целевые фреймы и элементы привязки

9.3 Базовый фрейм

9.4 Плавающие фреймы

9.1 Формат задания фреймовой структуры.

Фреймы – способ организации структуры сайта, при котором web-страница дробится на ряд отдельных составляющих и «собирается» в главном окне браузера из нескольких независимых или вложенных окон. При таком представлении каждый компонент страницы является самостоятельным документом HTML и встраивается в ту область экрана, которая задается директивой <FRAMESET>. Данный способ применяется в основном для дробления web-страницы на логические разделы: например, в верхнем фрейме выводится рекламный баннер, в левом – элементы навигации, в правом – основной текст страницы, в нижнем – сообщение об авторских правах и адрес электронной почты разработчика ресурса. При этом нажатие на любую из навигационных кнопок приводит к изменению содержимого лишь одного окна, все остальные фреймы остаются без изменений. Если содержимое фрейма не умещается в видимые границы окна, браузер отображает полосы прокрутки.

В случае применения фреймов в индексном html-файле (index.html) раздел <BODY> заменяется на раздел <FRAMESET>. Все остальные документы, отображаемые в отдельных фреймах, имеют стандартное представление.

Структура html-документа, содержащего фреймы:

```
<HTML>
  <HEAD>
    <TITLE>
      Заголовок
    </TITLE>
  </HEAD>
  <FRAMESET rows="значение"
    cols="значение"
    frameborder="значение"
    border="значение"
    bordercolor="значение">
    <FRAME name="имя"
      target="значение"
```

```

scrolling="yes, no или auto"
src="URL"
frameborder="значение"
noresize
marginwidth="значение"
marginheight="значение">
<NOFRAMES>
Текст, отображаемый в броузерах,
не поддерживающих фреймы.
</NOFRAMES>
</FRAMESET>
</HTML>

```

Тег <FRAMESET>, задающий параметры отображения всех фреймов в документе, содержит необходимое количество тегов <FRAME>, описывающих каждый отдельный фрейм. Причем некоторые атрибуты, применяемые совместно с командой <FRAME>, могут отменять параметры, заданные в атрибутах команды <FRAMESET>. Директива описания фреймов допускает вложение одного элемента <FRAMESET> в другой.

Атрибут тега <FRAMESET>:

- `rows` определяет количество и размеры горизонтальных фреймов. Значением этого атрибута является список размеров каждого фрейма, разделенных запятыми.
- `cols` определяют количество вертикальных отрезков во фреймовом наборе.
- `frameborder` определяет границы фреймов. Параметр этого атрибута может принимать одно из двух значений: 1, если разделители нужны (используется по умолчанию), или 0, если программист желает сделать их невидимыми.
- `border` указывает толщину разделителей в пикселах.
- `bordercolor` определяет цвет разделителей.

Параметры атрибутов `rows` и `cols` можно задавать в пикселах, процентах и в относительных длинах. Если атрибут `rows` в теге <FRAMESET> не указан, каждый столбец будет занимать всю длину страницы, если не указан атрибут `cols`, каждая строка растянется на всю видимую ширину экрана. Параметры фреймов задаются слева направо для горизонтальных элементов и сверху вниз для вертикальных. Например, запись <FRAMESET cols="50%, 50%"> разделит страницу вертикально на две равные части, запись <FRAMESET

rows="250, 10%, *"> задаст горизонтальное разделение страницы на три участка: первый – с фиксированной высотой 250 пикселей, второй – высотой в 10 % от оставшегося пространства, третий же займет все пространство, которое останется свободным после создания первых двух окон. Разделения web-страницы на сетку 2x3 ячейки можно добиться, например, применением следующих значений атрибутов директивы <FRAMESET>:

```

<FRAMESET rows="40%, 60%"
           cols="33%, 34%, *">.

```

Атрибуты тега <FRAME>:

- `name` – задает уникальное имя для данного конкретного фрейма, набранное с использованием символов латинского алфавита.
- `target` – содержит информацию о целевом фрейме (значение по умолчанию – "contents"). Целевые фреймы для отображения документов можно назначать не только в зависимости от их позиции в окне браузера. Целевой фрейм определяется дополнительными значениями атрибута: `_blank` – устанавливает загрузку документа в новое окно; `_self` – документ, с которым установлена гиперсвязь, открывается в текущем фрейме; `_parent` – отображение осуществляется в родительском окне фреймов, если текущий фрейм не имеет предков, то в текущем окне; `_top` – при активизации гиперссылки фреймы перестают отображаться, а содержимое целевого документа выводится в отдельном окне.
- `scrolling` – определяет наличие ("yes"), отсутствие ("no") или отображение по мере необходимости ("auto") полос прокрутки в текущем фрейме.
- `src` – задает адрес открываемой в данном фрейме страницы.
- `frameborder` – определяет границы фреймов и аналогичен соответствующему атрибуту тега <FRAMESET>
- `noresize` – устанавливает запрет на изменение пользователем размеров окна фрейма.
- `marginwidth` определяет размер в пикселах пустого пространства, оставляемого во фрейме в качестве правого и левого полей.
- `marginheight` задает верхний и нижний отступы.

Директива <NOFRAMES> позволяет записать произвольное количество html-кода, который будет динамически преобразован в самостоятельную web-страницу в случае, если клиентский браузер не поддерживает фреймы.

9.2 Целевые фреймы и элементы привязки. При активизации ссылки выполняется загрузка страницы в определенный фрейм, который называется *целевым*. Целевому фрейму нужно присвоить уникальное имя с помощью атрибута name тега <FRAME>.

Представление гиперссылок для страниц, использующих фреймы, также требует обязательного использования атрибута target:

```
<A href="URL"
  target="значение">
  Текст или элемент IMG
</A>
```

Если вместо значения атрибута target подставить имя фрейма, заданное в качестве параметра атрибута name тега <FRAME>, при активизации ссылки целевой документ загружается в окне с указанным именем.

9.3 Базовый фрейм. Документ, на который указывает ссылка, загружается в целевой фрейм. Этот фрейм определяется атрибутом target. Для того, чтобы все документы, соответствующие ссылкам, загружались в один и тот же фрейм, можно воспользоваться элементом <BASE>.

Фрейм, в который производится загрузка по гиперссылкам, называется *базовым фреймом*. Он задается значением атрибута target в одиночном теге <BASE>. То есть, задание базового фрейма позволяет только один раз в документе указать конкретный фрейм загрузки ссылок, а не записывать его в каждом элементе привязки.

9.4 Плавающие фреймы. Используются для размещения в обычном HTML-документе (в пределах элемента <BODY>) можно с помощью элемента IFRAME. Фреймы, создаваемые этими элементами, называются *плавающими*. Задание плавающего фрейма аналогично вставке обычного рисунка.

Формат задания плавающего фрейма:

```
<IFRAME name="значение"
  src="URL"
  width="значение"
  height="значение"
  align="значение"
  marginwidth="значение"
  marginheight="значение"
  frameborder="значение">
```

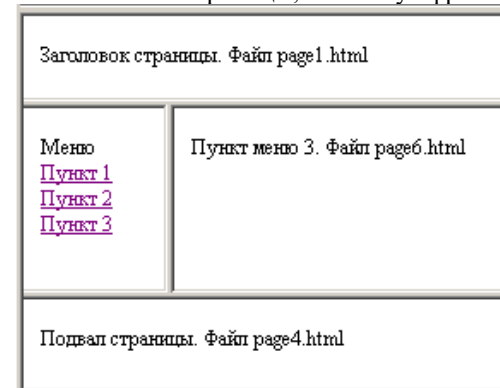
```
scrolling="значение">
</IFRAME>
```

В отличие от обычных фреймов, размеры плавающего фрейма не могут быть изменены пользователем.

Атрибуты тега:

- name – задает имя целевого фрейма;
- src – определяет адрес загружаемого файла;
- width – определяет ширину плавающего фрейма;
- height – определяет высоту плавающего фрейма;
- align – устанавливает выравнивание фрейма относительно окна браузера;
- marginwidth, marginheight – определяют размеры свободных полей между рамкой фрейма;
- frameborder – устанавливает вывод рамки фрейма;
- scrolling – определяет наличие полосы прокрутки.

Пример 1. Компонировка страницы, используя фреймы.



Файл frame1.html

```
<HTML>
<HEAD>
  <TITLE>Первый вариант компоновки</TITLE>
</HEAD>
<FRAMESET ROWS="54, *, 54">
  <FRAME name="banner" src="page1.html">
  <FRAMESET cols="100, *">
    <FRAME name="menu" src="page2.html">
    <FRAME name="contents" src="page3.html">
```

```

</FRAMESET>
<FRAME name="end" src="page4.html">
<NOFRAMES>
    Изображение фреймов невозможно
</NOFRAMES>
</FRAMESET>
</HTML>

```

Файл page1.html

```

<HTML>
<BODY>
    Заголовок страницы. Файл page1.html
</BODY>
</HTML>

```

Файл page2.html

```

<HTML>
<BODY>
    Меню<BR>
<A href="page3.html" target="contents">Пункт 1</A><BR>
<A href="page5.html" target="contents">Пункт 2</A><BR>
<A href="page6.html" target="contents">Пункт 3</A><BR>
</BODY>
</HTML>

```

Файл page3.html

```

<HTML>
<BODY>
    Пункт меню 1. Файл page3.html
</BODY>
</HTML>

```

Файл page4.html

```

<HTML>
<BODY>
    Подвал страницы. Файл page4.html
</BODY>
</HTML>

```

Файл page5.html

```

<HTML>
<BODY>
    Пункт меню 2. Файл page5.html
</BODY>
</HTML>

```

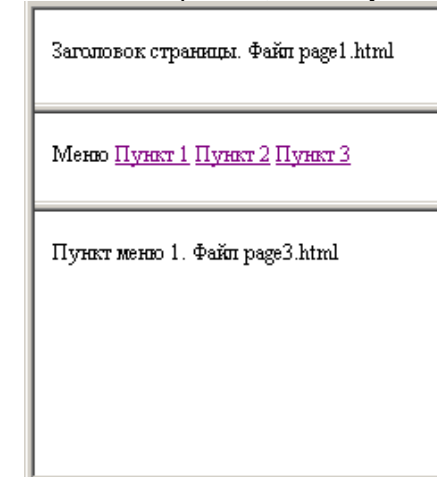
Файл page6.html

```

<HTML>
<BODY>
    Пункт меню 3. Файл page6.html
</BODY>
</HTML>

```

Пример 2. Компоновка страницы, используя базовый фрейм.



Файл frame2.html

```

<HTML>
<HEAD>
    <TITLE>Второй вариант компоновки</TITLE>
</HEAD>
<FRAMESET rows="54,54,*">
    <FRAME name="banner" src="page1.html">
    <FRAME name="menu" src="page2.html">
    <FRAME name="contents" src="page3.html">
</FRAMESET>
</HTML>

```

Файл page2.html

```

<HTML>
<BASE target="contents">
<BODY>
    Меню
    <A href="page3.html">Пункт 1</A>
    <A href="page5.html">Пункт 2</A>

```

```
<A href="page6.html">Пункт 3</A>
</BODY>
</HTML>
```

На первый взгляд фреймы кажутся весьма привлекательными с точки зрения удобства обслуживания сайта: например, для того чтобы изменить состав навигационных элементов, нет необходимости вносить изменения во все страницы вашего ресурса, достаточно лишь подправить документ, отображающийся в соответствующем фрейме.

Недостатки фреймовой структуры документа.

Во-первых, фреймы весьма сложны в разработке и настройке: достаточно трудно оптимизировать такую страницу для корректного отображения в разных версиях браузеров со всеми допустимыми экранными разрешениями. В некоторых фреймах возможно периодическое появление полос прокрутки, затрудняющих просмотр документа, а жесткий запрет на отображение средств скроллинга может повлечь за собой исчезновение части отображаемых во фреймах страниц за границей видимой области экрана.

Во-вторых, фреймы распознаются не всеми браузерами.

В-третьих, применение фреймов существенно затрудняет управление навигацией сайта при помощи встроенных в браузер кнопок «Вперед» и «Назад», поскольку они рассчитаны, в первую очередь, на работу с документами размером в целый экран, но никак не с отдельными окнами, в которые загружено множество документов. Нажатие на одну из этих кнопок иногда может привести к совершенно непредсказуемым последствиям.

В-четвертых, страницы, использующие фреймы, не всегда корректно индексируются поисковыми серверами. Проблема заключается в том, что большинство поисковых машин индексирует не страницу, а текст.

Поэтому использовать фреймы лучше всего только тогда, когда это действительно необходимо, например, при создании web-чата или интерактивной доски объявлений. Во всех остальных случаях от их применения лучше отказаться.

10 ФОРМЫ

10.1 Формат задания формы

10.2 Элементы формы

10.1 Формат задания формы.

Формой называется объединение логически связанных элементов управления в документе. Формы являются основным средством для ввода и обработки данных.

Формы применяются для передачи данных от html-документа интерактивным элементам сайта, например сценариям. Поместив в форму какие-либо значения, посетитель сервера нажимает мышью на соответствующую кнопку, после чего введенная им информация передается скрипту, который принимает управление процессом обработки данных. Подобный принцип реализован в многочисленных электронных конференциях, досках объявлений, гостевых книгах и web-чатах Всемирной сети.

Формат задания формы:

```
<FORM name="имя_формы"
      action="URL"
      method="метод_передачи_данных"
      enctype="тип_кодировки"
      target="значение">
```

Содержание формы

```
</FORM>
```

В качестве параметра атрибута `action` в кавычках указывается строка вызова скрипта, который использует данная форма (при нажатии кнопки `Submit`), например, `"http://www.myserver.by/имя_сценария.php"`. Если в качестве значения атрибута `action` указать обращение к электронной почте, например,

```
action="mailto:ivanov@gsu.unibel.by"
```

браузер автоматически отправит результаты, введенные в форму, по указанному адресу. Для корректной интерпретации данных используется параметр `enctype="text/plain"`

Если же атрибут `action` отсутствует, то в качестве значения `action` подставляется URL самого документа, т.е. после отправки формы текущая страница перезагружается, возвращая все элементы формы к их значениям по умолчанию.

Значение атрибута `method` устанавливает метод передачи данных из формы на сервер: "GET" с помощью стандартного интерфейса HTTP или "POST" – по каналам электронной почты.

При задании значения "GET" передача данных происходит в один этап адресной строкой. Пары «имя=значение» присоединяются в этом случае к адресу после вопросительного знака и разделяются между собой амперсандом (&). Метод GET используется для передачи данных с небольших форм с короткими полями.

При задании значения "POST" передача данных происходит, как минимум, в два этапа: браузер устанавливает связь с сервером, указанным в атрибуте `action`; затем отдельной передачей происходит посылка дополнительных данных. Переданные данные не отображаются в командной строке. Метод POST используется для передачи данных форм, имеющих много длинных полей.

Атрибут `enctype` устанавливает тип для данных, отправляемых вместе с формой. Обычно не требуется определять значения параметра `enctype`, однако если используется поле для отправки файла (`INPUT type=file`), то следует задать параметр `enctype="multipart/form-data"`.

Атрибут `target` определяет окно, в которое будет загружаться итоговая web-страница. Значения этого атрибута такие же, как и при задании фреймовой структуры.

Содержание формы описывается тегом `<INPUT>`, запись которого в общем виде следующая:

```
<INPUT type="тип_элемента"
      name="имя"
      value="строка"
      checked
      size="целое_число"
      maxlength="целое_число"
      align="значение"
      src="URL"
      tabindex="значение">
```

Атрибуты тега `INPUT`:

- `type` задает тип элемента формы;
- `name` задает уникальное имя для каждого элемента формы;
- `value` указывает первоначальное значение текущего поля;

- `checked` устанавливает выделенный объект из нескольких в случае, если значением атрибута `type` является `radio` или `checkbox`;
- `size` определяет размер текстового поля в символах;
- `maxlength` определяет максимально возможную длину текстового поля в символах для полей ввода текста;
- `align` определяет положение элементов формы на web-странице;
- `src` используется совместно с атрибутом `type=image` и задает URL нужного изображения;
- `tabindex` позволяет установить порядок перемещения фокуса по элементам формы при нажатии клавиши Tab.

10.2 Элементы формы. На рис. 10.1 представлены основные элементы формы.

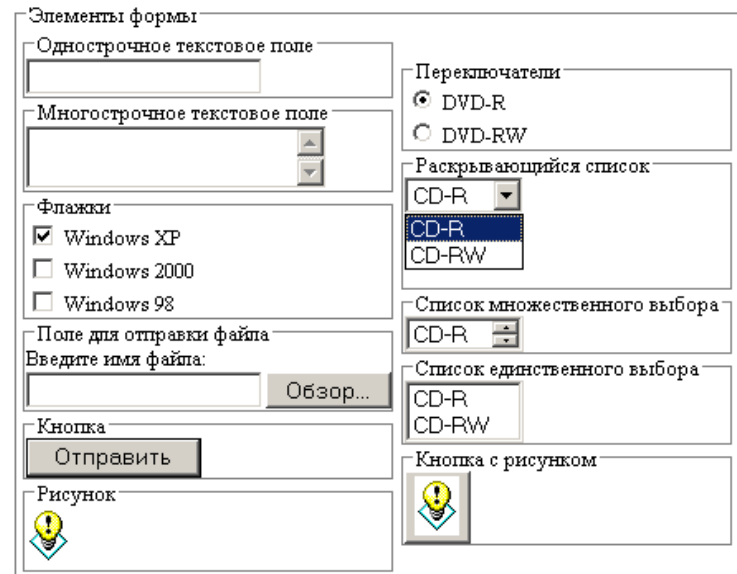


Рис. 10.1 Основные элементы формы

Однострочное текстовое поле. Оно называется полем ввода или полем редактирования и предназначено для ввода пользователем строки текста. Формат записи:

```
<INPUT type="text"
      name="имя_поля">
```

```
value="начальный текст, содержащийся в поле"
size="ширина поля"
maxlength="максимальное количество вводимых символов">
```

Пример:

```
<INPUT type=text
size=40
name=user_name
value="Введите ваше имя">
```

Поле для ввода пароля. Это обычное текстовое поле, вводимый текст в котором отображается звездочками. Формат задания:

```
<INPUT type=password
name="имя_поля"
value="начальный текст, содержащийся в поле"
size="ширина поля"
maxlength="максимальное количество вводимых символов">
```

Скрытые поля. Это поля, которые не отображаются на странице и встраиваются в HTML-файл, когда необходимо передать серверу техническую информацию. Скрытые поля служат доступной альтернативой файлам cookies – специальным файлам, в которых сохраняются индивидуальные настройки пользователя и позволяющим, например, восстановить последнее состояние формы при повторном посещении пользователем содержащей эту форму страницы. Формат задания:

```
<INPUT type=hidden
name="имя_поля"
value="текст, содержащийся в поле">
```

Пример:

```
<INPUT type=hidden
name="form1"
value="с3576-236-2113">
```

Многострочное текстовое поле. Используется для передачи текста большого размера. Формат задания:

```
<TEXTAREA name="имя элемента"
rows="целое число"
cols="целое число"
wrap=значение
disabled
readonly>
```

Текст, выводимый в текстовом поле по умолчанию

```
</TEXTAREA>
```

Атрибуты тега:

- rows и cols указывают соответственно максимально допустимое количество строк вводимого текста и символов в строке. В случае если набираемый пользователем текст не умещается в видимую часть текстового контейнера, по краям поля появляются вертикальные и горизонтальные полосы прокрутки.
- wrap управляет переносом слов и имеет следующие значения:
 - wrap=off – запрет автоматического переноса; при этом сохраняются переносы, определенные пользователем;
 - wrap=virtual – перенос слов при отображении браузером, а серверу введенные данные передаются одной строкой;
 - wrap=physical – сохраняется перенос слов как при отображении браузером, так и при передаче серверу.
- disabled – блокирует доступ и изменение текстового поля. Поле *отображается серым цветом*, недоступно для активации пользователем, и не может получить фокус. Состояние этого поля можно изменять с помощью скриптов.
- readonly – текстовое поле недоступно для изменения пользователем, в него не допускается вводить новый текст или модифицировать существующий, оно не может получить фокус. Однако поле отображается обычным цветом.

Пример:

```
<TEXTAREA name="message"
rows=25
cols=40>
```

Введите сюда текст сообщения

```
</TEXTAREA>
```

Поле выбора файлов. Генерирует на экране кнопку, при нажатии на которую на экране появляется Проводник Windows, позволяющий присоединить к отсылаемым на сервер данным любой файл с локального компьютера пользователя. Рядом с кнопкой отображается небольшое текстовое поле, куда автоматически заносится имя отсылаемого файла и путь к нему на локальном диске. Поле выбора файлов работает корректно лишь при методе пересылки Post и формате кодировки multipart/form-data. Формат задания:

```
<INPUT type=file
name="имя"
size="ширина поля"
maxlength="максимальная длина текста">
```

Пример:

```
<FORM enctype="multipart/form-data"
  action="URL"
  method=POST>
  Введите имя файла:<BR>
  <INPUT type=file name=myfile> <BR>
  <INPUT type=submit value="Отправить">
</FORM>
```

Флажки. Флажки используют, когда необходимо выбрать два или более варианта из предложенного списка. Элемент представляет собой простую форму выбора, принимающую одно из двух состояний: «отмечено» – «не отмечено». Несколько флажков могут объединяться в группу, которая будет отвечать набору параметров выбора. Данный элемент оперирует с **булевыми переменными**, то есть переменными, каждая из которых может принимать значение true или false. Формат задания:

```
<INPUT type=checkbox
  name="имя флажка или группы флажков"
  value="значение установленного флажка"
  checked
  title="всплывающая подсказка">
```

По умолчанию начальное положение флажка считается не установленным. Чтобы задать начальное положение установленного флажка, надо дополнить его атрибутом checked.

Значением установленного флажка является строка, заданная атрибутом value.

Пример задания группы флажков:

```
<INPUT type=checkbox
  name=system
  value="WXP" checked> Windows XP<BR>
<INPUT type=checkbox
  name=system
  value="W2000"> Windows 2000<BR>
<INPUT type=checkbox
  name=system
  value="W98"> Windows 98<BR>
```

В рассмотренном примере надписи рядом с флажками созданы как простой текст. Для того, чтобы флажок устанавливался не только щелчком непосредственно по квадратику флажка, но и щелчком по надписи, необходимо связать надпись с флажком с помощью тега LABEL, в котором содержится ссылка на связанный элемент управления с

помощью атрибута for. Этому атрибуту ставится в соответствие идентификатор id.

Пример связывания надписи с флажком:

```
<INPUT type=checkbox
  id=WindowsXP
  name=system
  value="WXP" checked>
<LABEL for=WindowsXP>Windows XP</LABEL><BR>
<INPUT type=checkbox
  id=Windows2000
  name=system
  value="W2000">
<LABEL for=Windows2000> Windows 2000</LABEL><BR>
<INPUT type=checkbox
  id=Windows98
  name=system
  value="W98">
<LABEL for=Windows98> Windows 98</LABEL><BR>
```

Переключатели (кнопки выбора или радиокнопки) применяется в случае, когда какая-либо логическая переменная может принимать только одно значение из множества возможных. Формат задания:

```
<INPUT type=radio
  name="имя переключателя или группы"
  value="значение установленного переключателя"
  checked
  title="всплывающая подсказка">
```

Все элементы radio одной группы обозначаются одним и тем же значением атрибута name. Использование радиокнопок требует явного указания значений атрибута value, одна из кнопок должна быть выделена атрибутом checked. Если атрибут checked не присвоен ни одному из переключателей группы, браузер при загрузке установит по умолчанию первый переключатель. При обработке формы на сервере будет отправлено значение установленного переключателя.

Пример использования:

```
<INPUT type=radio
  name=DVD
  value="DVDR" checked> DVD-R<BR>
<INPUT type=radio
  name=DVD
  value="DVDRW" checked> DVD-RW<BR>
```


Как и в случае флажков, надписи можно связать с соответствующими переключателями так, чтобы каждый переключатель устанавливался при щелчке по надписи. Для связывания каждому переключателю должен быть присвоен уникальный идентификатор, а все переключатели должны образовывать группу с определенным именем.

Пример связывания надписи с переключателем:

```
<INPUT type=radio
      id=disk1
      name=DVD
      value="DVDR" checked>
<LABEL for=disk1> DVD-R</LABEL><BR>
<INPUT type=radio
      id=disk2
      name=DVD
      value="DVDRW" checked>
<LABEL for=disk2> DVD-RW</LABEL><BR>
```

Кнопки. Это элементы управления, которые используются для представления формы (кнопка submit), сброса данных формы (кнопка reset), создания эффектов для кнопки (кнопка button). Кнопку можно создать двумя способами:

1. Использование тега INPUT. Формат задания:

```
<INPUT type=button
      name="имя кнопки"
      value="надпись на кнопке">
```

2. Использование тега BUTTON. На таких кнопках можно размещать любые элементы HTML, в том числе изображения и таблицы и изменять вид кнопки. Формат задания:

```
<BUTTON>
Надпись или изображение
</BUTTON>
```

В HTML предусмотрены два типа кнопок, которые создаются без использования значения button. Это кнопки специального назначения: Подача запроса (submit) и Сброс (reset).

Кнопка submit предназначена для запуска процедуры передачи формы на сервер. Формат задания:

```
<INPUT type=submit
      name="имя кнопки"
      value="надпись на кнопке">
```

```
или
<BUTTON type=submit>
Надпись на кнопке
```

```
</BUTTON>
```

Если атрибут value отсутствует, то кнопка по умолчанию имеет надпись «Подача запроса». В версиях Internet Explorer 4.0 и выше кнопка submit может работать как кнопка по умолчанию, то есть она активизируется при нажатии клавиши Enter. В форме можно применять несколько кнопок submit.

Кнопка reset предназначена для приведения формы в начальное положение (сброс всех введенных данных). Формат задания:

```
<INPUT type=reset
      name="имя кнопки"
      value="надпись на кнопке">
```

или

```
<BUTTON type=reset>
Надпись на кнопке
</BUTTON>
```

Если атрибут value отсутствует, то кнопка по умолчанию имеет надпись «Сброс».

Кнопка с изображением. Создает кнопку отсылки, аналогичную элементу submit, но с использованием графического изображения. Обычно применяется в случаях, когда стандартная серая прямоугольная кнопка «не вписывается» в дизайн сайта. Формат записи:

```
<INPUT type=image
      name="имя кнопки"
      src="URL изображения"
      value="надпись на кнопке">
```

В этом случае тег INPUT может содержать все атрибуты тега IMG.

Пример задания кнопок:

```
1. Обычная кнопка
<INPUT type=button
      name=press
      value="Нажми меня!">
```

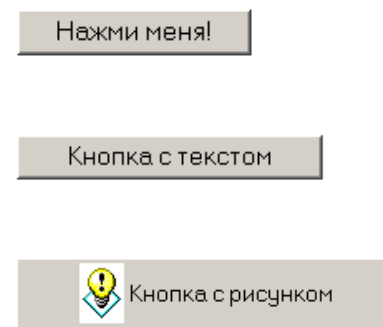
```
2. Обычная кнопка
<Button>
```

```
Кнопка с текстом
</Button>
```

```
3. Кнопка с рисунком
<Button>
```

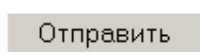
```
<IMG src="tips.gif"
      align=absmiddle>
```

```
Кнопка с рисунком </Button>
```



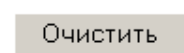
4. Кнопка submit

```
<INPUT type=submit  
value="Отправить">
```



5. Кнопка reset

```
<INPUT type=reset  
value="Очистить">
```



6. Кнопка с изображением

```
<INPUT type=image  
src="tips.gif">
```



Списки. Эти элементы представляют пользователю список вариантов для выбора. Существует три типа списков:

- *раскрывающийся список*, представляющий собой однострочное поле с треугольной стрелкой, которая раскрывает список;
- *поле-список*, в котором на экран выводится заданное число строк; для просмотра всех строк список может быть снабжен полосой прокрутки;
- *список со множественным выбором*, позволяющий благодаря полосе прокрутки просматривать все позиции списка и выбирать одновременно несколько позиций.

Формат записи:

```
<SELECT name="имя списка"  
size="целое число"  
multiple>  
<OPTION value="значение" selected>
```

Пункт 1

```
</OPTION>  
</SELECT>
```

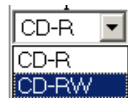
Атрибуты тега:

- `multiple` включает режим выбора нескольких элементов из списка, т.е. определяет список со множественным выбором;
- `size` устанавливает высоту списка. Если значение `size=1`, то список становится раскрывающимся. При добавлении параметра `multiple` список отображается как «крутилка»;
- `selected` делает текущий элемент списка выделенным.

Примеры задания списка:

1. Раскрывающийся список

```
<SELECT name=CD>  
<OPTION value=1>CD-R</OPTION>  
<OPTION value=2>CD-RW</OPTION>  
</SELECT>
```



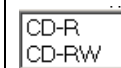
2. Список множественного выбора

```
<SELECT multiple size=1>  
<OPTION value=1>CD-R</OPTION>  
<OPTION value=2>CD-RW</OPTION>  
</SELECT>
```



3. Список единственного выбора

```
<SELECT size=2>  
<OPTION value=1>CD-R</OPTION>  
<OPTION value=2>CD-RW</OPTION>  
</SELECT>
```



Группирование элементов управления. Чтобы страница имела законченный вид, элементы формы должны быть распределены по группам. Формат задания группы:

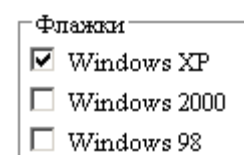
```
<FIELDSET>  
<LEGEND>  
Легенда группы элементов  
</LEGEND>
```

```
...  
</FIELDSET>
```

Теги `<LEGEND>` вводят надпись, которая помещается в разрыв рамки, обрамляющей группу.

Пример:

```
<FIELDSET>  
<LEGEND>Флажки</LEGEND>  
<INPUT type=checkbox checked>  
Windows XP<BR>  
<INPUT type=checkbox>  
Windows 2000<BR>  
<INPUT type=checkbox>  
Windows 98<BR>  
</FIELDSET>
```



11 МЕТА-ОПРЕДЕЛИТЕЛИ

11.1 МЕТА-определители, использующие атрибут name.

11.2 МЕТА-определители, использующие директиву http-equiv.

МЕТА-определители – это элементы заголовков HTML-документов, несущие исключительно служебные функции. Они не влияют на отображение web-страниц в браузере и не могут каким-либо образом изменить их содержимое. МЕТА-определители описывают свойства документа.

МЕТА-определители записываются в заголовке документа HTML между тегами <HEAD> и </HEAD> после тега <TITLE>.

Существуют два вида МЕТА-определителей: МЕТА-определители, использующие атрибут name и МЕТА-определители, использующие директиву http-equiv.

11.1. МЕТА-определители, использующие атрибут name. Формат записи:

```
<META name="значение"  
      content="значение2">
```

Виды МЕТА-определителей:

- <META name="description"
 content="краткое описание содержимого сайта">

Эта директива применяется для облегчения и ускорения процесса индексации ресурса поисковыми машинами. Большинство автоматических поисковых серверов оснащены специальными программными модулями, отыскивающими во Всемирной сети web-страницы, соответствующие введенному пользователем тематическому запросу. В этом случае в динамически создаваемый поисковым сервером файл отчета включается либо информация, встречающаяся первой в тексте документа HTML, либо данные, указанные в теге META. Таким образом, наличие МЕТА-определителя увеличивает шансы ресурса попасть в первую десятку сайтов, которые «найдёт» поисковый сервер.

- <META name="keywords"
 content="перечень ключевых слов через запятую">

Тег содержит перечисление ключевых слов, на которые, должны «реагировать» поисковые машины, включая сайт в файл отчета. Не рекомендуется указывать в нем более тридцати ключевых слов подряд. Поскольку ряд поисковых машин не учитывает регистр, в котором записаны ключевые слова, рекомендуется указывать в поле content

оба варианта их записи: как в строчном, так и в заглавном регистре. Хорошим тоном считается указывать в атрибуте content описание сайта на русском и английском языках.

Пример:

```
<HEAD>  
<TITLE>Разведение белых медведей в домашних условиях</TITLE>  
<META name="description"  
      content="Сайт, посвященный особенностям разведения бе-  
лых медведей в домашних условиях">  
<META name="keywords"  
      content="белые медведи, БЕЛЫЕ МЕДВЕДИ, медведь,  
МЕДВЕДЬ, разведение, РАЗВЕДЕНИЕ, животные, ЖИВОТНЫЕ">  
</HEAD>
```

- <META name="abstract"
 content="описание содержимого сайта и перечень ключевых выражений через запятую">

Директива рассчитана на обработку поисковыми серверами, имеющими отличный от стандартного алгоритм индексации html-документов. Данный тег используется редко.

- <META name="subject"
 content="главная тема сайта, выраженная одним-двумя словами">
- <META name="page-type"
 content="Тип, к которому относится страница">

Директива применяется с целью автоматического занесения ссылки на сайт в какую-либо тематическую рубрику поискового сервера. Наиболее «нейтральным» вариантом представления поля content является фраза «information resource».

- <META name="distribution" content="Global">
- <META name="rating" content="General">

Последние два META-тега определяют область действия информации, представленной на сайте. Рекомендуется оставить их «как есть».

- <META name="title" content="Заголовок web-страницы">
- <META name="page-topic"
 content="Заголовок web-страницы">

Эти МЕТА-определители предназначены для поисковых машин, не умеющих работать с тегом <TITLE>. В поле content данных директив можно скопировать текстовое содержимое, записанное в заголовке страницы.

- `<META name="site-created" content="Дата">`
Декларирует дату создания web-сайта, которая иногда отображается в файле отчета некоторых поисковых серверов в качестве дополнительной информации о ресурсе. Дата записывается в формате МЕСЯЦ-ДЕНЬ-ГОД с ведущим нулем, например, 16-12-2006.
- `<META name="expires" content="Дата">`
Предполагаемая дата закрытия сайта, после которой упоминание о нем будет удалено из базы данных поискового сервера. Дата записывается в том же формате, что и дата создания страницы.
- `<META name="revisit" content="N days">`
- `<META name="revisit-after" content="N days">`
Эти теги несут абсолютно идентичную функциональную нагрузку и указывают поисковому серверу, через сколько дней следует вновь обратиться к ресурсу для его переиндексации. Данные META-определители применяют, если страница регулярно обновляется, причем эти обновления затрагивают ее тематическое содержание, указанное в других META-тегах. Включением данной команды в код HTML можете избежать процедуры повторной регистрации сайта в поисковых машинах – сервер сделает это автоматически. Вместо ключа N в поле content данной подставляется любое целое число от 1 до 30.
- `<META name="robots" content="noindex">`
Директива запрещает поисковым машинам индексировать данную web-страницу.
- `<META name="robots" content="nofollow">`
Тег запрещает поисковому серверу следовать по гиперссылкам, установленным на данной странице.
- `<META name="content-language" content="язык">`
Директива декларирует язык, на котором представлена опубликованная на сайте информация. В поле content вносится название языка, написанное в английской транскрипции, например «russian» или «english».
- `<META name="resource-type" content="document">`
Указание на тип ресурса. Рекомендуется оставить без изменений.
- `<META name="audience" content="all">`
Определяет аудиторию сайта.
- `<META name="robots" content="index, all">`
Указания на правила регистрации ресурса поисковыми серверами. При указании в поле content параметра «all» будут проиндексированы все документы сайта, при указании параметра «index» – только индексные страницы.

- `<META name="Copyright" content="Имя вебмастера, Copyright 2006">`
- `<META name="Autor" content="Имя вебмастера, Copyright 2006">`
Декларируют ссылку на авторские права. Эта информация отображается в файле отчета некоторых поисковых машин.
- `<META name="reply-to" content="адрес электронной почты">`
Указание на адрес электронной почты автора проекта. Иногда показывается в файлах отчета поисковых серверов.
- `<META name="Owner" content="Название компании или имя владельца">`
- `<META name="Address" content="Адрес офиса">`
- `<META name="Owner-type" content="corporation, company/small или personal">`
Информация о владельце ресурса. В поле content первого META-определителя заносится полное название фирмы-владельца ресурса или имя владельца, если он – частное лицо, второй тег содержит адрес офиса, по которому можно обратиться за дополнительной информацией, третий указывает на тип фирмы: corporation – для больших корпораций, company/small – для малых компаний и, наконец, personal – для частных предпринимателей и физических лиц.
- `<META name="Publisher-name" content="Название компании или имя лица, опубликовавшего сайт">`
- `<META name="Publisher-type" content="corporation. company/small или personal">`
- `<META name="Publisher- location" content="ru">`
Для первых двух тегов справедливо все то же самое, что и для предыдущих META-определителей, за исключением того, что они представляют информацию не о владельце ресурса, а о том, кто его опубликовал. В общем случае издатель web-страницы не обязательно должен быть ее владельцем, например, когда какая-либо фирма заказывает себе сайт у стороннего web-мастера или дизайн-агентства. Как раз на такую ситуацию и рассчитаны данные директивы. В поле content META-тега с именем «Publisher-location» заносится обычно обозначение домена первого уровня страны издателя сайта.
- `<META name="home-url" content="URL">`

Указывается адрес ресурса в сети Интернет, который издатель сайта считает своей домашней страничкой. Обычно этот тег используется для представления URL разработчика web-сайта, если сам сайт был создан для сторонней фирмы или частного лица.

- `<META name="Generator" content="Название программы">`

Редкий и абсолютно бесполезный тег, не несущий ни малейшей смысловой нагрузки. Данный META-определитель был придуман разработчиками html-редакторов для рекламы собственных программных продуктов. Практически все программы, предназначенные для верстки web-страниц, вставляют этот тег в код HTML автоматически, не спрашивая разрешения у пользователя.

11.2 META-определители, использующие директиву http-equiv. В отличие от первого вида META-тегов, имеющих чисто информационное назначение, теги второй категории играют определенную роль при передаче документов посредством протокола HTTP. Эти директивы используются для придания web-страницам некоторых примечательных свойств, которые в ряде случаев могут быть полезны web-дизайнеру.

- `<META http-equiv="Refresh" content="N; URL=адрес">`

Автоматический переход браузера по указанному в поле content адресу через N секунд. Данная функция, называемая «редиректом», или «перенаправлением» пользовательского браузера на какой-либо ресурс, применяется тогда, когда сайт сменил адрес, переместился на другой хост. В этом случае на сервере со старым адресом размещается файл index.html, содержащий описанную выше директиву и соответствующую поясняющую надпись, например: «Извините, наш сайт переехал по адресу <адрес>. Через несколько секунд данная страница будет загружена автоматически». Иногда такая команда применяется также для автоматического переключения пользователя с заглавной («splash») на стартовую страницу сервера.

- `<META http-equiv="Refresh" content="N">`

Текущая web-страница будет автоматически перезагружаться в браузер каждые N секунд.

- `<META http-equiv="Expires" content="Tue, 02 Jan 2007 01:00:00 GMT">`

Определяет дату «устаревания» страницы для протокола HTTP. Зачастую в данной директиве указывают заведомо устаревшую дату: такой подход позволяет избежать **кэширования** документа на диск. Большинство браузеров сохраняют загружаемые из Интернета html-документы в специальную папку на жестком диске пользовательского

компьютера при повторном обращении к сайту открывают эту web-страницу уже с виччестера, что позволяет значительно сократить время ее считывания. Данная процедура называется «кэшированием».

- `<META http-equiv="Content-type" content="text/html; charset=KOI8-R">`

Осуществляет жесткую установку так называемого «**чар-сета**», или пользовательской кодировки страницы. Встретив в заголовке документа такую команду, пользовательский браузер автоматически переключается на заданную в теге кодировку. Однако практически все современные http-серверы поддерживают функцию автоматического изменения кодировки документов в зависимости от установленных в пользовательском браузере предпочтений.

- `<META http-equiv="Content-type" content="text/html">`

На запрос браузера сервер сформирует ответ с указанием, что данная страница является HTML-документом.

- `<META http-equiv="Content-Script-Type" content="text/javascript">`

Определяет применение сценариев на языке JavaScript.

- `<META http-equiv="Content-style-Type" content="text/css">`

Определяет, что в документе используются фрагменты CSS.

ЛИТЕРАТУРА

1. Айзекс С. Dynamic HTML / Пер. С англ. – СПб.: BHV. – Санкт-Петербург, 2000. – 496 с.: ил.
2. Браун М.Р., Хоникатт Д. Использование HTML 4 / Пер с англ. – 4 изд., спец. – М.: Вильямс, 2000. – 784 с.: ил.
3. Гаевский А.Ю., Романовский В.А. Самоучитель по созданию Web-страниц: HTML, JavaScript и Dynamic HTML. – К.: А.С.К., 2002. – 472 с.
4. Холмогоров В. Основы web-мастерства. Учебный курс. – СПб.: Питер, 2001. – 352 с.: ил.
5. Пауэл Т. HTML: справочник программиста / Т.А.Пауэл, Д.Уитворт. – М.: АСТ, 2003. – 383 с.
6. Полянский А. Хитрости, трюки и секреты создания web-страниц на базе языка гипертекстовых документов HTML 3.0-4.0 – М.: Познавательная книга плюс, 2000. – 240 с.

Учебное издание

Ружицкая Елена Адольфовна
Сафонов Игорь Владимирович

СПЕЦИАЛИЗИРОВАННЫЕ ЯЗЫКИ РАЗМЕТКИ ДОКУМЕНТОВ (HTML)

*Тексты лекций по спецкурсу для студентов 3 курса специальности
1–40 01 01 01 «Программное обеспечение информационных техноло-
гий» специализации «Компьютерные системы и Internet-технологии»*

В авторской редакции

Подписано в печать 22.12.2006 г. (№ 78). Формат издания 60x84 1/16.
Бумага писчая №1. Печать на ризографе. Гарнитура Таймс.
Усл.п.л. 5,3. Уч-изд.л. 4,1. Тираж 20 экз.

Учреждение образования
«Гомельский государственный университет
имени Франциска Скорины»,
246019 г.Гомель, ул. Советская, 104

Отпечатано в учреждении образования
«Гомельский государственный университет
имени Франциска Скорины»,
246019 г.Гомель, ул. Советская, 104