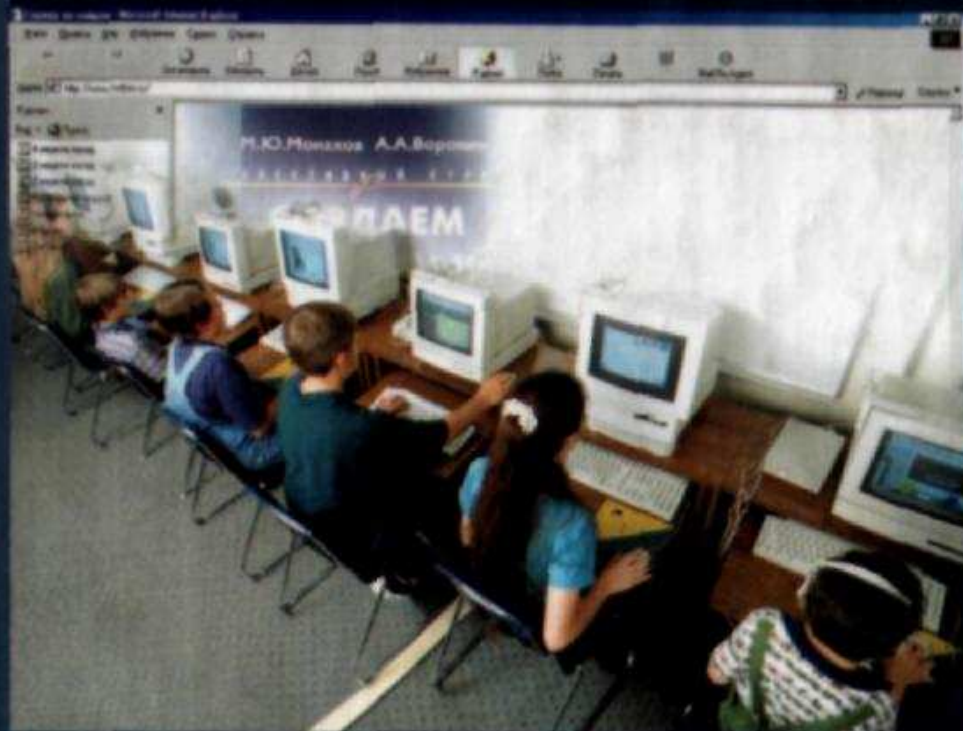


М. Ю. Монахов А. А. Воронин

Э Л Е К Т И В Н Ы Й   К У Р С

# СОЗДАЕМ ШКОЛЬНЫЙ САЙТ В ИНТЕРНЕТЕ

Учебное пособие



БИНОМ

**Э Л Е К Т Р О Н Н Ы Й      К У Р С**

**И Ф О**

**М. Ю. Монахов, А. А. Воронин**

# **СОЗДАЕМ ШКОЛЬНЫЙ САЙТ В ИНТЕРНЕТЕ**

**Учебное пособие**

Издание подготовлено при содействии  
НФПК – Национального фонда подготовки кадров



Москва  
**БИНОМ. Лаборатория знаний**  
2005

УДК 004.9  
ББК 32.97  
М77

**Монахов М. Ю.**

**М77** Создаем школьный сайт в Интернете. Элективный курс: Учебное пособие / М. Ю. Монахов, А. А. Воронин. — М.: БИНОМ. Лаборатория знаний, 2005. — 128 с.: ил.  
ISBN 5-94774-158-X

Учебное пособие позволяет получить профессиональные навыки создания сайтов в Интернете. Оно поможет сформировать у обучаемых творческий подход, способность к самостоятельному и инициативному решению проблем, умение использовать типовые инструментально-технологические средства и эффективно работать в неоднородных командах, что требуется для личностного развития и профессионального самоопределения.

Каждая тема учебного пособия представляет собой законченный учебный модуль, включающий теоретический материал, задания для самостоятельной работы, темы рефератов.

К учебному пособию предлагается CD-ROM, содержащий результаты выполнения упражнений, демонстрационные плакаты, спецификацию языка HTML.

Для учащихся старших классов информационно-технологического, физико-математического, естественно-научного и гуманитарного профилей.

УДК 004.9  
ББК 32.97

По вопросам приобретения обращаться:  
«БИНОМ. Лаборатория знаний» (095)955-0398,  
e-mail: Lbz@aha.ru  
<http://www.Lbz.ru>

ISBN 5-94774-158-X

© Монахов М. Ю., Воронин А. А., 2005  
© БИНОМ. Лаборатория знаний, 2005

# Оглавление

Предисловие . . . . .	4
Глава 1. Осваиваем Интернет . . . . .	7
Глава 2. Создаем среду сайта . . . . .	19
Глава 3. Определяем функции сайта . . . . .	32
Глава 4. Проектируем внешний вид сайта . . . . .	43
Глава 5. Проектируем карту сайта . . . . .	51
Глава 6. Создаем свою первую веб-страницу . . . . .	59
Глава 7. Подключаем графические объекты . . . . .	71
Глава 8. Выполняем верстку страниц . . . . .	81
Глава 9. Собираем и публикуем сайт на веб-сервере . . . . .	95
Глава 10. Оживляем сайт . . . . .	107
Глава 11. Защищаем сайт . . . . .	116
Что дальше? . . . . .	124
Список рекомендуемой литературы . . . . .	125
Алфавитный указатель . . . . .	126

# Предисловие

Интернет постепенно входит в нашу жизнь. Возможности, предоставляемые им, привлекают все больше и больше пользователей. Многие организации начинают обращаться к глобальной сети с целью размещения рекламы, для обмена информацией и т. д.

Информация в Интернете организована в виде системы связанных Web-страниц. Страницы объединяются в сайты.

Сайт — это совокупность информационных единиц — страниц, объединенных общей целью.

Курс «Создаем школьный сайт в Интернете» — элективный, т. е. курс по выбору учащегося. Он входит в состав профиля обучения на старшей ступени школы. Рекомендуемые профили — естественно-научный, физико-математический, информационно-технологический и гуманитарный. Курс базируется на программе по информатике для средней общеобразовательной школы и предполагает повышение уровня образования за счет углубленного изучения материала по информационным технологиям.

Данное учебное пособие поможет вам узнать:

- историю развития, возможности, типовые инструменты и средства глобальной сети Интернет;
- основные этапы и задачи проектирования модели Web-сайта как системы;
- основные способы создания Web-страниц;
- основные конструкции языка гипертекстовой разметки документов HTML;
- основные способы защиты информации в Интернете;
- способы эффективной работы в команде.

Вы научитесь:

- выполнять поиск необходимой информации в Интернете;
- выполнять системное проектирование модели Web-сайта, выделяя и реализуя элементы, связи, функции;
- создавать Web-страницы, собирать и устанавливать Web-сайт, принимать меры по защите информации;
- работать в команде над одним проектом, выполняя разные роли.

Вы также приобретете способность к самостоятельному творческому и инициативному решению проблем, умение использовать типовые инструментально-технологические средства и эффективно работать в неоднородных командах, что способствует развитию личности и профессиональному самоопределению.

Курс построен по модульному принципу. Каждая тема (глава) представляет собой законченный учебный модуль, включающий в себя теоретический материал, практические задания, задания для самостоятельной работы, темы рефератов.

К учебному пособию прилагается компакт-диск, содержащий:

- в папках **Chapter** <номер главы> — файлы, используемые для иллюстрации отдельных вопросов в главах книги. Это промежуточные и итоговые результаты выполнения упражнений, демонстрационные плакаты (см. ссылки в соответствующих главах);
- в папке **HTML40r** — спецификацию языка HTML 4.0 (рекомендация W3C от 18 декабря 1997 г.) на русском языке.

### Пиктограммы, используемые в тексте



*Практическая работа на компьютере.*



*Упражнение, выполняемое без компьютера.*



*Вопросы и задания. На каждый из вопросов можно найти ответ в теоретической части раздела.*



*Темы рефератов.* Написание реферата поможет получить более глубокие знания по изучаемой теме.



*Ссылка на компакт-диск.* Указание файла на компакт-диске, прилагаемом к учебному пособию.



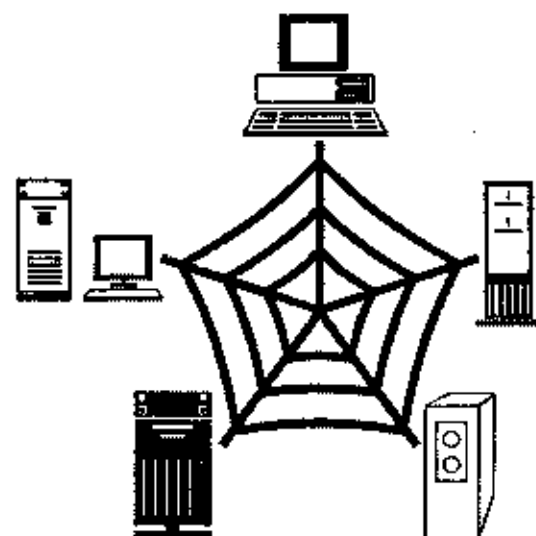
*Коротко о главном.* Основные знания, которые должен получить учащийся, прочитав главу учебного пособия.



*Демонстрационные плакаты.* Указаны названия плакатов, поясняющих материалы книги. Плакаты сохранены в форматах Visio и Corel Draw и содержатся в файлах с расширениями vsd и cdr в соответствующих главам папках прилагаемого компакт-диска.

## Глава 1

# ОСВАИВАЕМ ИНТЕРНЕТ



Создание Интернета началось в 70-х гг. прошлого века с идеи создания компьютерной сети, способной осуществлять обмен информацией при определенной степени повреждений. Основываясь на этой идее, по заказу Министерства обороны США была создана сеть ARPAnet, которая стала экспериментальной площадкой для научных исследований компьютерных сетей в военно-промышленной сфере.

Основными принципами, по которым велось создание ARPAnet, стали равноправность компьютеров и возможность работы сети при повреждении ее частей. Равноправность заключалась в возможности установления связи любого компьютера с любым другим компьютером сети.

Спустя некоторое время с ростом использования компьютеров в различных областях появилось желание объединять их в локальные вычислительные сети. Позже появилась потребность объединения локальных сетей с помощью сети ARPAnet. Однако использовать инфраструктуру сети ARPAnet не удалось в связи с возникшими проблемами управления. Поэтому была предпринята попытка создания другой сети, работающей по принципам ARPAnet. Второй проект получил название NSFnet и разрабатывался по инициативе Национального научного фонда. Новая сеть использовала в качестве каналов передачи данных телефонные линии.

С ростом количества подключений к сети возрастало и количество передаваемой информации. Этот процесс продолжается и сегодня, что приводит к необходимости постоянного усовершенствования линий связи и управляющих машин.

Со временем такая сеть получила название Интернет (Internet). Интернет можно представить как «сеть сетей».

Обмен информацией в сети Интернет осуществляется на основе протокола IP (Internet Protocol), описывающего правила и методы работы сети.



## Структура сети Интернет

Если бы сеть Интернет существовала сама по себе, то, скорее всего, она уже израсходовала бы свои ресурсы (исчерпала возможности), что привело бы к частым сбоям и, в конечном счете, пользователи стали бы отказываться от ее использования. За работоспособность и управляемость сети Интернет отвечает несколько организаций, в функции которых также входит и перспективное планирование сети, позволяющее ей развиваться согласно возникающим у пользователей потребностям. Такие организации являются в большинстве случаев добровольным собранием энтузиастов. Их совместная работа позволяет конечным пользователям, сидящим за своими персональными компьютерами, получать доступ к информационным ресурсам сети.

Среди этих организаций можно выделить:

- ISOC (Internet Society — Интернет-сообщество) — добровольная организация, отвечающая за развитие, стандартизацию и распространение технологий обмена информацией;
- IAB (Internet Architecture Board — совет по архитектуре Интернета) — входит в ISOC и отвечает за разработку стандартов, по которым работает сеть;
- IETF (Internet Engineering Task Force — оперативный инженерный отряд Интернета) — множество таких отрядов, входящих в состав ISOC, отвечают за решение технических проблем в работе сети.

Также существует множество других крупных и мелких организаций, которые отвечают за работу сети в определенных регионах, городах и т. д. как в сфере управления, так и в области технических решений.

## Виды доступа к Интернету

Для пользователя в большей степени важно не то, как организована сеть или по каким правилам протекает обмен информацией в ней, а как получить доступ к ресурсам сети. Здесь ему предоставляется множество различных способов подключения к сети, которые определяются возможностями пользователя и его персонального компьютера и в основном — необходимыми финансовыми затратами.

Доступ к сети Интернет обычно осуществляется через поставщика услуг, или провайдера (*service provider*). Провайдер помимо доступа обычно предоставляет еще множество других услуг, таких как почтовый ящик, хостинг и т. п.

Основные виды доступа:

- **Непосредственный доступ.** Данное подключение предпочтительно для организаций, имеющих локальную вычислительную сеть. Оно обеспечивает доступ ко всем ресурсам и возможностям сети Интернет. Установив у себя определенное программное и аппаратное обеспечение, получив выделенный канал связи (скорость которого зависит от его цены), вы сами можете стать поставщиком услуг (провайдером) и самостоятельно управлять доступом к сети. Явным недостатком является высокая стоимость данного вида доступа.
- **SLIP (Serial Line IP — IP для последовательных линий) и PPP (Point to Point Protocol — протокол «точка-точка»).** Вид доступа, использующий обычные телефонные линии и высокоскоростные модемы. Подключение реализуется к сети организации или к провайдеру (а через него — к сети Интернет) в качестве полноправного пользователя. Однако данные протоколы применимы только для отдельных пользователей, их использование для подключения к локальной сети не рекомендуется. Это объясняется их низкой эффективностью и скоростью обмена при подключении более одного компьютера. Протокол PPP является более поздней разработкой и предоставляет большее количество возможностей по сравнению с протоколом SLIP. Существует также протокол CSLIP (**Compressed SLIP — сжатый SLIP**), представляющий собой измененный протокол SLIP для медленных линий связи. В итоге получается достаточно дешевый вид подключения с вполне приемлемым качеством.
- **Dial-Up Access (доступ по вызову).** Основан на идее подключения к сети через другой компьютер. Для этого необходимо, чтобы вызываемый компьютер имел доступ к сети Интернет и разрешал удаленную работу пользователей. В итоге получается, что вы работаете с Интернетом не на своем компьютере, а на компьютере, который вызываете. Многие организации предоставляют этот вид доступа для своих сотрудников, так как он позволяет использовать все имеющиеся на удаленном компьютере программное обеспе-

чение и оборудование. Однако такой доступ приводит к повышению требований для компьютера, к которому осуществляется доступ (обычно это мощный сервер).

- **Доступ через другие сети.** Это вид доступа, с помощью которого можно осуществлять получение файлов через электронную почту через специальные серверы. Получив по электронной почте запрос, специальный сервер выполнит инструкции, указанные в письме, и отправит вам результат. А доступ к электронной почте может быть значительно более дешевым.

## Принципы функционирования сети

Обмен информацией в сети базируется на понятии «протокол». Протокол представляет собой свод правил, по которым может осуществляться такой обмен. Как отмечалось ранее, сеть Интернет использует протокол IP (Internet Protocol) — протокол маршрутизации. Помимо IP используется также протокол TCP (Transmission Control Protocol) — транспортный протокол. Он, в отличие от протокола IP, отвечающего за передачу отдельных пакетов (IP-пакетов), обеспечивает передачу больших объемов информации. Согласно протоколу TCP большой файл разбивается на мелкие IP-пакеты, которые по отдельности транспортируются разными путями к компьютеру-получателю. Затем на компьютере-получателе исходный файл собирается из частей (пакетов) в правильной последовательности. Через TCP обеспечивается работа целого семейства других протоколов, таких как протокол передачи файлов. Можно сказать, что совместно эти протоколы (TCP/IP) обеспечивают работу всех основных сервисов сети Интернет.

Информацию приходится передавать через множество узлов и сетей. Для правильной передачи необходимо определить путь, по которому должны пройти пакеты. Это приводит к необходимости получения информации о структуре сети и связях между ее узлами. Протокол IP обеспечивает передачу информации между компьютерами сети. Процесс вычисления пути следования пакетов называется маршрутизацией.

Этот процесс начинается с пользователя, когда он указывает адрес источника, к которому хочет обратиться. Каждый узел сети Интернет обладает уникальным адресом. Этот адрес представляет собой последовательность из четырех чисел, разделен-

ных точками, например 192.168.0.1. Здесь каждое число служит для уточнения местоположения компьютера, которому принадлежит адрес. Например, первое число может определять страну или город, второе — район и т. д.

Использование числовых адресов не всегда бывает удобным. Для упрощения работы пользователя числовым адресам присваивают словесные имена. Для осуществления преобразования этих имен в числовые адреса была создана доменная система имен (DNS — Domain Name System). Например, адресу 192.168.0.1 может соответствовать доменное имя `comp1.school1.city1.ru`. Здесь слева направо определяется принадлежность имени: в нашем примере доменное имя обозначает компьютер с именем `comp1`, который принадлежит организации `school1`, расположенной в городе `city1`, находящемся в России. В процессе определения адреса все происходит следующим образом: сначала происходит обращение к DNS-серверу, определяемому крайним правым именем (`ru`); он «знает», как обратиться к следующему DNS-серверу (`city1`) и запрашивает у него адрес; далее второй DNS-сервер аналогично обращается к следующему DNS-серверу (`school1`) и т. д., до последнего DNS-сервера, на котором хранится таблица соответствия имен и адресов. Так протекает типовой процесс определения адреса, однако в реальности он может идти значительно быстрее, так как адрес может быть обнаружен на местных DNS-серверах. Например, если обращение происходит с компьютера с адресом `comp2.school1.city1.ru`, то DNS-сервер с именем `school1` уже может «знать» адрес `comp1.school1.city1.ru`.

Для обращения к ресурсам сети Интернет используются адреса URL (Universal Resource Locator — универсальный локализатор ресурса). В большинстве случаев при обращении к Интернету в качестве адреса указывается адрес конкретного документа (рис. 1.1).

Существуют различные типы источников, которые будут рассмотрены далее.

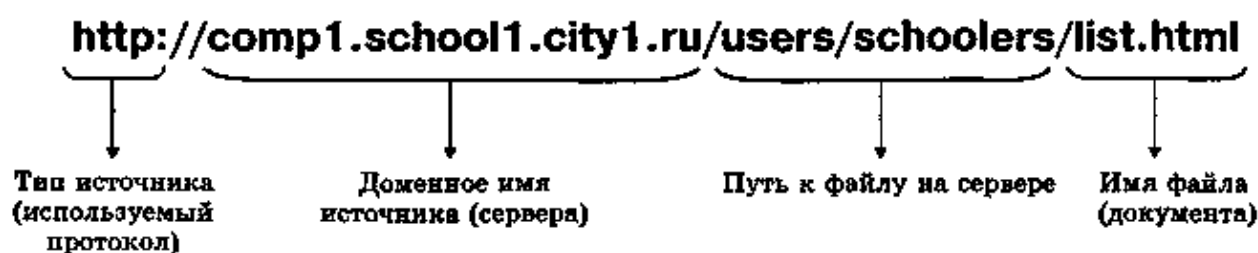


Рис. 1.1



**Упражнение 1.** Создайте и настройте подключение к сети Интернет по протоколу PPP. (Дополнительную информацию по настройке такого подключения можно взять у провайдера.)



**Упражнение 2.** Предложите варианты URL для начальной страницы сайта вашей школы.



На компакт-диске, прилагаемом к учебному пособию, имеются плакаты, демонстрирующие виды доступа к сети Интернет и принципы функционирования системы доменных имен. Плакаты содержатся в файлах poster1-1.vsd, poster1-2.vsd, poster1-1.cdr и poster1-2.cdr в папке chapter1.

## Сервисы Интернета

Рассмотрим основные сервисы сети Интернет. Среди них можно выделить удаленный доступ, передачу файлов, электронную почту, доски объявлений, поиск данных и программ, поиск людей, Gopher, WAIS, всемирную паутину (WWW), общение.

Удаленный доступ позволяет пользователям работать на удаленном компьютере. При этом пользователь получает в свое распоряжение практически все ресурсы удаленного компьютера, в том числе подключенное к нему периферийное оборудование. Для пользователя работа на удаленном компьютере осуществляется с помощью специальной оболочки, имитирующей терминал удаленного компьютера. Работа осуществляется по протоколу telnet, который реализует поддержку удаленного доступа через сеть Интернет.

Пример URL: telnet://school1.city1.ru.

Передача файлов осуществляется по протоколу FTP (File Transfer Protocol — протокол передачи файлов). Основными назначениями этого протокола являются передача файла с одного компьютера на другой и доступ к файловым архивам. Файлы могут передаваться в двух режимах: двоичном (побитно, содержимое файла значения не имеет) и текстовом (посимвольно с автоматической перекодировкой данных согласно кодовой таблице, используемой на конкретном компьютере, для текстовых файлов). Большую часть этих функций в настоящее время можно осуществить через протокол HTTP (HyperText Transfer Protocol — протокол передачи гипертекста). Также существует

возможность получения файлов через специальные серверы при использовании электронной почты.

Пример URL: `ftp://school1.city1.ru/file.exe`.

Электронная почта (e-mail — electronic mail) используется для обмена сообщениями. Она может также использоваться для обмена файлами, организации конференций и почтовых рассылок (передача сообщений по списку адресов). Обмен сообщениями осуществляется через почтовые серверы, основное назначение которых заключается во временном (до запроса пользователем) хранении полученных писем и пересылке отправляемых писем по указанным адресам. Адрес абонента на почтовом сервере представляется в следующем формате: имя\_абонента@доменное\_имя\_почтового\_сервера.

Пример: `schooler@school1.city1.ru`

Доски объявлений (USENET) предназначены для обмена сообщениями в пределах дискуссионных групп по различным темам, где каждая тема подразделяется на несколько категорий. Когда кто-то оставляет сообщение, оно автоматически рассылается всем участникам. Тем самым обеспечивается быстрая рассылка сообщений.

Поиск данных и программ осуществляется через систему Archie. Она представляет собой поисковую систему по файлам, расположенным на анонимных файловых серверах. Периодически эта система опрашивает ftp-серверы для получения списка доступных файлов и их описаний. Поиск может осуществляться как по имени файла, так и по его атрибутам либо описанию. Доступ к этой системе осуществляется через специальные Archie-серверы. Возможен доступ к ним по электронной почте.

Поиск людей может быть реализован при использовании служб whois, finger, fred. Поиск обычно осуществляется через одноименные программы. Также существует объединенный интерфейс KIS (Knowbot Information Service — информационная служба баз знаний), через который можно осуществить поиск практически по всем информационным базам данных в сети Интернет.

Оболочка Gopher представляет собой некое средство объединения возможностей сети Интернет. Представленная в виде вложенных меню, она позволяет получить доступ к telnet, ftp, электронной почте и другим ресурсам. Многие ресурсы располагаются на различных gopher-серверах. Однако этот факт «прозрачен» для пользователя, так как тот работает с единой систе-

мой меню и для него все выглядит так, как если бы находилось на одном сервере. Используется протокол Gopher.

Wais представляет собой интерактивную диалоговую систему поиска по ключевым словам. Для работы с такой системой желательно иметь быстрое соединение с Интернетом.

WWW (World Wide Web, «Всемирная паутина») представляет собой единую среду, наполненную гипертекстовыми документами. Последние связаны между собой ссылками, по которым осуществляется переход от одного документа к другому. World Wide Web предоставляет широкие возможности для доступа к другим сервисам Интернета (e-mail, поиску документов и людей, поиску и передаче файлов и т. д.), фактически вытеснив ряд прежних сервисов (таких как Gopher, Wais, Archie и пр.), а также для оформления (дизайна) документов и использования в них различной мультимедиа-информации. В WWW поддерживается множество протоколов, однако наиболее часто используется протокол HTTP (HyperText Transfer Protocol — протокол передачи гипертекстовых документов).

Пример URL: <http://school1.city1.ru>

Большое распространение в сети получили службы общения. В отличие от электронной почты, такое общение осуществляется в реальном времени. Оно может быть организовано различными способами, в том числе с использованием звука, видео и обычного текста. При использовании звука и видео требуются достаточно быстрые каналы связи с Интернетом, а для обмена текстовыми сообщениями достаточно и медленного канала связи. Среди подобных систем можно выделить ICQ, IRC, NetMeeting, IPhone (IP-телефонию) и др.

Существуют также другие сервисы, предоставляемые пользователям сети, например, онлайн-переводчики, аудио- и видеоархивы и др. В дальнейшем будут рассматриваться в основном технологии WWW.

## **Программы просмотра (браузеры)**

Для просмотра гипертекстовых документов используется специальное программное обеспечение. Такие программы называются браузерами (browser). Среди наиболее популярных браузеров можно назвать Netscape Navigator, Microsoft Internet Explorer, Opera. Некоторые браузеры представляют собой целые комплексы, включающие в себя средства для работы с электронной

почтой, функции передачи файлов (для работы с ftp-серверами), редакторы гипертекстовых документов и пр.

Следует выделить две разновидности браузеров: онлайнные (on-line) и оффлайнные (off-line).

Онлайнные браузеры, к которым относятся и упомянутые выше, требуют подключения к сети и работают с ней в реальном времени по запросу пользователя.

Оффлайнные браузеры сначала накапливают требуемые ресурсы (копии документов) на пользовательском компьютере, чтобы впоследствии их можно было использовать и без подключения к сети Интернет. В качестве примеров можно назвать Offline Explorer, BlackWindow и др.

Рассмотрим типовые элементы браузеров на примере Internet Explorer версии не ниже 5.0 (рис. 1.2).

В верхней части рабочего окна находится меню программы, элементы управления доступом к ресурсам сети Интернет, средства навигации и настройки.

Чтобы получить доступ к какому-либо ресурсу в сети Интернет (веб-странице), необходимо указать его адрес: ввести его в строке адреса или выбрать соответствующую гиперссылку в тексте другого документа.

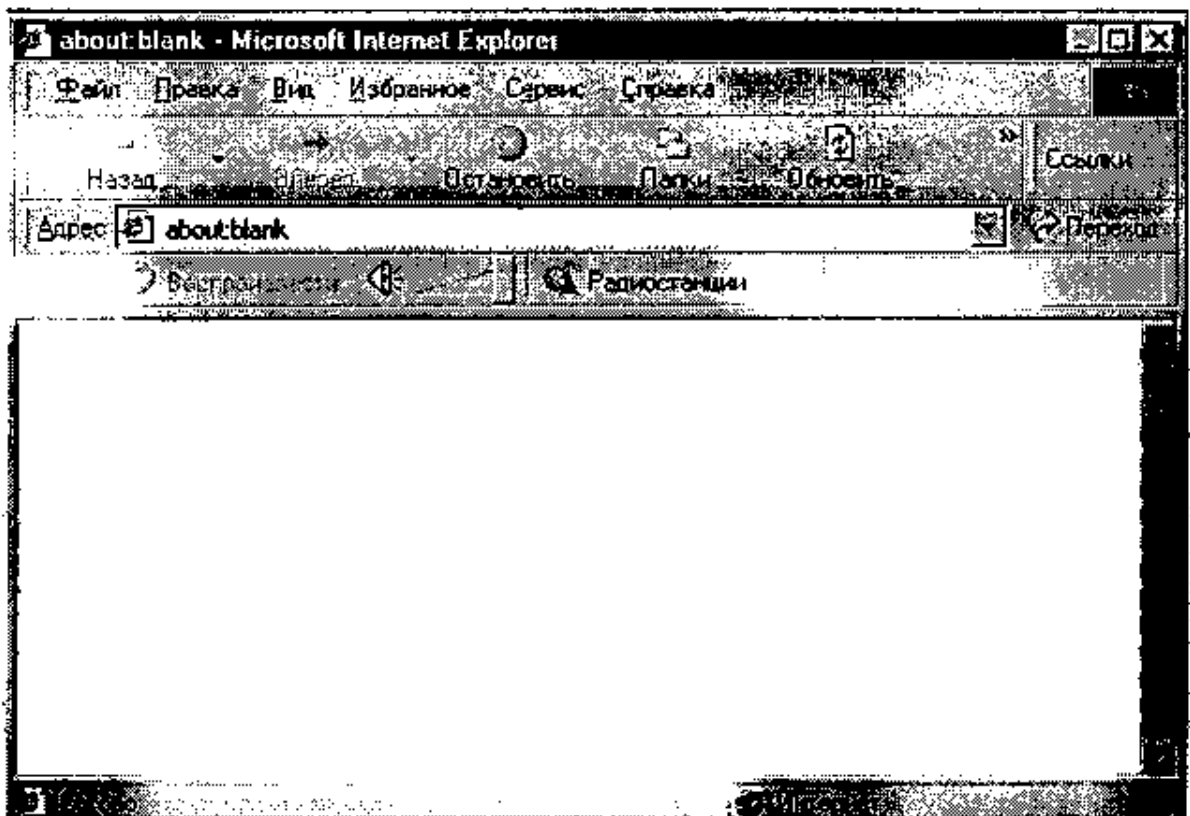












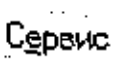
Рис. 1.2



Дальнейшее продвижение (навигация) по страницам осуществляется с помощью гиперссылок, объявленных на отображаемой странице. Такой гиперссылкой может быть кнопка с надписью, картинка или фрагмент текста, который обычно выделяется цветом и подчеркиванием (при наведении на гиперссылку форма указателя мыши меняется).

Основные элементы управления браузером приведены в табл. 1.1.

Таблица 1.1

Элемент	Назначение
 Назад	Возврат к ранее просмотренным страницам
	Выбор из нескольких элементов (например, из списка ранее посещенных страниц)
 Остановить	Остановка (прерывание) загрузки текущей страницы
 Обновить	Обновление текущей страницы
 Домой	Переход на «домашнюю» страницу (которая указана в качестве стартовой)
 Печать	Печать текущей страницы
 Файл	Файловые операции (сохранение, открытие, печать и др.)
 Прзвк	Команды работы с текстом страницы (копирование, поиск и др.). Текст на странице, просматриваемой в Интернете, доступен только для чтения
 Вид	Команды управления отображением текста страницы на экране (размер шрифта, кодировка и др.)
 Избранное	Формируемый пользователем список адресов понравившихся страниц. В дальнейшем для перехода на выбранную страницу достаточно выбрать ее название в этом списке
 Сервис	Команды настройки браузера



## Коротко о главном

1. Сеть Интернет является объединением множества сетей. Взаимодействие осуществляется на основе протокола IP. С точки зрения пользователя, Интернет можно представить как огромное, постоянно развивающееся хранилище информации.
2. Управление сетью и ее развитие осуществляется организациями на добровольных началах.
3. Обращение к компьютерам в сети осуществляется по числовым адресам. Существует служба доменных имен, которая осуществляет преобразование словесных адресов в числовые. Чтобы пользователи сети Интернет могли обращаться к вашему сайту, сначала необходимо получить для него адрес у провайдера и зарегистрировать этот адрес на DNS-сервере.
4. Существует несколько типов подключения к сети, которые различаются в основном стоимостью и скоростью подключения. Для самостоятельного подключения к сети достаточно подключения по протоколу PPP с автоматическим присвоением адресов.
5. Наиболее распространенные сервисы Интернет — WWW и электронная почта. Для работы с выбранным сервисом необходимо специальное программное обеспечение. При использовании WWW это браузер.



## Вопросы и задания

1. Назовите сеть, которая определила основные принципы работы сети Интернет.
2. Назовите организации, которые осуществляют обслуживание и развитие сети Интернет.
3. Как осуществляется адресация в Интернете?
4. Назовите типы подключения к сети Интернет и их особенности.
5. Перечислите основные сервисы сети Интернет.
6. В чем состоит основное назначение программ просмотра (браузеров)?



## Практическое задание

Изучите возможности браузера Internet Explorer. Установите в его настройках:

- а) в качестве страницы, с которой начинается обзор («домашней страницы»), — начальную страницу вашего сайта;
- б) программы Интернета — используемый редактор HTML и программу электронной почты.



## Темы рефератов

1. История создания сети Интернет.
2. Способы подключения компьютера к сети (локальной и Интернет).
3. Эталонная модель взаимодействия открытых систем OSI.
4. Браузеры и их характеристики.



## Демонстрационные плакаты

1. Виды доступа к сети Интернет.
2. Принципы работы службы доменных имен. Структура доменного имени.

## Глава 2

# СОЗДАЕМ СРЕДУ САЙТА



В сети Интернет существует возможность получить во временное пользование место для своего сайта, однако это приводит к дополнительным «расходам» разного рода при разработке и дальнейшей поддержке сайта. Поэтому лучше создать для своего сайта собственный Web-сервер.

Преимущества создания собственного сервера:

- упрощается создание и отладка сайта (устранение неточностей и ошибок);
- упрощается обновление страниц;
- при использовании «чужого» сервера необходимо учитывать имеющиеся ограничения (например, обязательный показ рекламных баннеров);
- наш Web-сервер создается в учебных целях, т. е. он постоянно будет перестраиваться, устанавливаться заново и т. п. А «чужой» сервер — это только дисковое пространство с ограниченными возможностями и значительными ограничениями по администрированию.

Сервер представляет собой специализированное устройство или узел, выполняющий определенные функции, запрашиваемые другими узлами (компьютерами пользователей). Основной функцией Web-сервера является предоставление информации по запросам пользователей.

Сервер может представлять собой компьютер общего назначения, для которого добавлены функции Web-сервера. Многозадачная операционная система тратит часть своих ресурсов на выполнение запросов пользователей. Тогда Web-сервер можно представить, с одной стороны, как специализированное устройство (в качестве которого может выступать персональный компьютер конкретного пользователя), а с другой стороны, как специальное программное обеспечение.

Среди наиболее распространенных Web-серверов можно назвать Apache и IIS (Internet Information Services). Далее рассмотрен процесс установки и настройки Web-сервера Apache.



На компакт-диске, прилагаемом к учебному пособию, имеется плакат, демонстрирующий процесс установки и запуска HTTP-сервера Apache. Плакат содержится в файлах poster2-1.vsd и poster2-1.cdr в папке chapter2.

## Функции Web-сервера

Основная функция Web-сервера — предоставление информации пользователям по их запросам.

В обязанности Web-сервера также входит преобразование записей пути к запрашиваемым файлам, переданных через URL, в формат, принятый для используемой операционной системы. Разные операционные системы предполагают различные форматы представления пути и имени файла. Например: в операционной системе Windows — `c:\folder\file.exe`; в Unix-подобной операционной системе — `/usr/home/folder/file.exe` и т. д. Выделяя из URL путь к файлу, Web-сервер осуществляет его преобразование в требуемый формат, добавляя необходимые для поиска файла данные.

Если указанный файл или группа файлов, запрошенных пользователем, найдены, то осуществляется их передача на пользовательский компьютер, где браузер выполняет компоновку всех полученных файлов в один документ и демонстрирует его пользователю.

К функциям Web-сервера можно также отнести функцию определения имени индексного файла, если оно явно не указано в URL. Например, в адресе `http://www.school1.city1.ru` имя конкретного файла не указано, поэтому Web-сервер автоматически выполняет загрузку страницы с именем `index.htm` или `index.html` (при ее наличии).

Web-сервер выполняет и другие функции, рассмотрение которых не входит в наши задачи.

## HTTP-сервер Apache. Установка и настройка

Web-сервер Apache является одним из наиболее широко распространенных. По функциональным возможностям этот сервер не уступает многим дорогостоящим коммерческим продуктам и может быть настроен практически для любой системы. Процесс установки и настройки рассматривается здесь для версии 1.3.27 и операционной системы Windows<sup>1</sup>.



Дистрибутив Web-сервера Apache находится на компакт-диске, прилагаемом к учебному пособию, в папке Chapter 2 под именем `apache_1.3.27-win32-x86-no_src.msi`.

После запуска программы установки вы увидите окно приветствия (рис. 2.1). Во всех последовательно появляющихся окнах нажимайте кнопку `Next >` (Далее) для продолжения процесса установки (кнопка `Cancel` позволяет прервать процесс установки).

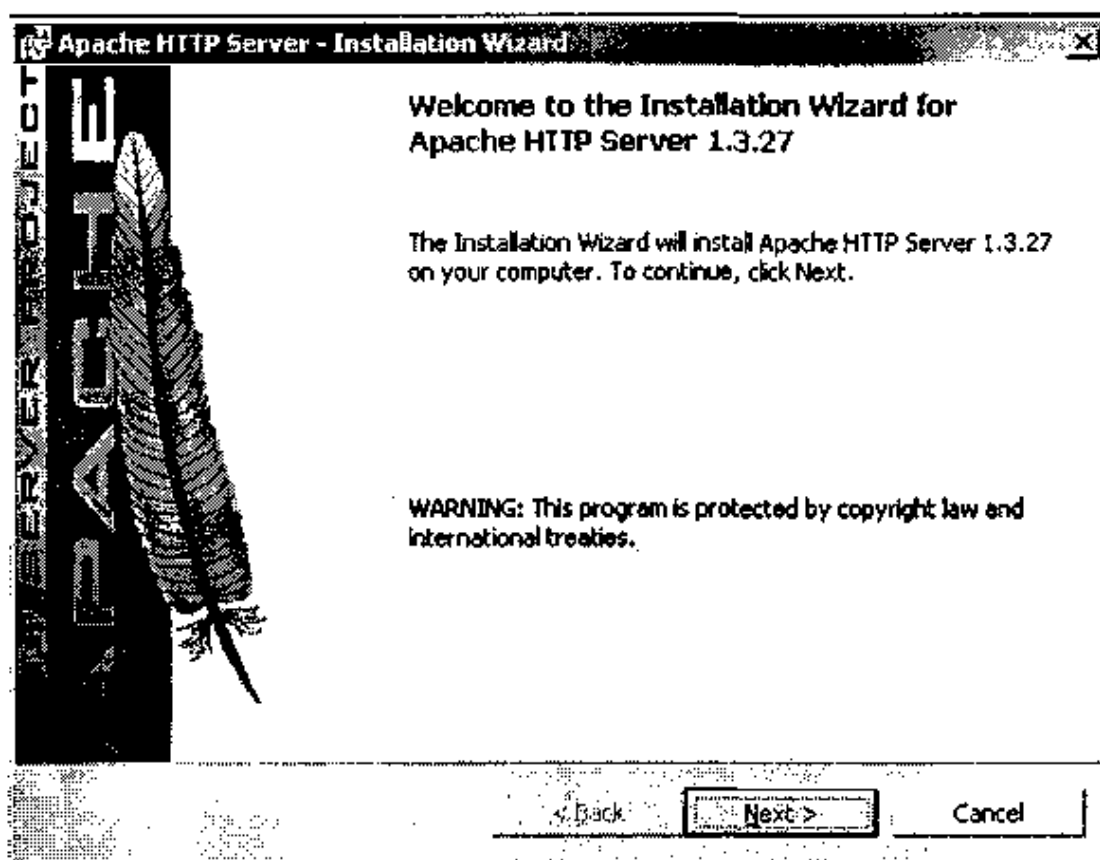


Рис. 2.1

<sup>1</sup> Указанная версия сервера Apache (1.3), согласно ее описанию, применима для Windows 95/98/2000/NT, однако рекомендуется ее использование в среде Windows начиная с версии 2000 и старше.

В следующем окне (рис. 2.2) приведено лицензионное соглашение на использование программного обеспечения. Выбрав радиокнопку **I accept the terms in the license agreement**, вы соглашаетесь с условиями лицензии, выбрав же **I do not accept the terms in the license agreement**, — отвергаете их (тогда процесс установки будет прерван). Вы можете вернуться к предыдущему окну, воспользовавшись кнопкой **< Back (Назад)**.

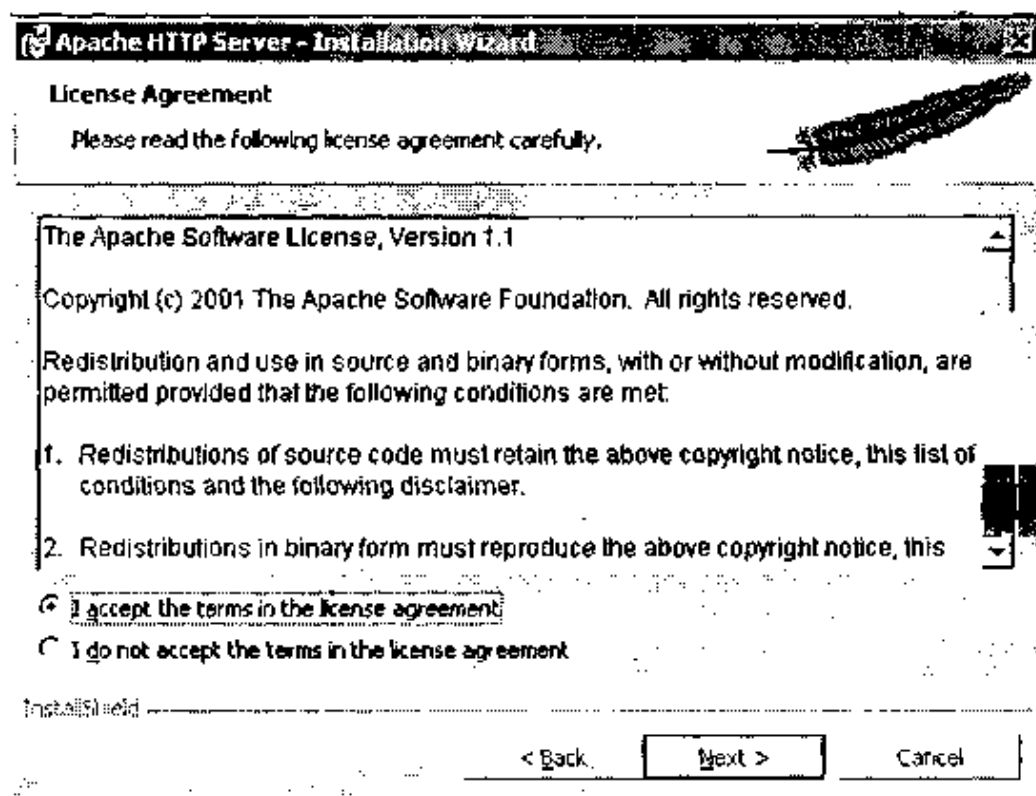


Рис. 2.2

Далее программа установки представит вам информацию о назначении устанавливаемого программного обеспечения, а также об особенностях данной версии (рис. 2.3).

В следующем окне (рис. 2.4) надо ввести основные параметры устанавливаемого Web-сервера. Необходимо указать имя домена (поле **Network Domain**), к которому принадлежит сервер, имя Web-сервера (поле **Server Name**) и адрес электронной почты администратора Web-сервера (поле **Administrator's Email Address**), по которому можно будет связаться с вами, если пользователи обнаружат проблемы с Web-сервером. Рекомендуется указать все эти параметры, так как отсутствие некоторых из них может привести к невозможности запуска Web-сервера после завершения установки. Параметр **Install Apache HTTP Server programs and shortcuts** необходимо установить в положение

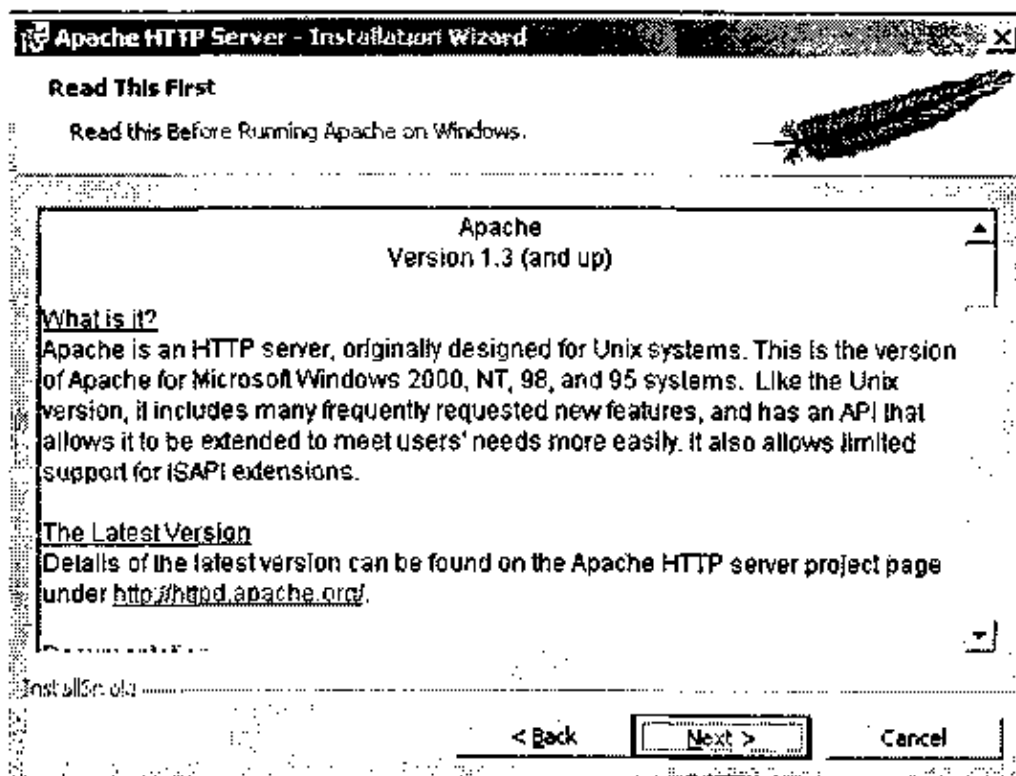


Рис. 2.3

**Run as a service for All Users** (по умолчанию), чтобы Web-сервер запускался автоматически при запуске операционной системы у всех пользователей системы, или в положение **Run when started manually, only for me**, чтобы Web-сервер был доступен только для текущего пользователя и запускался вручную.

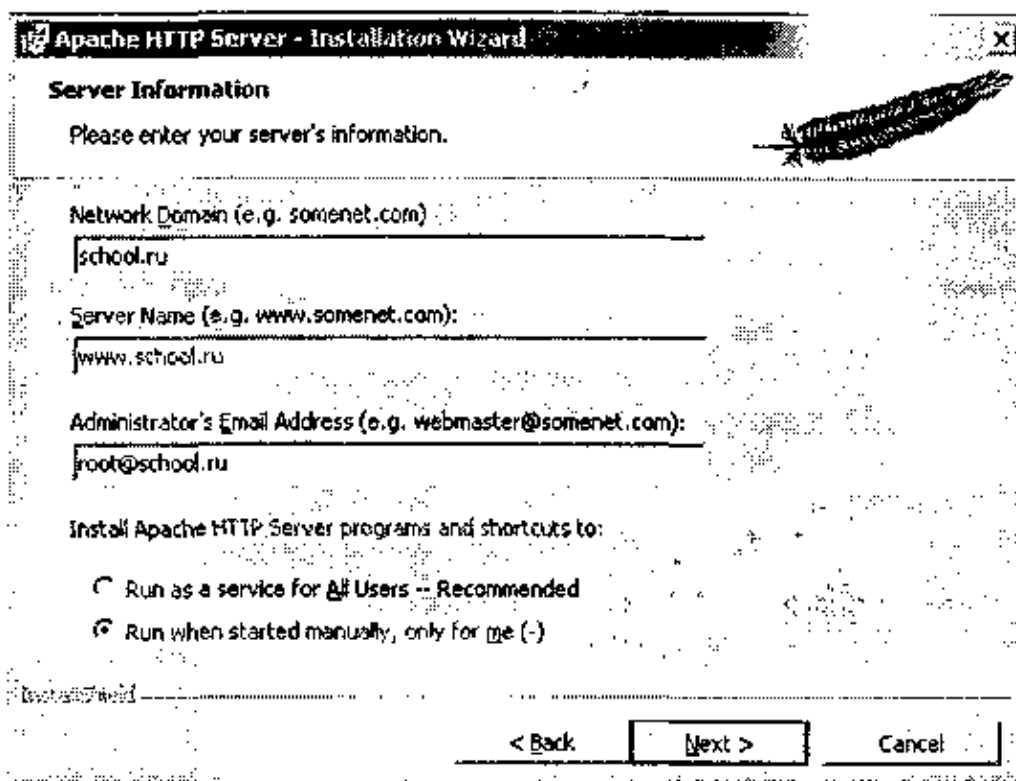


Рис. 2.4



В следующем окне (рис. 2.5) предлагается выбрать тип установки **Complete** (для установки всех компонентов программного обеспечения) или **Custom** (для установки только выбранных вами компонентов — они будут указаны в дальнейшем).

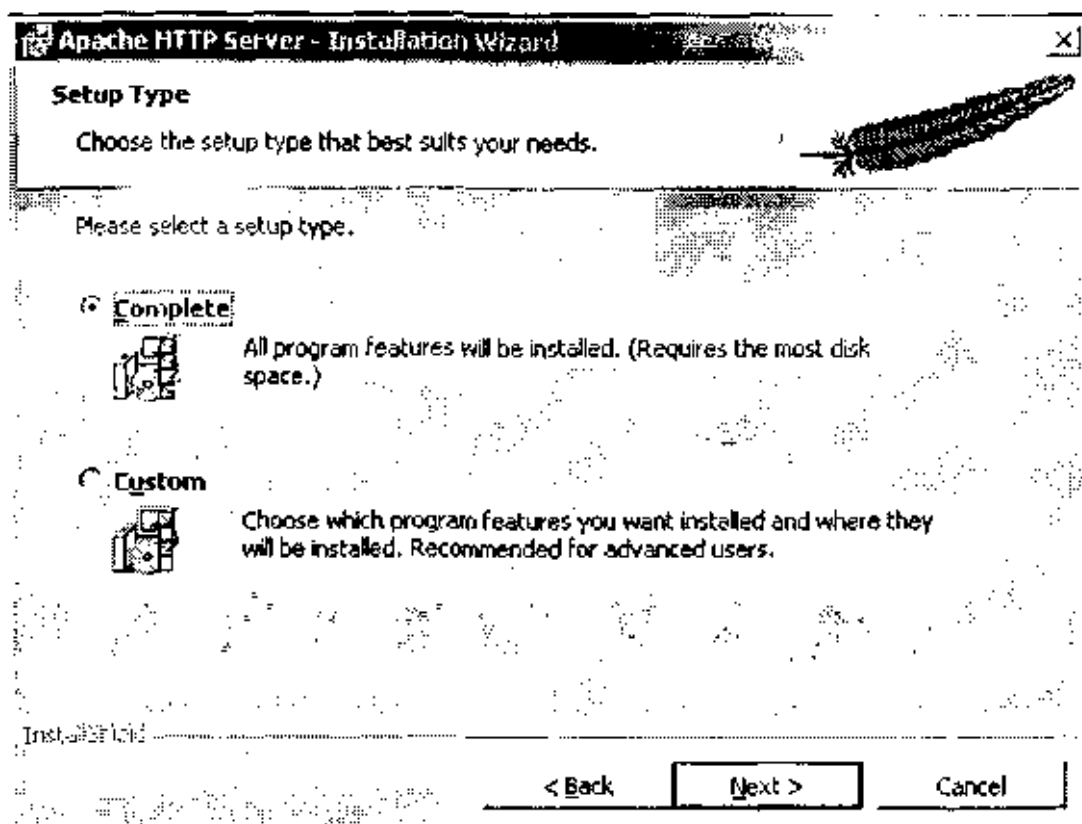


Рис. 2.5

Выбрав тип **Complete**, вы перейдете в окно выбора папки для установки (рис. 2.6). Если нужно произвести установку в другой каталог, щелкните на кнопке **Change** и введите путь к каталогу (поле **Folder name**) или выберите требуемый каталог в имеющемся списке (рис. 2.7).

Выбрав тип **Custom**, вы перейдете в окно выбора компонентов установки (рис. 2.8). Для каждого из предложенных компонентов можно задать один из следующих типов установки:

- **This feature will be installed on local hard drive** — компонент устанавливается на жесткий диск компьютера;
- **This feature, and all subfeatures, will be installed on local hard drive** — компонент и все его составляющие устанавливаются на жесткий диск компьютера;
- **This feature will be installed when required** — компонент будет установлен, когда он потребуется (при первом обращении);
- **This features will not be available** — компонент будет недоступен (не будет установлен вообще).

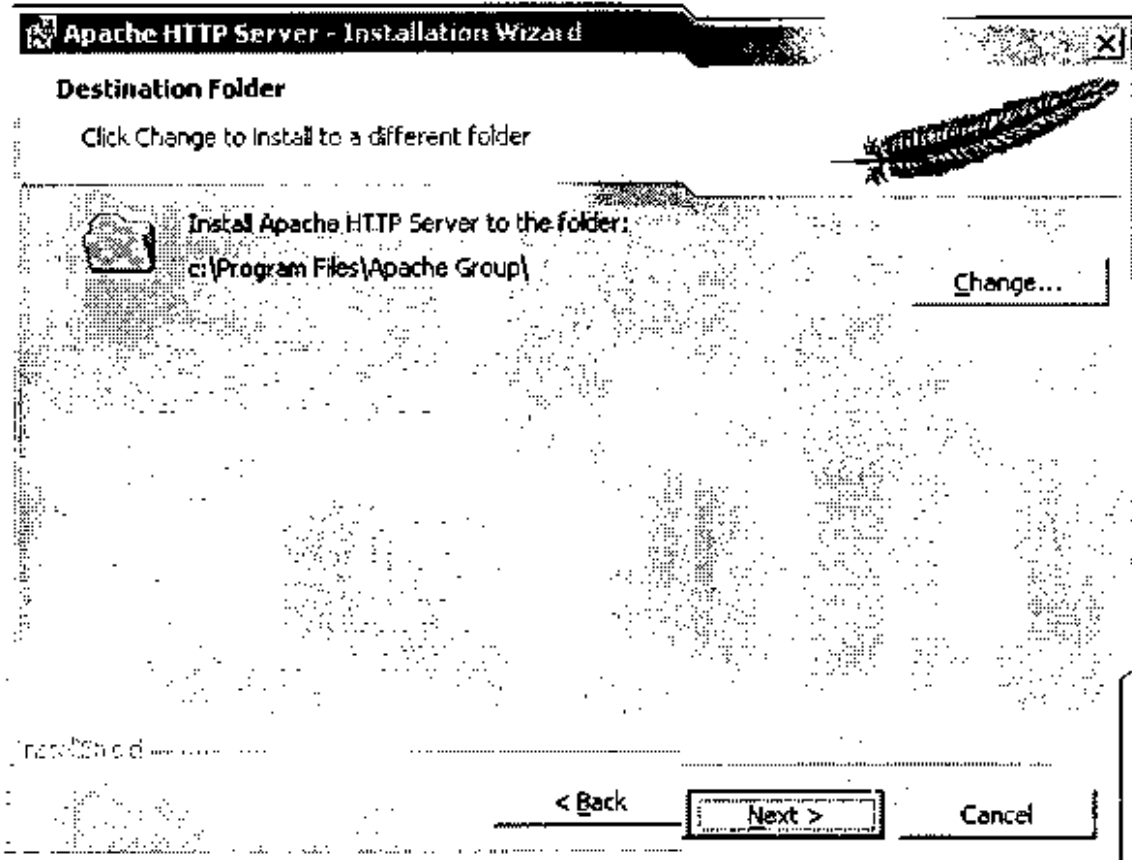


Рис. 2.6

Здесь же при помощи кнопки **Change** внизу справа можно также изменить каталог для установки сервера.

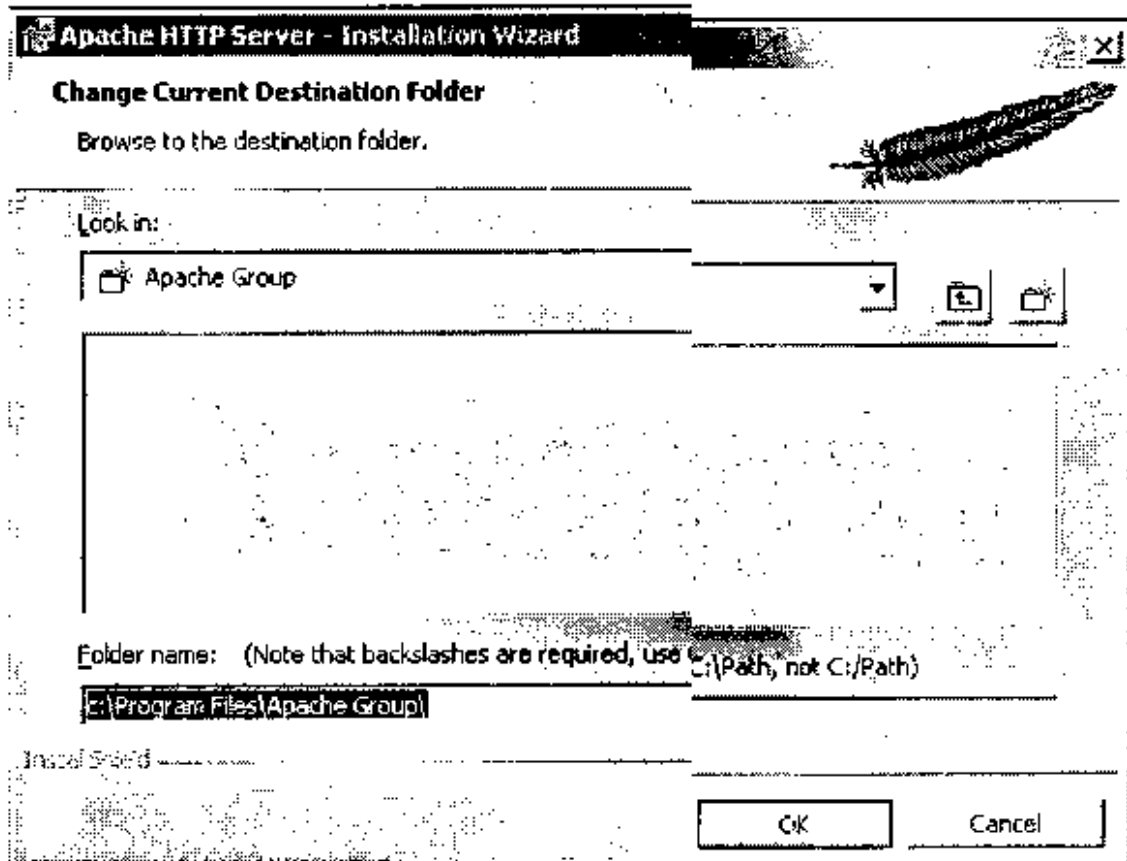


Рис. 2.7

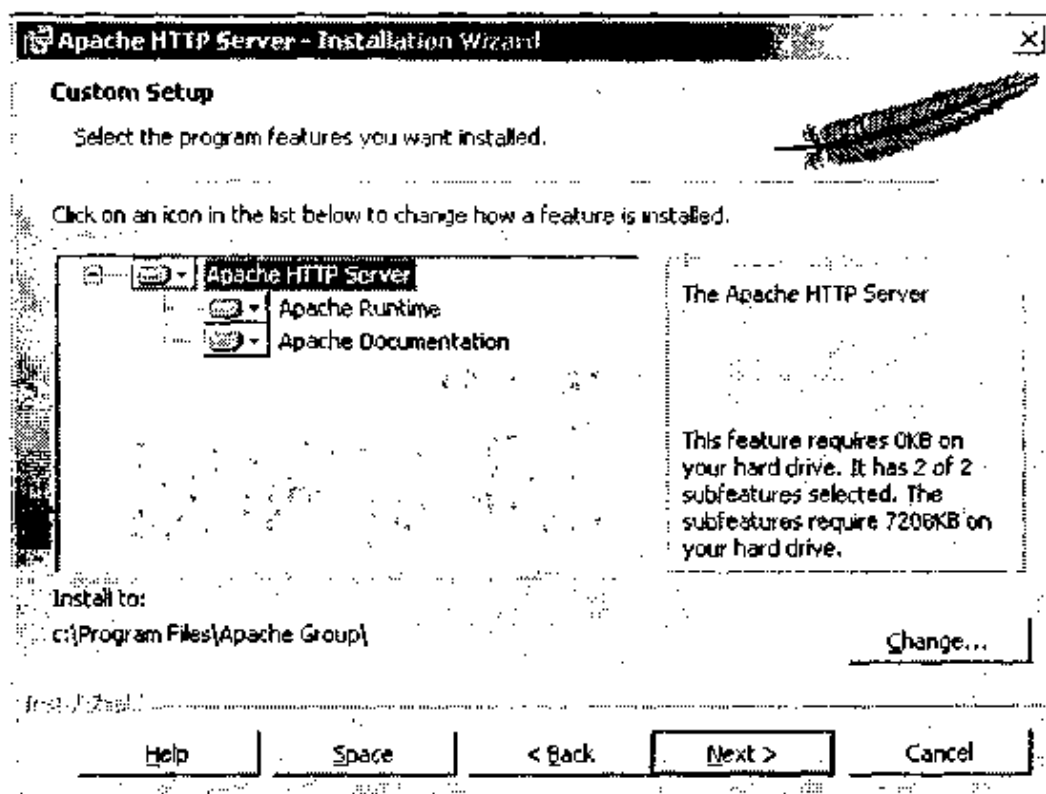


Рис. 2.8

Далее программа установки выдает сообщение о готовности к установке (рис. 2.9). Если все предыдущие этапы выполнены правильно, вы увидите в следующем окне динамический индикатор, который показывает процесс установки (рис. 2.10).

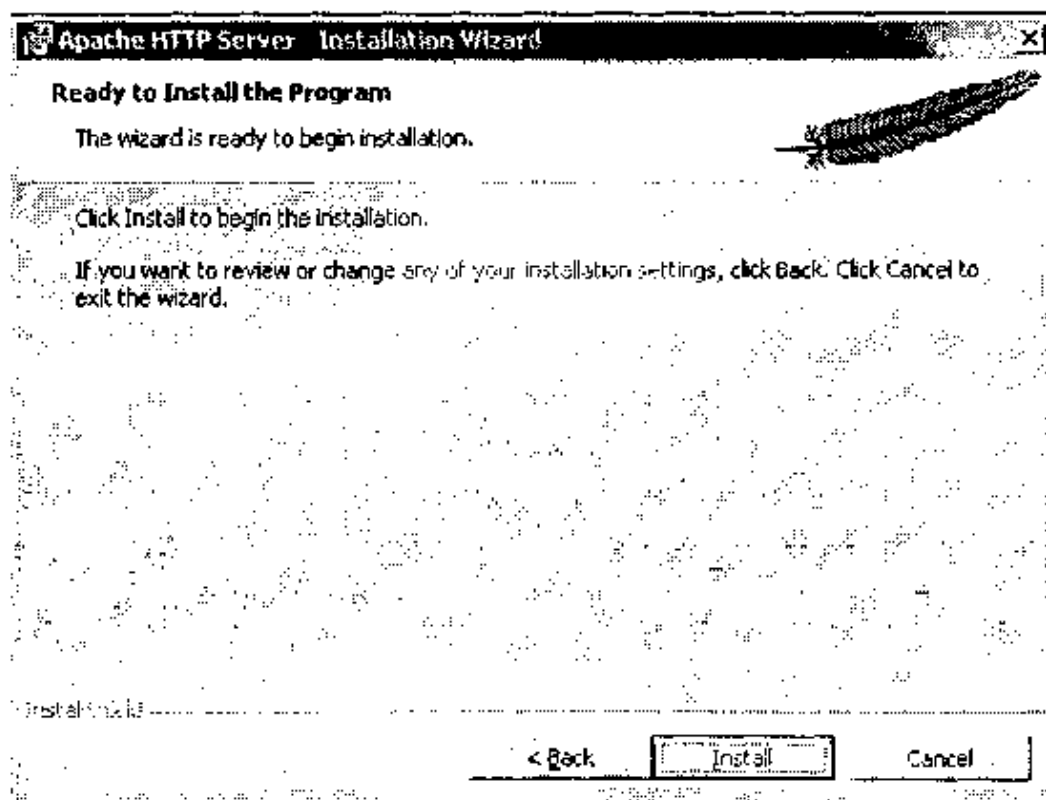


Рис. 2.9

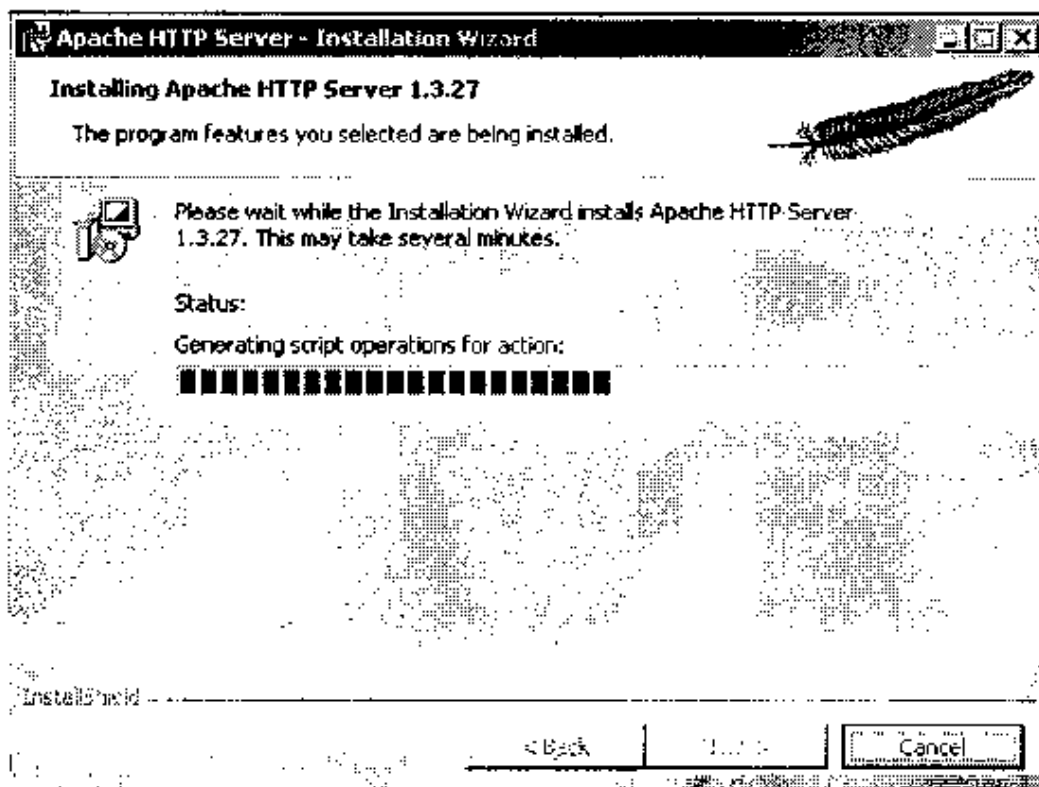


Рис. 2.10

В любой момент вы можете прервать процесс установки, щелкнув на кнопке **Cancel** (Отмена).

После успешного же завершения установки вы увидите заключительное окно, изображенное на рис. 2.11.

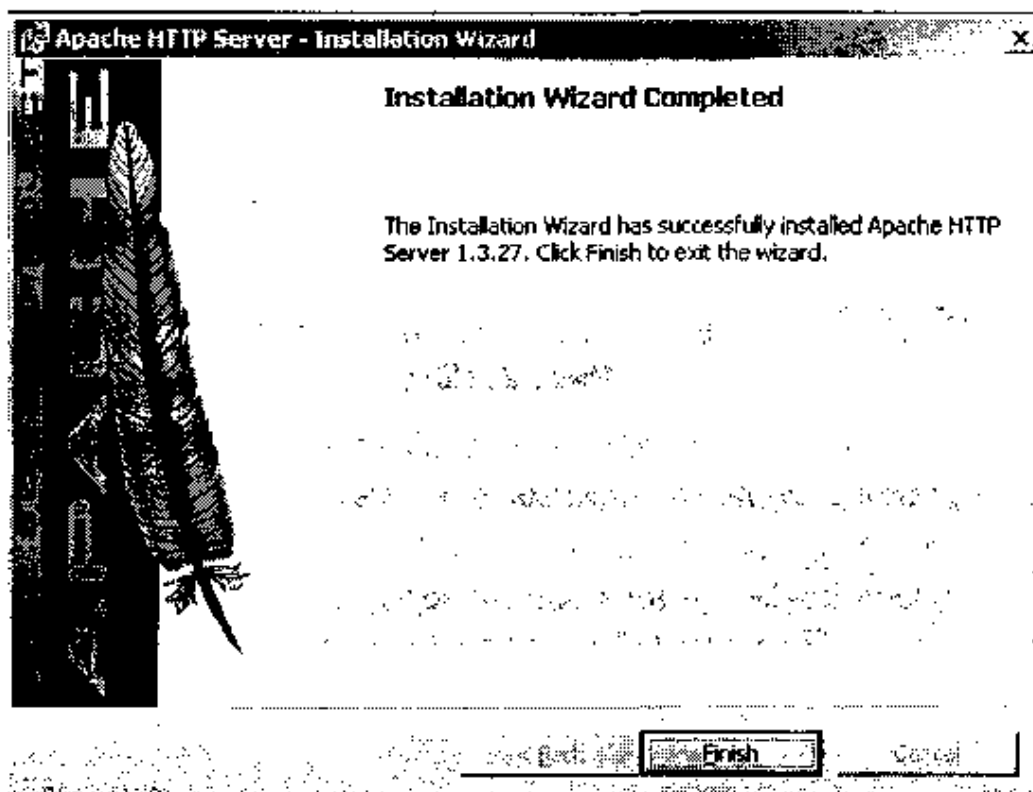


Рис. 2.11

Щелкните на кнопке **Finish** (Готово) в этом окне. Теперь Web-сервер Apache установлен на вашем персональном компьютере.

Откройте в меню **Пуск, Программы** группу **Apache HTTP Server**. Здесь присутствуют элементы, необходимые для настройки и запуска Web-сервера. Выбрав элемент **Start Apache in Console** (при установке параметра **Install Apache HTTP Server programs and shortcuts to** в положение **Run when started manually, only for me**, — см. рис. 2.4), вы запустите Web-сервер в консоли (рис. 2.12). Закрыв это окно, можно остановить работу Web-сервера. Если консоль не появилась, значит один из параметров сервера не задан, либо задан неверно. Если параметр **Install Apache HTTP Server programs and shortcuts to** был установлен в положение **Run as a service for All Users**, то сразу после установки Web-сервера будет выполнена попытка его запуска.

Управление работой сервера осуществляется через группу **Control Apache Server** (меню **Пуск, Программы**, группа **Apache HTTP Server**). Здесь присутствуют три ярлыка: **Restart** (перезапуск Web-сервера), **Start** (запуск Web-сервера) и **Stop** (остановка Web-сервера).



Рис. 2.12

Обращаться к серверу можно, введя в качестве адреса в браузере:

- слово «**Localhost**» — на компьютере, на котором установлен Web-сервер. Пример URL: `http://localhost/`;
- IP-адрес — в пределах локальной сети школы или на текущем компьютере. Пример URL для текущего компьютера: `http://127.0.0.1/`. Пример URL для локальной сети (IP-адрес целевого компьютера — `192.168.0.10`): `http://192.168.0.10/`.

Если Web-сервер установлен корректно, то после ввода в браузере адреса `http://localhost/` вы увидите тестовую страницу установки Apache (рис. 2.13). Ее отсутствие говорит о *неправильном определении параметров Web-сервера*.

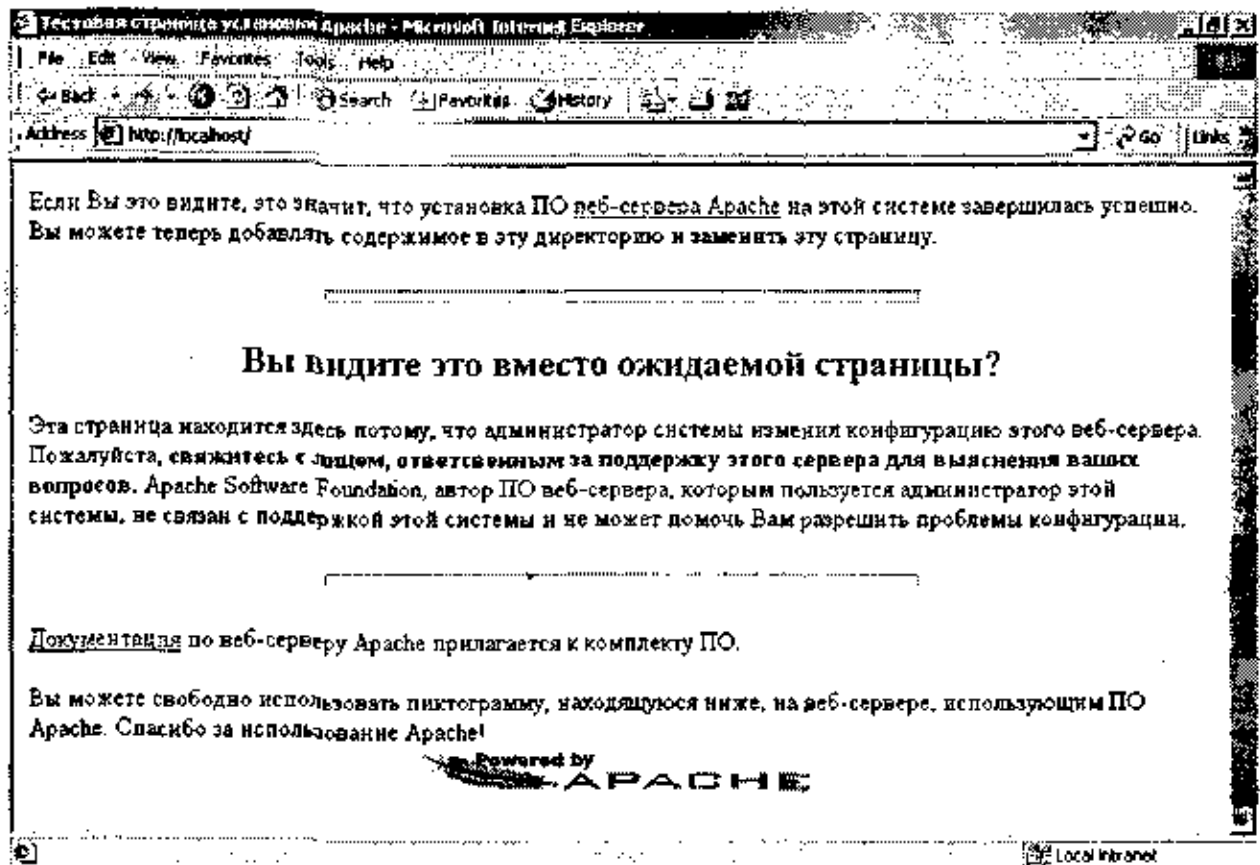


Рис. 2.13

Настройка Web-сервера осуществляется в основном путем определения директив в файле `httpd.conf`, либо обратившись к нему через ярлык **Edit the Apache httpd.conf Configuration File** (меню Пуск, Программы, группа Apache HTTP Server, Configure Apache Server), либо отыскав его внутри папки `conf` папки установки Apache HTTP Server.

Основные директивы файла `httpd.conf`:

- **ServerRoot** — полный путь к каталогу сервера (сам сервер, конфигурационные файлы, каталог с html-файлами и пр.);
- **Port** — порт, через который производится обращение к Web-серверу (по умолчанию — 80);
- **ServerAdmin** — адрес электронной почты администратора Web-сервера;
- **ServerName** — имя Web-сервера (оно обязательно должно присутствовать);

- **DocumentRoot** — «корень» сайта (папка, в которой размещаются документы, доступные для чтения);
- **DirectoryIndex** — список файлов в текущем каталоге, выдаваемых пользователю при отсутствии явного указания имени файла в URL;
- **Directory** — контейнер для определения атрибутов отдельных папок Web-сервера. Его элементы определяют параметры доступа (защиты) для этих папок и состав компонентов (скриптов), доступных для выполнения в этих папках;
- **ErrorLog** — путь и имя файла для вывода сообщений об ошибках при работе Web-сервера.

Полную информацию по настройке Web-сервера можно найти в документации, ссылка на которую имеется на тестовой странице установки Apache (см. рис. 2.13) или в папке установки Apache HTTP Server (по умолчанию — C:\Program Files\Apache Group\Apache\htdocs\manual\).



## Коротко о главном

1. Web-сервер — это программно-аппаратный комплекс, включающий в себя подключенный к Интернету компьютер и необходимое программное обеспечение.
2. Основная функция Web-сервера — предоставление документов по запросу пользователей.
3. Web-сервер в процессе работы осуществляет преобразование записей пути к файлам.
4. Адрес (URL) сервера на компьютере, где установлен этот сервер: `http://localhost` или `http://127.0.0.1`.



## Вопросы и задания

1. В чем заключаются основные функции Web-сервера?
2. Какие Web-серверы вы знаете?
3. Прокомментируйте процесс установки Web-сервера Apache.
4. Назовите основные параметры настройки Web-сервера.
5. Как проверить работоспособность Web-сервера?



## Практическое задание

Установите Web-сервер Apache в папку Web корневого каталога. Определите параметры `ServerName` и `ServerAdmin`. Задайте следующий список индексных файлов: `index.htm`, `index.html`. Проверьте работоспособность установленного сервера. Установите сервер на автозагрузку (для систем на базе Windows 95/98/ME или в случае первоначальной установки с ручным запуском) или как службу (для систем на базе Windows NT и при установке в качестве сервиса).



## Темы рефератов

1. Этапы развития Web-серверов. Их компоненты и функции.
2. Сравнение различных Web-серверов.



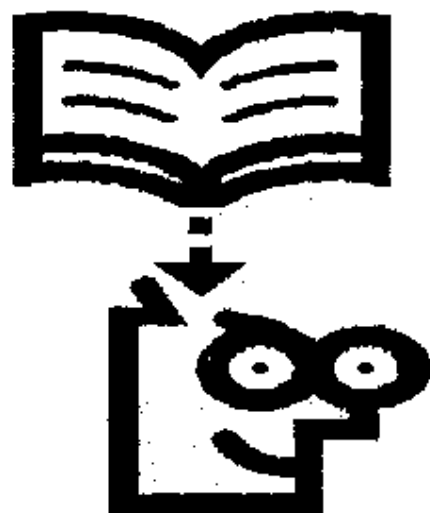
## Демонстрационный плакат

Процесс установки и запуска HTTP-сервера Apache.



## Глава 3

# ОПРЕДЕЛЯЕМ ФУНКЦИИ САЙТА



Определение функций сайта является важным этапом его создания. От этого зависит, какая информация будет представлена на сайте, ее объем и множество других параметров. Функции сайта прежде всего определяются характером информации, которая будет на нем размещена, и целью создания сайта. В основном ресурсы сети Интернет разрабатываются для развлечения, информирования, общения посетителей сайтов.



На компакт-диске, прилагаемом к учебному пособию, имеется плакат, демонстрирующий основные виды сайтов. Плакат содержится в файлах poster3-1.vsd и poster3-1.cdr в папке chapter3.

## Виды сайтов

Сайты сети Интернет можно разделить по следующим категориям:

- **Персональные (личные) сайты.** Личные сайты размещаются в основном на бесплатных серверах (например, на сайте [www.parod.ru](http://www.parod.ru) — рис. 3.1), из-за чего на них много различного рода рекламной информации (баннерной рекламы). Использование Мастеров для создания страниц такого сайта сильно упрощает их проектирование, особенно для пользователей, не имеющих навыков работы с подобными серверами, создания и кодирования веб-документов. Это своего рода «визитные карточки», которые создаются из чистого интереса и редко имеют практическую значимость. Посещаемость большинства подобных сайтов крайне мала. Типичный личный сайт состоит из нескольких страниц (их редко бывает больше 5) с относительно простым дизайном. Разновидностью подобных сайтов являются «виртуальные резюме», используемые для предоставления краткой информации о владельце (в том числе для ссылки на них при поиске работы). Время существования таких сайтов обычно невелико.

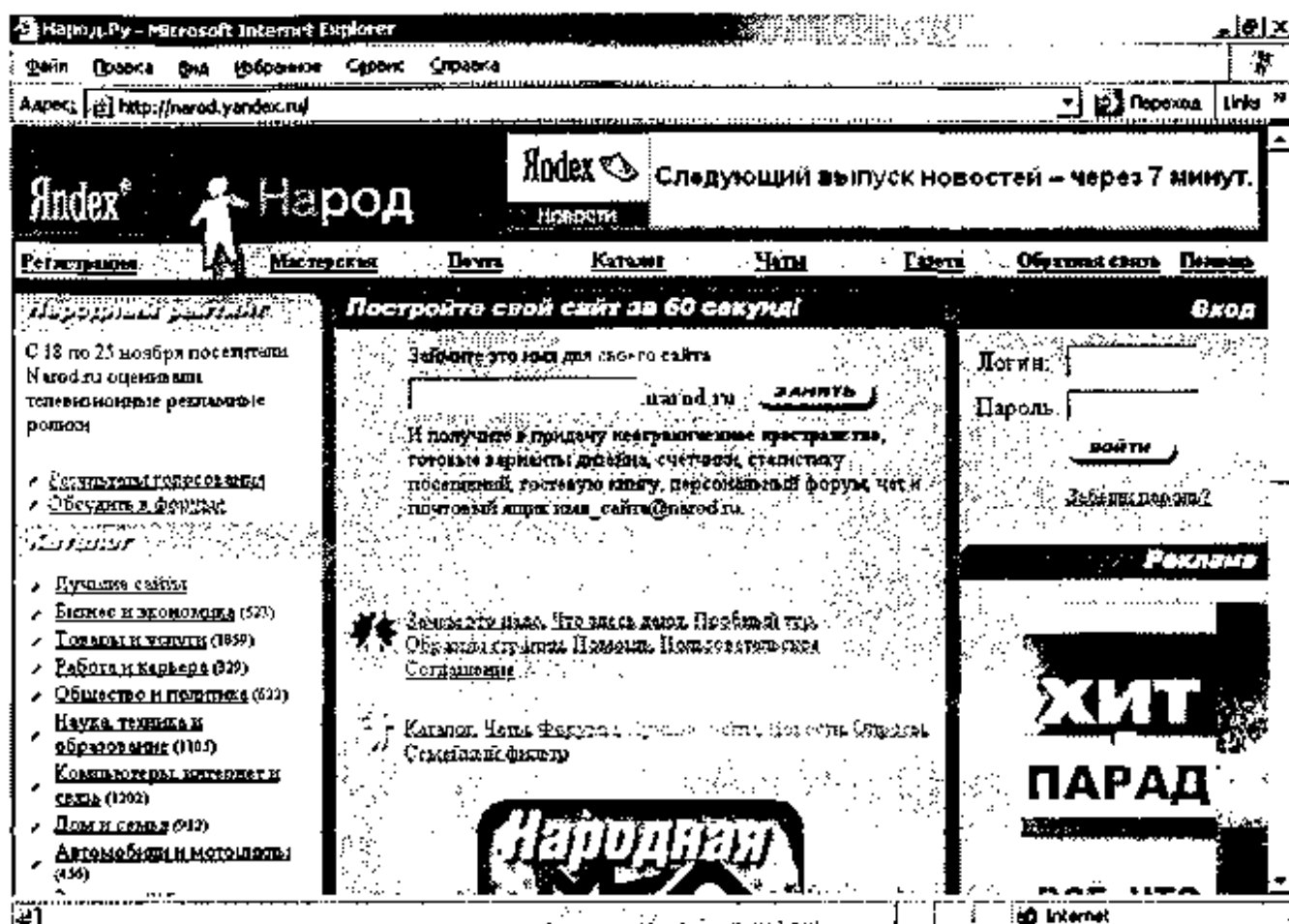


Рис. 3.1

- Службы новостей, электронные доски объявлений, электронные версии печатных изданий ([www.lenta.ru](http://www.lenta.ru), [www.gazeta.ru](http://www.gazeta.ru)) и т. д. — рис. 3.2). Их создание — работа очень трудоемкая. Такой сайт представляет собой оболочку с постоянно меняющейся информацией. Определенной тематики у большинства подобных сайтов не существует, основной их функцией является «временное» предоставление актуальной информации. (Понятие «временное» здесь относительно, так как практически все такие системы имеют архивы, в которых хранится уже не актуальная информация.)

Разновидностью описываемых сайтов являются специализированные службы новостей. Основное их отличие заключается в конкретизации тематики информации, размещаемой на сайте. Например, это может быть сайт, предоставляющий информацию только в области компьютерных технологий ([www.compulenta.ru](http://www.compulenta.ru) — рис. 3.3).

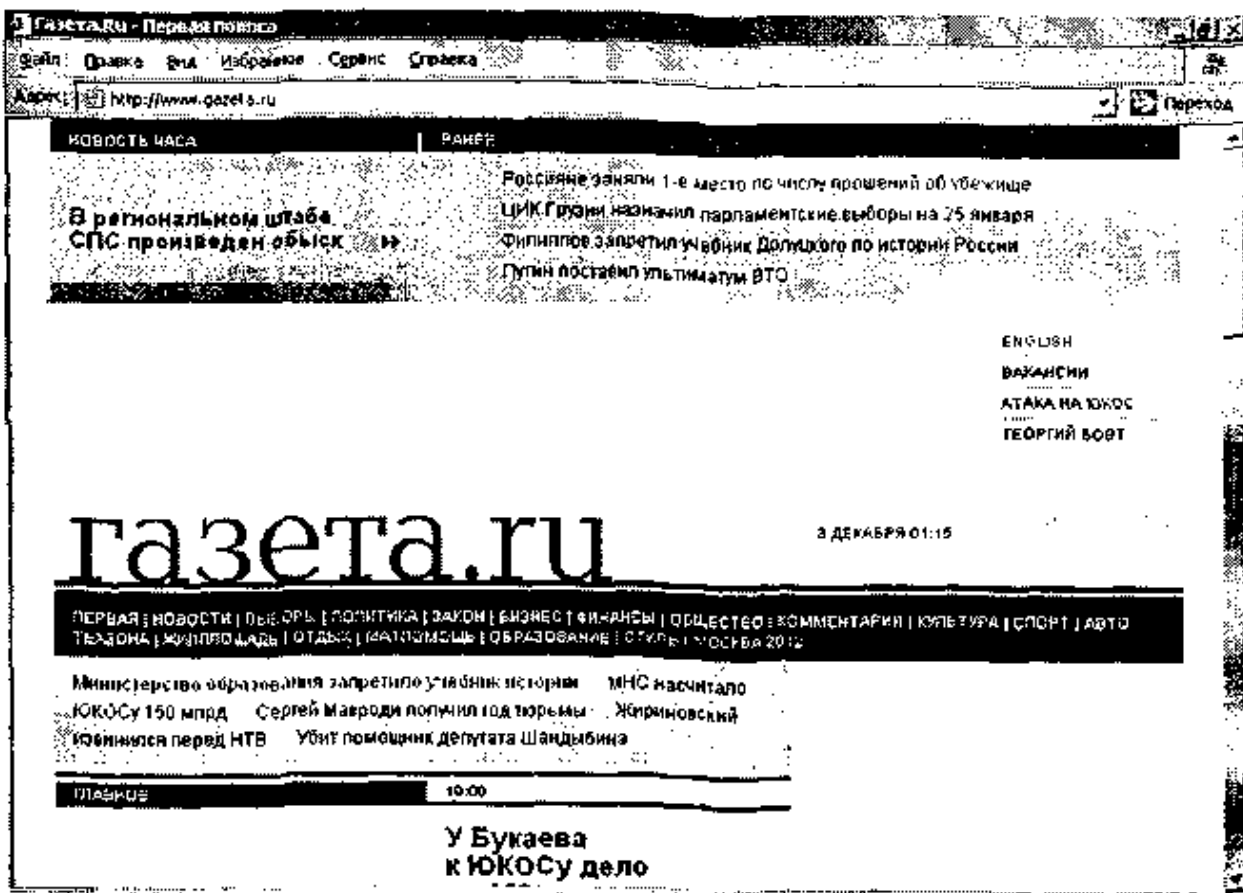


Рис. 3.2

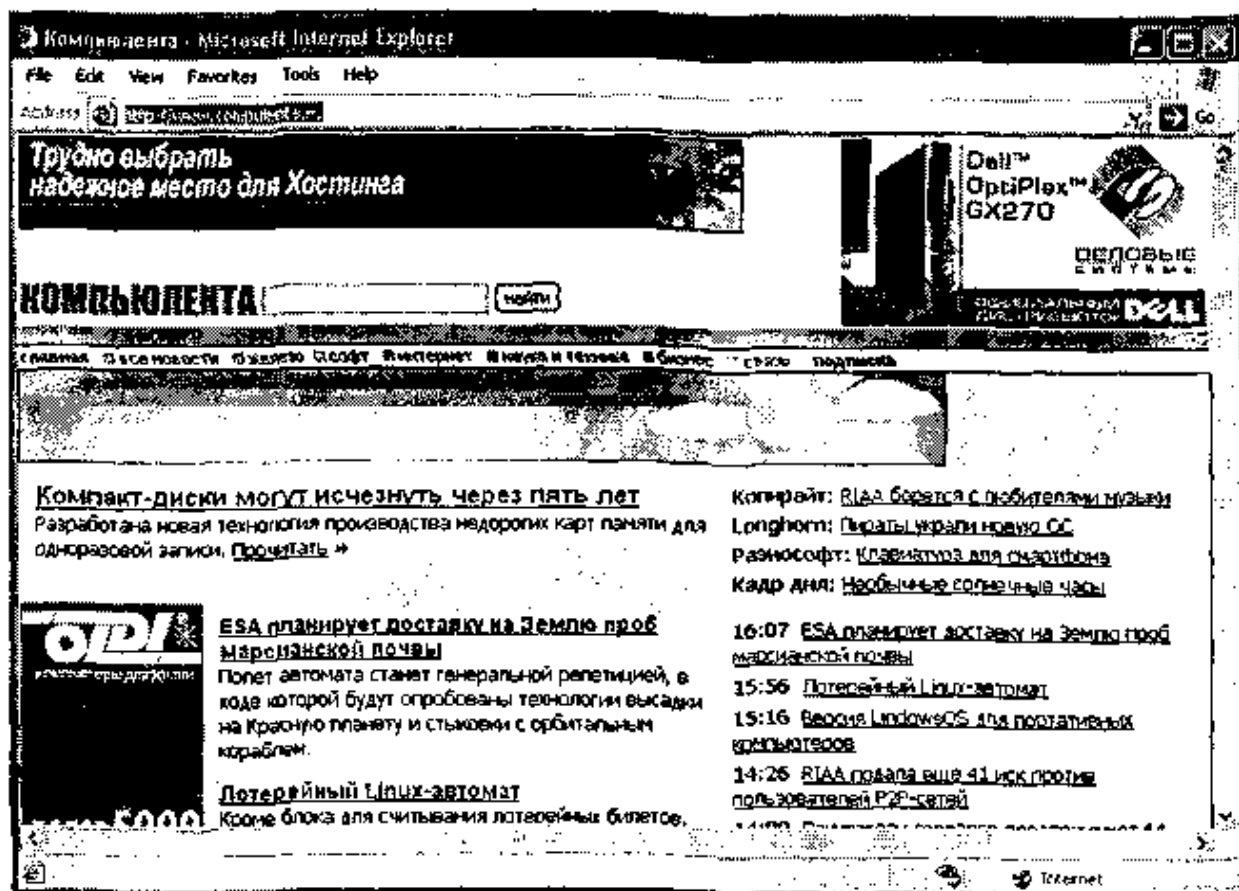


Рис. 3.3

- Сайты — сборники документов определенной тематики и виртуальные библиотеки. Содержат материалы по выбранной тематике (например, [www.citforum.ru](http://www.citforum.ru) — рис. 3.4). Основной принцип, по которому проектируются подобные сайты (равно как и службы новостей, поисковые системы и электронные магазины), — это преобладание информационного содержания над формой его представления. Сайты — сборники документов используются для предоставления различной информации справочного характера, для которой вопрос оперативной актуальности не ставится. Многие материалы хранятся на таких сайтах годами.

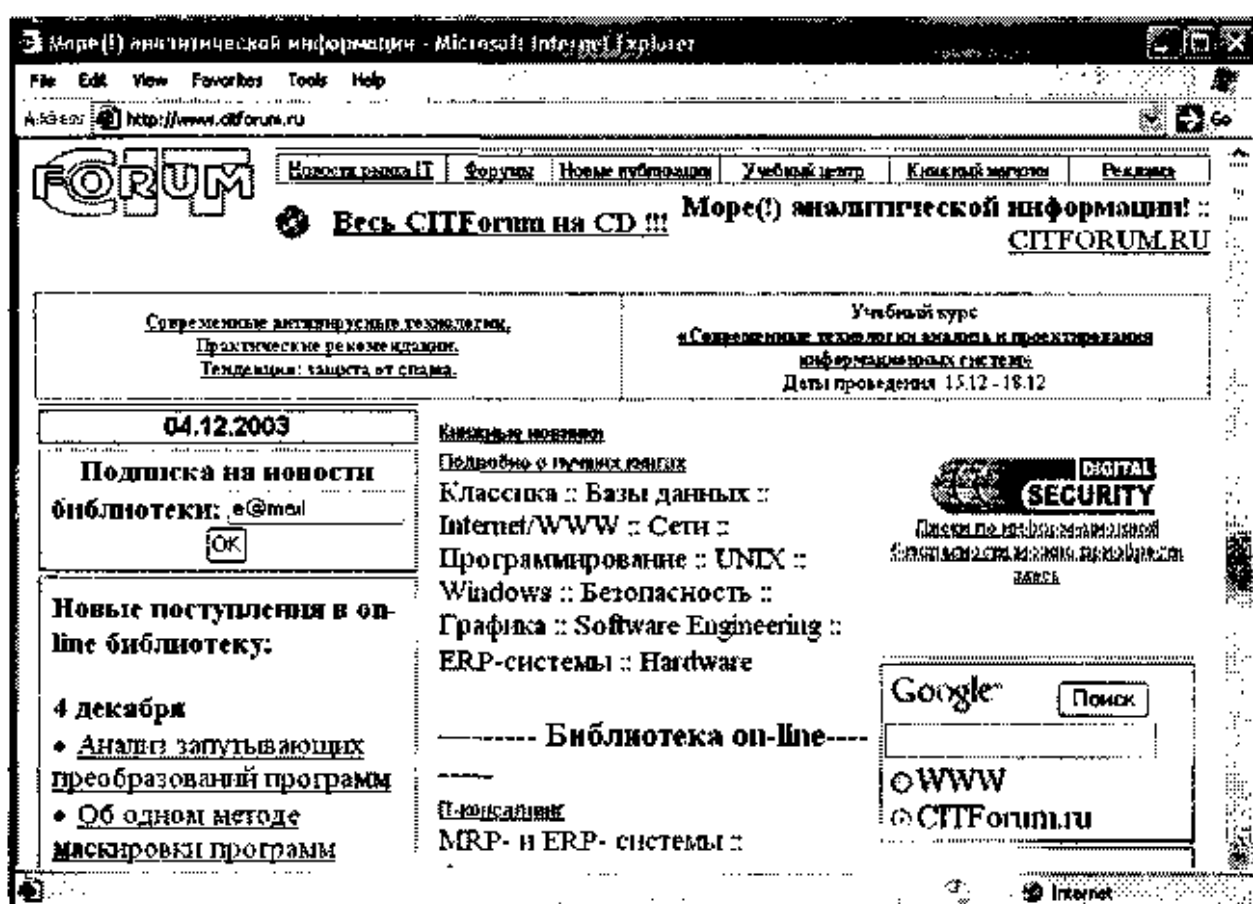


Рис. 3.4

- Поисковые системы обеспечивают возможность навигации среди веб-страниц, количество которых постоянно увеличивается. Такая навигация обеспечивается за счет периодического индексирования сайтов сети Интернет (записи информации о них) в огромные базы данных. Поиск и предоставление адресов по содержимому веб-страниц — основная функция подобных систем. В большинстве случаев поисковые системы по принципам работы с ними и дизайну — это аналоги иерархических каталогов библиотек. Начальные страницы со-

держат основные рубрики, которые шаг за шагом уточняются (примеры: [www.google.ru](http://www.google.ru) — рис. 3.5, [www.ya.ru](http://www.ya.ru) — рис. 3.6), пока не будет достигнут нужный результат, либо можно сразу ограничить список сайтов в текстовой форме, определив параметры поиска. В данную группу также можно отнести поисковые энциклопедические системы (например, <http://search.km.ru>).

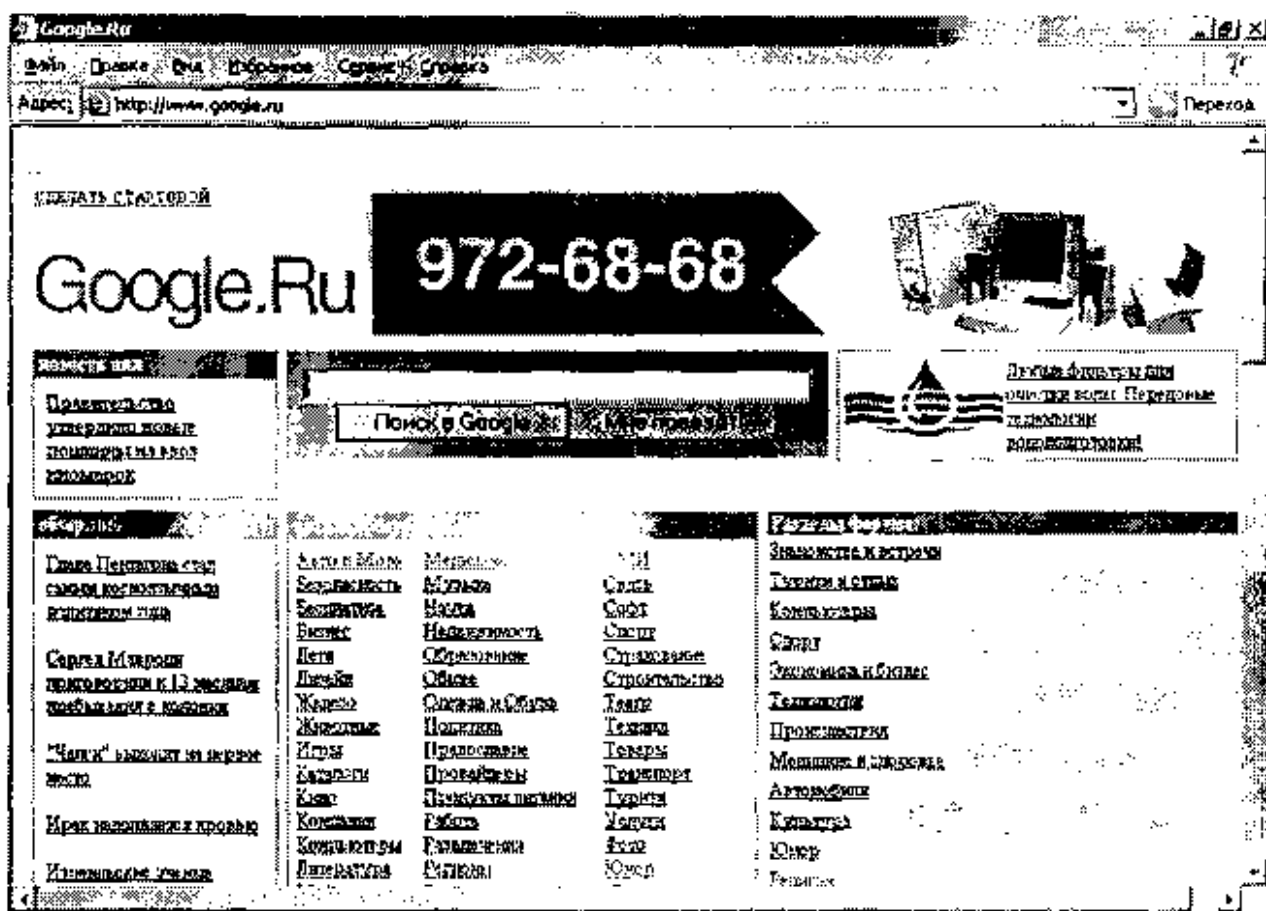


Рис. 3.5

- Интернет-магазины (рис. 3.7) содержат сведения о различных товарах и услугах, снабжаются иллюстративными материалами и предоставляют возможность оформить покупку через сеть Интернет. Разработка подобных систем требует особых знаний и навыков. Обычно на таких сайтах имеются каталоги, по которым можно выбирать различные товары.
- Сайты, обеспечивающие интерфейс с различными приложениями и службами. Наибольшее распространение получили сайты, обеспечивающие доступ к электронной почте (рис. 3.8). Они создаются для упрощения работы с wybranymi системами, открывая их для большего количества потенциальных пользователей.

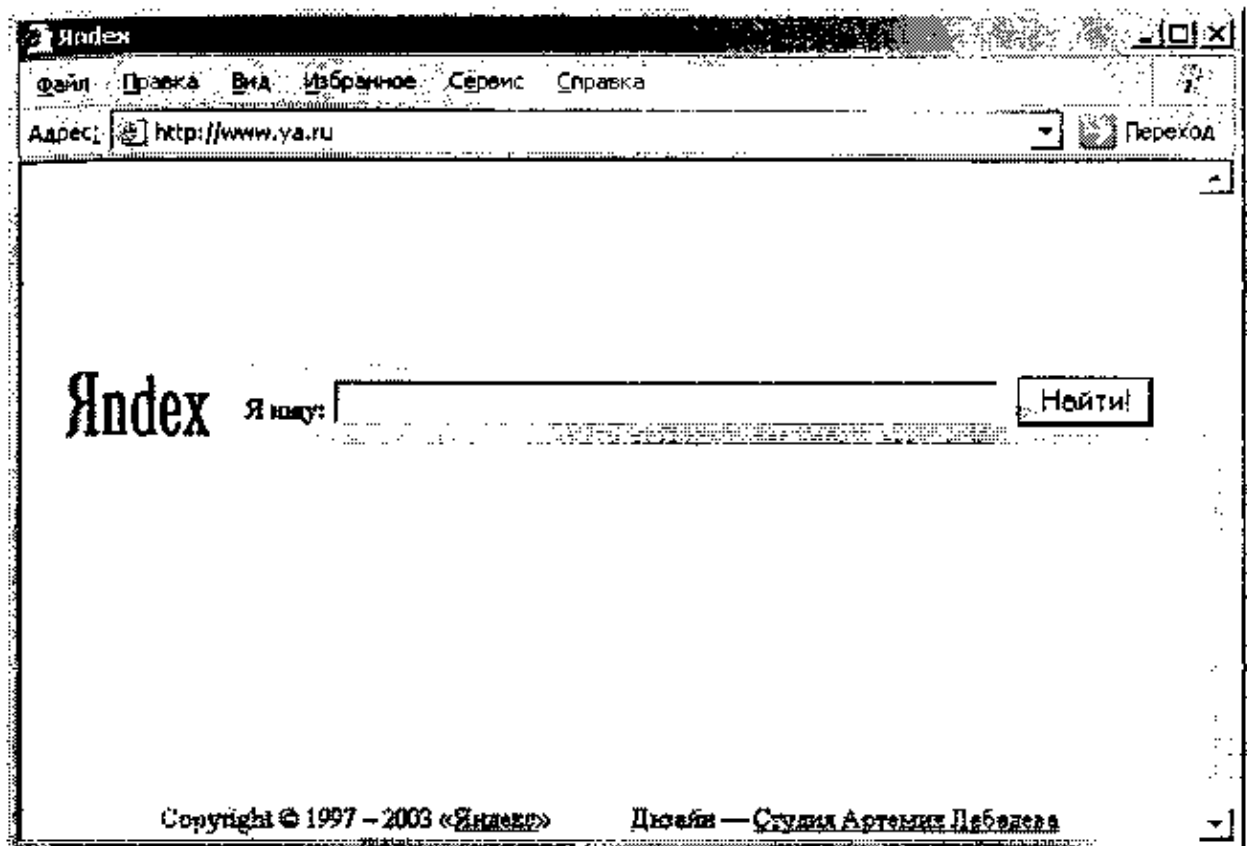


Рис. 3.6

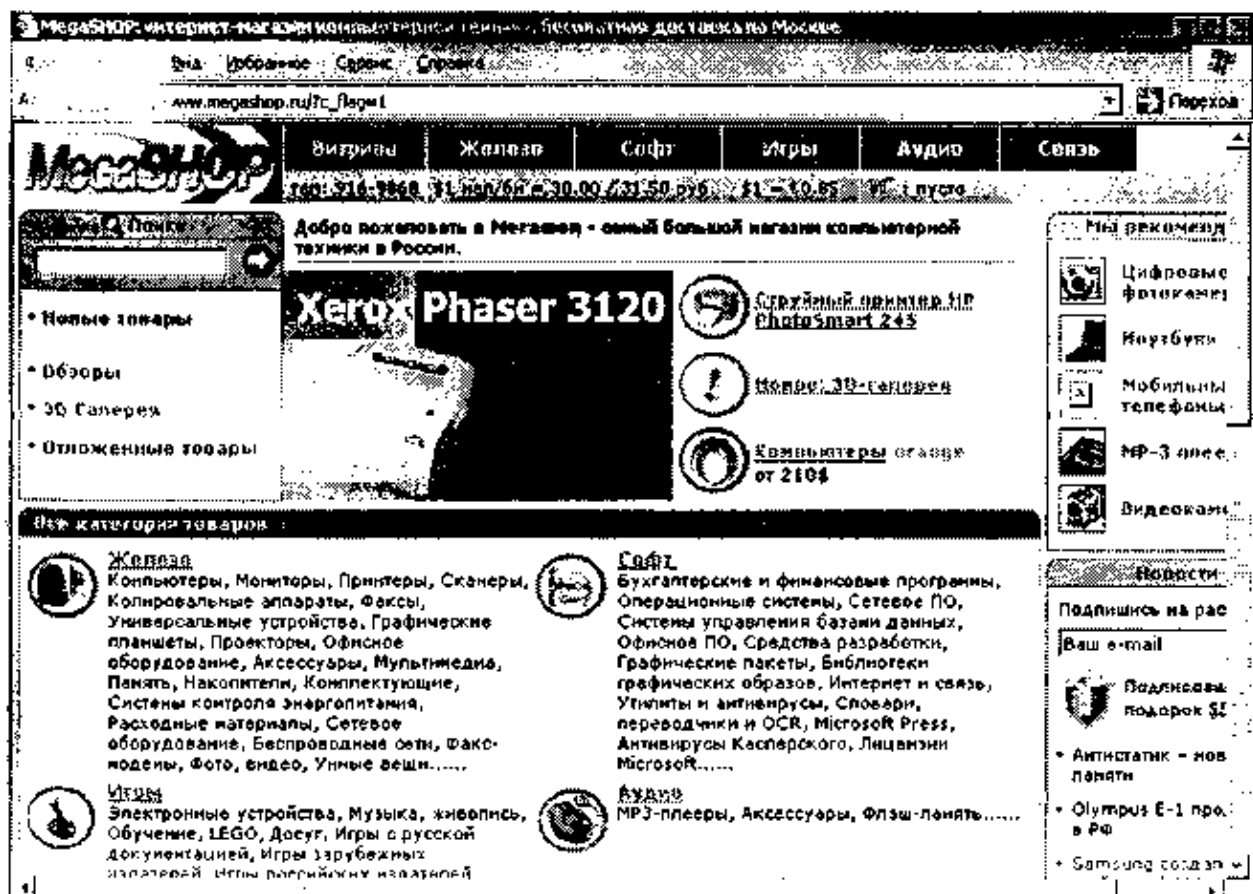


Рис. 3.7

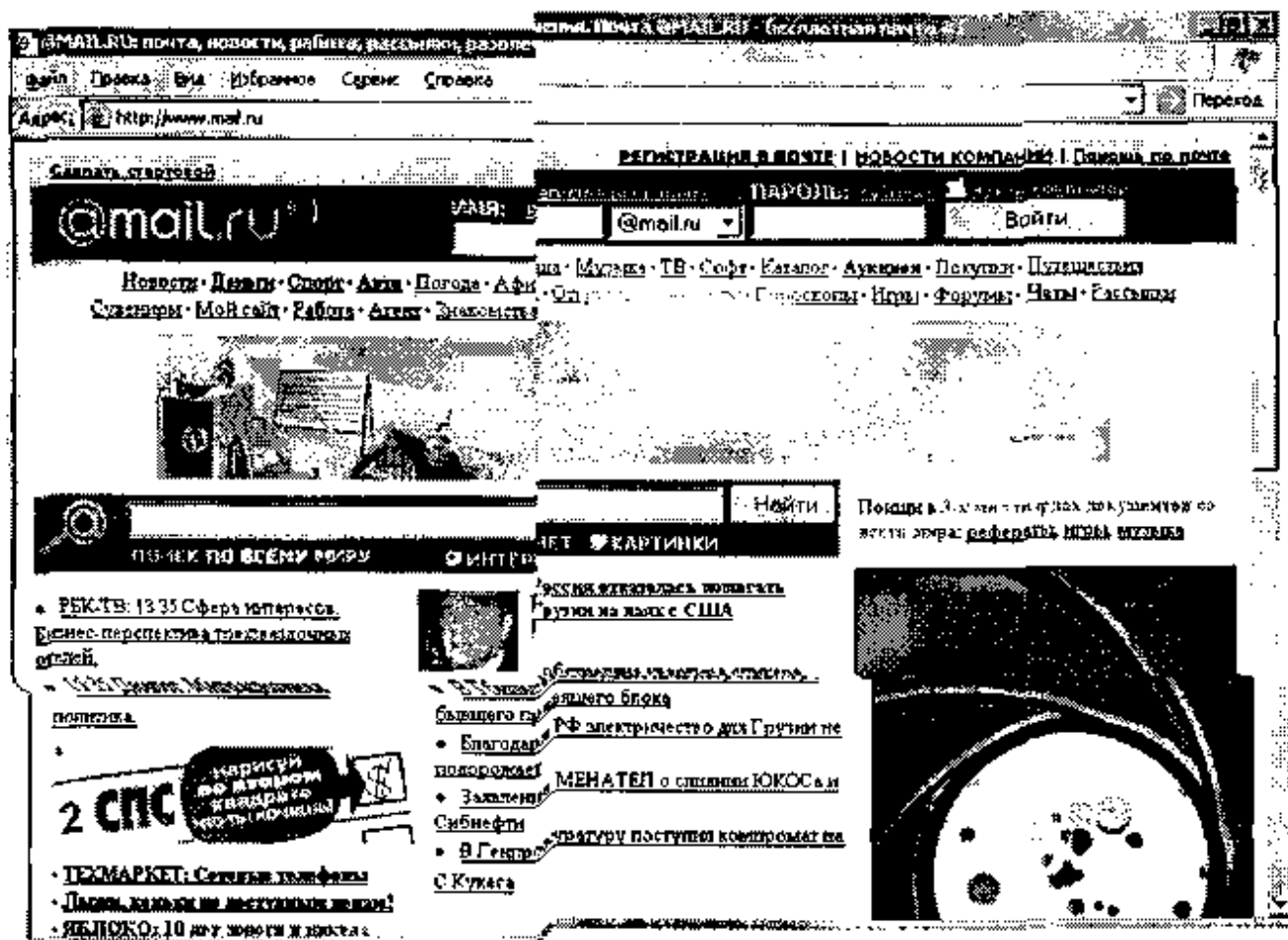


Рис. 3.8

- **Корпоративные сайты** создаются в основном в информационных целях. Однако подобные сайты могут быть снабжены дополнительными функциями, обеспечивая в едином интерфейсе доступ к различным сервисам (электронная почта, удаленный доступ), выполнять функции «архивных» сайтов и организационные функции (рис. 3.9). В связи с этим структура многих подобных сайтов предусматривает веб-страницы, открытые как для всех пользователей (в основном выполняющие информационные функции), так и только для определенных групп пользователей (в частности, для сотрудников организации).

- «Архивные» сайты используются для хранения и предоставления различных файлов. В большинстве случаев это ftp-серверы, однако многим подобным серверам добавляется веб-интерфейс (пример — [www.freeware.ru](http://www.freeware.ru)). Доступ к таким ftp-серверам также возможен из браузера (рис. 3.10).

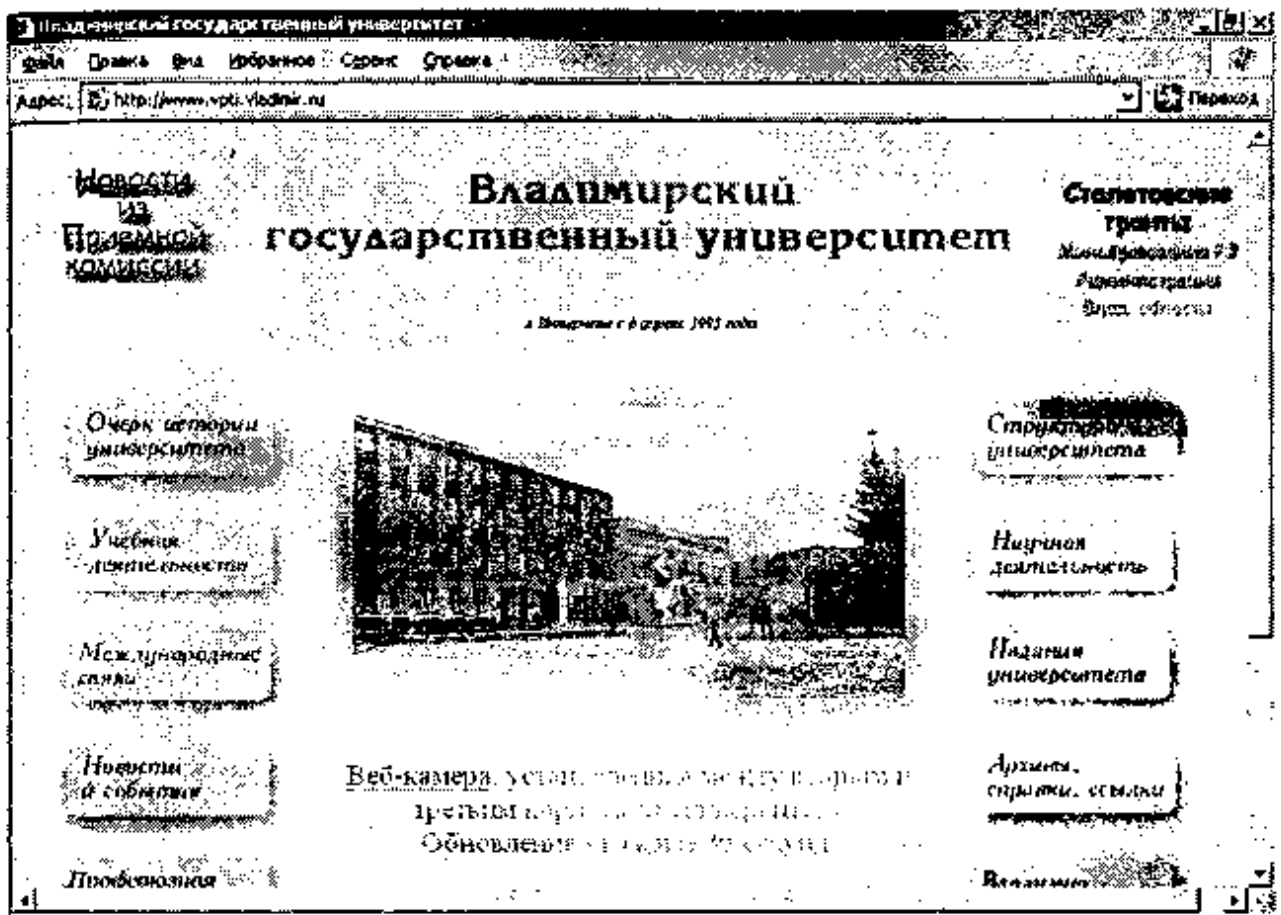


Рис. 3.9

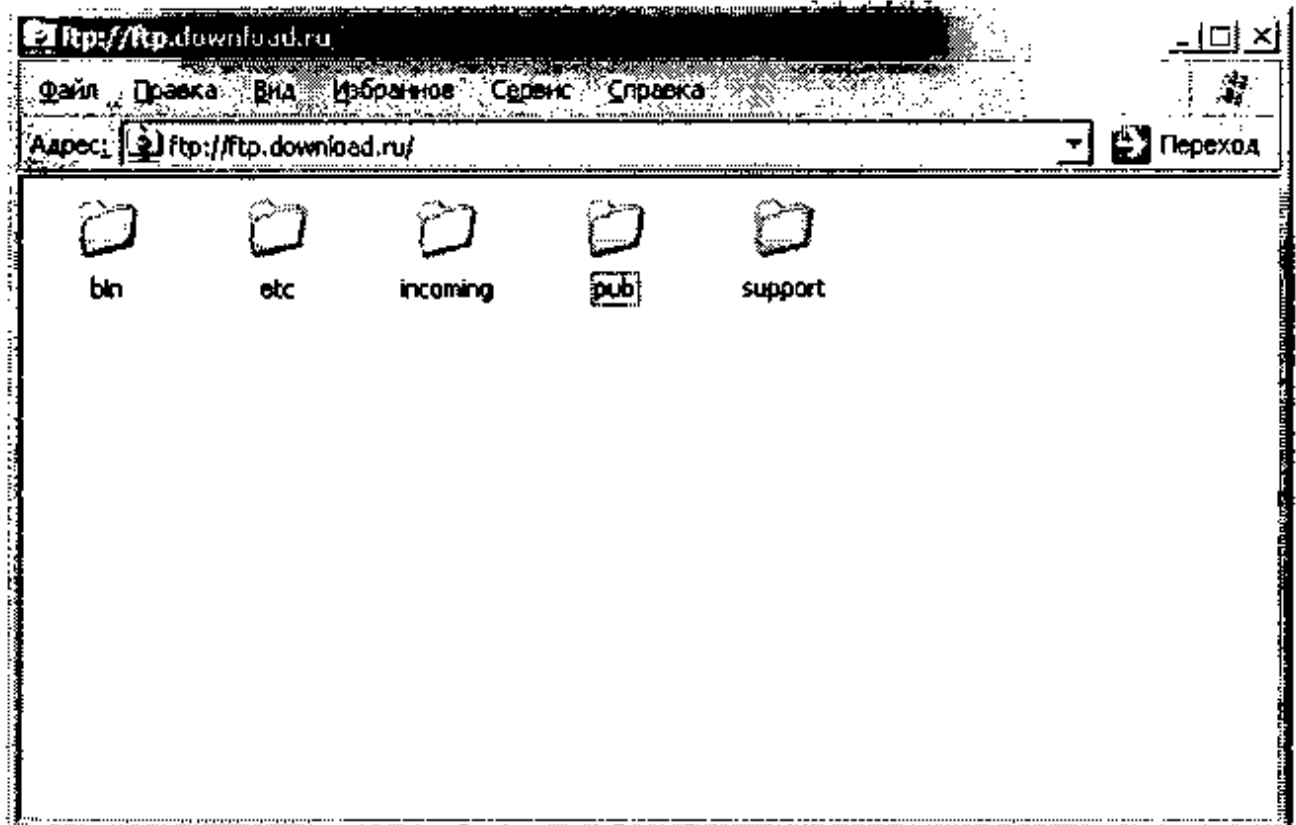


Рис. 3.10



- Ресурсы для общения основаны на идее двухсторонней связи. Они предоставляют пользователю возможность вводить те или иные данные с клавиатуры, влиять на процесс выполнения определенных действий. Это могут быть различные форумы, конференции и т. п. Основная функция здесь — это обмен сообщениями.

## Функции сайта

Создавая сайт, необходимо прежде всего ответить на следующие вопросы:

- каков будет возраст посетителей сайта, их уровень образования и потребности;
- каковы будут глубина сайта, его внутренние и внешние ссылки, ресурсы, которые на нем будут представлены;
- каким будет наполнение страниц, их фон, используемая графика, звук или видео;
- как будет осуществляться навигация, управление, ввод данных на страницах сайта, а также отмена неправильно выполненных действий.

*Рассмотрим пример.* Начнем определение функций с выделения основных групп посетителей сайта:

- ученики;
- учителя;
- администрация школы;
- родители;
- все прочие, кто интересуется школой, например, предполагая устроить в нее своих детей.

Каждая из этих групп посетителей определяется различными потребностями. Для каждой группы необходимо определить права доступа к выбранной информации, ее содержание и детальность представления.

Для ученика подобной информацией может служить расписание занятий, выданные задания и результаты их проверки. Сайт также может предоставлять ученикам возможность обсудить те или иные задания во внеурочное время.

Для учителя сайт может стать неким аналогом доски объявлений, на которую «вывешиваются» материалы и задания для

углубленного изучения, результаты проверок контрольных работ, предоставляя интересующимся ученикам дополнительные возможности при изучении той или иной дисциплины. Сайт также может использоваться в качестве электронной библиотеки и для организации интерактивных тестовых мероприятий или форумов.

Для родителей на сайте может быть представлена информация, касающаяся программы обучения по каждой дисциплине, планируемых мероприятий и условий участия в них. Один из разделов сайта может представлять собой электронный вариант классного журнала, в котором проставляются результаты успеваемости и посещаемости.

Для администрации школы сайт может служить аналогом «визитной карточки» школы, где отражены основные достижения учебного заведения за последнее время, история и традиции школы, обязательный минимум образования и профильные дисциплины, преподавательский состав. Также на сайте может быть представлена информация о самом учебном заведении, например схема проезда, количество классов, лабораторий и другие сведения, информация об общешкольных мероприятиях.

К функциям такого сайта может быть также добавлена возможность обмена сообщениями.

Однако не следует забывать, что некоторые из таких функций могут оказаться не реализуемыми, например, по техническим причинам (скажем, нет возможности организовать «архивный» сайт, если дискового пространства хватит только для установки сервера).

## **Подведем итоги**

Таким образом, определение функций сайта — важная задача его разработки. На данном этапе определяются вид сайта, ресурсы, которые будут представлены на нем или к которым будет осуществляться доступ. Все это в конечном счете влияет на объем работ по созданию сайта, а также определяет дальнейший процесс его развития. Анализ различных видов сайтов, представленных в сети Интернет, позволит вам лучше понять процесс определения функций.



## Коротко о главном

1. Сайты сети Интернет создаются для реализации различных функций, которые определяют их вид.
2. Пользователи часто создают персональные странички с информацией о себе, пользуясь Мастерами и услугами бесплатных серверов.
3. Многие организации создают рекламные сайты для привлечения клиентов.
4. Корпоративные сайты предоставляют широкий набор функций, включая возможности удаленного доступа к различным информационным ресурсам организации.
5. Основная функция поисковых систем — поиск сайтов по запросам пользователей.



## Вопросы и задания

1. В чем заключается цель определения функций сайта?
2. Какое влияние функции сайта оказывают на его вид?
3. Назовите виды информационных сайтов.
4. Определите назначение и функции поисковых систем.
5. Назовите назначение и функции корпоративных сайтов.



## Практическое задание

Определите функции и вид создаваемого вами школьного сайта. Опишите причины выбора вида вашего сайта и объясните его функции.



## Темы рефератов

1. Обзор личных сайтов по определенной тематике.
2. Обзор поисковых систем.
3. Обзор информационных ресурсов периодических и специализированных изданий.

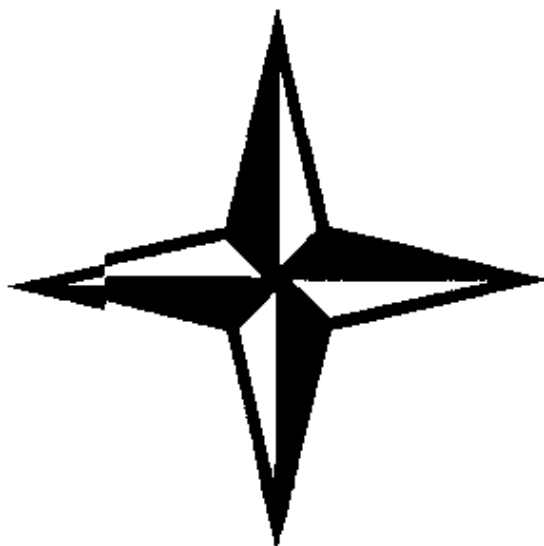


## Демонстрационный плакат

Виды сайтов.

## Глава 4

# ПРОЕКТИРУЕМ ВНЕШНИЙ ВИД САЙТА



Стартовая страница — это первое, что видит пользователь после ввода адреса вашего сайта в адресной строке браузера и нажатия клавиши Enter. Важно произвести на посетителя хорошее впечатление, чтобы задержать его на вашем сайте, чтобы у посетителя появилось желание посмотреть другие страницы этого сайта.

Следует учесть, что у большинства пользователей пропадает интерес к сайту, если время на загрузку страницы превышает 30 секунд. Кроме того, у пользователя может возникнуть неприятное впечатление, если на просматриваемой странице нет сведений о сайте или если на сайте имеются «мертвые ссылки» (на несуществующие страницы данного сайта или на другие сайты).

## Создание стартовой страницы

Стартовая (начальная) страница обычно содержит информацию (текст и, возможно, графические элементы) о тематике и назначении сайта. При этом можно рекомендовать начинающим разработчикам придерживаться следующих принципов создания эффективной и удобной для пользователей стартовой страницы.

- Важно обеспечить для пользователя быструю идентификацию сайта.
- Графические элементы, расположенные на начальной странице, должны загружаться достаточно быстро.
- На стартовой странице коммерческого сайта следует поместить наиболее важную информацию о вашей организации или услугах, предоставляемых ею. Если эта организация обладает хорошо знакомым логотипом (товарным знаком), то его присутствие обеспечивает наилучшую узнаваемость сайта.

- Следует помнить, что стартовая страница — это не обложка книги или журнала: кроме чисто иллюстративных функций, она также выполняет и практические функции, определяя *ссылки на основные разделы сайта*. Поставьте себя на место пользователя и постарайтесь продумать, что бы вы хотели видеть на этой странице.
- При создании сайта следует учесть, что пользователи будут переходить на выбранные страницы в поисках нужной им информации, следовательно нужно обеспечить на стартовой странице удобные *механизмы поиска* и перехода к другим страницам, а сама начальная страница должна играть роль оглавления вашего сайта.
- Опишите функции вашего сайта несколькими короткими, но предельно точными и конкретными фразами, не заставляя пользователей тратить время на выяснение деталей. *Информацию о новинках* на сайте также рекомендуется помещать на стартовой странице, для чего необходимо учесть эту возможность при разработке общего дизайна страниц.
- Разрабатывайте страницы с учетом *типичного разрешения дисплея*, установленного у большинства пользователей, чтобы избежать появления полос скроллинга в развернутом окне браузера.
- Не забудьте поместить на стартовую страницу (равно как и на остальные) *контактную информацию* — только адрес электронной почты, либо ссылку на отдельную страницу с полной контактной информацией. Рекомендуется также поместить на странице адрес электронной почты администратора сайта («веб-мастера»). Полезно также дать пользователям возможность подписаться на рассылки с вашего сайта, если таковые планируются.
- Для графических ссылок (в виде рисунков) следует не включать *обрамление специальной рамкой*, стандартно используемой для графических ссылок. Однако в некоторых случаях сама форма таких ссылок (например, в виде кнопок) является интуитивно понятной, а наличие обрамления приводит к нарушению общей концепции страницы (рис. 4.1).

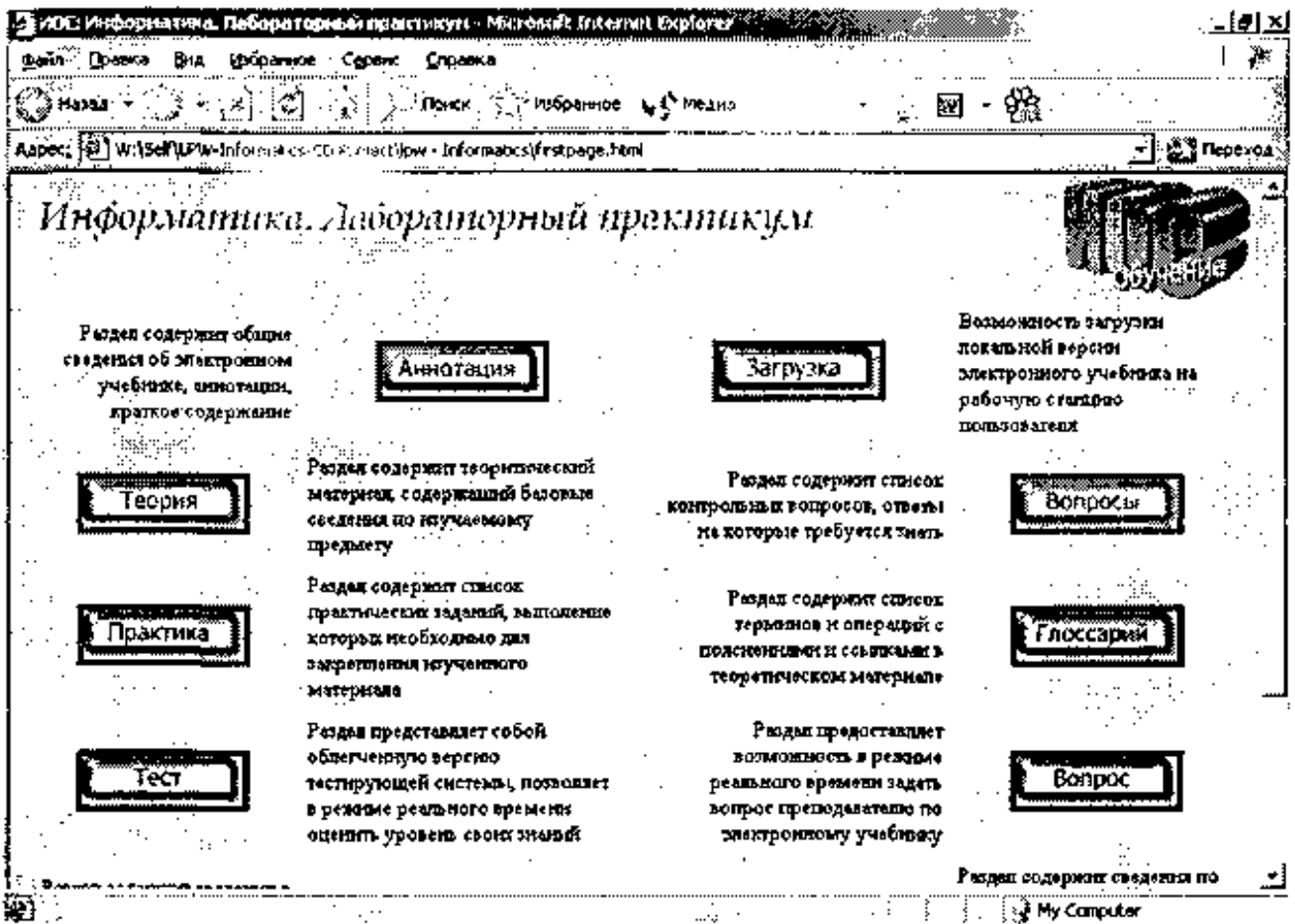


Рис. 4.1

В этом случае следует назначить для подобных элементов (при написании соответствующих HTML-тегов) границу нулевой толщины.

- Проверьте уже созданную страницу на всех доступных вам (по размеру диагонали) мониторах и во всех наиболее популярных браузерах. Желательно также провести такую проверку при различных разрешениях дисплея (рис. 4.2). Это поможет выявить ошибки дизайна (слишком маленькие или слишком большие элементы), определить, как лучше разместить (и представить) основные элементы, чтобы их было видно при любых разрешениях и на различных мониторах. Это также позволит выявить основные ошибки при написании HTML-кода, так как каждый браузер обрабатывает такие ошибки по-своему.

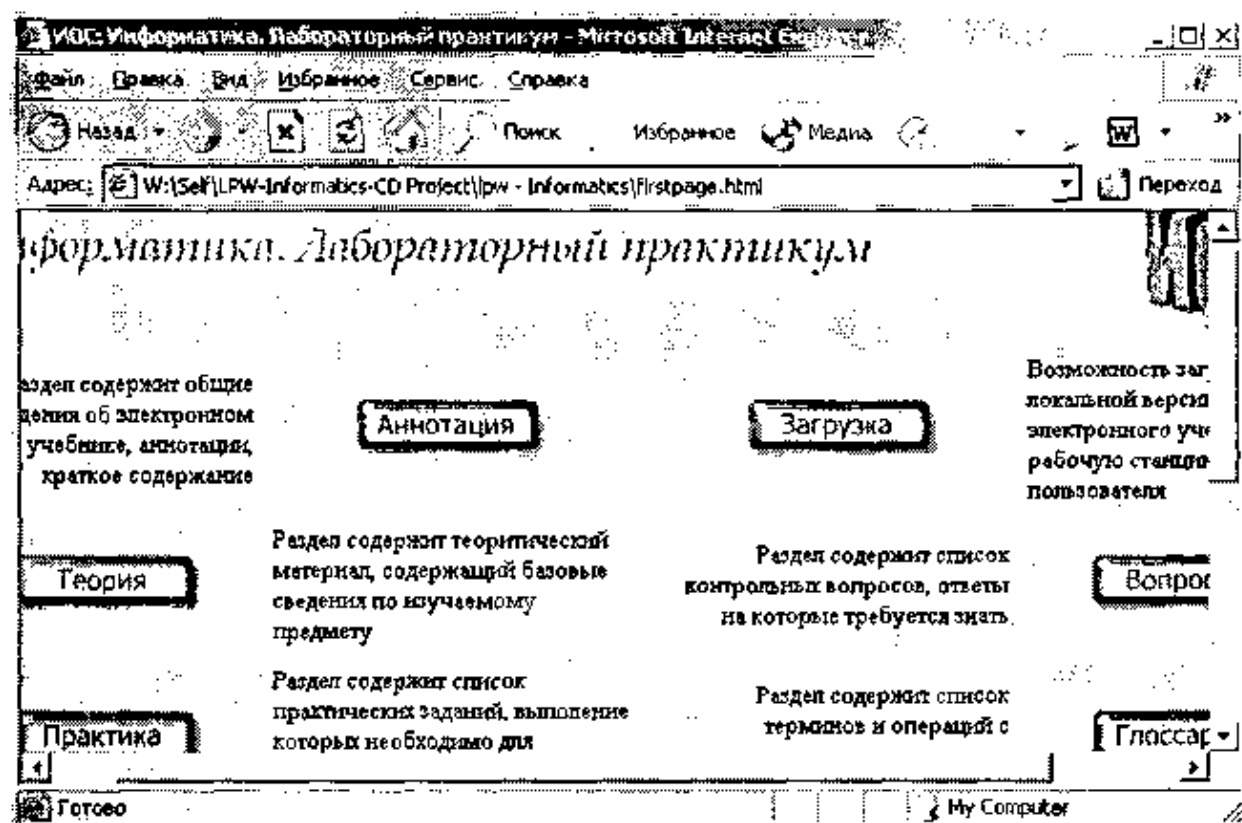


Рис. 4.2

## Прочие страницы сайта: дизайн и навигация

Стартовая страница задает общий тон всего сайта, но она обычно используется только как «обложка», без возможности возврата на нее при перемещении на другие страницы сайта.

**Навигация по сайту.** Для всех других страниц сайта становится актуальным вопрос навигации. Поэтому необходимо тщательно продумать расположение элементов навигационных панелей и ссылок на всех его страницах.

Для организации быстрой и удобной навигации можно использовать все преимущества, которые дают ссылки и «якоря»<sup>1</sup> (например, для перемещения в начало страницы или к ее оглавлению). Однако использование текстовых ссылок не всегда оправданно, так как они требуют много места и могут «не вписаться» в дизайн. В таких случаях можно использовать небольшие иллюстрации, в том числе анимированные. Вообще же возможности персональных компьютеров постоянно растут, вместе с ними

<sup>1</sup> «Якорем» называется метка, добавляемая на веб-страницу, на которую позже можно запрограммировать точный переход по гиперссылке с той же или другой страницы.

растут и возможности браузеров. Ушло то время, когда текстовые ссылки были основным элементом дизайна веб-страниц, хотя и нельзя сказать, что сегодня от них пора отказаться бесповоротно. Но с ростом возможностей браузеров стала проявляться тенденция самого широкого использования графических элементов — и не только в качестве иллюстраций. Например, технология Macromedia Flash позволяет сделать сайт практически полностью графическим и интерактивным.

Заметим, что для всех страниц сайта также действительны все требования и рекомендации, относящиеся к стартовой странице и описанные ранее.

**Карта сайта.** При проектировании сайта важно продумать взаимосвязь всех элементов и страниц — разработать *карту сайта*. Это позволит определить параметры навигационных элементов сайта, которые должны присутствовать на его страницах.

**Разработка страниц средствами пакета Microsoft Office.** Для быстрого создания веб-страниц можно использовать различные приложения для обработки текстовой, табличной, графической и другой информации, например пакет MS Office (рис. 4.3).

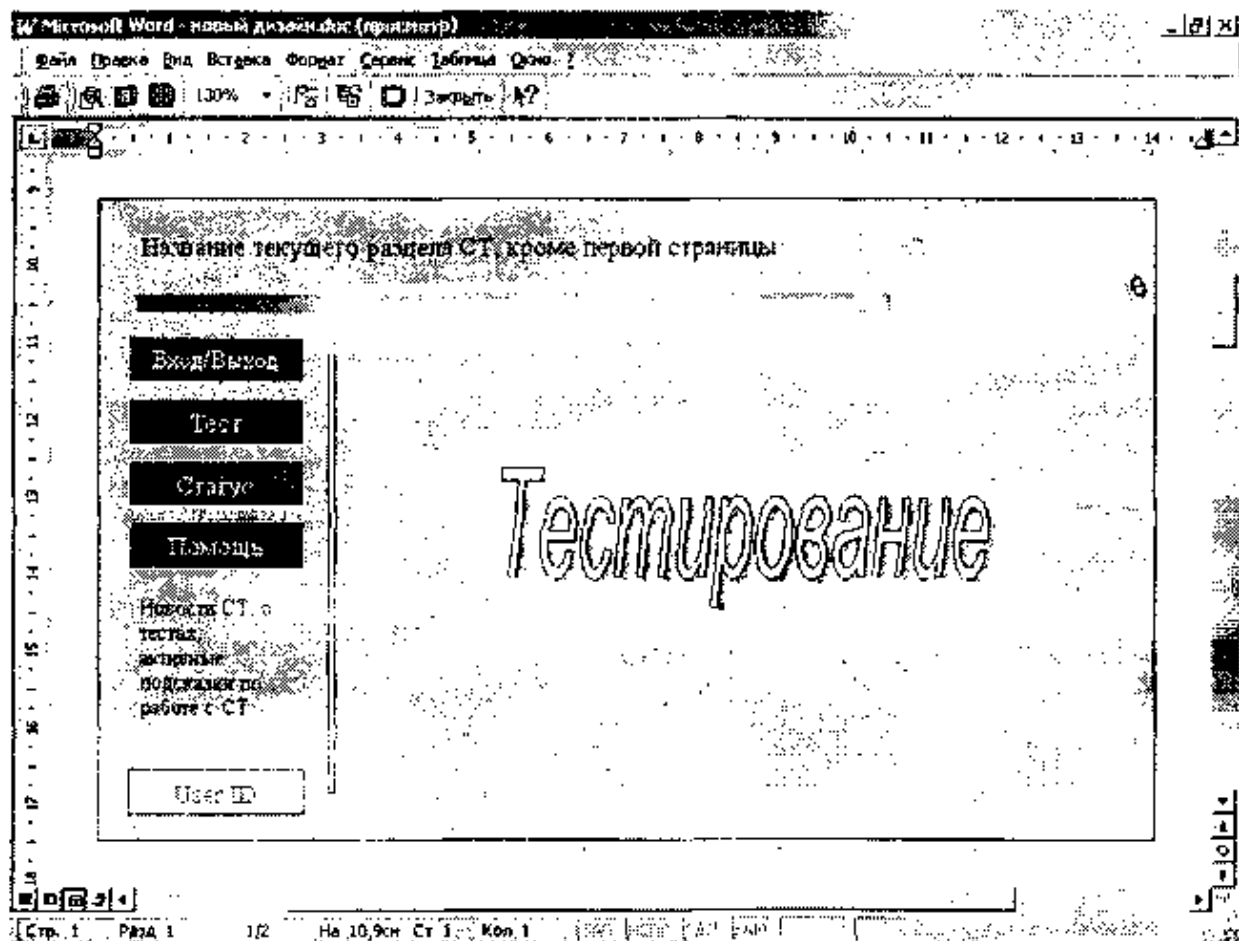


Рис. 4.3



Такие системы предоставляют широкие возможности по созданию веб-документов с помощью встроенных Мастеров, в том числе с выбором предусмотренных в них вариантов общего дизайна.

Ярким примером такой программы может стать приложение PowerPoint из состава пакета MS Office. В нем содержится множество готовых макетов, на основе которых можно создавать свои страницы (слайды). Кроме того, подобные программы можно использовать и для проверки орфографии в тексте веб-страниц или при создании эскизов их дизайна.

Существует также целый ряд различных программ для создания различных элементов дизайна веб-страниц (например, Хага WebStyle). Но нужно не забывать, что основная цель веб-дизайна заключается не в использовании превосходной графики, а в создании удобного, быстрого и интуитивно понятного интерфейса доступа к предоставляемой вами информации. Кроме того, следует предусмотреть, чтобы у браузера не возникало проблем с отображением элементов дизайна из-за сложной структуры HTML-документа.

Не надо забывать и про используемую на сайте цветовую гамму. Для всех элементов она должна быть гармоничной, параметры текста (шрифт, размер, выравнивание) должны совпадать для однотипных элементов страниц.

## Пример разработки эскиза веб-страницы



**Упражнение 1.** Разработка эскиза страницы сообщения об ошибке.

1. Посетитель сайта должен знать, где он находится, поэтому на создаваемой странице должен присутствовать общий заголовок сайта. Для отделения заголовочной части документа от его информационного содержания используем разделительную линию.
2. Определим тему разрабатываемой веб-страницы. В данном случае это особенно необходимо, так как пользователь получает страницу не с тем содержимым, которое он запрашивал. Поэтому надо не только сообщить пользователю об ошибке, но и дать краткое описание этой ошибки. Сообщение об ошибке в данном случае и будет играть роль заголовка страницы.

3. Добавим также в текст сообщения рекомендации для пользователя по устранению ошибки и попросим сообщить об ошибке администратору сайта. (Это необходимо для выявления и устранения подобных ошибок в будущем.)
4. Предоставим пользователю контактную информацию.

Один из возможных вариантов реализации подобной страницы показан на рис. 4.4.

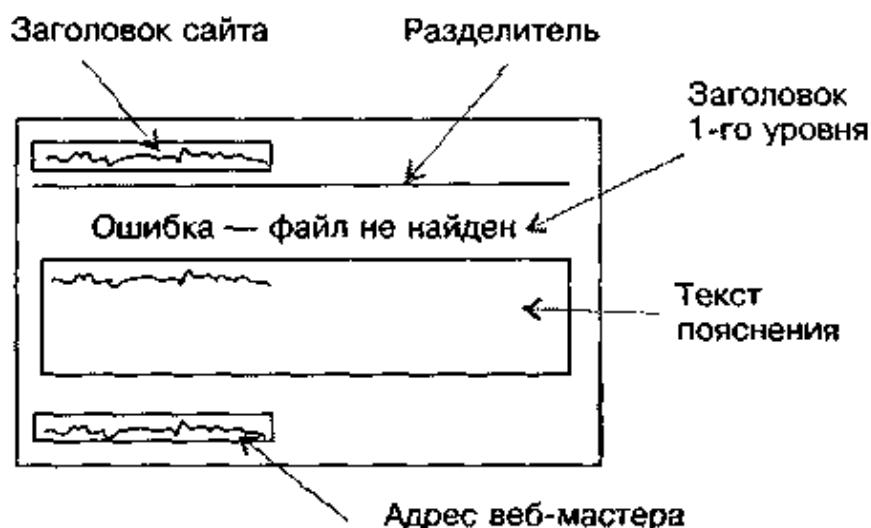


Рис. 4.4



## Коротко о главном

1. Стартовая страница — это «визитная карточка» вашего сайта. Она создает первое впечатление об основном содержании сайта и его информационной важности.
2. Все страницы сайта должны проектироваться с учетом эргономических параметров, определяющих удобство восприятия информации пользователем, а также с учетом технических характеристик пользовательских компьютеров.



## Вопросы и задания

1. В чем заключаются эргономические требования к оформлению веб-страниц?
2. Какие технические характеристики компьютеров и линий связи необходимо учитывать при проектировании веб-страниц?

3. Назовите типичные ошибки при разработке веб-страниц.
4. Укажите примерную последовательность действий при разработке внешнего вида веб-страницы.
5. Опишите влияние функций сайта на внешний вид его страниц.



### Практическое задание

Проработайте внешний вид основных страниц школьного сайта на основе функций, определенных в главе 3.



### Темы рефератов

1. Влияние цветовых оттенков на восприятие информации сайта.
2. Возможные способы расположения элементов веб-страниц (обзор существующих сайтов).

## Глава 5

# ПРОЕКТИРУЕМ КАРТУ САЙТА



Анализируя различные сайты, имеющиеся в сети Интернет, можно сделать следующий вывод: в большинстве случаев основным критерием оценки их качества является внешний вид (дизайн веб-страниц). Однако в подавляющем большинстве случаев сайт представляет собой не одну страницу или документ, а совокупность веб-страниц различного типа. Возможно также наличие видео- и аудиофрагментов, анимаций, файлов, доступных для скачивания, интерактивных компонентов и пр.

В Интернете можно найти сайты как с хорошо продуманной структурой, так и с полным отсутствием таковой. Многие сайты бесконтрольно и беспланово растут, еще больше усложняя взаимодействие пользователя с ними. Создавая же сайты с продуманной структурой, их разработчики заранее планируют дальнейший рост сайта, закладывая в существующую его структуру возможности для будущего развития.

В этой главе мы рассмотрим, как осуществляется планирование и создание структуры сайта и организация сервера для реализации подобной структуры.



На диске имеется плакат, демонстрирующий основные виды (архитектуру) карт сайта, описанные в данной главе. Плакат содержится в файлах poster5-1.vsd и poster5-1.cdr в папке chapter5.

## Карта сайта

Карта сайта нужна не только для организации эффективного представления информации в браузере. Она также позволит вам как разработчику сайта в дальнейшем намного легче производить изменения на нем.

Информацию на сайте можно условно разбить на несколько групп, где доступ к одной группе возможен через другую. Среди подобных групп выделим стартовую страницу, страницы разделов сайта, страницы данных и страницы ресурсов.

*Стартовая страница* (см. главу 4) содержит основные сведения о вашем сайте, а также ссылки на основные его разделы. Такой структуры стартовых страниц придерживаются практически все разработчики сайтов, если только вообще информацию, представляемую на сайте, можно разбить на разделы. Обычно со стартовой страницы можно обратиться непосредственно к конкретным ресурсам. Но ее основная цель — заинтересовать пользователя для перехода к разделам сайта, в связи с чем хорошо продуманное выделение таких разделов может напрямую отражаться на его посещаемости. В качестве ссылок на стартовой странице могут использоваться как текст, так и графические иллюстрации.

Основное назначение *страниц разделов* — это организация доступа к конкретным документам и ресурсам. На таких страницах также возможно присутствие ссылок на внешние (по отношению к вашему сайту) источники информации. При создании подобных ссылок следует учесть, что материалы, на которые вы ссылаетесь, могут оказаться недоступными (если данные повреждены или находятся в переработке, соответствующий сервер выключен и т. п.), ресурсы могут быть перемещены на другие адреса (для скачиваемых файлов часто возможно изменение имени файла при смене версии программы или драйвера), либо «целевой» сайт может быть перегружен графикой. В большинстве случаев между страницами разделов предусматриваются перекрестные ссылки, дающие возможность переходить от одного раздела сайта к другому без возврата на стартовую страницу.

Остальные страницы сайта обычно представляют собой информационные (*страницы данных*) и *страницы ресурсов*. В большинстве случаев ресурсами здесь являются каталоги файлов (программ) и медиа-каталоги (графика, видеофайлы, звуковые файлы и пр.).

## Виды карт сайтов

Существует четыре основных разновидности карты сайта.

1. **Линейная** (рис. 5.1). Представляет собой серию последовательно связанных документов, в каждом из которых возможен только переход к следующему документу или возврат к предыдущему. Встречается довольно редко, обычно на сайтах с небольшим количеством страниц (например, на презентационных сайтах).

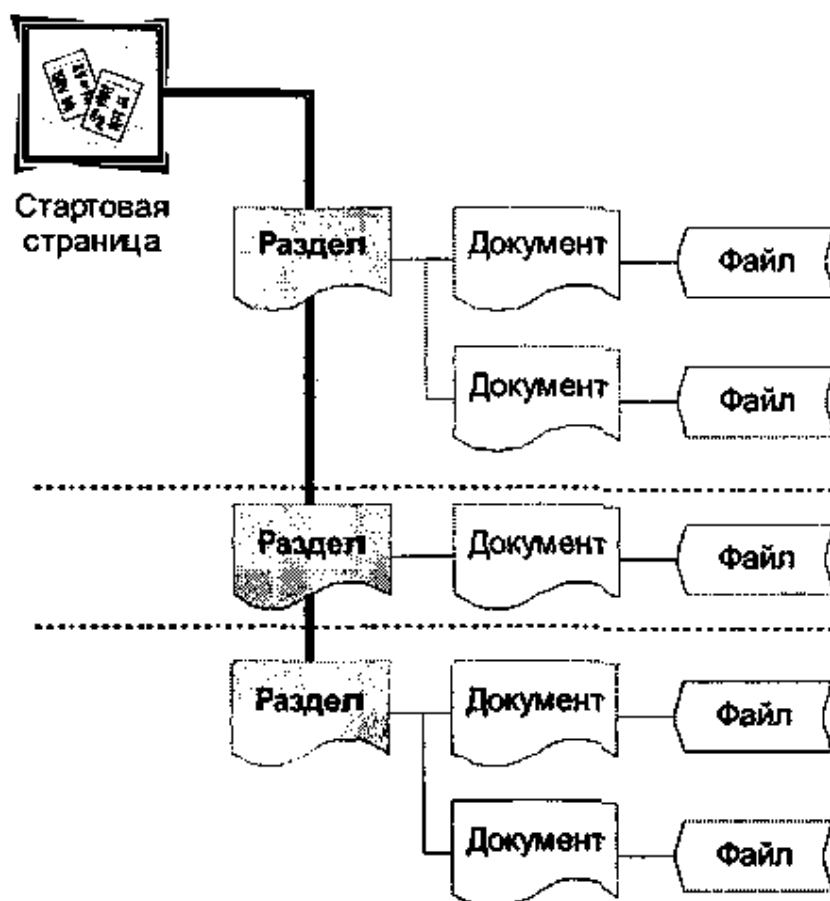


Рис. 5.1

2. Иерархическая (рис. 5.2). В большинстве своем повторяет файловую структуру соответствующего сервера. Создание такой структуры в чистом виде возможно для небольших сайтов, тогда как на крупных сайтах к ней добавляются также перекрестные ссылки между некоторыми документами.
3. Пирамидальная (рис. 5.3), где множество ресурсных файлов образуют подобие пирамиды. Она нашла отражение на множестве сайтов, которые, помимо информационной части, содержат также единый каталог для хранения файлов программ, доступных для скачивания (например, на сайтах производителей компьютерного оборудования).
4. Диаметральная (рис. 5.4).

Выбор конкретной структуры карты сайта определяется функциями сайта, объемом и содержанием размещаемых на нем материалов, планами дальнейшего развития сайта. Последнее позволит вам в будущем избавиться от проблем, связанных с реконструкцией сайта и его переводом на другую структуру, что, в свою очередь, даст возможность существенно снизить расходы денег, времени и сил на эти работы.

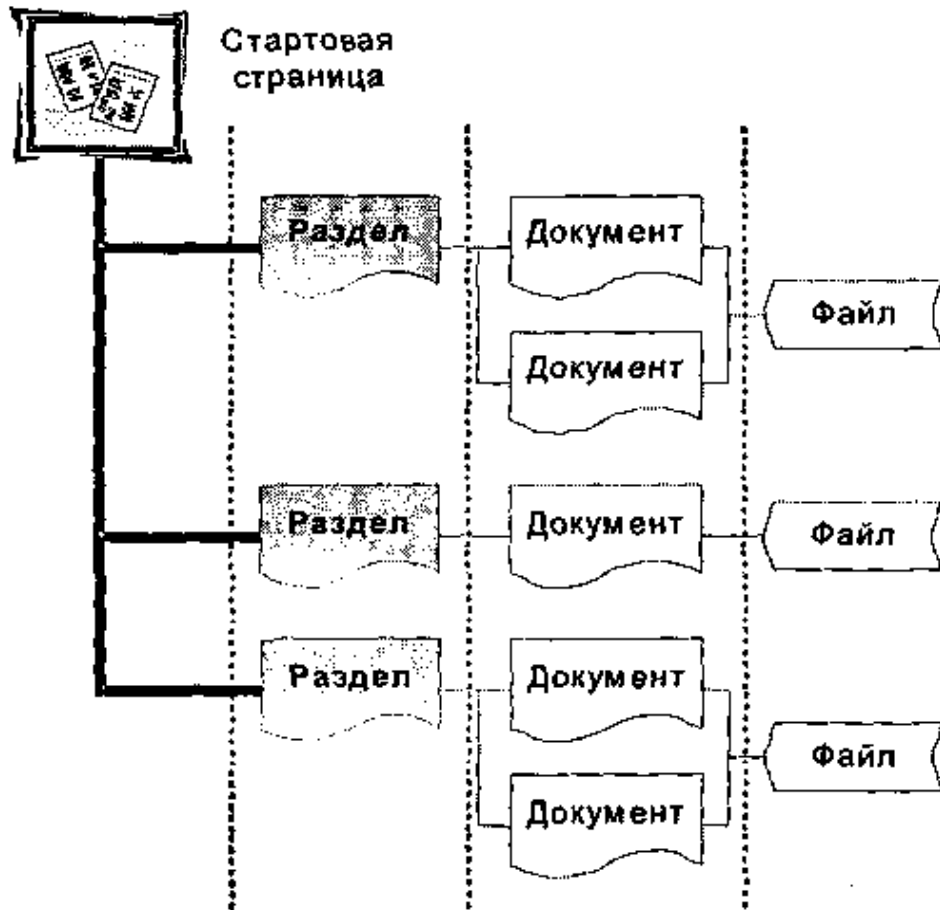


Рис. 5.2

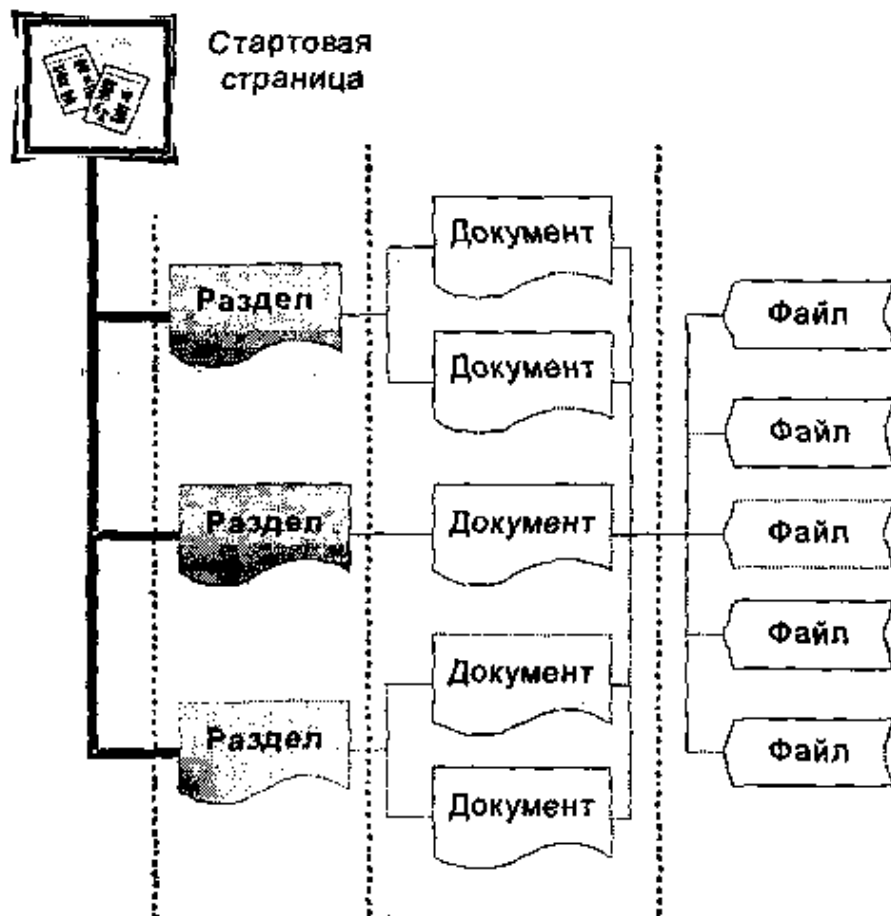


Рис. 5.3

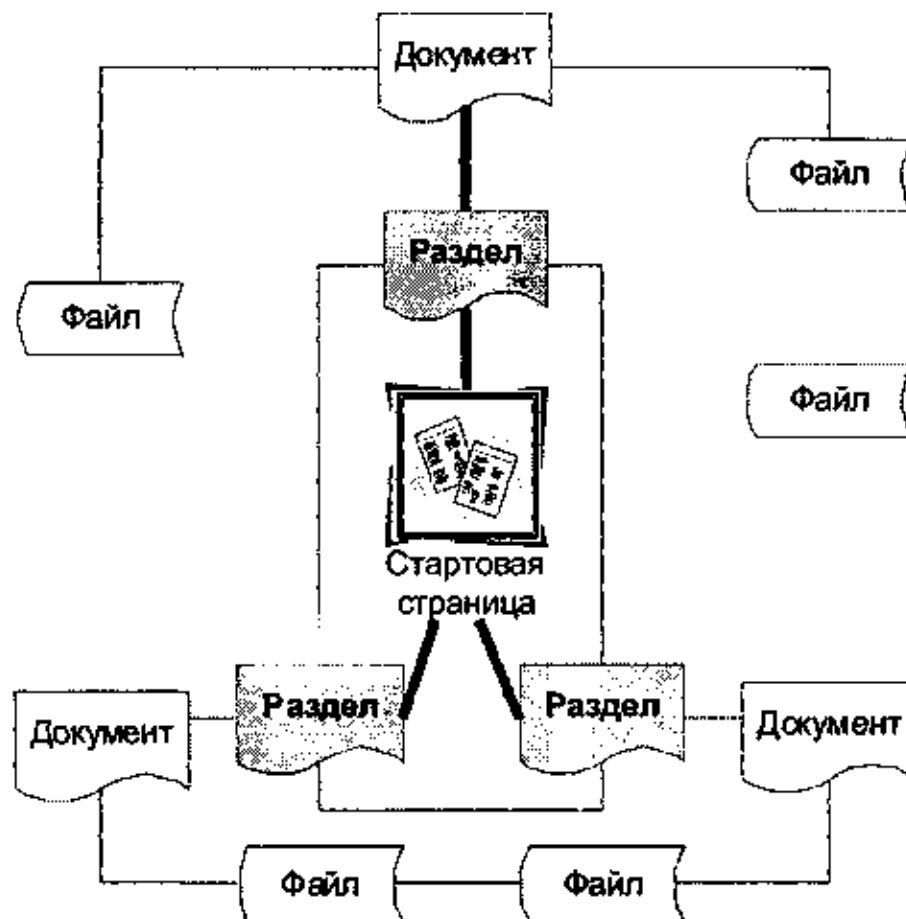


Рис. 5.4

## Файловая структура сайта

Проблема правильного размещения файлов сайта на диске сервера обычно возникает только при их большом количестве. Если сайт содержит лишь несколько страниц с минимумом графических файлов, то заботиться о правильной организации файлов не приходится — можно хранить все эти файлы в одном каталоге. Но проблема сразу становится насущной, если сайт начинает расти, — ведь рост количества файлов приводит к увеличению сложности всех работ с сайтом (чтение 50 имен файлов через ftp занимает меньше времени, чем чтение 500 имен!). Поэтому правильная организация файловой структуры сервера поможет снизить затраты при работе с сайтом, а также обеспечить его рост в дальнейшем.

Обычно файлы на сервере сортируют по двум основным категориям: по функциям или по их типу. При этом имена каталогов отражают назначение файлов: каталог веб-страниц, каталог графики, каталог скриптов (программ, выполняемых сервером или браузером), файлы для загрузки на клиентский компьютер и т. д.



Корневой каталог сайта при этом обычно содержит лишь несколько файлов, главным из которых является индексный, загружаемый по умолчанию. (Если просмотреть каталог установки HTTP-сервера Apache, то и там можно найти аналогию с приведенным выше разделением файлов: каталог `htdocs` предназначен для хранения HTML-документов, каталог `cgi-bin` — для скриптов и т. д.)

В зависимости от типа сайта и его функций может получиться довольно большое количество таких каталогов, однако это намного лучше (с точки зрения управления сайтом), чем если бы все файлы располагались в одном каталоге. Создание же вложенных папок позволит еще больше упростить понимание принадлежности файлов. Например, если графика используется только в HTML-документах, то целесообразно создать каталог для графики вложенным в каталог для HTML-документов. Графику, используемую в конкретном веб-документе (рисунки, графики и т. п.), целесообразно расположить в одном каталоге, отражающем имя документа. Если же какие-то графические элементы используются несколькими документами, то рекомендуется размещать их в отдельном каталоге, чтобы они были легко доступны из всех этих документов. Например, графические элементы, определяющие общий дизайн веб-страниц, лучше разместить в каталоге в корне сайта, что облегчит доступ к ним с любой веб-страницы.

Желательно также предварительно определить формальные правила, по которым будут назначаться имена файлов. Это позволит уменьшить время, которое тратится на поиск файла (например, при обновлении дизайна) и избежать возможных проблем с одинаковыми именами у нескольких файлов. Во многих случаях имена файлов различаются только числами (порядковыми номерами), с постоянной текстовой частью (`graph1.gif`, `graph2.gif` и т. д.). Обычно этот принцип применяется для наименования большого количества иллюстраций, относящихся к одному и тому же документу. Кроме того, рекомендуется во всех файлах, которые относятся к одному документу, использовать имя этого документа. Например, если файл `graph1.gif` относится к документу, файл которого имеет имя `func.html`, то рекомендуется назвать файл графики `func_graph1.gif`. Часто в имени файла используются «префиксные» и «постфиксные» текстовые фрагменты, например имена всех файлов с изображениями кнопок могут иметь типовой вид `button_????.gif`, где `????` — название конкретной кнопки.

Следует также не забывать о специфике представления имен файлов в различных операционных системах. Некоторые из этих ОС являются регистрозависимыми, другие — нет. Например, Unix-подобные операционные системы регистрозависимы и в них имена файлов `file.txt` и `FILE.txt` являются различными. Операционные же системы Windows регистронезависимы. Это следует учесть и при создании файлов, и при определении ссылок на них. Не рекомендуется также использовать в имени файла знаки и буквы национальных раскладок клавиатуры (в данном случае — русские буквы), так как подобные знаки преобразуются в коды, которые могут быть неправильно преобразованы или восприняты HTTP-сервером (он работает с английской раскладкой). Это объясняется тем, что для кодирования национальных раскладок используются различные кодовые таблицы, так что одному и тому же коду в различных кодовых таблицах могут соответствовать различные символы. (Для кодирования русского алфавита используются кодировки `windows-1251` и `koi-8r`. При этом кодовая таблица `windows-1251` используется в операционных системах Windows, а кодировка `koi-8r` — в Unix-подобных системах.)



### Коротко о главном

1. Карта сайта определяет структуру и взаимосвязь его элементов. Она может иметь линейную, иерархическую, пирамидальную или диаметральною архитектуру.
2. Правильно организованная файловая структура сайта — основа для эффективного дальнейшего его развития.
3. Разрабатывая файловую структуру, следует учитывать особенности используемой операционной системы и функции компонентов сайта.



### Вопросы и задания

1. Что такое карта сайта?
2. Назовите и прокомментируйте основные виды структуры карты сайта.
3. Перечислите основные рекомендации при назначении имен файлов и каталогов сайта.
4. Какие особенности операционных систем следует учитывать при формировании файловой структуры сайта?



## Практическое задание

Разработайте карту школьного сайта (в соответствии с одной из описанных ее архитектур) и макет организации файловой структуры сайта.



## Темы рефератов

1. Организация файловой структуры для операционной системы Windows.
2. Организация файловой структуры для Unix-подобных операционных систем.
3. Кодовые таблицы.



## Демонстрационный плакат

Виды карт сайтов.

## Глава 6

# СОЗДАЕМ СВОЮ ПЕРВУЮ ВЕБ-СТРАНИЦУ



Создание веб-страниц возможно с использованием автоматизированных средств или в режиме ручной теговой разметки (используя стандартные средства редактирования документов).

Применение автоматизированных средств позволяет создавать веб-страницы в минимальные сроки и без знания языка разметки HTML. Основной принцип работы таких систем — WYSIWYG («What You See Is What You Get» — «что видим, то и получаем»). Однако профессиональные разработчики сайтов редко используют автоматизированные средства непосредственно для создания веб-страниц, реализуя с их помощью лишь функции управления.

Режим теговой разметки требует более глубоких знаний — как минимум, знания специального языка HTML и принципов представления информации в различных браузерах.

На практике же при создании своего сайта рекомендуется выбирать наиболее подходящий способ в зависимости от специфики задачи.

В этой главе описывается процесс создания (с помощью автоматизированных средств и в режиме теговой разметки) веб-страницы «Сообщение об ошибке», эскиз которой был разработан в главе 4, а также приведены упражнения, которые рекомендуется выполнить по образцу, и варианты групповых и индивидуальных заданий-проектов.



На диске имеются плакаты, демонстрирующие процесс создания веб-страницы. Плакаты содержатся в файлах poster6-1.vsd, poster6-2.vsd, poster6\_3.vsd, poster6-1.cdr, poster6-2.cdr и poster6\_3.cdr в папке chapter6.

## Автоматизированный способ создания веб-страниц

Самым простым способом создания веб-страницы является использование редактора Word из пакета Microsoft Office версии 97/2000/XP/2003<sup>1</sup>. Помимо прочих основных функций, в этом текстовом редакторе существует возможность сохранения документов как веб-страниц.

Среди специализированных же инструментальных средств разработки веб-страниц и сайтов можно назвать Macromedia Dreamweaver, Microsoft FrontPage и др.

Запустим редактор Microsoft Word и создадим страницу, соответствующую ранее разработанному эскизу (рис. 6.1).

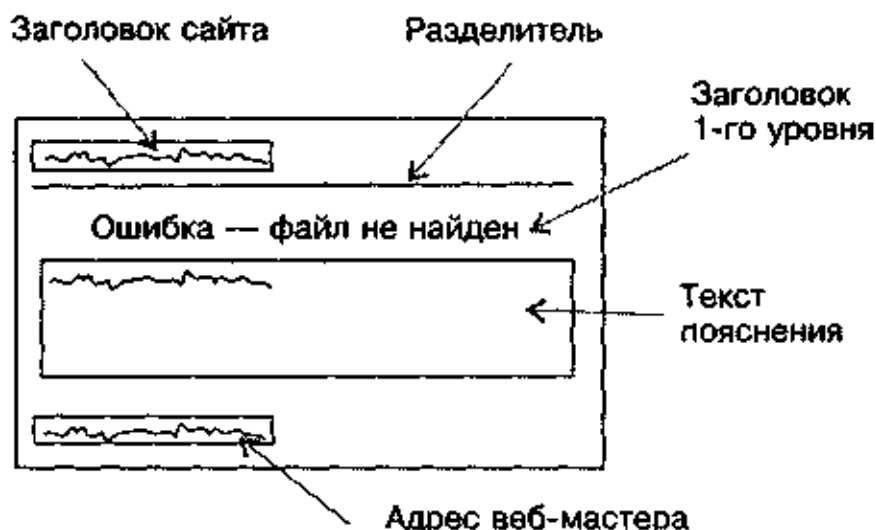


Рис. 6.1

Напомним, что текст пояснения должен быть написан с таким расчетом, чтобы дать потенциальному пользователю необходимый минимум информации об ошибке, а в некоторых случаях — и возможность сообщить об ошибке разработчику сайта («веб-мастеру»). Например, текст пояснения может быть следующим:

Извините, но запрошенная вами страница не найдена. Проверьте: возможно, Вы указали неправильный адрес страницы.

Если эта ошибка появилась повторно, просьба сообщить об этом группе поддержки сайта по адресу: [webmaster@servername.ru](mailto:webmaster@servername.ru). В письме укажите, пожалуйста, адрес страницы, на которой возникла ошибка.

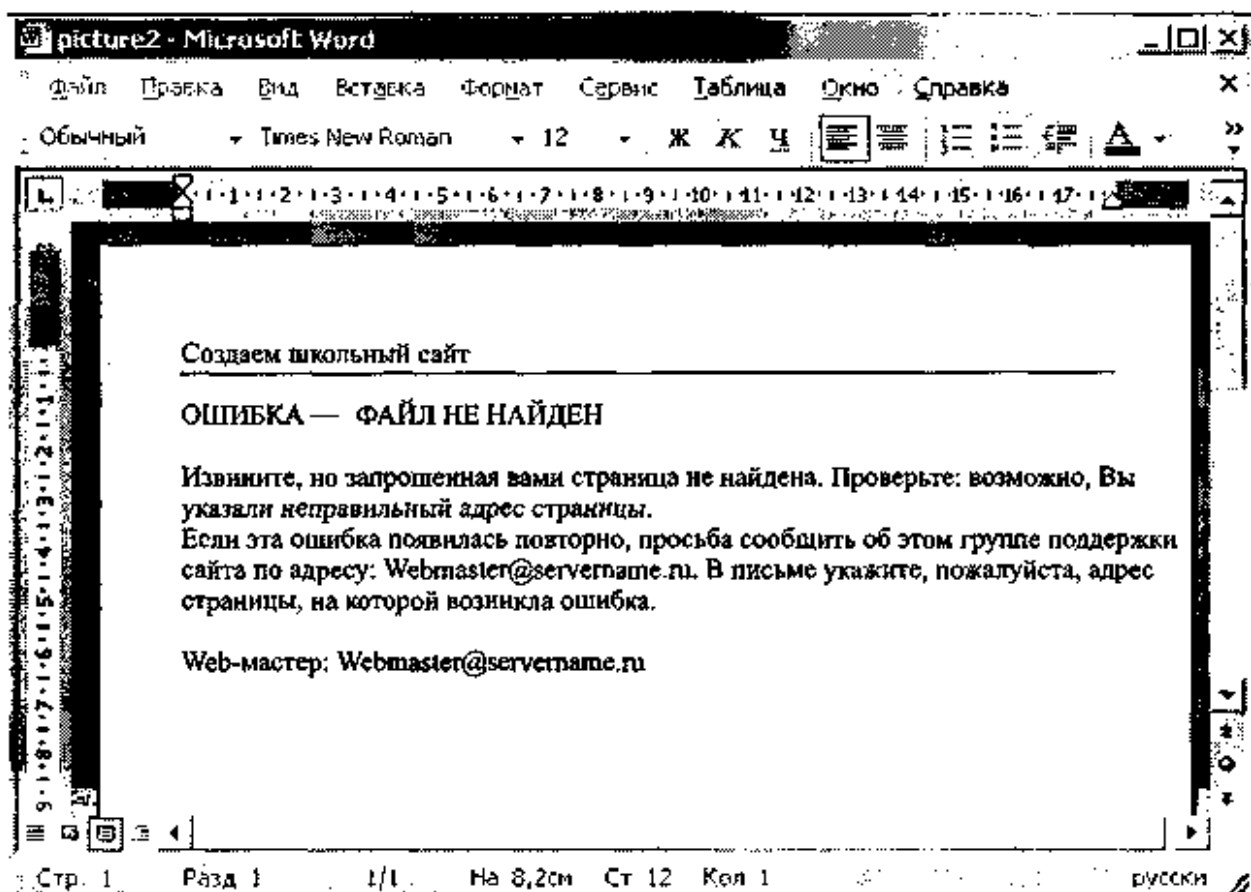
<sup>1</sup> Недокументированная возможность сохранения документа в формате HTML имеется в версии Word 7.0 (95). Для этого достаточно выбрать в списке типов для сохраняемого файла пункт Microsoft Word 2.0. — *Прим. ред.*



**Упражнение 1.** Создайте в редакторе Microsoft Word текстовый документ, соответствующий рис. 6.2, и сохраните его под именем picture2.doc.



Образец этого документа имеется на прилагаемом диске в папке chapter6 под именем picture2.doc.



**Рис. 6.2**



**Упражнение 2.** Выполните форматирование текста в соответствии с эскизом страницы (см. рис. 6.1). Для этого можно воспользоваться стандартными средствами форматирования, предусмотренными в Word. В процессе форматирования не забудьте залить фон страницы нужным цветом (меню **Формат**, **Фон**). В результате документ должен иметь вид, показанный на рис. 6.3. Сохраните этот документ под именем picture3.doc.



Образец этого документа имеется на прилагаемом диске в папке chapter6 под именем picture3.doc.

Чтобы преобразовать созданный документ в веб-страницу, его необходимо сохранить в соответствующем формате. При этом, так как многие пользователи при перемещении по открытым страницам ориентируются по их заголовкам, обращайте внимание на правильное определение этих заголовков. В нашем случае необходимо вместо заголовка страницы, предлагаемого программой Word, ввести текст, отражающий смысл данной страницы.

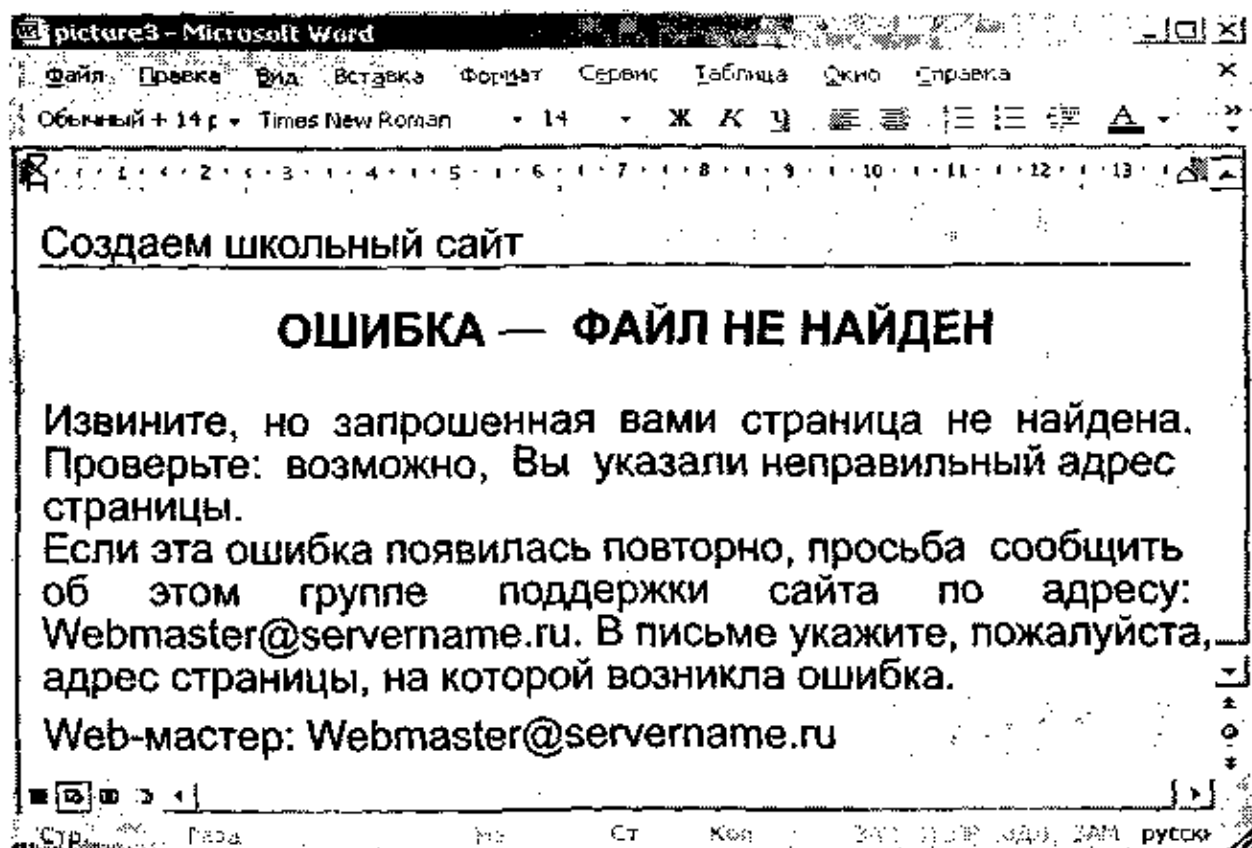


Рис. 6.3


 **Упражнение 3.** Сохраните документ picture3.doc как веб-страницу. Для этого в редакторе Word выберите в меню **Файл** пункт **Сохранить как веб-страницу**. Измените предлагаемый заголовок: в диалоговом окне сохранения нажмите кнопку **Изменить** в строке **Заголовок** и введите текст: «Ошибка — файл не найден». Укажите для файла страницы имя 404.htm (по коду соответствующей стандартной ошибки) и нажмите кнопку **Сохранить** (рис. 6.4).



Рис. 6.4



Образец этого документа имеется на прилагаемом диске в папке chapter6 под именем 404.htm.

Страница готова. Теперь ее можно загрузить в браузер для проверки работоспособности и корректности. В браузере Internet Explorer результат должен быть аналогичен показанному на рис. 6.5.

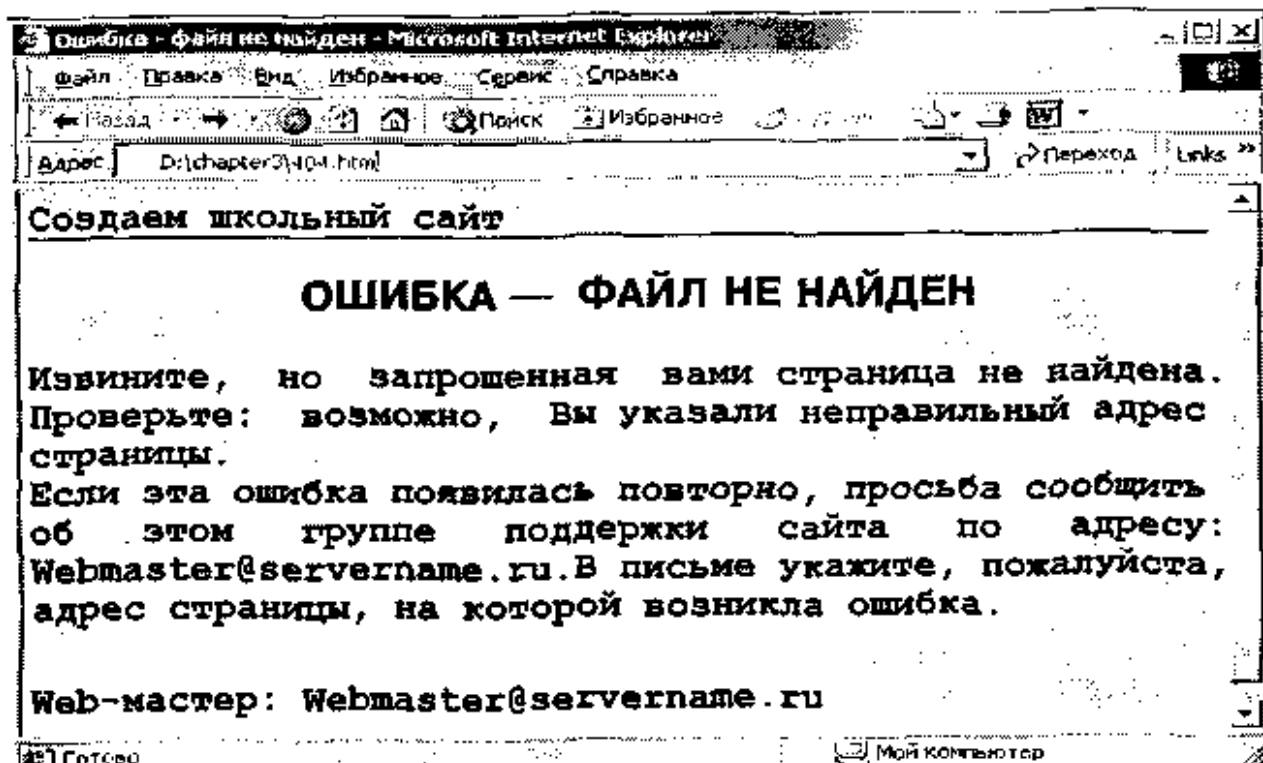


Рис. 6.5

## Режим теговой разметки

Создание веб-страницы без использования автоматизированных средств требует предварительных знаний о так называемом языке гипертекстовой разметки — HTML (Hypertext Markup Language), широко применяемом для представления информации в сети Интернет.

**Структура HTML-документа.** Процесс создания веб-страницы заключается во встраивании в ее текст специальных команд, называемых *тегами* (tag), которые задают правила вывода на экран выбранного текста. Для ввода тегов можно использовать любой текстовый редактор (например, Notepad, входящий в комплект стандартных программ Microsoft Windows). Все теги обрамляются угловыми скобками, внутри которых записывается та или иная команда (например, <br> — тег перевода строки).



Во многих случаях требуется использовать *парные открывающие и закрывающие теги (контейнеры)*. Закрывающий тег отличается от открывающего наличием знака «/» после первой угловой скобки. Например, открывающий тег абзаца — `<p>`, а закрывающий — `</p>`. Закрывающий тег показывает, где нужно завершить выполнение команды. Пара из открывающего и закрывающего тега образует блок. Внутри блока могут использоваться другие теги и блоки, а также сам этот блок может быть частью другого блока.

Любая веб-страница содержит следующие блоки:

- 1) блок HTML — определяет, что текст внутри него является веб-страницей. Открывающий тег `<html>` находится в первых строках любой веб-страницы, а закрывающий `</html>` — в самом конце страницы. Блок HTML содержит в себе два других основных блока: HEAD и BODY;
- 2) блок HEAD — содержит команды, касающиеся всей страницы, дополнительную и служебную информацию (указание используемой кодировки кириллицы, сведения об авторе, информацию для поисковых серверов, заголовки страницы и т. п.) и записывается до блока BODY. Информация из блока HEAD не отображается на странице. Открывающий тег — `<head>`, закрывающий — `</head>`;
- 3) блок BODY — содержит текст и команды, которые непосредственно отображаются на экране. Все, что нужно показать пользователю, находится в данном блоке. Открывающий тег — `<body>`, закрывающий — `</body>`.

Простейший HTML-документ можно создать, записав в текстовом редакторе следующие строки:

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

Так как все форматирование в языке HTML осуществляется командами, то весь текст HTML-документа может быть представлен как одна строка. Поэтому следующая запись тоже является корректной:

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

Однако для обеспечения лучшей читаемости HTML-текста веб-страниц рекомендуется при его написании использовать те же принципы размещения блоков, что и при написании листингов на структурных языках программирования.



**Упражнение 4.** Создайте в редакторе Notepad текстовый документ, соответствующий рис. 6.6. Сохраните его под именем blank2.htm. Так как весь имеющийся текст должен быть виден пользователю, его необходимо поместить в блок BODY.



Образец документа имеется на прилагаемом диске в папке chapter6 под именем blank2.htm.

```

blank - Notepad
File Edit Format Help
<html>
<head>
</head>
<body>
Создаем школьный сайт
ОШИБКА — ФАЙЛ НЕ НАЙДЕН
Извините, но запрошенная вами страница не найдена. Проверьте: возможно, Вы указали
неправильный адрес страницы. Если эта ошибка появилась повторно, просьба сообщить
об этом группе поддержки сайта по адресу:
Webmaster@servename.ru. В письме укажите, пожалуйста, адрес страницы, на которой
возникла ошибка.
Web-мастер: Webmaster@servename.ru
</body>
</html>
  
```

Рис. 6.6

На рис. 6.7 показано, как эта страница будет отображена в веб-браузере.

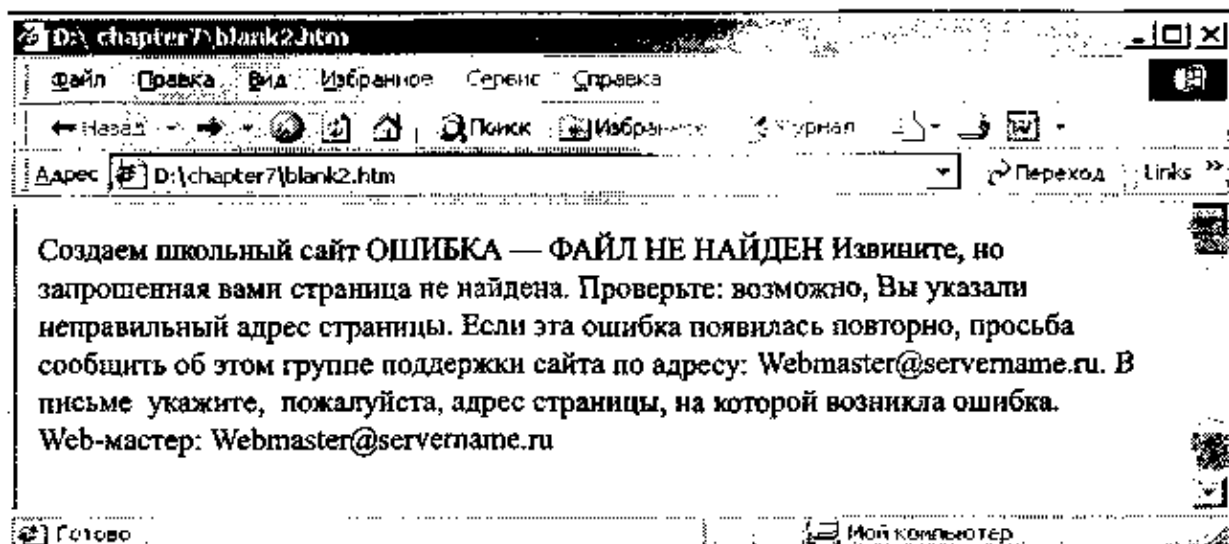


Рис. 6.7

**Форматирование текста.** Как видно на рис. 6.7, браузер воспроизводит весь текст, содержащийся в блоке BODY, сплошным потоком, игнорируя имеющиеся в HTML-тексте переходы на новую строку. Чтобы придать тексту требуемый вид, воспользуемся тегами:

`<p>` — тег абзаца; используется совместно с закрывающим тегом `</p>`. Каждый абзац начинается с новой строки. Между абзацами браузер оставляет небольшой промежуток (межабзацный отступ);

`<b>` — полужирный текст; используется совместно с закрывающим тегом `</b>`. Любой текст, находящийся внутри этого блока, выводится полужирным;

`<hr>` — разделительная горизонтальная линия;

`<br>` — перевод строки;

`<h1>` — заголовок 1-го уровня; используется совместно с закрывающим тегом `</h1>`.

Чтобы выполнить форматирование имеющегося HTML-документа (файл `blank2.htm`), разместим теги в нужных местах текста:

```
<html>
<head>
  </head>
  <body>
    <p><b>Создаем школьный сайт</b></p><hr>
    <h1>ОШИБКА — ФАЙЛ НЕ НАЙДЕН</h1>
    <p>Извините, но запрошенная вами страница не найдена.
    Проверьте: возможно, Вы указали неправильный адрес
    страницы. <br> Если эта ошибка появилась повторно, просьба
    сообщить об этом группе поддержки сайта по адресу:
    webmaster@servername.ru. В письме укажите, пожалуйста,
    адрес страницы, на которой возникла ошибка. </p>
    <p>Web-мастер: webmaster@servername.ru</p>
  </body>
</html>
```



**Упражнение 5.** Выполните форматирование текста (файл `blank2.htm`). Сохраните файл под именем `404-self.htm`. Вид страницы в веб-браузере должен соответствовать рис. 6.8.



Образец документа имеется на прилагаемом диске в папке chapter6 под именем 404-self.htm.

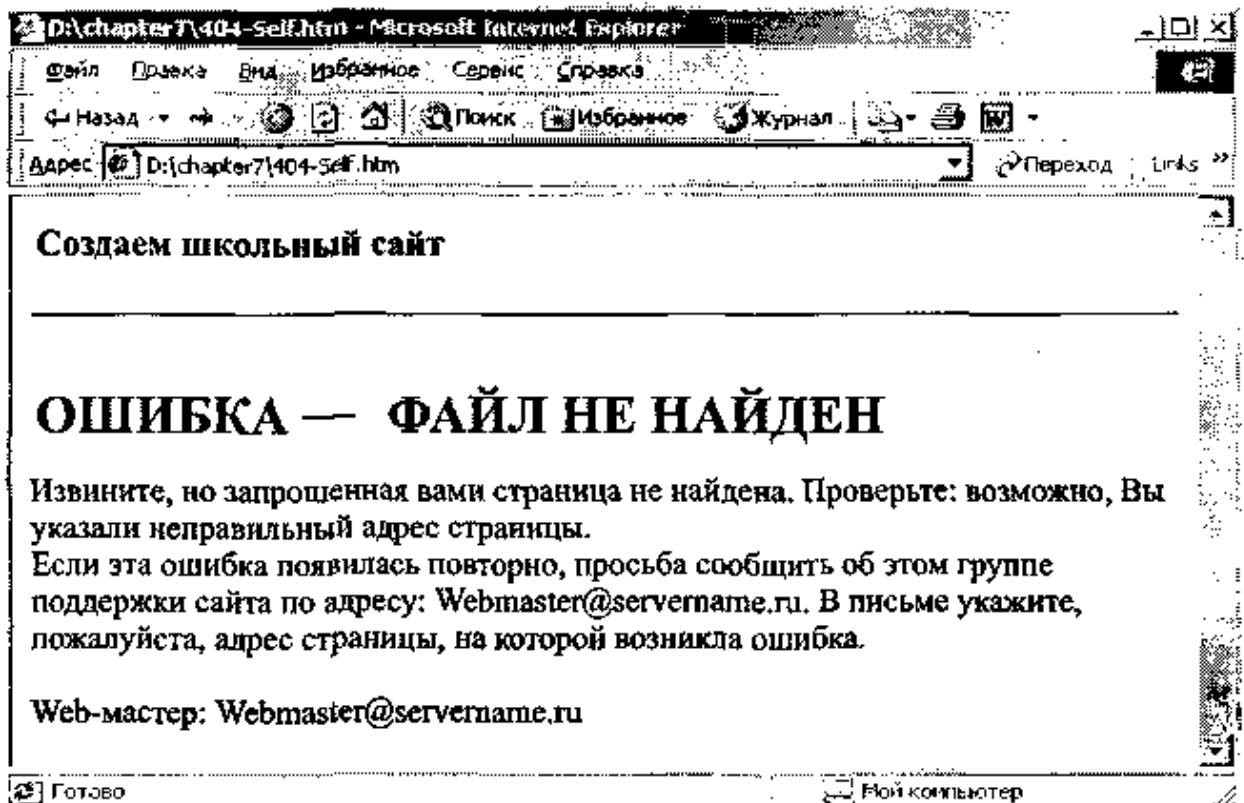


Рис. 6.8

**Параметры тегов.** Теги могут содержать в себе дополнительные инструкции (параметры), определяющие правила их использования. Например, параметр `align` задает выравнивание по левому или правому краю страницы, по центру или по ширине (значения `left`, `right`, `center`, `justify`, соответственно). По умолчанию используется выравнивание по левому краю. Эти инструкции аналогичны переменным и записываются внутри тега через пробел. Например, код `<p align=right>Текст</p>` означает выравнивание абзаца по правому краю.

Чтобы определить цвет фона страницы, нужно в составе тега `<body>` записать инструкцию `bgcolor`. Цвет в большинстве случаев задается на основе трех основных цветов: красного, зеленого и синего (цветовая модель RGB), где яркость каждой из этих цветовых составляющих определяется числами в диапазоне от 0 до 255 в шестнадцатеричной системе счисления. Перед числовой записью цвета всегда ставится знак «#». Например, зеленый цвет задается числом `#00FF00`, определяющим максимальную интенсивность зеленой и отсутствие других составляющих.

Добавим инструкции в теги, имеющиеся в HTML-коде нашей веб-страницы, а тег `<hr>` перенесем внутрь блока абзаца для уменьшения отступа. В результате должен получиться следующий текст:

```
<html>
  <head>
  </head>
  <body bgcolor="#FFCC99">
    <p align=left><b>Создаем школьный сайт</b><hr></p>
    <h1 align=center>ОШИБКА — ФАЙЛ НЕ НАЙДЕН</h1>
    <p align=justify>Извините, но запрошенная вами страница
    не найдена. Проверьте: возможно, Вы указали неправильный
    адрес страницы. <br> Если эта ошибка появилась повторно,
    просьба сообщить об этом группе поддержки сайта по адресу:
    webmaster@servername.ru. В письме укажите, пожалуйста,
    адрес страницы, на которой возникла ошибка.</p>
    <p align=left>Web-мастер: webmaster@servername.ru</p>
  </body>
</html>
```



**Упражнение 6.** Выполните форматирование текста (файл 404-self.htm). Сохраните файл под именем 404-self-final.htm. Вид страницы в веб-браузере должен соответствовать рис. 6.9.



Образец документа имеется на прилагаемом диске в папке chapter6 под именем 404-self-final.htm.

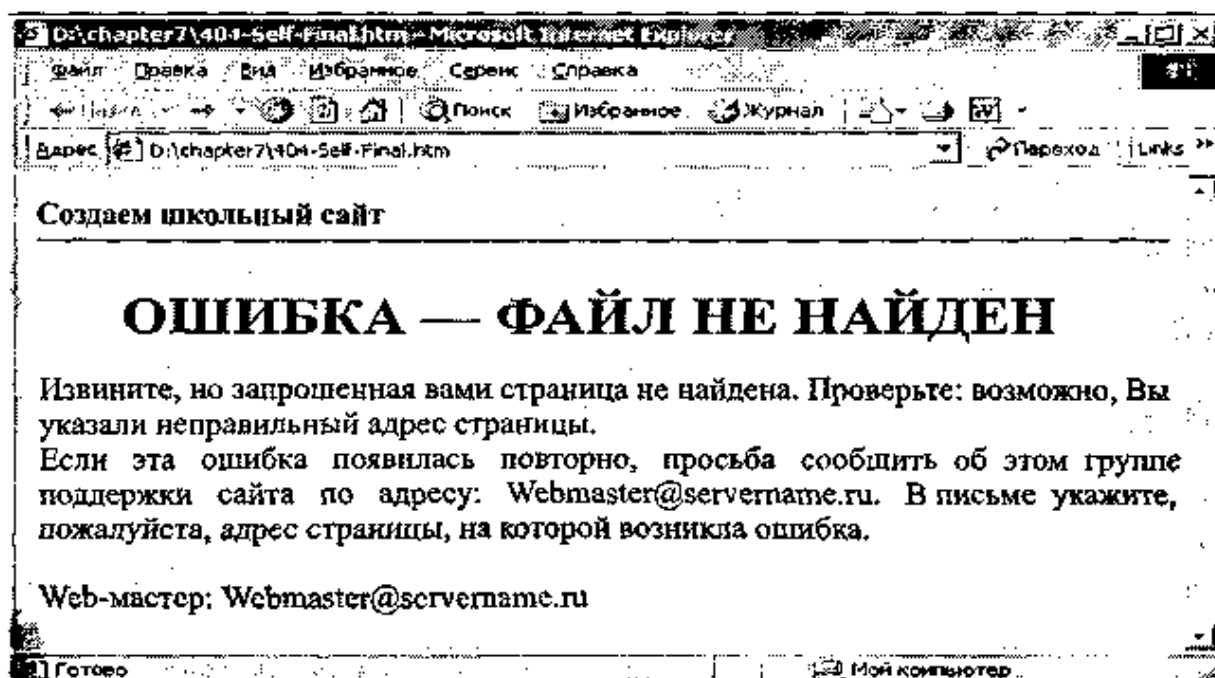


Рис. 6.9

Если теперь сравнить файлы 404.htm и 404-self-final.htm, то можно увидеть существенное различие в их объеме (5636 и 546 байтов, соответственно). Это различие определяется тем, что при создании страницы в режиме теговой разметки мы добавляем в текст только необходимые теги, тогда как при генерации HTML-кода автоматизированными средствами создания веб-страниц в них добавляется много лишних тегов и инструкций.

Для разработчика это различие очень важно: уменьшив размер файла страницы, можно сократить (и во многих случаях весьма существенно) время ее загрузки. Это создаст благоприятное впечатление у посетителей вашего сайта.



### Коротко о главном

1. Создать веб-страницу можно двумя способами: автоматизированным и в режиме ручной теговой разметки.
2. Автоматизированный способ основан на применении специализированных инструментальных средств, а также стандартных текстовых редакторов с встроенными средствами форматирования и сохранения веб-документов.
3. В основе режима теговой разметки лежит использование языка гипертекстовой разметки. Для представления информации в сети Интернет в большинстве случаев применяется язык HTML.
4. Вид информации на веб-странице в HTML определяется тегами — встраиваемыми в текст специальными командами.
5. Теги содержат в себе инструкции (параметры), определяющие правила их использования.



### Вопросы и задания для самоконтроля

1. Назовите основные различия между автоматизированным способом и режимом ручной теговой разметки при создании веб-страниц.
2. Какие инструментальные средства можно использовать для автоматизированного создания страниц?
3. Какие средства можно использовать для создания страниц в режиме теговой разметки?

4. Что такое тег?
5. Назовите основные блоки HTML-документа.
6. Что такое инструкция тега?
7. Опишите назначение тегов `<p>`, `<b>`, `<hr>`, `<h1>`.



### Практическое задание

Создайте веб-страницу, на которой представлены краткие сведения о каком-либо человеке (автобиография, биография писателя и т. п.). В работе используйте редактор Microsoft Word (минимальный уровень) или язык HTML (повышенный уровень).



### Темы рефератов

1. История языков гипертекстовой разметки.
2. Среда разработки гипертекстовых документов Microsoft FrontPage.
3. Среда разработки гипертекстовых документов Macromedia Dreamweaver.

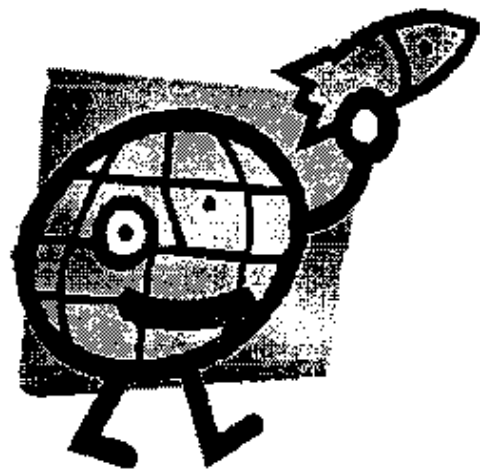


### Демонстрационные плакаты

1. Создание веб-страницы автоматизированным способом.
2. Создание веб-страницы в режиме теговой разметки.
3. Основные конструкции HTML.

## Глава 7

# ПОДКЛЮЧАЕМ ГРАФИЧЕСКИЕ ОБЪЕКТЫ



Компьютерная графика обладает огромным потенциалом, ведь один рисунок может дать больше информации для пользователя, чем целая страница текста.

Для представления графических элементов в Интернете используется три основных формата растровой графики: GIF (Graphics Interchange Format), JPEG (Joint Photographic Experts Group) и PNG (Portable Network Graphics).

Поддержка форматов GIF и JPEG осуществляется всеми графическими браузерами, формат же PNG имеет несколько меньшее распространение. Другие форматы кодирования изображений могут поддерживаться некоторыми браузерами (например, многие версии Internet Explorer допускают работу с форматом BMP), но в других браузерах такие картинки могут не отображаться.

Форматы GIF, JPEG и PNG имеют некоторые особенности, которые в значительной степени определяют их использование в тех или иных случаях.

- **Формат GIF** поддерживает не более 256 цветов, чего в большинстве случаев бывает достаточно для кодирования небольших изображений с приемлемым качеством. Часто количество цветов уменьшают сознательно для снижения общего объема файла (зачем хранить информацию о 256 цветах, когда в рисунке используется максимум 10 цветов?). Формат GIF является поточным форматом, позволяющим воспроизводить изображение уже полученной части рисунка, пока загружается его оставшаяся часть. Немаловажной причиной частого использования этого формата является и наличие в палитре атрибута «прозрачности», который позволяет сделать определенный цвет (или несколько цветов) прозрачным при наложении изображения на уже существующую подложку. Интересной особенностью формата GIF является прогрессивная развертка, позволяющая начать вывод изображения сразу



после начала загрузки, постепенно улучшая его качество в процессе загрузки. Другим частым применением формата GIF является анимированная графика, определяемая возможностью присутствия в графическом файле соответствующих управляющих команд и хранения нескольких изображений (кадров анимации). Поэтому если необходимо разместить на странице небольшой по объему рисунок (например, логотип) со средним качеством или с элементами анимации, то использование формата GIF наиболее приемлемо.

- **Формат JPEG** является в некотором смысле противоположностью формата GIF. В отличие от него, основное назначение формата JPEG заключается в кодировании полноцветных изображений (до 16,7 млн. оттенков). Использование сжатия с потерей качества позволяет добиться серьезного уменьшения объема графического файла с приемлемым уровнем потери качества изображения. Для хранения фотоизображений это наиболее подходящий формат. Но следует не забывать, что при сохранении изображения в формате JPEG происходит потеря качества, поэтому в процессе обработки изображения рекомендуется использовать другой формат, не использующий сжатие либо использующий сжатие без потерь, и только по завершении работы сохранить изображение в формате JPEG.
- **Формат PNG** обладает многими возможностями форматов GIF и JPEG; он как бы впитал в себя их лучшие стороны, однако из-за большого распространения форматов GIF и JPEG он применяется сравнительно редко.

При создании графики, ориентированной на использование в веб-страницах, важными являются следующие параметры.

- **Размер изображения.** Пользовательские дисплеи отличаются размерами по диагонали и используемым разрешением. Не рекомендуется использовать изображения, превышающие размеры окна браузера, развернутого на полный экран при типовом разрешении 800×600 точек.
- **Разрешение изображения (dpi, «dot per inch»** — количество точек на дюйм). Большое разрешение позволяет передать изображение с высочайшим качеством, однако это приводит к увеличению размеров файла. Кроме того, круг пользователей, которые смогут оценить это качество по достоинству, довольно мал. Стандартным считается разрешение 72 точки на дюйм.

- **Глубина цвета.** Использование большого количества цветов в изображениях в большинстве случаев не оправдывает себя (если только это не изображения фотографического качества). Большинство графических элементов может быть представлено с глубиной цвета в 8 битов (256 цветов).



На компакт-диске, прилагаемом к учебному пособию, имеется плакат, демонстрирующий процесс добавления изображения на веб-страницу. Плакат содержится в файлах poster7-1.vsd и poster7-1.cdr в папке chapter7.

## Использование редактора MS Word

Возьмем в качестве исходного материала веб-страницу «Сообщение об ошибке», кодирование которой на языке HTML изучалось в главе 6. Сделаем заголовок сайта в виде графического объекта (рис. 7.1).

Заголовок сайта

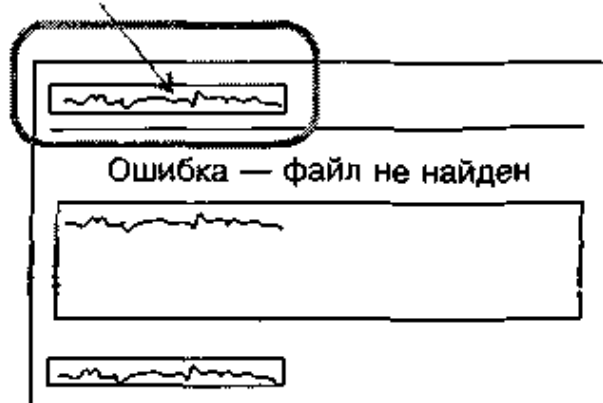


Рис. 7.1



**Упражнение 1.** Создайте рисунок (логотип сайта) в любом доступном графическом редакторе и сохраните его под именем logo.bmp. Преобразуйте рисунок в формат GIF, воспользовавшись стандартным графическим редактором MS Paint; для этого нужно выбрать в меню **Файл** пункт **Сохранить как** и тип файла GIF. Сохраните рисунок под именем logo.gif.



Образцы рисунка имеются на прилагаемом диске в папке chapter7 под именами logo\_1.gif, logo\_2.gif и logo\_3.gif.



**Упражнение 2.** Откройте в редакторе MS Word созданный ранее HTML-документ (образец имеется на прилагаемом диске в папке chapter6 под именем 404-self-final.htm).

Вставьте рисунок в текст документа в соответствии с эскизом страницы (рис. 7.1), воспользовавшись пунктом **Рисунок** в меню **Вставка**. Выберите далее пункт вложенного меню **Из файла** и в окне **Добавление рисунка** (рис. 7.2) выберите файл logo.gif. Сохраните документ под именем picture3.htm. (Возможный вид результата при использовании графического файла logo\_3.gif показан на рис. 7.3.)



Образец документа имеется на прилагаемом диске в папке chapter7 в файле picture3.htm.

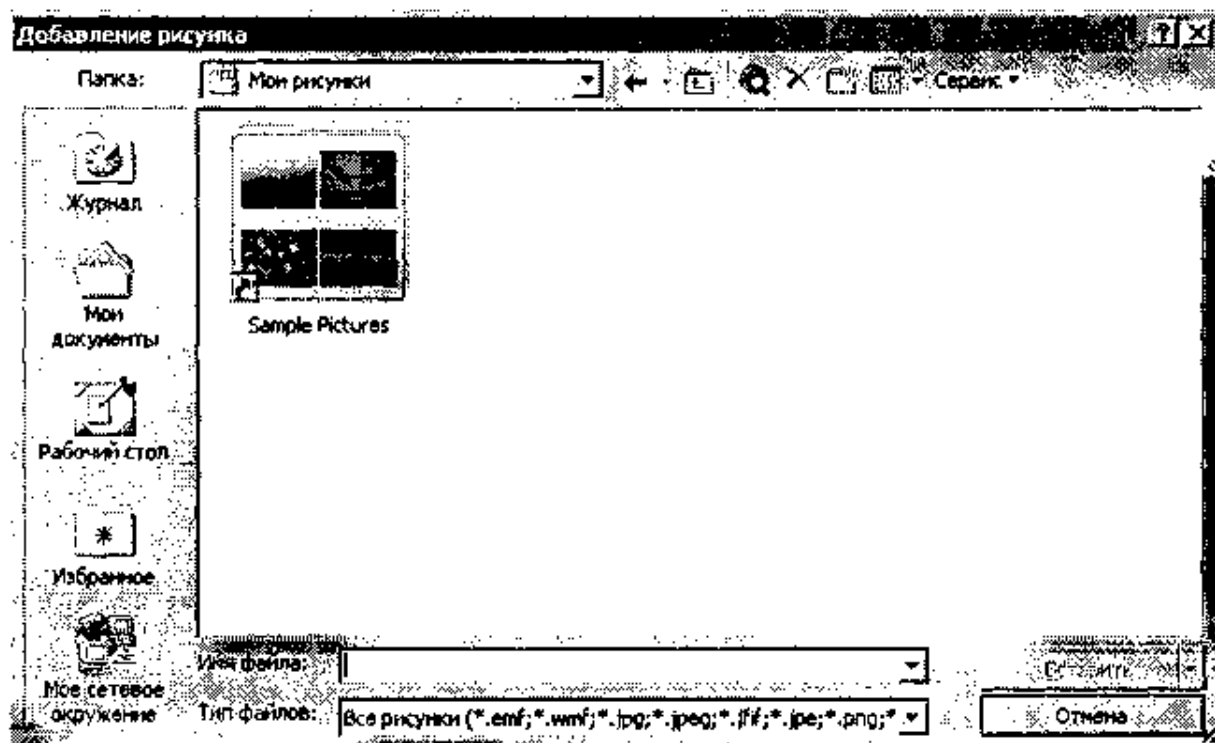


Рис. 7.2.

Если теперь просмотреть каталог, куда был сохранен файл, то можно отметить появление в нем подкаталога picture3.files. Он автоматически создается редактором Word и используется для хранения изображений, используемых на веб-странице picture3.htm.

Белый фон под названием сайта необходимо сделать прозрачным. Такую возможность предоставляет формат GIF. Для этого необходимо использовать профессиональные графические

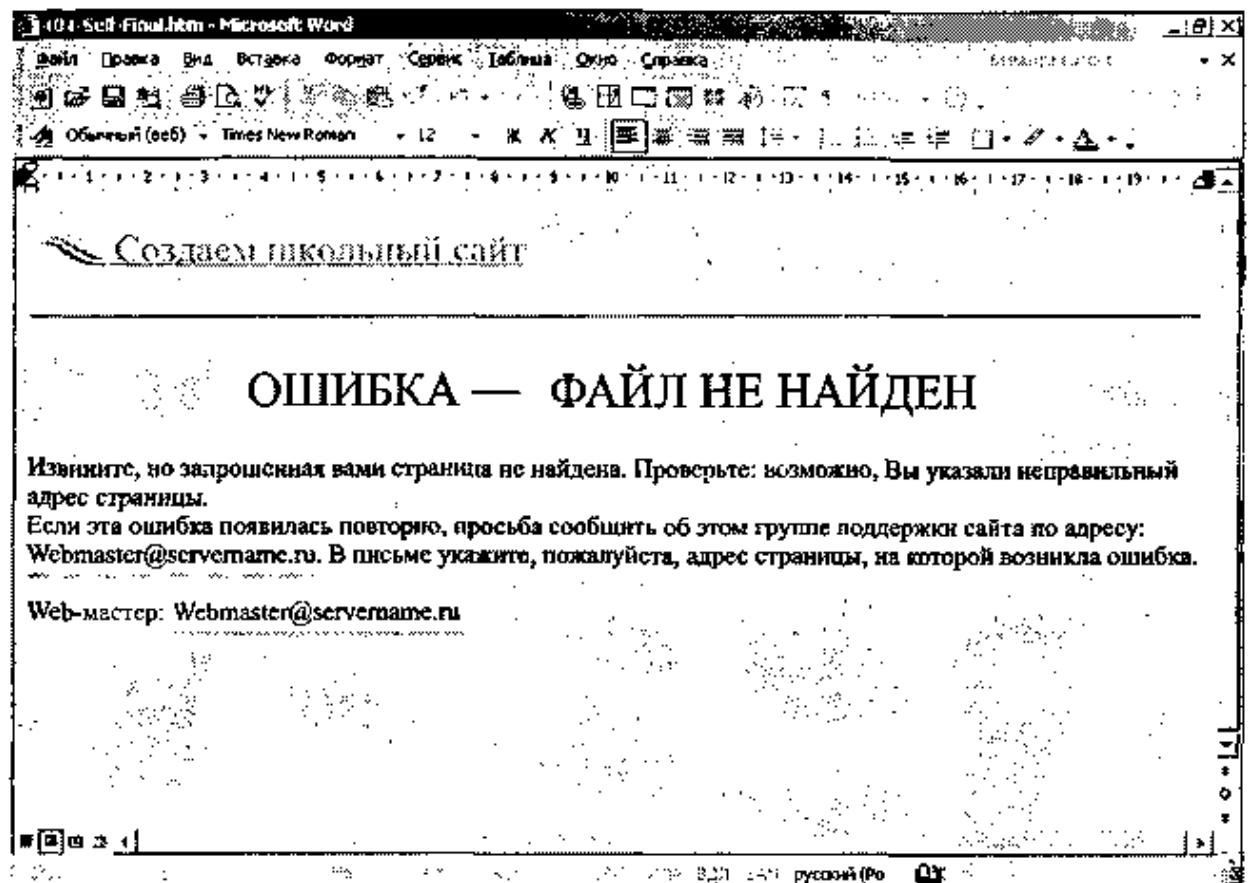


Рис. 7.3

редакторы или специальные программы. Альтернативная возможность — залить фон изображения цветом фона веб-страницы в обычном графическом редакторе, хотя результат подобной операции будет значительно хуже, чем при использовании прозрачного фона.



**Упражнение 3.** Откройте созданное изображение в редакторе Paint. Чтобы добиться наиболее полного соответствия цвета фона, необходимо изменить палитру. Для этого выберите пункт **Изменить палитру** в меню **Палитра**, выберите свободный элемент в области **Дополнительные цвета** и нажмите кнопку **Определить цвет**. Задайте параметры цветового оттенка: красный = 255, зеленый = 204, синий = 153 (в веб-странице эти параметры были определены кодом #FFCC99:  $FF_{16} = 255_{10}$ ,  $CC_{16} = 204_{10}$ ,  $99_{16} = 153_{10}$  для соответствующих основных цветов). Нажмите кнопки **Добавить в набор** и **ОК**. Далее выберите на панели инструментов элемент **Заливка** и залейте фон выбранным цветом.



Образец рисунка имеется на прилагаемом диске в папке `chapter7` под именем `logo_3_f.gif`.

Поместите только что созданный рисунок на веб-страницу (см. упражнение 2). Результат (при использовании файла logo\_3\_f.gif) представлен на рис. 7.4.



Образец документа имеется на прилагаемом диске в папке chapter7 под именем picture3\_f.htm.

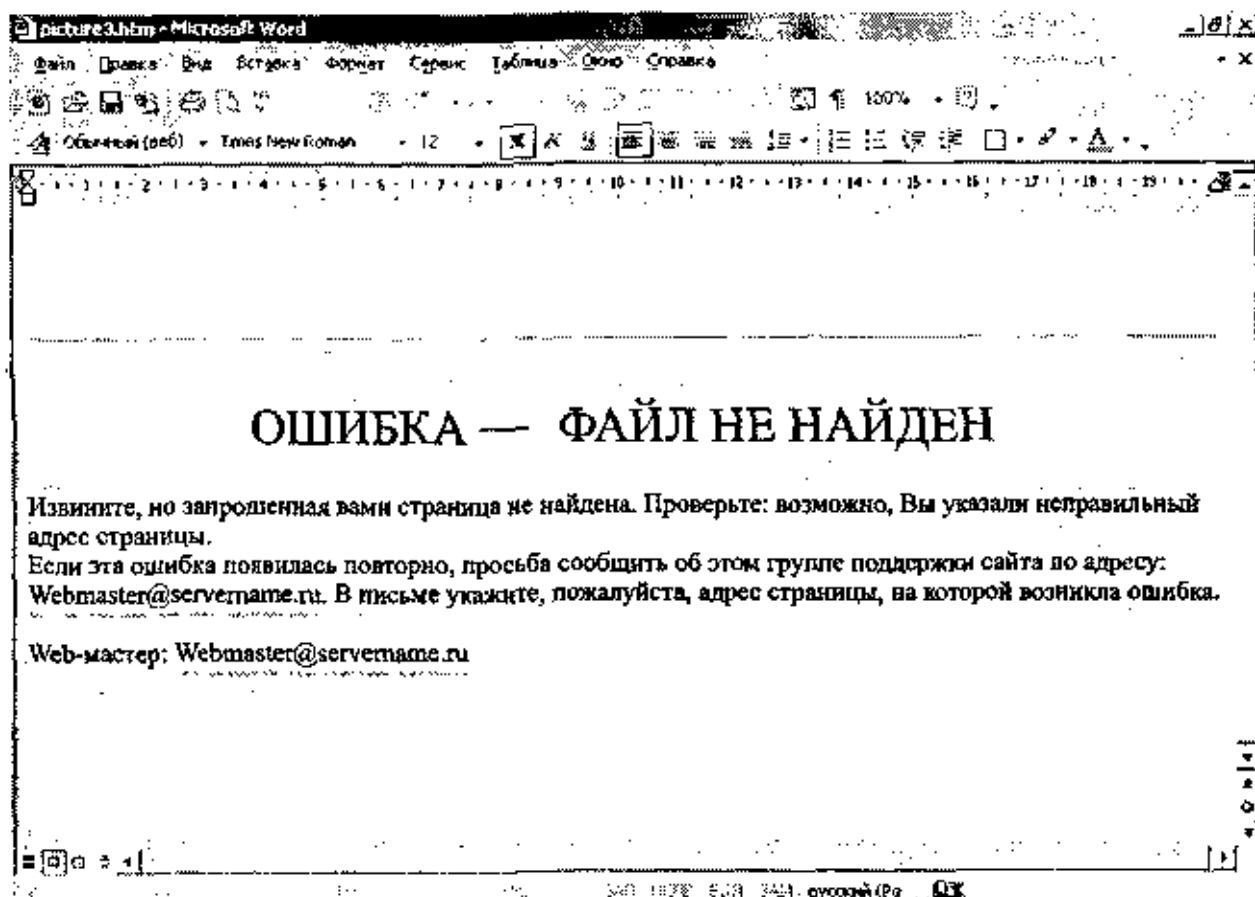


Рис. 7.4

Описанный выше способ обладает существенным недостатком: при сохранении документа редактор Word осуществляет его преобразование, что приводит к увеличению объема файла (объем picture3.htm составляет 4785 байтов тогда как исходный файл, на базе которого он был создан, занимает только 546 байтов).

## Создание прозрачного фона в редакторе Adobe PhotoShop

Рассмотрим процесс создания прозрачного фона с помощью графического редактора Adobe PhotoShop (версия 7.0). Редактор PhotoShop — это прекрасный графический редактор для обработки оцифрованных изображений, снабженный большим количеством фильтров.

Процесс создания прозрачного фона состоит из следующих этапов:

- 1) открыть рисунок;
- 2) выбрать пункт **Сохранить для Web** в меню **Файл**;
- 3) в появившемся окне (рис. 7.5) выбрать закладку **2-Up**, разделив окно изображения на две части. В верхней содержится исходное изображение, в нижней — оптимизированное. На панели настроек справа вверху можно, меняя параметры изображения, оптимизировать его по объему и качеству;

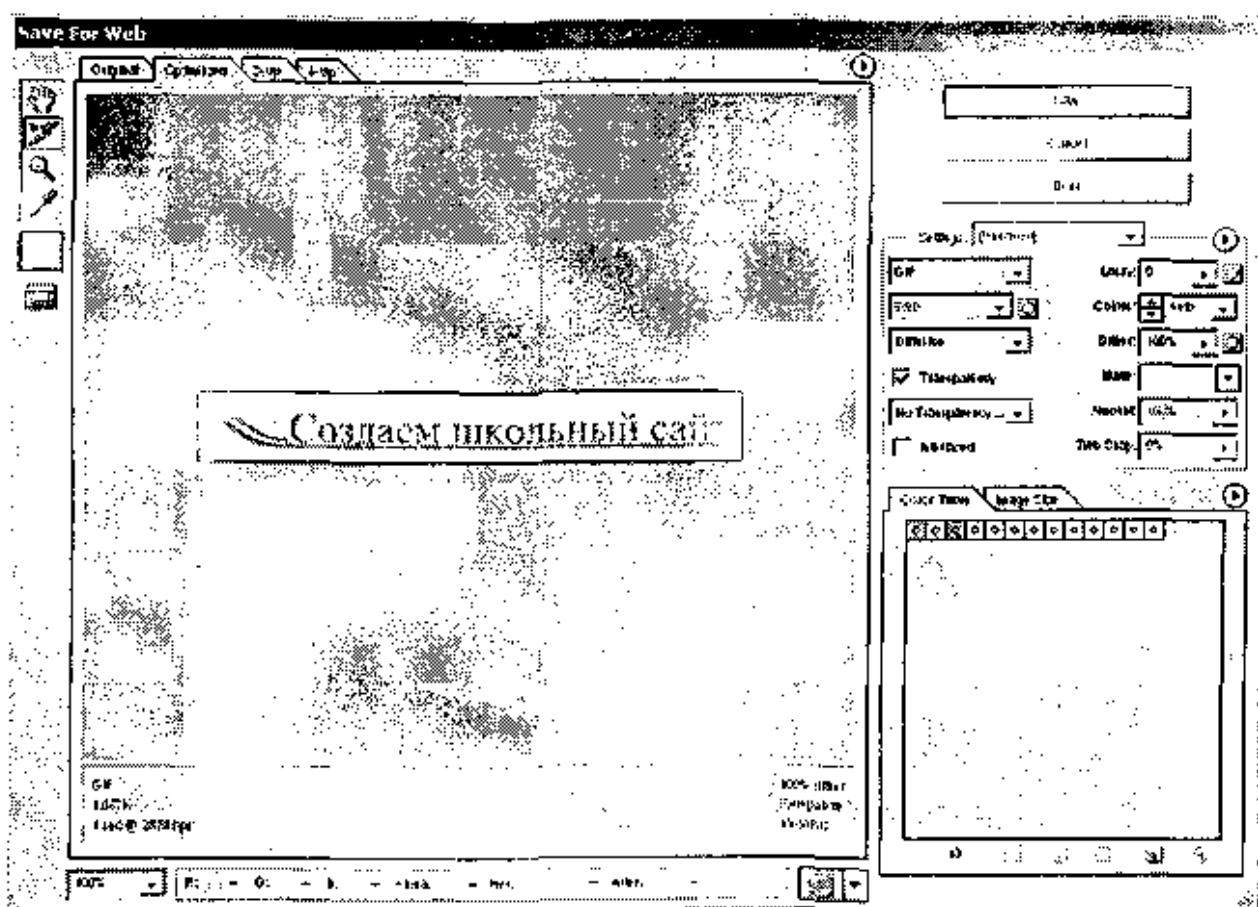


Рис. 7.5

- 4) в таблице цветов (**Color Table**) выбрать цвет, который необходимо сделать прозрачным, и нажать кнопку **Map/unmap selected colors to/from transparent**, расположенную под таблицей цветов (□). Если потребуется опять сделать цвет непрозрачным, надо еще раз нажать эту же кнопку. При выборе белого цвета в нашем примере оптимизированное изображение должно измениться аналогично рис. 7.6;
- 5) нажать кнопку **Save** в окне сохранения изображения для **Web** и затем сохранить изображение в указанной папке.



Рис. 7.6

## Теговое включение графики

В языке HTML существует отдельный тег для вставки в текст изображений — `<img>`. Он используется без закрывающего тега, а изображение размещается там, где этот тег вставлен в текст.

Параметры изображения и имя файла, в котором оно хранится, определяются инструкциями тега (атрибутами).

Инструкция `src` определяет имя графического файла. Пример:

```

```



**Упражнение 4.** Осуществите форматирование текста исходной веб-страницы (файл имеется на прилагаемом диске в папке `chapter6` под именем `404-self-final.htm`). Сохраните результат работы в файле с именем `404-self-img.htm`. Вид полученной страницы в веб-браузере должен соответствовать рис. 7.7.



Образец документа имеется на прилагаемом диске в папке `chapter7` под именем `404-self-img.htm`.

Среди других инструкций тега `<img>` существуют следующие:

- `alt` — альтернативное текстовое описание рисунка;
- `height` — переопределение высоты рисунка;
- `width` — переопределение ширины рисунка;
- `border` — задание ширины границы.

Общий объем веб-страницы с учетом изображений теперь составляет:

- при использовании редактора Word — 8355 байтов (файл `picture3_f.htm` и каталог `picture3_f.files`);

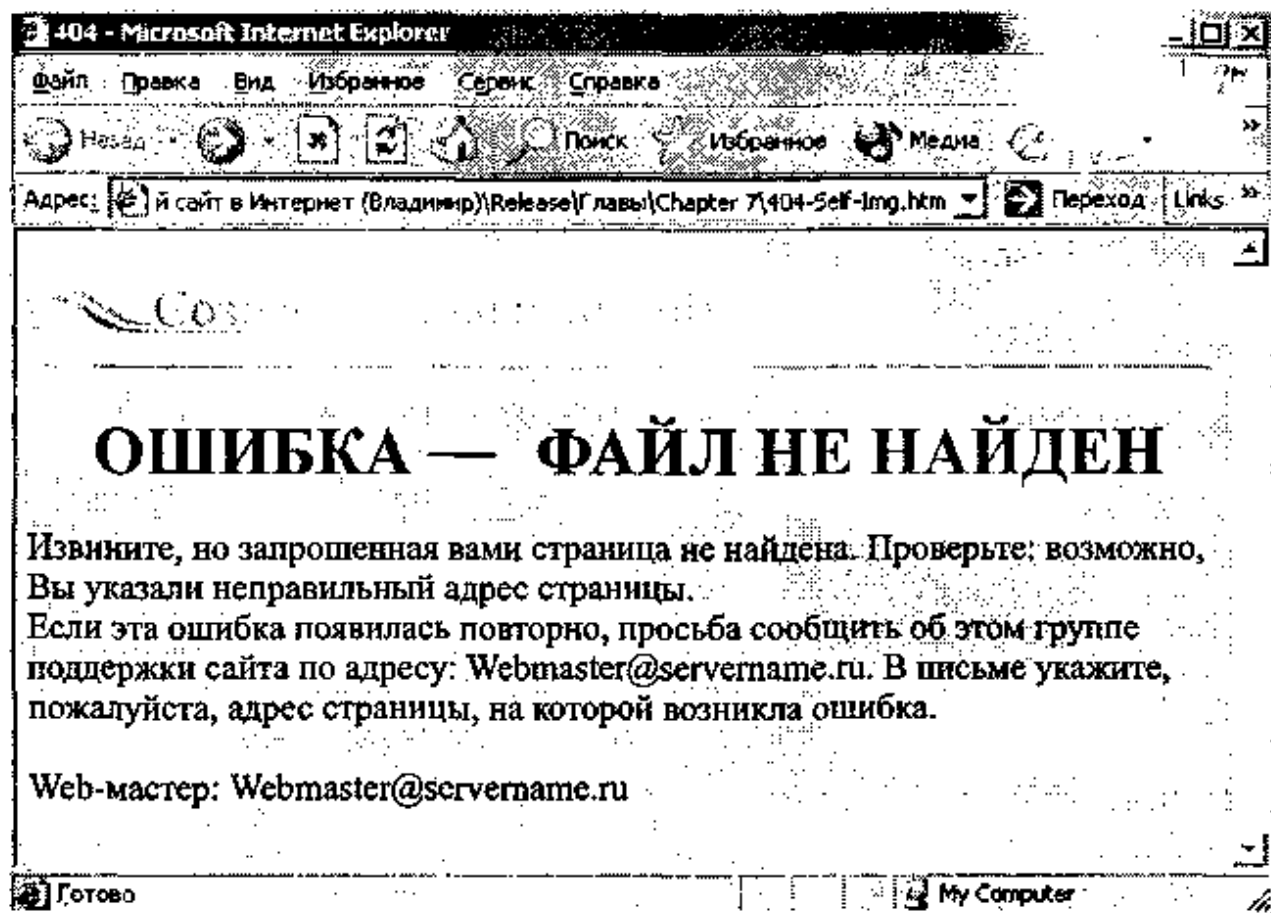


Рис. 7.7

- при теговом включении графики — 3954 байта (файлы 404-self-img.htm и logo\_3\_f.gif).

Нетрудно оценить, насколько более экономным по объему файлов и по времени загрузки страницы является ручной способ по сравнению с автоматизированным. Вместе с тем, также нетрудно заметить и большее удобство (и экономию времени) при использовании автоматизированного способа для разработки большого числа содержащих графику веб-страниц.



## Коротко о главном

1. Существует три основных формата растровой графики, используемых для представления графических элементов в сети Интернет: GIF (Graphics Interchange Format), JPEG (Joint Photographic Experts Group) и PNG (Portable Network Graphics), обладающих своими преимуществами и недостатками.
2. При создании графических изображений, ориентированных на использование в веб-страницах, необходимо учитывать размер и разрешение изображения, а также глубину передаваемого цвета.



3. Применение автоматизированных средств позволяет быстрее добавлять графические элементы на веб-страницу.
4. В языке HTML существуют специальные средства для вставки графических объектов.



## Вопросы и задания

1. Назовите основные форматы растровой графики, используемые в Интернете.
2. В чем состоят различия графических форматов, используемых в Интернете?
3. Какие параметры изображений следует учитывать при проектировании графических объектов веб-страниц?
4. Какие вы знаете средства автоматизированного размещения графических объектов на веб-странице?
5. Какие средства языка HTML позволяют вставлять в текст изображения?



## Практическое задание

Создайте веб-страницу, на которой демонстрируются графики базовых математических функций. В работе используйте редакторы Microsoft Word и PaintBrush (минимальный уровень) или HTML и профессиональный графический редактор (повышенный уровень; можно использовать редактор Adobe PhotoShop).



## Темы рефератов

1. Формат GIF.
2. Формат JPEG.
3. Методы сжатия графической информации.
4. Оптимизация графики для использования на веб-страницах.

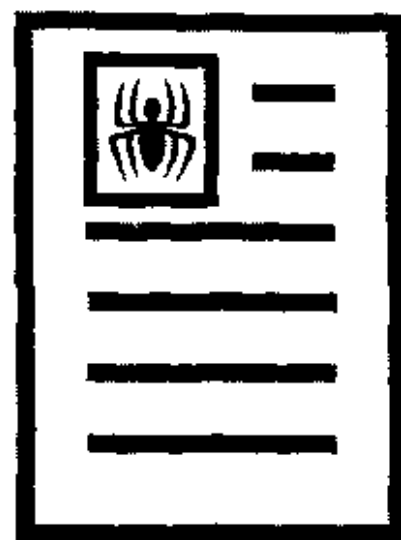


## Демонстрационный плакат

Добавление графических элементов на веб-страницу.

## Глава 8

# ВЫПОЛНЯЕМ ВЕРСТКУ СТРАНИЦ



Верстка (размещение элементов на странице) в веб-документах является понятием относительным. Расположение тех или иных элементов здесь определяется относительно положения других элементов. Структурированность документа позволяет повысить уровень восприятия представленной в нем информации.

Для верстки здесь в основном используются два подхода: невидимые таблицы и фреймы. *Фреймы* представляют собой вложенные окна, позволяющие разбить окно браузера на несколько подокон.

Структура документа, определенная расположением и характеристиками ячеек таблицы либо фреймами, может использоваться для расположения различных элементов. Интересной особенностью является автоматическое форматирование таблиц и фреймов браузером в пределах его рабочего окна для обеспечения наилучшего представления информации. Хотя существует и возможность жесткого задания размеров элементов таблиц и фреймов в пикселях, разработчикам в данном случае рекомендуется указывать относительные размеры элементов таблиц и фреймов.

Следует отметить, что полное отображение таблицы возможно только после ее полной загрузки, в связи с чем использование больших по объему таблиц является нецелесообразным. Решением подобной проблемы (если создание больших таблиц все же необходимо) может стать использование нескольких меньших таблиц, которые в совокупности образуют единую таблицу.

В данной главе рассматриваются способы создания таблиц в веб-документах как с помощью текстового редактора MS Word, так и путем вставки соответствующих тегов. Создание же фреймов рассматривается только на уровне кодирования тегов, так

как визуальное проектирование фреймовой структуры возможно только при наличии профессиональных HTML-редакторов и не предусмотрено в MS Word.

## Визуальное создание таблиц в редакторе MS Word

Редактор MS Word помимо кнопок в панели инструментов содержит меню **Таблицы**, с помощью которого можно создавать и редактировать таблицы различного вида, и предоставляет множество возможностей для оформления таблиц. Основным преимуществом здесь является их визуальное создание в полном соответствии с принципом «Что вижу, то и получаю», заложенным в основе редактора Word.

Процесс создания таблиц для веб-страницы не особенно отличается от процесса создания обычных таблиц. При этом следует помнить, что многие элементы «оформительского» характера не поддерживаются языком HTML и его расширениями, следовательно, они не могут быть отображены браузером. При сохранении документа Word в формате HTML-документа, редактор пытается осуществить это преобразование наиболее полно, но если оно невозможно, то элемент преобразуется к другому, наиболее похожему на него допустимому элементу.

Рассмотрим пример. Пусть требуется создать таблицу, аналогичную показанной на рис. 8.1.

	Текстовые редакторы	Блокнот	Word
Возможности			
Поддержка форматирования		-	+
Поддержка таблиц		-	+
...		...	...

Рис. 8.1

При создании подобной таблицы в редакторе MS Word необходимо выполнить следующие действия (для версии Word 2000):

- 1) в меню **Таблицы** выбрать пункт **Вставить** и далее — пункт **Таблица**;
- 2) задать количество строк, равное 4, количество столбцов, равное 3, и стиль таблицы **Сетка 7** (доступ к списку стилей обеспечивает кнопка **Автоформат**);
- 3) ввести данные в соответствующие ячейки таблицы.

Эта таблица в первой ячейке первой строки содержит элемент (наклонную разливку), который не может быть закодирован средствами языка HTML. Если выполнить сохранение такого документа в качестве веб-страницы (меню **Файл**, пункт **Сохранить как веб-страницу**), то в браузере она будет выглядеть аналогично рис. 8.2.



Образец документа имеется на прилагаемом диске в папке chapter8 под именем picture1.1.htm.

Текстовые редакторы	Блокнот	Word
Возможности		
Поддержка форматирования	-	+
Поддержка таблиц	-	+
...	...	...

Рис. 8.2

Это правило относится и ко многим другим возможностям редактора MS Word и объясняется тем, что при создании редактора Word преследовались другие цели; лишь впоследствии к его функциям была добавлена возможность преобразования документов в формат HTML.

## Теговое создание таблиц

Для построения таблиц используется группа тегов, которые определяют параметры таблицы целиком, ее отдельных строк и столбцов.

**Определение параметров таблицы.** Создание таблицы начинается с добавления тега `<table>`, который вместе с завершающим тегом `</table>` является контейнером, определяющим параметры всех ячеек таблицы. Эти теги указывают границы представления таблицы в теле веб-документа.

Перечислим наиболее часто используемые атрибуты тега `<table>`.

- Атрибут `border` используется для определения толщины рамки таблицы (в пикселях). Например, чтобы задать толщину, равную 5 пикселям, тег `<table>` должен иметь вид (конструкция `<!-- ... -->` представляет собой комментарий и не отображается браузером):

```
<table border="5">  
  <!-- тело таблицы -->  
</table>
```

Чтобы задать отсутствие рамки таблицы, необходимо определить ее толщину равной нулю. Отсутствие рамки не влияет на структуру таблицы. Подобные таблицы часто называют «невидимыми».

- Атрибут `width` определяет рекомендуемую ширину таблицы — относительную (в процентах от размеров окна браузера) или фиксированную (в пикселях). При отсутствии данного атрибута ширина таблицы определяется ее содержимым. Следует отметить, что задание ширины таблицы равной 100 % приводит к появлению в окне браузера полос прокрутки, поэтому рекомендуется использовать немного меньшее значение. Например, чтобы задать относительную ширину таблицы, равную половине ширины окна браузера, тег `<table>` должен иметь следующий вид:

```
<table border="5" width="50%">  
  <!-- тело таблицы -->  
</table>
```

Аналогичная конструкция с указанием фиксированной ширины таблицы (если считать, что окно браузера развернуто на весь экран при разрешении 800×600) может иметь вид:

```
<table border="5" width="400">  
  <!-- тело таблицы -->  
</table>
```

- Атрибут `cellspacing` определяет интервал между ячейками (как горизонтальный, так и вертикальный), а также интервал между рамкой таблицы и граничными ячейками (черные стрелки на рис. 8.3).
- Атрибут `cellpadding` определяет интервал между границей ячейки и ее содержимым (серые стрелки на рис. 8.3 для ячейки 2).

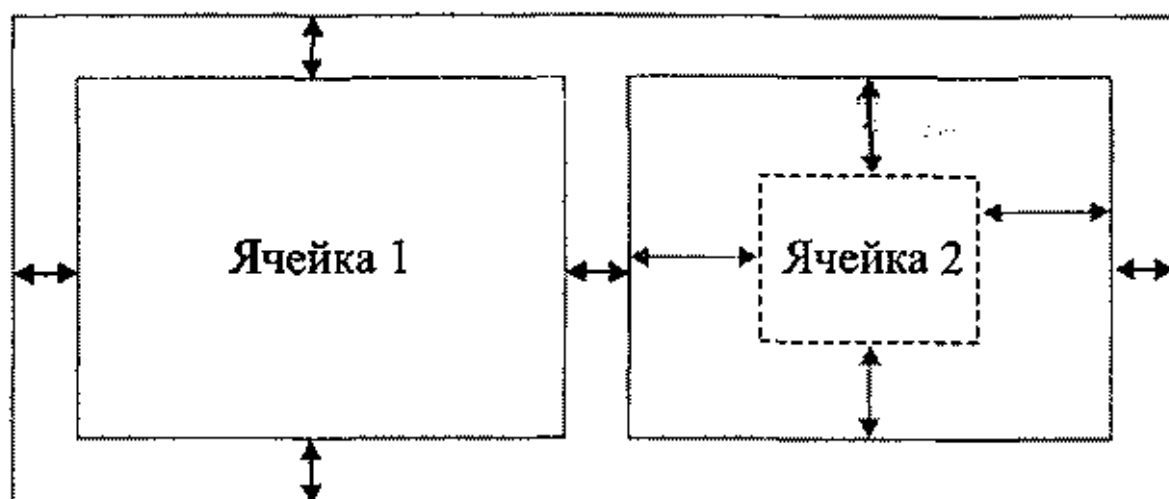


Рис. 8.3

- Атрибут `bgcolor` определяет цвет фона того элемента, в котором он указывается (например, цвет фона для всей таблицы).

**Определение параметров строки таблицы.** После определения общих параметров для всей таблицы необходимо определить параметры для каждой ее строки и ячейки.

Строки в таблицу добавляются с помощью тегов `<tr>` и `</tr>`. Они являются контейнером для отдельных ячеек. Атрибут `align` может использоваться здесь для выравнивания текста в пределах строки.

**Определение параметров ячейки таблицы.** После определения строк в них добавляются отдельные ячейки. Для этого используются теги `<td>` и `</td>`. С помощью атрибута `width` можно определять ширину ячейки, а атрибут `align` позволяет выравнивать текст в пределах ячейки.

Ячейки в таблице можно *объединять*. Для этого используются атрибуты `colspan` (объединение по ширине) и `rowspan` (объединение по высоте). При этом нужно удалить из конструкции таблицы лишние теги ячеек.

Примеры.

1. Чтобы определить структуру таблицы, представленной на рис. 8.4, а (объединение двух ячеек в строке), текст HTML-документа должен иметь следующий вид:

```
<table border="1">
  <tr>
    <td>1</td>
    <td>2</td>
    <td>3</td>
  </tr>
  <tr>
    <td colspan="2">4</td>
    <!-- ячейка 5 отсутствует! -->
    <td>6</td>
  </tr>
  <tr>
    <td>7</td>
    <td>8</td>
    <td>9</td>
  </tr>
</table>
```

1	2	3
4	6	
7	8	9

а)

1	2	3
4		6
7	8	9

б)

Рис. 8.4

2. Чтобы определить структуру таблицы, представленной на рис. 8.4, б (объединение двух ячеек в столбце), текст HTML-документа должен иметь вид:

```
<table border="1">
  <tr>
    <td>1</td>
    <td rowspan="2">2</td>
    <td>3</td>
  </tr>
  <tr>
    <td>4</td>
    <!-- ячейка 5 отсутствует! -->
    <td>6</td>
  </tr>
  <tr>
    <td>7</td>
    <td>8</td>
    <td>9</td>
  </tr>
</table>
```

Теги для создания таблиц могут содержать и множество других атрибутов, причем некоторые из них могут наследоваться из внешних тегов во вложенные. Примерами могут служить атрибуты выравнивания и фонового цвета.



**Упражнение 1.** Создайте HTML-документ, содержащий таблицу (рис. 8.5), и сохраните его с именем picture5.htm.

1	2		3
	4	5	
	6		

Рис. 8.5



Образец документа имеется на прилагаемом диске в папке chapter8 под именем picture5.htm.





**Упражнение 2.** Создайте HTML-документ, содержащий таблицу с фоновой закраской ячеек (рис. 8.6), и сохраните его с именем `picture6.htm`.



Рис. 8.6



Образец документа имеется на прилагаемом диске в папке `chapter8` под именем `picture6.htm`.

## Создание фреймовой структуры

Использование фреймов подразумевает создание нескольких независимых информационных документов, загружаемых каждый в свой фрейм, и создание специального файла, определяющего структуру фреймов. Например, для создания HTML-документа, содержащего два фрейма, необходимо создать три файла. Зато у фреймов есть преимущества, которых нет у таблиц. Так, созданием отдельных фреймов могут заниматься разные разработчики. Каждый фрейм обрабатывается браузером отдельно, что дает возможность сделать часть информации постоянно видимой в окне браузера. Существует также возможность управления одним фреймом из других, что широко применяется для создания меню, заголовков сайтов и других элементов, которые должны постоянно присутствовать на странице. Однако следует помнить (хотя сейчас это требование и не особенно актуально), что не все браузеры поддерживают фреймы, так что, используя их, вы ограничиваете круг потенциальных посетителей вашего сайта. Эту проблему можно решить, предусматривая во фреймовой структуре запасной бесфреймовый вариант.

Рассмотрим процесс создания фреймового документа на примере веб-страницы, эскиз которой показан на рис. 8.7.

В данном случае управляющим фреймом является второй. Определив элементы меню, мы можем в зависимости от ситуации менять с их помощью содержимое первого или третьего фрейма, не затрагивая при этом второй фрейм. Тем самым также снижается объем пересылаемой информации, так как часть документа (меню) уже загружена.

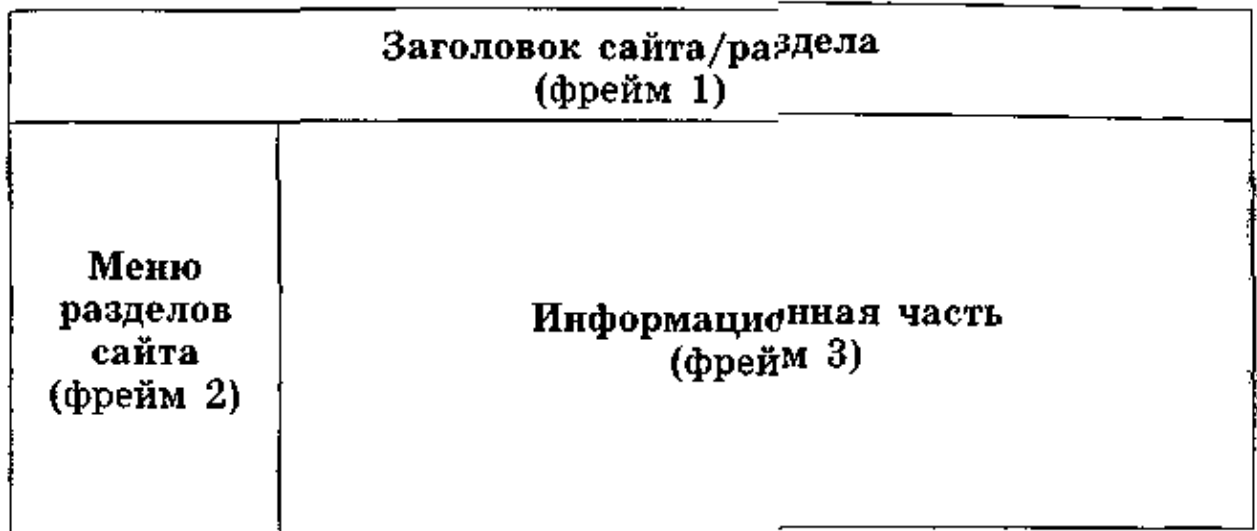


Рис. 8.7

Каждый фрейм — это отдельный HTML-документ. Веб-страницы, загружаемые во фреймы, представляют собой обычные HTML-документы. Пусть во фрейм 1 загружается файл `header.htm`, во фрейм 2 — файл `menu.htm`, во фрейм 3 — `info_1.htm`. Файлу структуры фреймов дадим имя `index_fr.htm`.

Текст HTML-документа `header.htm` при этом может иметь вид:

```
<html>
  <head>
    <title>Фрейм 1</title>
  </head>
  <body>
    <p align=left><hr></p>
    <!-- файл logo_3_f.gif — название сайта из главы 7 -->
  </body>
</html>
```

Текст HTML-документа `menu.htm` может иметь вид:

```
<html>
  <head>
    <title>Фрейм 2</title>
  </head>
  <body>
    <!-- создаем нумерованный список -->
    <p align=left>
      <ol>
        <li> О сайте
        <li> История школы
        <li> Объявления
        <!-- ... -->
```

```
    </ol>
  </p>
</body>
</html>
```

Текст HTML-документа info\_1.htm может иметь вид:

```
<html>
  <head>
    <title>Фрейм 3</title>
  </head>
  <body>
    <h2 align=right>История школы</h2>
    <p align=left>
      Школа была основана ...
    </p>
  </body>
</html>
```

После создания каждого отдельного фрейма их нужно связать в единую конструкцию. Для этого используется HTML-документ, в котором тело документа (раздел BODY) заменяется описанием фреймовой структуры.

Структура фреймов определяется с помощью тега <frameset>, который является контейнером для тегов, определяющих фреймы, — <frame>.

Тег <frameset> имеет два важных атрибута: rows и cols. Первый из них определяет количество горизонтальных фреймов (строк), второй — количество вертикальных (столбцов). Фреймы создаются слева направо (для столбцов) и сверху вниз (для строк). Если указаны оба атрибута, фреймы создаются слева направо в каждой строке. Размеры фреймов могут определяться тремя способами: фиксировано, относительно страницы, либо относительно других фреймов и страницы. Фиксированное определение размера фрейма означает задание его длины (или ширины) в пикселях, относительное — в процентах от размеров окна браузера. Символ "\*" при этом используется для обозначения всего оставшегося пространства окна браузера и вместе с числами может определять размеры фреймов. Так, если документ состоит из двух фреймов, кодируемых атрибутом rows="2\*,\*", то это означает, что окно браузера будет разделено на две строки, причем первая будет в два раза больше второй. Если размер фрейма не указывается, то он занимает все доступное пространство.

Фреймы могут быть вложены в другие фреймы. Уровень вложенности фреймов не ограничен. Описание вложенных фреймов может храниться как в файле общей структуры фреймов, так и в отдельных файлах.

Какой файл следует изначально загружать в определенный фрейм, также определяется тегом `<frame>`. Его атрибут `src` задает URL HTML-документа, загружаемого в данный фрейм.

Чтобы определить фреймовую структуру, показанную на рис. 8.7, текст HTML-документа должен иметь следующий вид:

```
<html>
  <head>
    <title>Создаем школьный сайт</title>
  </head>
  <frameset rows="100,*">
    <!-- первая строка -->
    <frame src="header.htm">
      <!-- вторая строка представляет собой группу
            вложенных фреймов -->
      <frameset cols="*,4*">
        <!-- первый столбец второй строки -->
        <frame src="menu.htm">
          <!-- второй столбец второй строки -->
          <frame src="info_1.htm">
        </frameset>
      </frameset>
    </frameset>
  </html>
```

В результате при загрузке этого файла в браузере должен получиться документ, аналогичный показанному на рис. 8.8.

Границы между фреймами могут быть использованы для изменения их размеров. Атрибут `noresize` тега `<frame>` используется, если необходимо запретить изменение размеров текущего фрейма.

Среди других атрибутов данного тега можно выделить:

- `scrolling` — определяет присутствие полос прокрутки и может принимать три значения: `yes` — полосы присутствуют, `no` — полосы отсутствуют, `auto` — полосы присутствуют, если весь текст не умещается во фрейм;
- `frameborder` — определяет показ границ фрейма и может принимать значение 1 (границы показаны) или 0 (границы отсутствуют). Если установлено значение 0, то граница может все равно появиться, если ее показ задан в других фреймах.

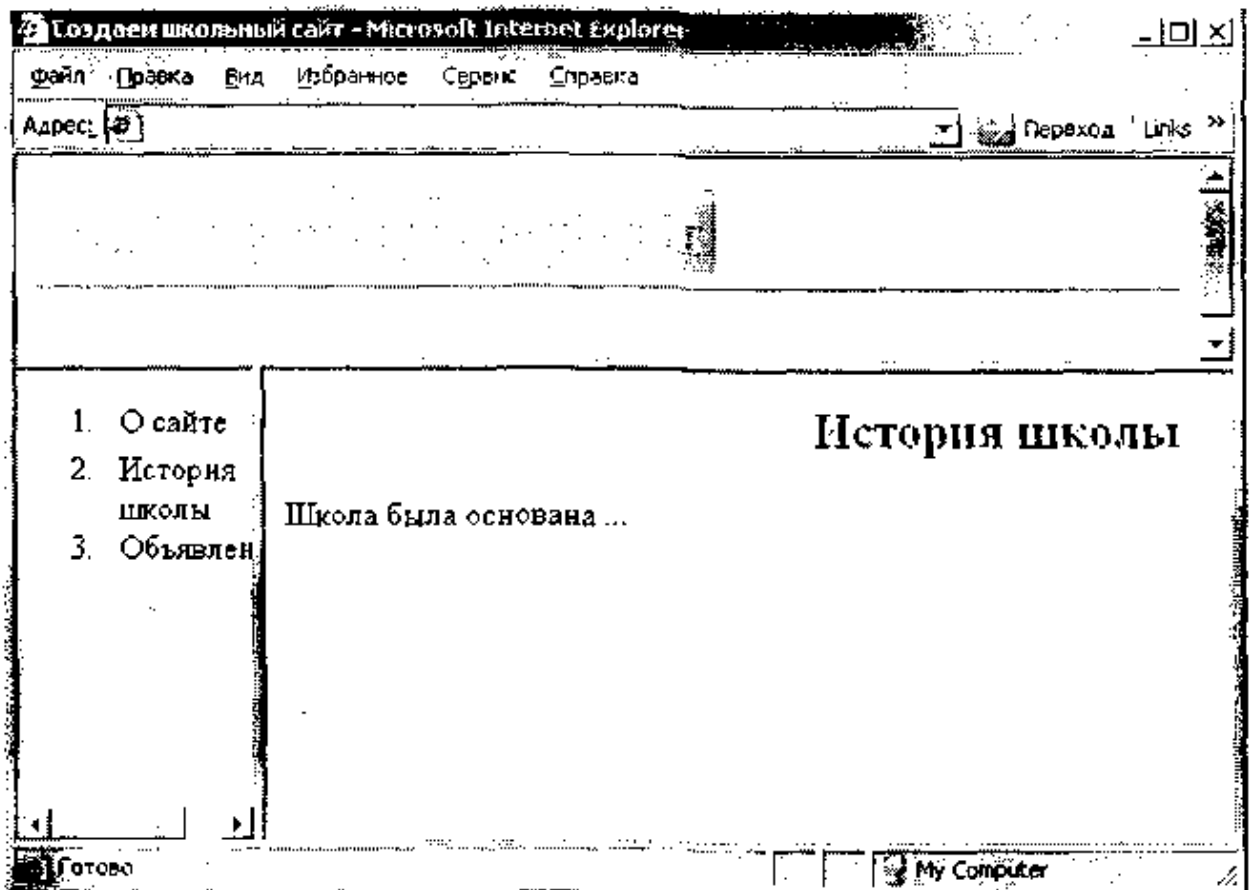


Рис. 8.8



**Упражнение 3.** Измените HTML-документ `index_fr.htm`, удалив из него границы фреймов и полосы прокрутки для фреймов 1 и 2. Сохраните файл под именем `index_fr_nd.htm`.



Образец документа имеется на прилагаемом диске в папке `chapter8` под именем `index_fr_nb.htm`.



**Упражнение 4.** Создайте фреймовый HTML-документ в соответствии с эскизом на рис. 8.9, где фреймы 3 и 4 являются группой фреймов, вложенных во фрейм 5 (структура этого фрейма определена в отдельном файле). Сохраните файлы под именами: `index_ex5.htm` — файл основной структуры фреймов, `frame_ex5_??.htm` — HTML-документы, загружаемые в соответствующие фреймы.



Образцы документов имеются на прилагаемом диске в папке `chapter8` под соответствующими именами.

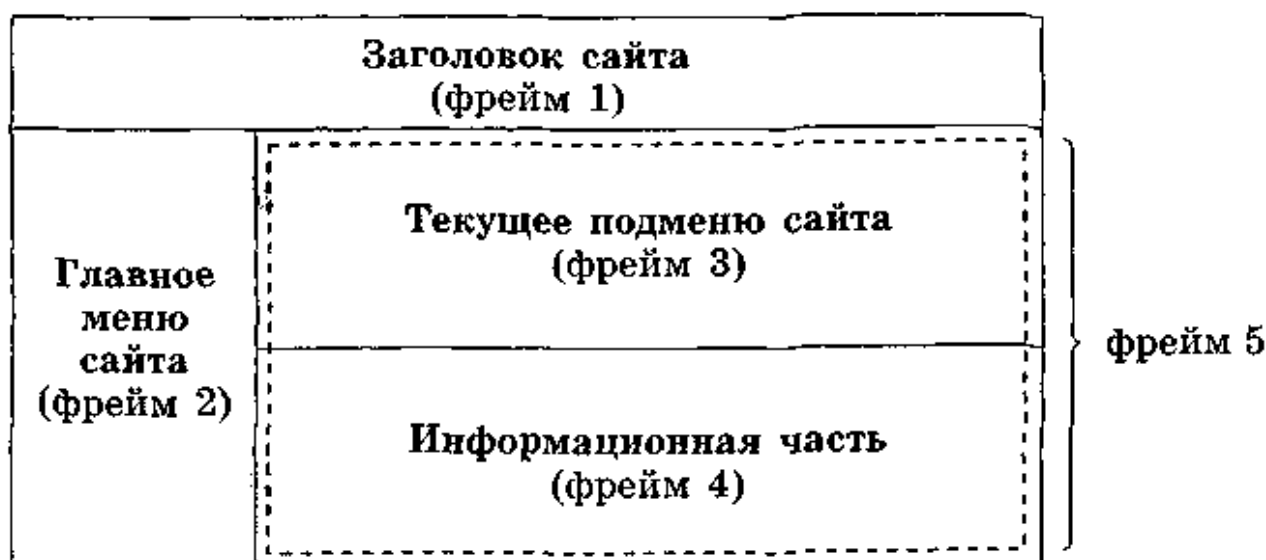


Рис. 8.9

При определении фреймовой структуры можно также использовать контейнер `<noframes>...</noframes>`. Его содержимое выводится в окне браузера, если данный браузер не поддерживает фреймы.

Чтобы информировать пользователя о невозможности просмотра фреймовой версии страницы, соответствующий HTML-документ должен иметь следующий вид:

```
<html>
  <head>
    <title>Создаем школьный сайт</title>
  </head>
  <frameset rows="100,*">
    <!-- ... -->
  </frameset>
  <noframes>
    Ваш браузер не может отобразить фреймы.
    Перейдите по адресу http://school1.city1.ru/index\_nf.htm
    к документу без использования фреймов.
  </noframes>
</html>
```



### Коротко о главном

1. В веб-документах нет жесткой координатной сетки, на основе которой можно было бы размещать элементы текста.

2. Расположение элементов в HTML-документах относительно.
3. Для верстки веб-страниц могут быть использованы невидимые таблицы и фреймы.
4. С помощью фреймов можно снизить объем передаваемой информации и разделять окно браузера на области, управлять которыми можно независимо.
5. Размеры элементов могут задаваться фиксированно или относительно.



## Вопросы и задания

1. Какие средства существуют в языке HTML для позиционирования элементов?
2. Как с помощью таблиц создать координатную сетку?
3. Прокомментируйте особенности использования таблиц и фреймов.
4. Как осуществляется объединение ячеек таблицы?
5. Назовите атрибуты, используемые во фреймах.



## Практическое задание

Осуществите разметку элементов дизайна и их верстку при использовании таблиц и фреймовой структуры.

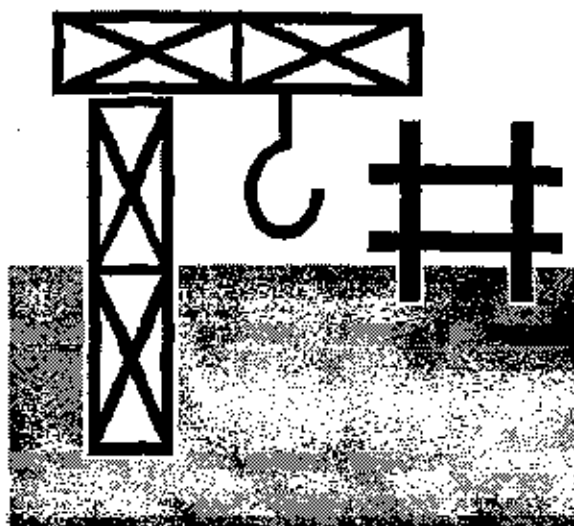


## Темы рефератов

1. Таблицы и фреймы в языке гипертекстовой разметки HTML.
2. Автоматизированные средства верстки, применяемые в издательской деятельности.

## Глава 9

# СОБИРАЕМ И ПУБЛИКУЕМ САЙТ НА ВЕБ-СЕРВЕРЕ



Гиперссылки являются наиболее важным элементом гипертекстовых документов. С их помощью реализуются связи между документами, предоставляется возможность создавать сложные документы, обращаться к ресурсам сети Интернет. В этом, собственно, и заключается основное различие обычных и гипертекстовых документов.

С одной стороны, ссылка — это указатель на определенный элемент текста. С другой стороны — это точка, на которую производится переход и которая получила название «якоря» (anchor). Ссылка может указывать как на элементы текущего документа, так и на любые ресурсы сети Интернет. В качестве ссылки может выступать текст или графический элемент.

Когда пользователь нажимает на ссылку, он (согласно терминологии языка HTML) начинает загрузку ресурса, на которую указывала ссылка. Если в качестве ресурса указан HTML-документ, то он может содержать «якорь», тогда при указании последнего в ссылке браузер осуществит сдвиг текста документа в окне до выбранной точки. Это открывает широкие возможности при создании оглавлений и организации навигации по документам. Ссылки в тексте документа обычно выделяются определенным образом, по умолчанию — цветом и подчеркиванием или рамкой (для графических элементов). При наведении курсора мыши на ссылку он изменяет свою форму. «Якорь» же — это скрытый элемент, который не отображается в тексте. Следует помнить, что использование в качестве «якоря» текста в национальной раскладке (на русском языке) или большого по объему не имеет смысла, так как этот элемент все равно является невидимым и к тому же может приводить к сбоям при неправильном определении.

При создании ссылок следует продумать, как пользователь будет возвращаться в случае необходимости к текущему доку-



менту. Для этого может понадобиться ссылка из документа, к которому он переходит, на текущий документ с соответствующим «якорем». Также можно использовать возможность браузера сохранять адреса посещенных страниц (использовать кнопку **Назад**) или открывать каждую ссылку в отдельном окне браузера.

После установки всех ссылок во всех документах необходимо полностью протестировать сайт, причем такое тестирование необходимо производить практически при любом изменении сайта. Если вы отредактировали документ, проверьте все ссылки, которые указывали на него, все ссылки с данного документа на другие и т. п. Далее этот вопрос будет рассмотрен более детально.

Когда вы сочтете, что сайт готов для публикации в сети Интернет, его необходимо скопировать на сервер. Если сервер — это компьютер, к которому вы имеете непосредственный доступ (установленный в пределах школы), то процесс публикации сайта представляет собой простое копирование файлов с одного компьютера на другой. Если же непосредственного доступа к серверу нет, то нужно использовать ftp-клиент.

Однако факт публикации сайта еще не является причиной завершения работ, связанных с его созданием. Необходимо провести тестирование всех страниц, проверить правильность работы всех элементов сайта и сервера.

## **Создание гиперссылок в редакторе MS Word**

Создание ссылок с помощью редактора MS Word основано на двух элементах: гиперссылках и закладках. Закладка при этом выполняет функции «якоря».

Рассмотрим в качестве примера создание страницы «История школы», содержащей оглавление и другие ссылки.

Выделим основные элементы текста и создадим оглавление. Например, в нем могут быть отражены история открытия школы, основные даты, текущее положение, планы развития, выпускники. Также на этой странице можно поместить ссылки на фотографии школы в различные периоды существования, по знаменательным датам, фотографии директоров школ и выпускников.

Создадим документ Word и введем текст оглавления и заголовков (рис. 9.1). Далее можно вводить текст в соответствующую

щие разделы страницы, если предполагается наличие ссылок в тексте (в рассматриваемом примере мы считаем, что в тексте ссылок нет).

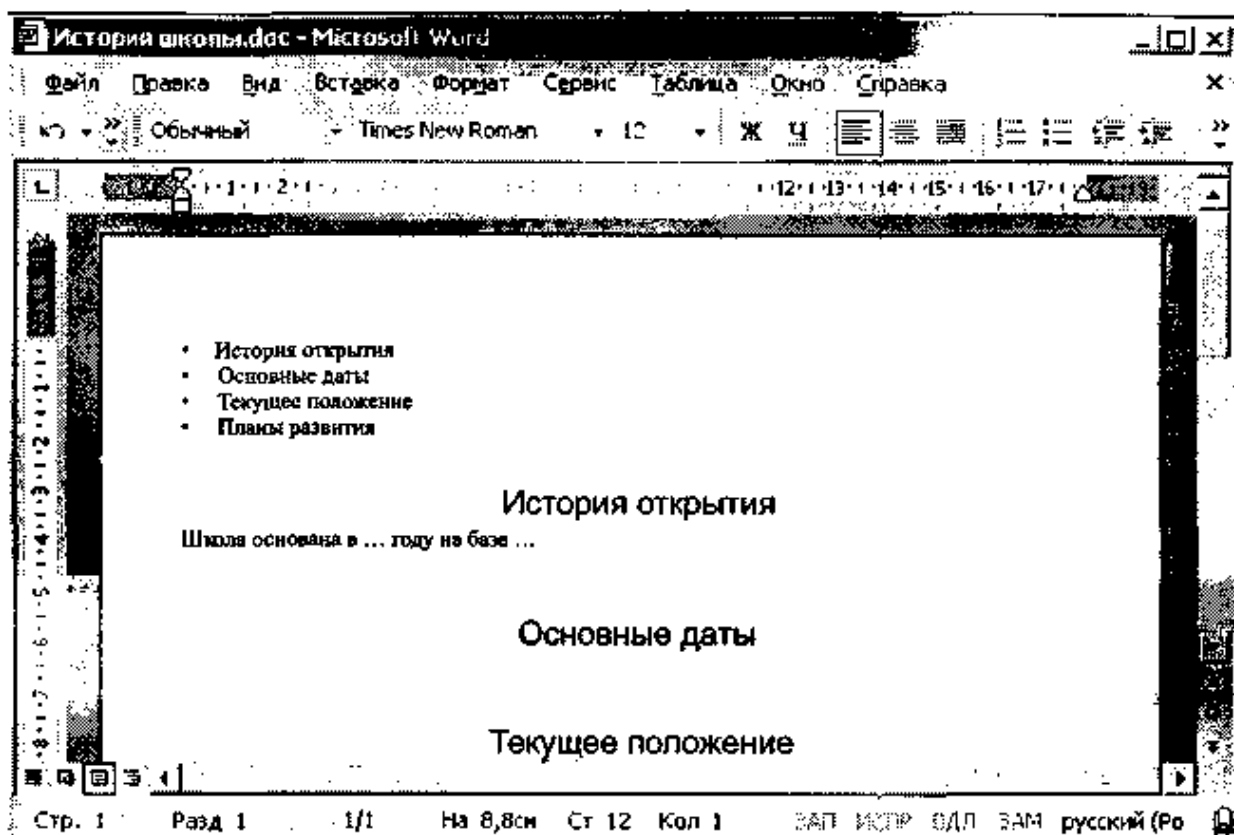


Рис. 9.1

Теперь необходимо расставить закладки в документе. Порядок создания закладок следующий:

- перемещаем курсор в позицию, где необходимо установить закладку (например, на первый символ заголовка «История открытия»);
- в меню **Вставка** выбираем элемент **Закладка** (рис. 9.2);
- вводим в соответствующее поле имя закладки (например, label1) и нажимаем кнопку **Добавить**.

Далее окно можно закрыть и перейти к созданию других закладок.

Определив все закладки в документе, перейдем к созданию гиперссылок. Добавление гиперссылок осуществляется с помощью пункта **Гиперссылка** меню **Вставка** (рис. 9.3).

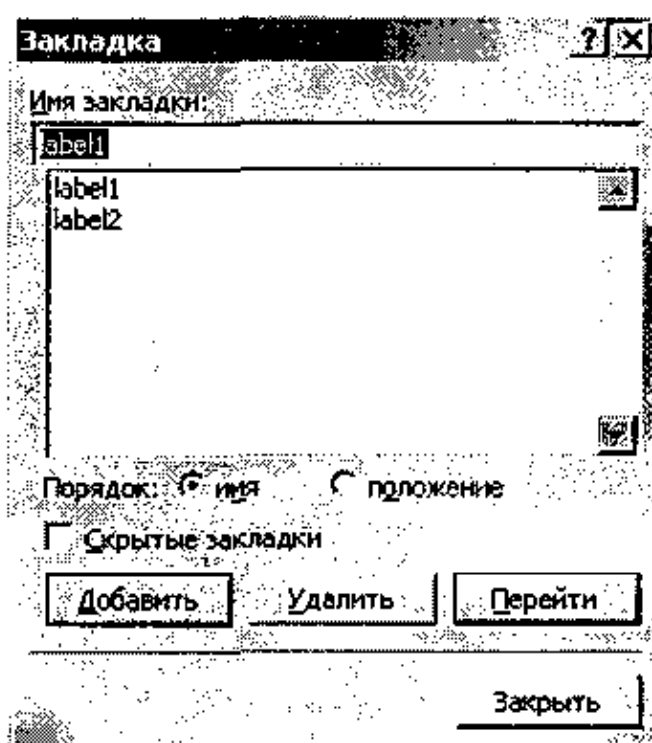


Рис. 9.2

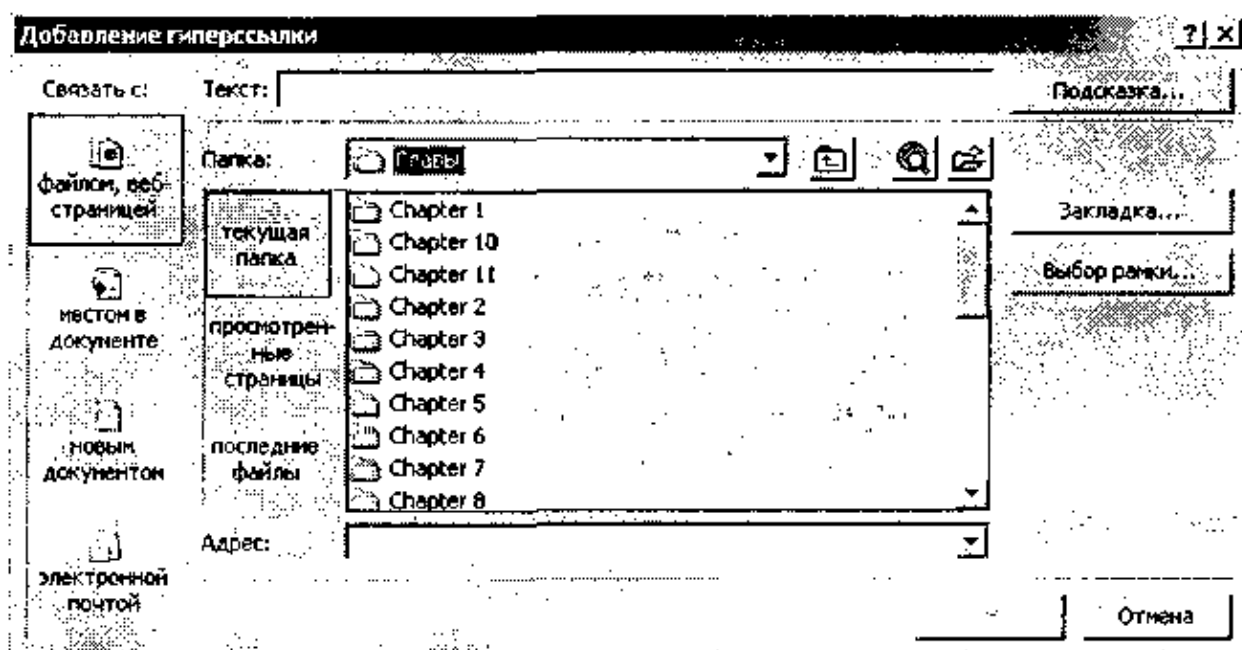


Рис. 9.3

Выделим текст, который будет являться ссылкой (например, слова «История открытия» в оглавлении). Выберем в меню Вставка пункт Гиперссылка. В окне Добавление гиперссылки перейдем на вкладку В местном документе, так как ссылки будут определены только в пределах текущего документа (рис. 9.4).

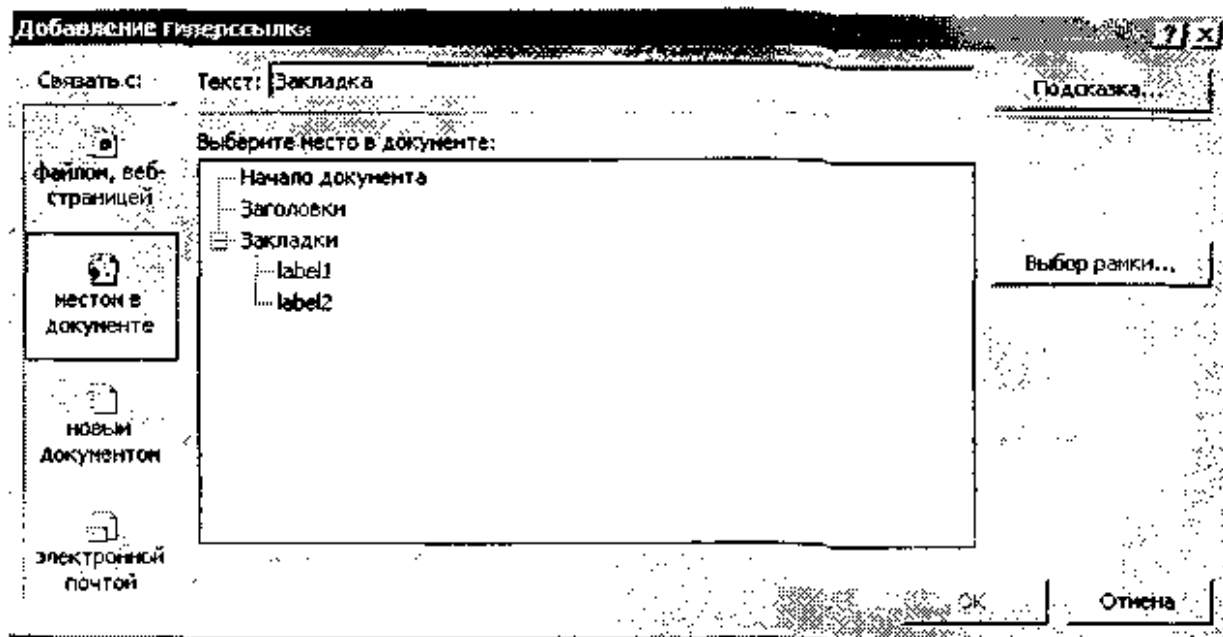


Рис. 9.4

Выберем из списка доступных закладок нужную (в данном случае — label1) и нажмем кнопку ОК. Здесь же можно задать подсказку к ссылке (кнопка Подсказка), определить, где открыть документ (кнопка Выбор рамки): в текущем окне браузера или в новом, и т. п.

Аналогично определяются остальные ссылки в текущем документе.

MS Word предоставляет также возможность создать ссылку на еще не существующий документ, после чего сразу же создать его. Для этого необходимо перейти на вкладку новым документом (рис. 9.5).

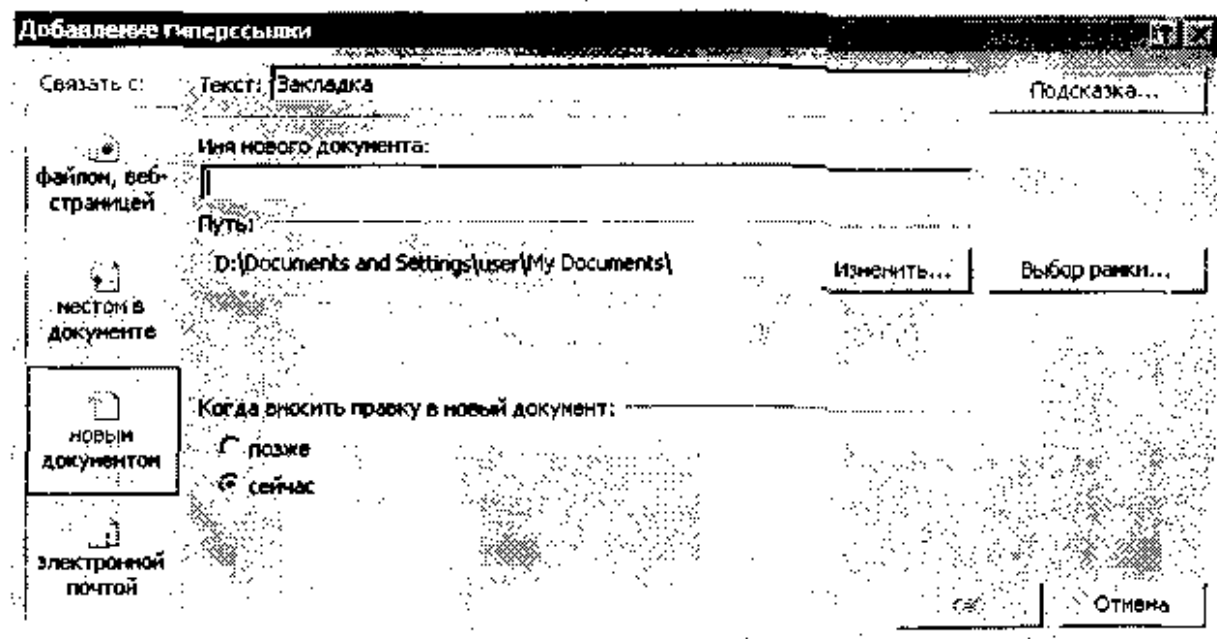


Рис. 9.5

Когда создается ссылка на другой документ (см. рис. 9.3), необходимо перейти на вкладку **файлом, веб-страницей**. В данном случае сначала необходимо выбрать файл или ввести его адрес. Если этот документ содержит закладки, то, используя кнопку **Закладка**, можно указать закладку в этом документе, на которую необходимо осуществить переход.

Ссылки на адрес электронной почты в основном используются для организации обратной связи пользователя с веб-мастером или для передачи контактной информации (например, для связи с директором школы). Для задания такой ссылки необходимо перейти на вкладку **электронной почтой** и ввести адрес электронной почты в соответствующее поле (рис. 9.6). Здесь же можно ввести строку темы письма, по которой вы сможете определять тип и важность сообщений, приходящих на ваш почтовый адрес. Например, одна группа писем может адресоваться веб-мастеру для сообщения об ошибках (тема «Ошибки»), другая — классным руководителям (по строке темы может определяться, кому именно адресовано письмо) и т. д.

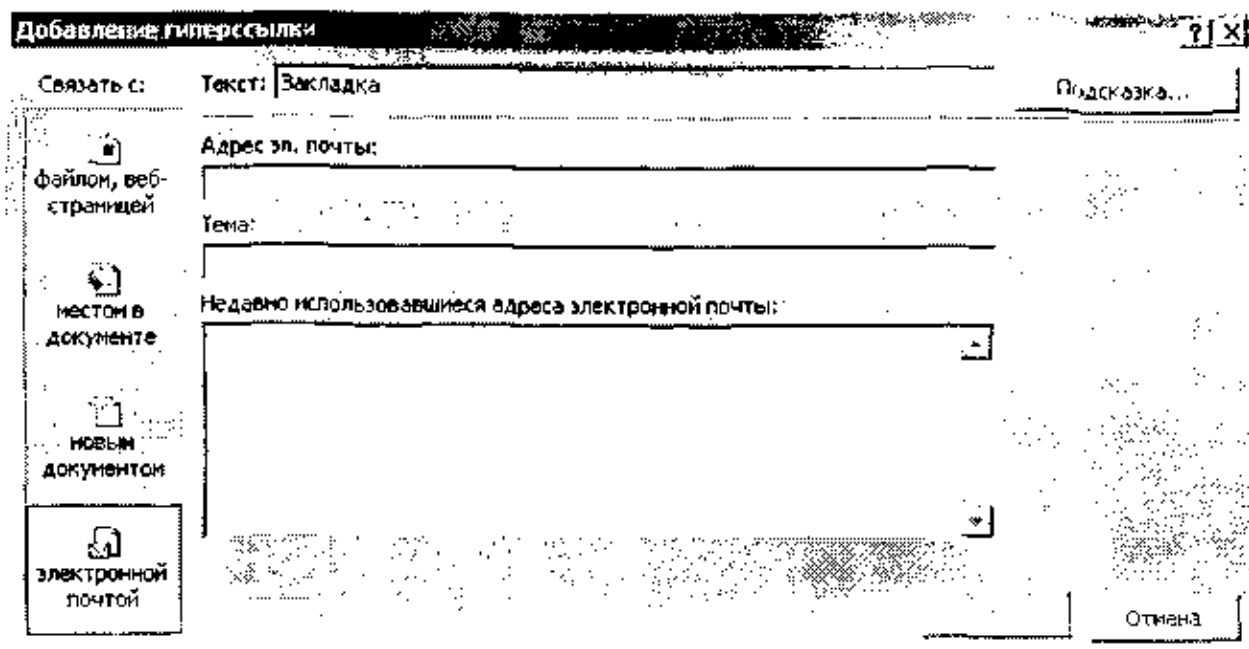


Рис. 9.6

В итоге должен получиться документ, аналогичный представленному на рис. 9.7 (ссылки выделены синим цветом и подчеркнуты).

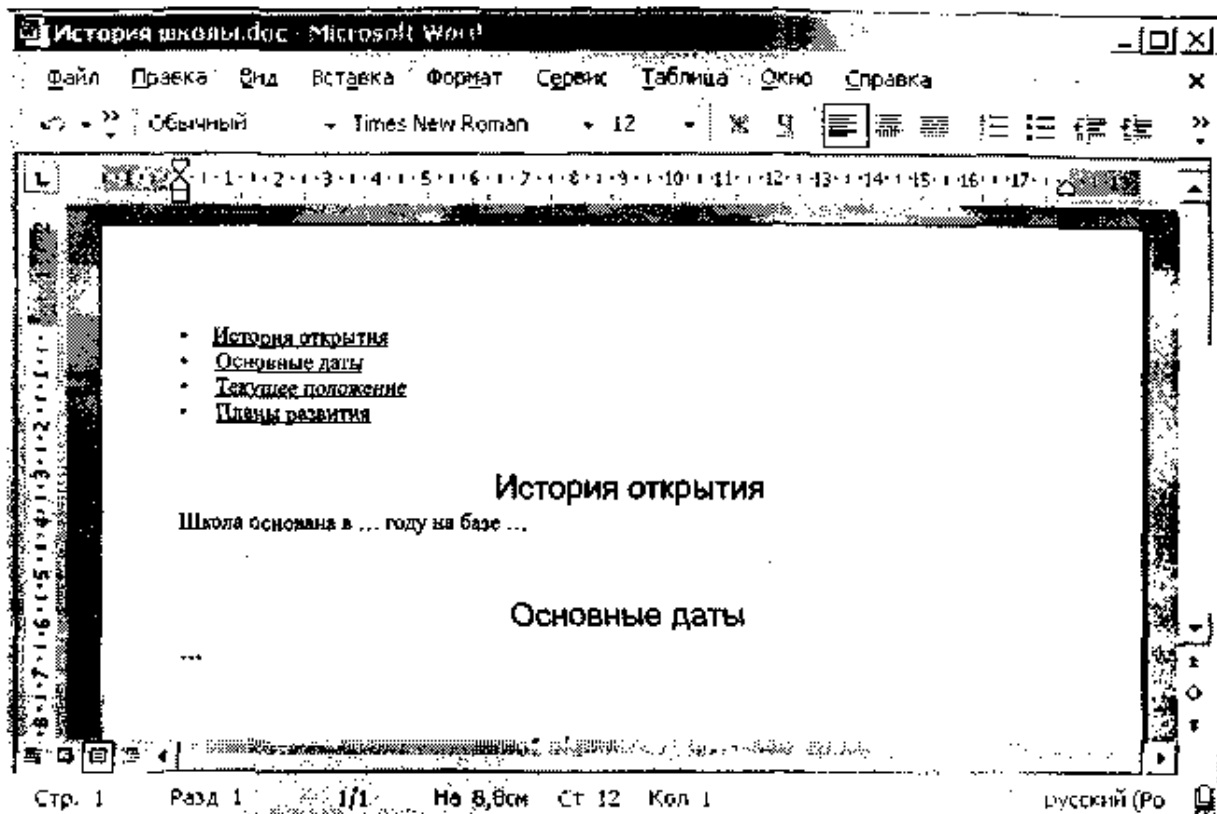


Рис. 9.7



**Упражнение 1.** Создайте в редакторе Microsoft Word текстовый документ предложенной структуры, содержащий сведения по истории школы. Определите гиперссылки и сохраните документ как веб-страницу.



Образец макета документа имеется на прилагаемом диске в папке chapter9 под именем История школы — ссылки.doc.

## Теговое создание гиперссылок

Для создания ссылок и «якорей» в HTML-документе используется тег `<a>`. При создании ссылки используется соответствующий контейнер: обрaмленный тегами `<a>` и `</a>` текст отображается браузером как ссылка. Обрамлять можно практически любой текст с сохранением его форматирования, а также другие элементы, явно отображаемые в документе (разумеется, невозможно сделать ссылкой невидимый текст, например, комментарий).

Чтобы создать текстовую ссылку, HTML-код может выглядеть следующим образом (текст ссылки будет полужирным и курсивным):

```
<a><b><i>Текстовая ссылка</i></b></a>
```

Чтобы создать графическую ссылку (из рисунка), HTML-код может выглядеть следующим образом:

```
<a></a>
```

Важным атрибутом тега `<a>` является атрибут `href`. Он определяет URL документа, который будет загружен при переходе по этой ссылке. Если, помимо имени HTML-документа, необходимо определить «якорь» внутри него, то имя последнего добавляется к URL после знака «#».

Ссылка может быть абсолютной, относительной и в текущем документе. Абсолютная ссылка требует указания полного URL для ресурса, на который она ссылается, относительная — указания URL относительно текущего документа (в пределах сайта рекомендуется именно такой тип ссылок, это облегчает перенос и изменение сайта). Ссылка же в пределах документа может состоять только из имени «якоря» и предваряющего его знака «#».

Другим важным атрибутом является атрибут `name`. Он определяет, что данный тег является «якорем» и задает его имя. Имя якоря должно быть уникальным в пределах документа.

Чтобы создать текстовую ссылку на определенный «якорь», HTML-код может выглядеть следующим образом:

```
<a href="doc2.htm#label1">Якорь label1 документа doc2.htm</a>
```

Соответственно, в самом документе «якорь» должен быть определен следующим образом:

```
<a name="label1"></a>Текст документа doc2.htm
```

Ссылка может одновременно быть и якорем. Для этого необходимо определить в ней оба атрибута: `href` и `name`. Вложенность ссылок не допускается.



**Упражнение 2.** Создайте HTML-документ согласно заданию из упражнения 1. Определите «якоря» и гиперссылки. Сохраните документ.



Образец документа имеется на прилагаемом диске в папке chapter9 под именем picture7.htm.

## Управление фреймами

Ссылки можно использовать для управления фреймами. Атрибут `target` тега `<a>` определяет имя фрейма, в котором должен быть открыт документ, на который указывает ссылка.

Для именования фреймов используется атрибут `name` тега `<frame>`. Существует также несколько зарезервированных имен фреймов:

- `_blank` — открыть документ в новом окне браузера;
- `_self` — открыть документ в том же фрейме, в котором определена ссылка;
- `_parent` — открыть документ в родительском фрейме (для вложенных фреймов);
- `_top` — открыть документ в текущем окне браузера (с удалением фреймовой структуры), т. е. занять все окно.

Все это можно широко использовать для создания меню в одних фреймах и вывода различной информации в других. Если у большей части ссылок атрибут `target` указывает на другие фреймы, то можно использовать тег `<base>` с атрибутом `target`, где и задать имя фрейма, в котором должны открываться ссылки. Тег `<base>` определяется в разделе `<head>`.



**Упражнение 3.** Откройте файлы упражнения 3 из главы 8. Определите во втором фрейме ссылки, которые будут открываться во фрейме 3. Сохраните фрейм 2 под именем `menu_a.htm`, а файл структуры — под именем `index_fr_nb_a.htm`.



Образцы документов имеются на прилагаемом диске в папке chapter9 под соответствующими именами.



## Копирование сайта на сервер

Рассмотрим процесс копирования сайта на удаленный сервер через ftp при использовании файлового менеджера FAR (версия 1.70). Предварительно нужно позаботиться об открытии ftp-доступа к серверу и создать пользователя.

Вызовем панель дисков (Alt + F1 или Alt + F2) и выберем в меню пункт ftp (рис. 9.8).

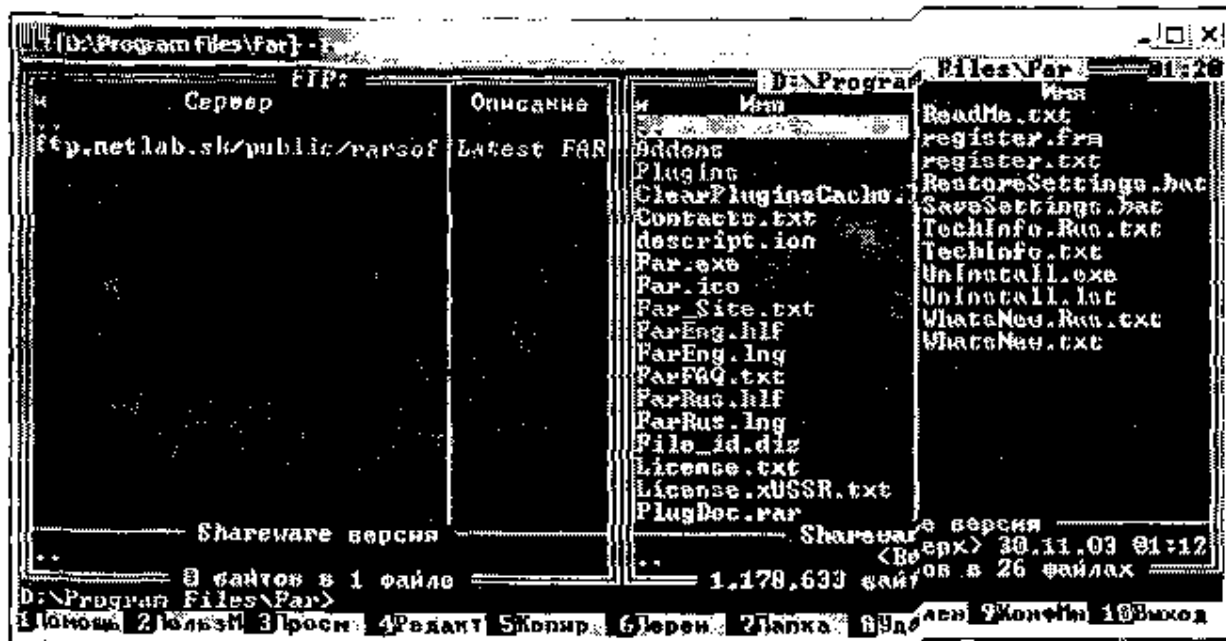


Рис. 9.8

Создадим новое подключение: нажмем комбинацию клавиш Shift + F4 и введем параметры для подключения к серверу (рис. 9.9) (назначение этих элементов и параметров можно узнать в справочной системе менеджера). В большинстве случаев достаточно определить имя сервера и установить флажок Спрашивать пароль непосредственно перед подключением. Сохраним введенное описание.

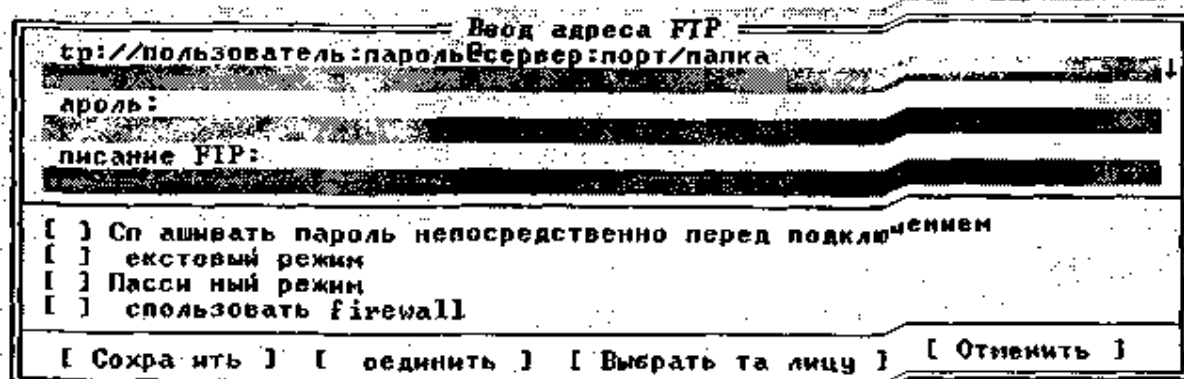


Рис. 9.9

При подключении запрашивается имя пользователя и пароль (рис. 9.10). Введем их и нажмем клавишу **Enter**.

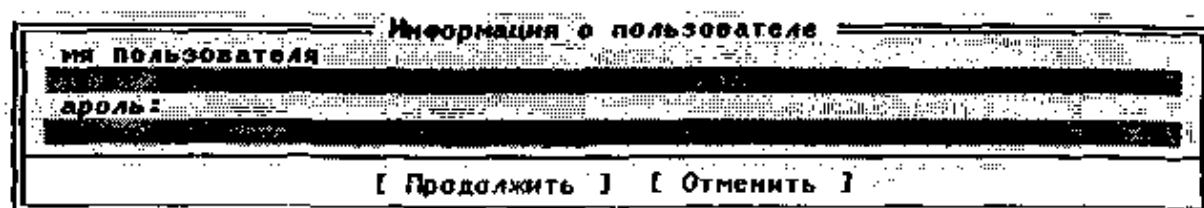


Рис. 9.10

Дальнейшие операции аналогичны обычному копированию файлов с одного диска на другой. Необходимо перейти в каталог, который является корнем сайта, и скопировать туда все требуемые файлы и подкаталоги.

## Тестирование сайта

Тестирование сайта предполагает выполнение следующих действий:

- После завершения кодирования (создания веб-страницы редактором), необходимо проверить правильность ее отображения в браузере. Желательно осуществить такую проверку в различных версиях наиболее популярных браузеров. Если после просмотра в одном из браузеров вы внесли изменения в код документа, его необходимо снова проверить во всех браузерах.
- Если вы осуществляете перенос файлов на другую систему, необходимо проверить работоспособность всех ссылок и правильность отображения всех страниц. Учтите, что документы могут быть уже сохранены в кеше браузера, поэтому рекомендуется очищать кеш перед проверкой. (Для загрузки страницы с сервера, игнорируя ее копию в кеше браузера, нужно нажать кнопку обновления содержимого окна браузера при нажатой клавише **Ctrl**.)
- Большинство выявляемых при тестировании ошибок — орфографические, поэтому проверяйте текст с самого начала.

Нужно также не забывать, что:

- информация в сети Интернет очень быстро меняется, следовательно, нужно периодически проверять внешние ссылки вашего сайта;
- информацию на сайте необходимо обновлять;
- необходимо отвечать на все электронные письма, которые вам приходят по поводу сайта.



### Коротко о главном

1. Связь документов в сети Интернет осуществляется при помощи гипертекстовых ссылок.
2. Ссылка также может выполнять функции якоря.
3. С помощью ссылок может быть реализован механизм управления фреймами.
4. Скопировать сайт на удаленный сервер можно с помощью ftp-клиента.
5. Необходимо производить тестирование сайта при любом его изменении.



### Вопросы и задания

1. Каким образом осуществляется взаимосвязь документов в Интернете?
2. Прокомментируйте основные этапы создания гиперссылок средствами MS Word.
3. В чем заключается теговое создание гиперссылок?
4. Как можно управлять фреймами посредством ссылок?
5. Назовите последовательность действий при копировании сайта на сервер.
6. Проанализируйте основные действия при тестировании сайта.



### Практическое задание

Скомпонуйте ваш сайт и опубликуйте его на сервере.

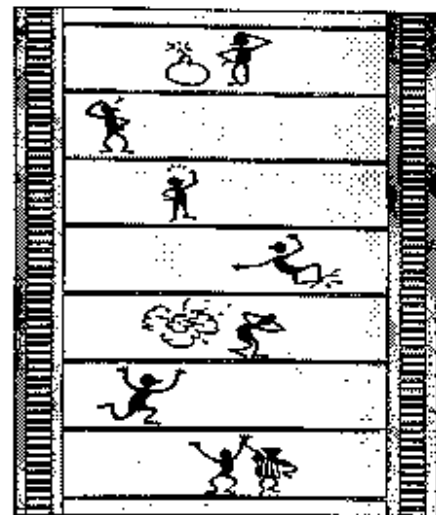


### Темы рефератов

1. Использование ftp-клиента.
2. Терминальные клиенты (удаленный доступ).

## Глава 10

# ОЖИВЛЯЕМ САЙТ



Анимация довольно широко распространена на сайтах в сети Интернет и в основном используется в рекламных целях. Для этого был даже придуман специальный термин — «баннер». Баннер представляет собой картинку (в большинстве случаев анимированную), являющуюся ссылкой, основное назначение которой заключается в привлечении пользователей на соответствующий сайт. С помощью баннерной рекламы можно обеспечить продвижение сайта в сети Интернет. Однако показы баннеров на популярных сайтах в большинстве случаев оплачиваются. Также, помимо непосредственной функции по привлечению пользователей, баннеры могут служить просто рекламой вашей организации (школы). Существуют целые системы обмена баннерами между сайтами.

В процессе создания баннера используется ряд правил, которые повышают его привлекательность для пользователя. К таким правилам можно отнести следующие:

- использование слов, побуждающих к действию, — вроде «нажми здесь» и т. п.;
- использование в тексте безотлагательных фраз — вроде «посети немедленно», «еще есть шанс» и т. п.;
- простота изложения — нет смысла выводить на баннере большие объемы текста и графики, ведь основная цель заключается в том, чтобы пользователь нажал на баннер для перехода на сайт, где информация будет раскрыта детально;
- широкое использование цвета, в большинстве случаев с таким расчетом, чтобы резко выделить баннер на общем фоне страницы, где он размещен;
- создание эффекта интерактивности, которое в большинстве случаев заключается в имитации стандартных элементов диалога (например, кнопок, флажков и т. д.).

Основными используемыми для баннеров размерами (в пикселях) являются: 88×31, 120×60, 100×100, 400×40, 400×50, 468×60. Во многих случаях также вводятся ограничения на максимальный размер файла баннера (не более 20 кб). В связи с этим важно правильно использовать имеющиеся ресурсы. Так, количество кадров анимации сказывается на объеме файла. Однако с точки зрения качества анимации количество кадров, отображаемых в секунду, не играет очень большой роли; главное здесь — дать возможность связывания изображений в единую логическую последовательность. Также нелишней является оптимизация цветовой палитры и качества изображения.

Редактор ImageReady позволяет оптимизировать и подготавливать изображения для сети Интернет, в том числе создавать анимацию. При использовании пакета графических программ Adobe в редакторе PhotoShop можно подготовить различные фазы (шаги) будущей анимации. Возможно также использование специализированных программ для создания анимаций на основе GIF, которые можно найти в сети Интернет. При использовании языка Java, языков сценариев JavaScript и ActiveX и технологии Flash также возможно создание анимированных элементов.

Рассмотрим процесс создания и оптимизации анимированного баннера при использовании пакета программ обработки изображений фирмы Adobe.

## **Создание анимированного изображения**

Процесс создания анимации можно начать с планирования общего ролика и подготовки всех его фрагментов в графическом редакторе. В данном случае можно использовать библиотеки изображений, вырезки из фотографий и другой имеющийся иллюстративный материал. Все это можно найти в виде файлов или отсканировать.

Создадим файлы исходных изображений. При этом необходимо определить размеры изображений, аналогичные размерам баннера. Для этих целей можно использовать графический редактор Adobe PhotoShop. Поместим изображения кадров анимации в отдельный каталог.

Запустим редактор Adobe ImageReady (рис. 10.1).

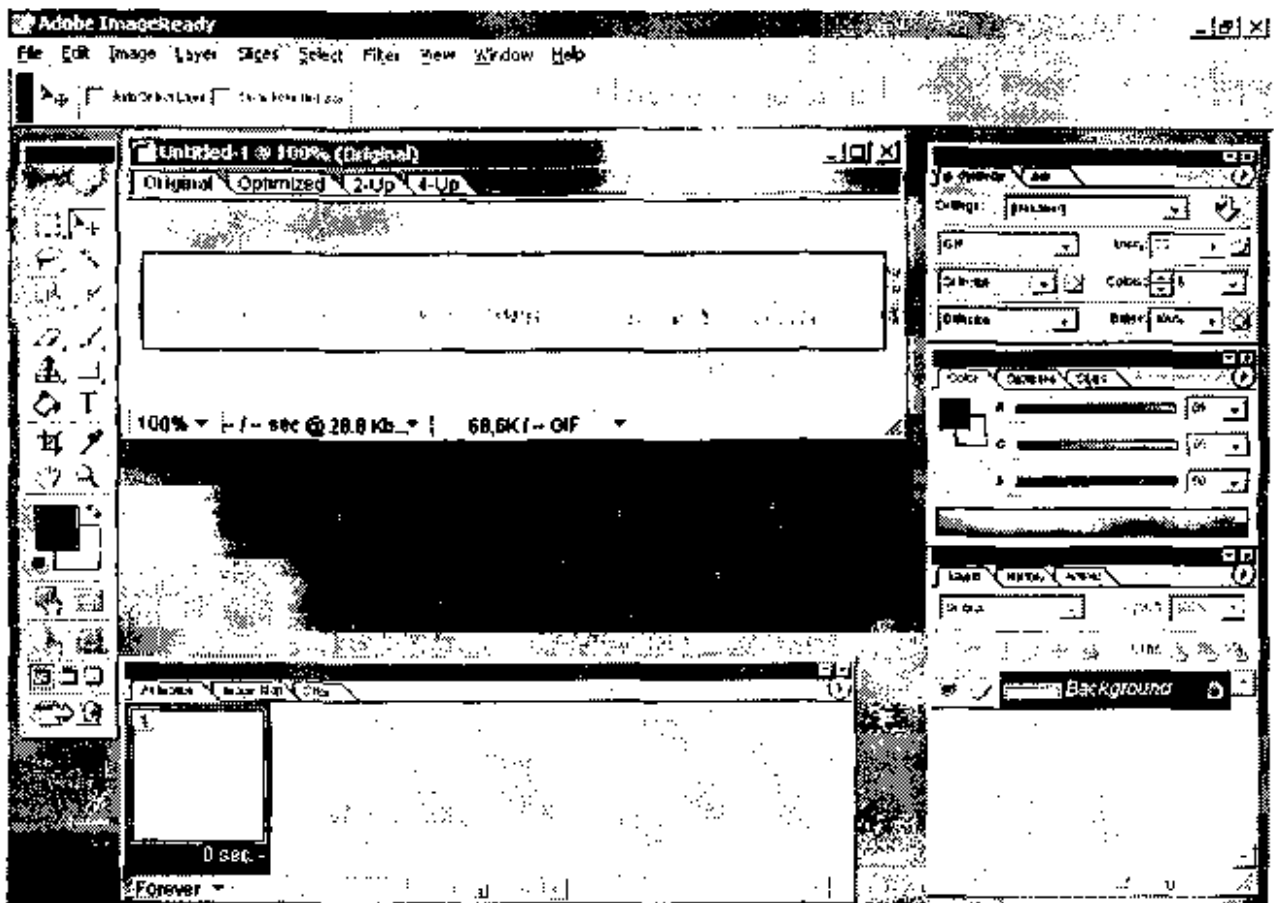


Рис. 10.1

Здесь необходимо вызвать панель **Animation** (Анимация) (рис. 10.2), если она еще не отображена на экране. Для этого необходимо в меню **Window** (Окно) выбрать «флажковый» пункт **Animation** (рядом с ним при последующем раскрытии меню **Window** будет отображаться «галочка»).

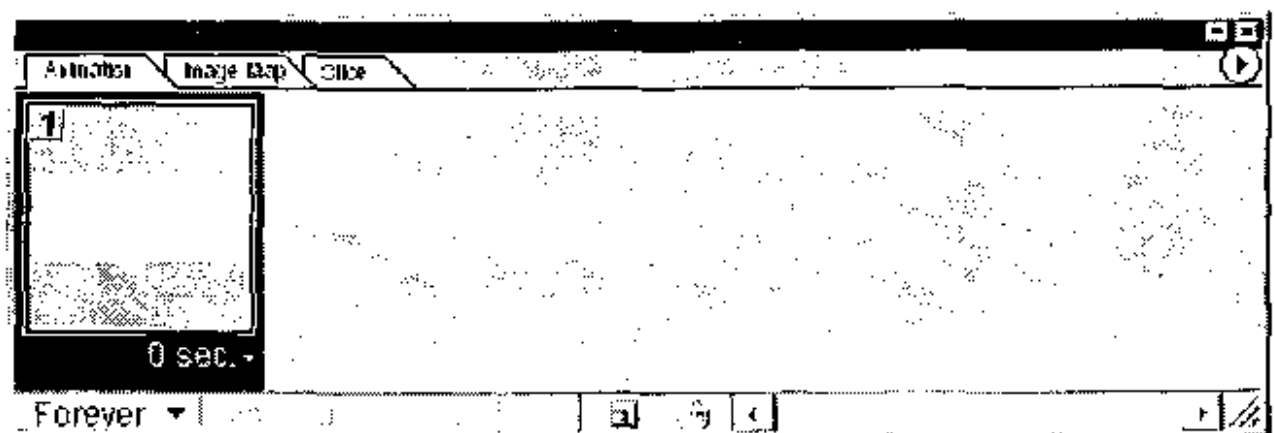



Рис. 10.2

Программа ImageReady также осуществляет импортирование изображений кадров в редактор. Для этого необходимо

в меню **File (Файл)** в подменю **Import (Импорт)** выбрать пункт **Folder as Frames (Папка как фреймы)** и указать каталог, где сохранены файлы кадров анимации.

По завершении импортирования файлов будет открыто окно редактирования изображений, а в панель **Animation** будут добавлены кадры анимации. Если необходимо добавить еще кадры в анимацию, то можно сделать это с помощью кнопки  на панели **Animation** (рис. 10.3), выбрав пункт меню **New frame (Новый кадр)**; аналогично производится удаление и копирование кадров.

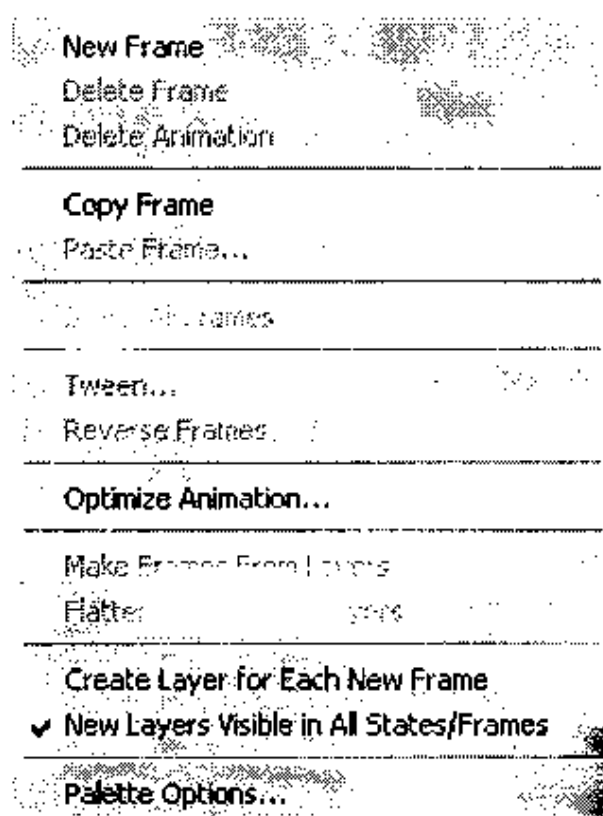


Рис. 10.3

Расставим теперь временные интервалы между моментами отображения фреймов, т. е. определим задержку для показа каждого кадра после его отображения. По умолчанию этот интервал равен нулю. Для его изменения необходимо нажать на текущее значение интервала (оно находится под изображением кадра в панели **Animation**) для соответствующего кадра. Появится меню, с помощью которого можно выбрать один из наиболее часто используемых интервалов или указать свой (рис. 10.4). Зададим значение задержки по 1 с для каждого кадра анимации.

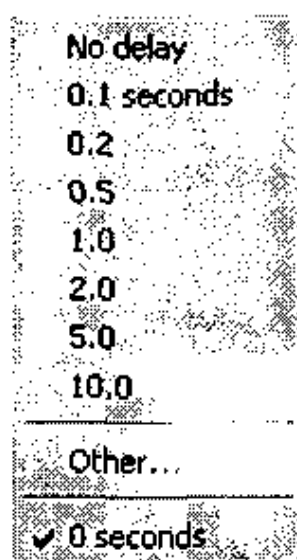



Рис. 10.4

На панели **Animation** присутствуют кнопки управления (рис. 10.5). Левая кнопка (1) используется для определения, сколько раз анимация будет повторяться с начала (количество отображений кадров анимации). После этого анимирование будет остановлено. Возможные значения: **Forever** — бесконечное повторение (анимация никогда не будет прерываться), **Once** — отобразить один раз (после показа последнего кадра анимация будет остановлена), **Other** — заданное количество повторений. Кнопки в виде «медиапанели» справа (2) используются для перемещения между кадрами и управления просмотром (слева направо): переход на первый кадр анимированного изображения, переход на предыдущий кадр, начало/остановка проигрывания кадров, переход на следующий кадр.



Рис. 10.5

 **Упражнение 1.** Создайте кадры для анимированного изображения. Пусть оно состоит из четырех кадров. На первом кадре должно быть представлено название учебного заведения и фраза, побуждающая к действию. На втором — фраза из истории школы, дающая представление о ее



надежности и качестве (простое изложение). На третьем и четвертом — указание, что выделяет эту школу из общего круга подобных организаций, и фразы, побуждающие к действию. Размеры баннера — 468×60. Сохраните кадры в отдельной папке.



Образцы кадров имеются на прилагаемом диске в папке chapter10 в файлах frame1.bmp (psd), frame2.bmp (psd), frame3.bmp (psd) и frame4.bmp (psd).



**Упражнение 2.** Создайте анимацию. Кадры, входящие в анимацию, должны последовательно сменять друг друга с интервалом в 1 с. Сохраните полученный баннер.



Образец баннера имеется на прилагаемом диске в папке chapter10 в файлах banner.psd и banner.gif.

## Оптимизация анимации

Для оптимизации необходимо вызвать панель **Optimize** (Оптимизация) (рис. 10.6), если она еще не отображена на экране. Для этого необходимо в меню **Window** (Окно) выбрать «флажковый» пункт **Optimize**.

На этой панели можно выбрать один из уже определенных шаблонов (типовых параметров изображения) или создать свой шаблон, используя элементы (1) на панели **Optimize** (рис. 10.6).

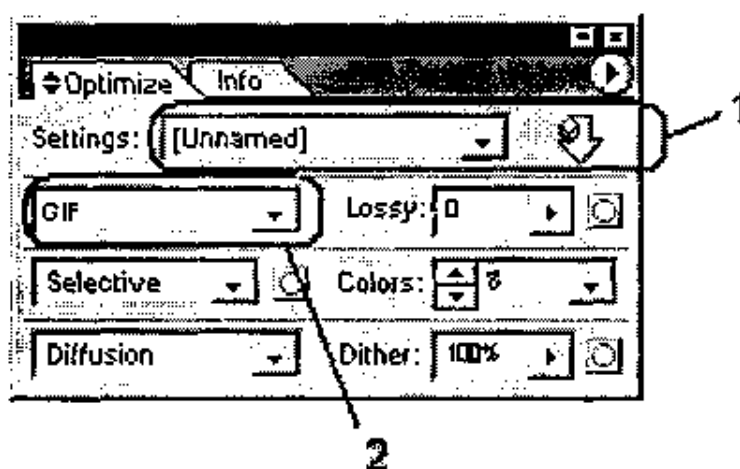


Рис. 10.6

Другой вариант оптимизации связан с определением параметров изображения. Для определения его формата используется

список (2) на рис. 10.6. Остальные элементы панели определяют уровень потерь, количество цветов, используемых в изображении, и смешивание цветов. Смешивание цветов (*dithering*) позволяет создать впечатление большего их количества по сравнению с реальной палитрой в данном изображении. Этот эффект достигается за счет группировки точек с цветами, имеющимися в цветовой палитре, смещение которых дает цвет, близкий к необходимому.

При оптимизации желательно переключить окно изображения в режим 2-Up (рис. 10.7). Это позволит, во-первых, визуально определять качество оптимизации изображения, а во-вторых, контролировать физические параметры изображения и получать ориентировочные данные по новому изображению.

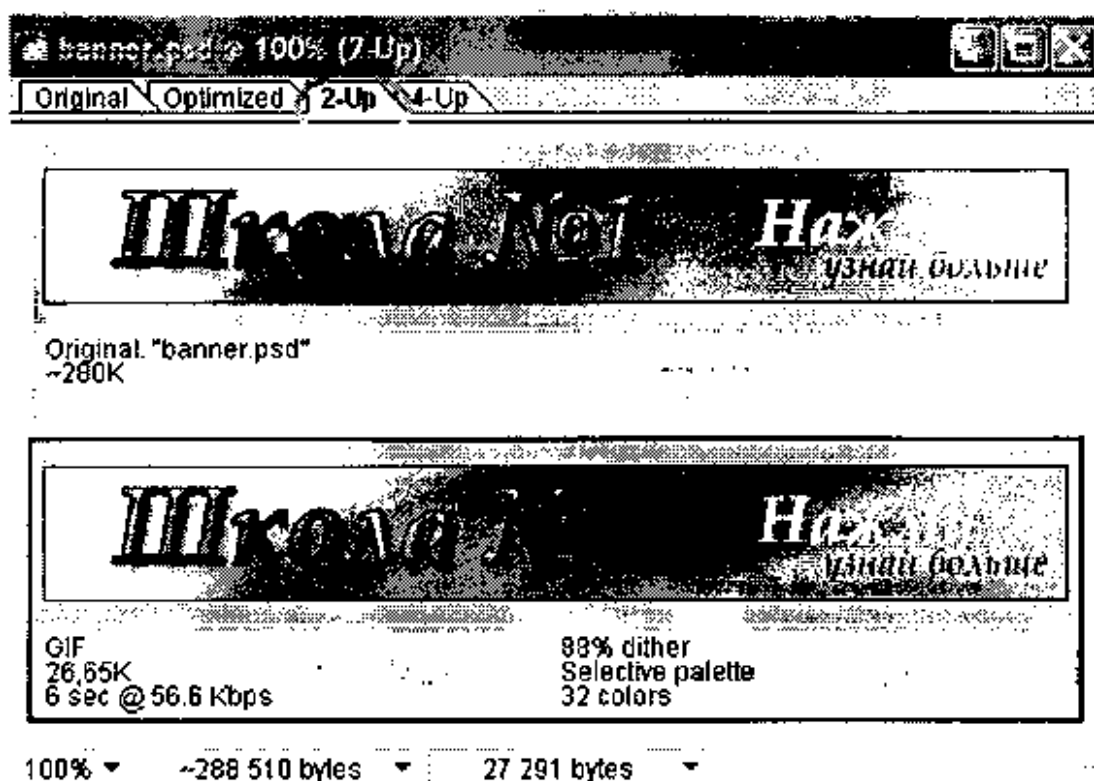


Рис. 10.7

К важным параметрам изображения следует отнести скорость его загрузки браузером. Щелкнув мышью на строке состояния окна изображения (см. рис. 10.7) можно выбрать параметры соединения с сетью Интернет (второй и третий выпадающие списки на рис. 10.8).

По завершении процесса оптимизации изображения сохраним его. Для этого используется пункт **Save Optimized as** в меню **File**.

Original/Optimized File Sizes
Optimized Information
Image Dimensions
Watermark Strength
Undo/Redo Status
Original in Bytes
Optimized in Bytes
Optimized Savings
Size/Download Time (9600 bps Modem)
Size/Download Time (14.4 Kbps Modem)
Size/Download Time (28.8 Kbps Modem)
✓ Size/Download Time (56.6 Kbps Modem/ISDN)
Size/Download Time (128 Kbps Dual ISDN)
Size/Download Time (256 Kbps Cable/DSL)
Size/Download Time (384 Kbps Cable/DSL)
Size/Download Time (512 Kbps Cable/DSL)
Size/Download Time (768 Kbps Cable/DSL)
Size/Download Time (1 Mbps Cable)
Size/Download Time (1.5 Mbps Cable/T1)
Size/Download Time (2 Mbps)

Рис. 10.8



**Упражнение 3.** Оптимизируйте анимированное изображение (файл *banner.psd* в папке *chapter 10*). Добейтесь снижения объема файла до 20 кб при приемлемом качестве изображения. Сохраните полученный баннер.



### Коротко о главном

1. Основное назначение анимированных изображений на сайте — привлечь внимание пользователя. Такие анимации получили название «баннеров».
2. Существует ряд правил, использование которых позволяет сделать ваш баннер привлекательным для пользователей.
3. Существует ряд автоматизированных программных средств, позволяющих создавать несложные анимации.



## Вопросы и задания

1. Что такое баннер?
2. Какие средства анимации позволяют привлечь пользователей к вашему сайту?
3. Какие автоматизированные средства создания анимаций для веб-сайта вы знаете?
4. Прокомментируйте типовую последовательность действий по созданию анимированного изображения (например, с помощью Adobe ImageReady).



## Практическое задание

Создайте собственный баннер.

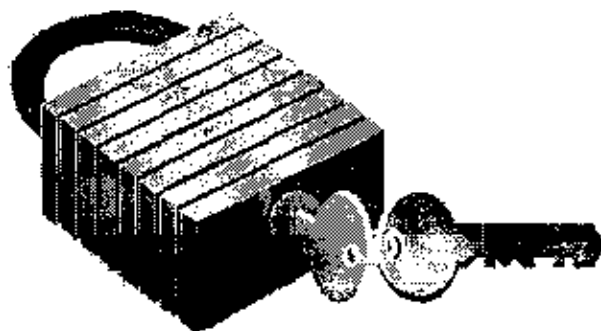


## Темы рефератов

1. Баннерная реклама.
2. Технология Flash.
3. Специализированные программы для создания анимированных изображений.

## Глава 11

# ЗАЩИЩАЕМ САЙТ



Ваш сайт может представлять собой возделенный объект для хакерских атак, целью которых может быть изменение содержимого сайта или вывод его из строя. В любом случае сбой в работе сайта наносит ущерб вашей репутации и приводит к снижению количества посетителей сайта, особенно если в результате атаки ваш сайт становится распространителем вирусов.

## Основные правила обеспечения безопасности сервера

Защиту сайта необходимо начинать с защиты сервера, на котором он опубликован.

Первое, что рекомендуется сделать, — это закрыть все неиспользуемые порты. В данном случае под *портом* понимается канал связи для организации обмена информацией между приложениями, такими как HTTP-сервер и браузер. Если вы используете HTTP-сервер, то можно закрыть практически все порты, за исключением 80-го, через который идет обмен данными по протоколу HTTP. Также можно оставить открытым порт 21 для обеспечения работоспособности протокола FTP, если имеется только удаленный доступ к серверу, — иначе особой необходимости в нем нет.

Далее следует остановить все сервисы (службы компьютера, работающие обычно автоматически в фоновом режиме), в которых отсутствует необходимость. Большинство подобных сервисов обычно и являются местом потенциальных атак (например, сервис удаленного управления реестром). К тому же остановка ненужных сервисов приведет к повышению производительности сервера, так как освободятся ресурсы, используемые этим сервисом. Если необходимость в работе некоторых сервисов все-таки

есть, то можно использовать безопасные соединения (например, с помощью SSL). Особенно это касается сервисов удаленного управления. Не рекомендуется использовать анонимный доступ к сервисам сервера (не сайта!). Особенно хороший эффект дает ограничение количества компьютеров, которым разрешен удаленный доступ. Таким образом можно ограничить возможность удаленного управления с компьютеров, находящихся за пределами вашей подсети.

Следующим этапом может стать настройка профилей пользователей. В данном случае необходимо ограничить круг пользователей, имеющих права на изменение конфигурационных файлов (права администратора), относящихся как к самой операционной системе, так и к HTTP-серверу. Имеет смысл также запустить практически все службы протоколирования (различные журналы событий). Информацию в этих журналах рекомендуется шифровать или сохранять на других компьютерах для определения в случае возникновения сбоя его причин и для невозможности изменения журналов (ведь чтобы изменить журнал на другом компьютере, необходимо будет «взломать» и его).

Рекомендуется также позаботиться об установке специальных сервисов (программ), запускаемых периодически и проверяющих целостность системы. Это может быть обычная проверка контрольных сумм конфигурационных файлов или файлов сайта. Необходимо проверять журналы, хотя бы для выявления возможных атак и принятия решений об изменении параметров защиты (например, смены паролей пользователей). Можно также делать периодическое копирование конфигурационных файлов (особенно!) и страниц сайта на CD. В любом случае рекомендуется сделать резервную копию оригинальной версии сайта.

После пуска сервера и HTTP необходимо периодически устанавливать публикуемые (на проверенных источниках) обновления программного обеспечения как для самого сервера, так и для HTTP.

В целях проверки безопасности сервера можно проверить его специальными программными средствами, выявляющими уязвимости в защите. Информацию об обнаруженных уязвимостях и путях их устранения можно узнать из бюллетеней CIAC (Computer Incident Advisory Capability — <http://www.ciac.org/ciac/>) и Microsoft (<http://www.microsoft.com/security/default.asp>).

В случае обнаружения взлома CIAC рекомендует:

- установить все обновления;
- проверить все профили пользователей;
- проверить все сервисы и порты;
- проверить наличие подозрительных файлов.

## Основные параметры защиты HTTP-сервера Apache

Первое, что необходимо сделать, — это запретить изменения в каталоге, где установлен HTTP-сервер Apache и где находятся конфигурационные файлы, для пользователей, не являющихся администраторами системы. Это необходимо сделать хотя бы для того, чтобы случайно не удалить эти файлы.

Далее необходимо осуществить настройку конфигурационных файлов. Они расположены в подкаталоге `conf` каталога, в котором установлен HTTP-сервер Apache, и имеют расширение `conf`.

Конфигурационный файл `access.conf` используется для определения прав доступа к каталогам и файлам сайта. Однако рекомендуется его не использовать, а параметры доступа определять в файле `httpd.conf`.

Конфигурационный файл `mime.conf` определяет типы расширений файлов.

Конфигурационный файл `srm.conf` определяет общие параметры структуры каталогов. Вместо него, так же как и вместо `access.conf`, рекомендуется использовать для определения соответствующих параметров файл `httpd.conf`.

Рассмотрим параметры, определяемые конфигурационным файлом `httpd.conf`, который задает основные настройки HTTP-сервера.

Этот файл может содержать секции `<Directory>`, `<Location>` и `<Files>`, в которых определяются параметры доступа к каталогам и файлам сайта. Секция `<Directory>` используется для определения параметров каталога, путь которого задан относительно файловой системы. Секция `<Location>` имеет то же назначение, что и секция `<Directory>`, но путь задается относительно каталога установки сайта. А секция `<Files>` определяет параметры доступа к файлу или группе файлов.

Чтобы определить параметры доступа к каталогу `cgi-bin`, секция `<Directory>` должна выглядеть следующим образом:

```
<Directory "C:/Program Files/Apache Group/Apache/cgi-bin">  
  # команды, относящиеся ко всем файлам, каталога cgi-bin  
  # и вложенных в него каталогов  
</Directory>
```

Аналогично для секции `<Location>`:

```
<Location /cgi-bin>  
  # команды, относящиеся ко всем файлам, по адресу  
  # http://servername/cgi-bin/  
  # и вложенных в него каталогов  
</Location>
```

Чтобы определить параметры доступа к файлу `admin.htm`, секция `<Files>` должна выглядеть следующим образом:

```
<Files "admin.htm">  
  # команды, относящиеся к этому файлу  
</Files>
```

Данная секция может находиться внутри секций управления каталогами или быть вынесена в отдельный файл (по умолчанию `.htaccess`), размещаемый в каталоге и определяющий параметры доступа к текущему каталогу и его файлам.

Секция `<Directory>` содержит следующие основные команды:

1. `Options`. Эта команда определяет, какие возможности сервера доступны в текущем каталоге. Среди значений команды возможны следующие:

- `ExecCGI` — разрешить выполнение `cgi`-программ в текущем каталоге и его подкаталогах. С помощью `cgi`-программ возможна обработка форм и создание генерируемых страниц. Однако применять `cgi`-программы следует осторожно, так как они могут использовать практически все возможности операционной системы. При недуманном назначении прав доступа и выполнении подобных программ может возникнуть ситуация, которая приведет к краху как сайта, так и сервера. Открывая доступ разработчикам `cgi`-программ или устанавливая



их, вы должны быть уверены, что они не приведут к сбою, и должны проверять каждую такую программу на предмет наличия уязвимостей в защите;

- **Includes** — разрешить использование SSI (Server Side Includes). Серверные вставки можно сравнить с фреймами, так как со стороны сервера веб-документ представляет собой набор файлов. Однако при передаче его пользователю все файлы собираются в единый документ. Использование SSI таит в себе риск потери производительности сервера. Это вызвано тем, что HTTP-серверу необходимо проверять содержимое выдаваемых пользователю файлов на предмет наличия в них директив (команд) SSI, что приводит к снижению его производительности относительно пользователя. Особенно остро это выражается, если расширение SSI-документов не отличается от обычных HTML-документов. К тому же из SSI-документов можно вызывать cgi-программы, что требует определения соответствующих прав доступа;
- **Indexes** — разрешить выдачу листинга каталога, если в нем не обнаружен индексный файл;
- **MultiViews** — разрешить поддержку различных языков. По умолчанию отключена и особой необходимости в ее включении нет, так как перекодирование происходит «на лету»;
- **All** — разрешить выполнение всех возможностей, кроме MultiViews.

2. **AllowOverride**. Эта команда используется для определения групп команд, используемых в локальных файлах доступа (по умолчанию `.htaccess`), размещаемых в каталогах. Имя файла доступа определяется с помощью команды `AccessFileName`. Среди возможных значений команды `AllowOverride` возможны следующие:

- **None** — файлы доступа игнорируются;
- **All** — возможно выполнение всех возможных команд, размещенных в файлах доступа;
- **AuthConfig** — возможно выполнение команд, связанных с идентификацией пользователей. Например, команды задания имен пользователей, которым разрешен доступ к файлу или каталогу;

- **FileInfo** — возможно выполнение команд, связанных с типами файлов. Например, изменение типов, языков, кодировок;
- **Indexes** — возможно выполнение команд, связанных с индексированием каталогов. Например, добавление описаний каталогов, определение индексных файлов;
- **Limit** — возможно выполнение команд, связанных с доступом с определенных компьютеров. Например, задание IP-адресов, с которых разрешено обращение в данный каталог;
- **Options** — возможно выполнение команды Options секции <Directory>.

Команды групп AuthConfig, FileInfo, Indexes, Limit и Options могут использоваться совместно.



**Упражнение 1.** Определите содержимое секции <Directory> для каталога htdocs. Запретите выполнение в этом каталоге cgi-программ и использование SSI, разрешите создание листинга каталога. Сохраните документ под именем httpd\_part1.txt.



Образец документа имеется на прилагаемом диске в папке chapter11 под именем httpd\_part1.txt.



**Упражнение 2.** Определите содержимое секции <Directory> для каталога htdocs. Запретите использование SSI, разрешите возможность ограничения IP-адресов, используйте идентификацию пользователей. Сохраните документ под именем httpd\_part2.txt.

Секции <Directory>, <Location> и <Files> также могут содержать следующие команды, определяющие права доступа:

1. **Allow.** Команда определяет компьютеры, которые могут обращаться к каталогу или файлу сайта. Имя компьютера может быть задано словесно или в виде IP-адреса. Также можно указывать группы компьютеров по диапазону словесных и IP-адресов. Например, строка Allow 192.168.0.1 определяет, что доступ возможен только для компьютера с IP-адресом 192.168.0.1, а строка Allow 192.168 указывает, что доступ возможен со всех компьютеров, IP-адреса которых начинаются с 192.168. Значение команды All определяет,

что доступ возможен для всех компьютеров. Символьное имя компьютера можно использовать, если команда `HostnameLookups` имеет значение `On` для обеспечения преобразования символьного имени в IP-адрес. Также можно определять доступ по значениям, передаваемым браузером в заголовке запроса.

2. `Deny`. Команда определяет компьютеры, которые не могут обращаться к каталогу или файлу. Синтаксис команды аналогичен синтаксису команды `Allow`.
3. `Order`. Команда определяет параметры использования команд `Allow` и `Deny`:
  - если команда имеет значение `Deny`, `Allow`, то обработка команды `Deny` производится перед командой `Allow`. По умолчанию доступ считается открытым, поэтому доступ получают компьютеры, имена (или адреса) которых отсутствуют в команде `Deny` или присутствуют в команде `Allow`;
  - если команда имеет значение `Allow`, `Deny`, то обработка команды `Allow` производится раньше команды `Deny`. По умолчанию доступ считается закрытым, поэтому компьютеры, которые отсутствуют в списке разрешенных командой `Allow` или присутствуют в списке команды `Deny`, не получают доступ к ресурсу;
  - если команда имеет значение `Mutual-failure`, то доступ получают только те компьютеры, которые присутствуют в списке разрешенных и отсутствуют в списке запрещенных.

**Важное замечание.** Чтобы изменения, внесенные в конфигурационные файлы, вступили в силу, требуется перезапуск HTTP-сервера.



Примеры использования команд `Allow`, `Deny` и `Order` можно посмотреть в конфигурационном файле `httpd.conf` в подкаталоге `conf` каталога установки HTTP-сервера Apache.



**Упражнение 3.** Определите содержимое секции `<Directory>` для подкаталога `private` каталога `htdocs`. Запретите доступ к этому каталогу для всех, кроме компьютеров сети `school1.city1.ru`. Сохраните документ под именем `httpd_part3.txt`.



Образец документа имеется на прилагаемом диске в папке chapter11 под именем httpd\_part3.txt.



## Коротко о главном

1. Сайт может представлять собой объект для хакерских атак, целью которых может быть как изменение содержимого, так и вывод сайта из строя. Поэтому сайт необходимо защищать.
2. Существуют специальные аппаратные и программные средства защиты компьютерной системы, с помощью которых можно обеспечить информационную безопасность сервера с вашим сайтом.



## Вопросы и задания

1. В чем заключается угроза хакерской атаки на ваш сайт?
2. Какие ресурсы вашего сайта требуют защиты от несанкционированного доступа и почему?
3. Перечислите основные правила обеспечения информационной безопасности сервера.
4. Прокомментируйте основные параметры защиты HTTP-сервера Apache.



## Практическое задание

Задайте параметры конфигурационного файла `httpd.conf`. Определите два раздела сайта с соответствующими параметрами: один откройте для всеобщего обозрения, другой — для использования только в пределах подсети школы. Настройте параметры сервера, сделайте резервную копию сайта.



## Темы рефератов

1. Информационные технологии для защиты информации.
2. Хакеры и хакерские атаки.
3. Парольная защита.
4. Компьютерные вирусы и способы борьбы с ними.

## Что дальше?

Область использования сайтов довольно обширна: это и рекламные площадки, и системы поддержки потребителей, и различного рода исследовательские проекты. Все больше людей осознают необходимость создания своего «представительства» в Интернете, ведь спектр услуг по предоставлению информации здесь гораздо шире, чем для многих других средств массовой информации. Это и «фокусировка» пользователя на определенный тип информации, и мощные механизмы поиска, и интерактивные опросные системы, и т. д.

Получив общее представление о процессе проектирования, создания и тестирования сайтов в Интернете, вы сможете перейти к изучению возможностей, связанных с созданием высокоинтерактивных сайтов, позволяющих адаптировать сайт под определенные требования посетителей, вовлекать его в процесс восприятия информации. Учащиеся, изучившие представленный материал, получают базовые навыки проектирования и создания сайтов небольшого объема, изучат основные принципы установки и настройки сервера. Все это послужит базой для дальнейшего, уже осознанного повышения квалификации в области разработки и поддержки сайтов.

В небольшой книге невозможно рассмотреть все аспекты создания сайта. За ее рамками осталось множество вопросов, посвященных созданию динамических сайтов, различным технологиям представления и обработки информации на сервере и пользовательском компьютере, использованию форм, серверных вставок и Macromedia Flash, организации интерактивных элементов, создания систем, подобных Интернет-магазинам и интерфейсным сайтам.

Помните: то, что вы освоили в данном курсе — это только начало!

## Список рекомендуемой литературы

1. Березин С., Раков С. Internet у вас дома. СПб.: ВHV-Санкт-Петербург, 1999.
2. Денисов А. Microsoft Internet Explorer 5: Справочник. СПб.: Питер, 2000.
3. Денисов А. Интернет: Самоучитель. СПб.: Питер, 2000.
4. Заботин Ю., Гроднева С. Интернет в вашем доме. Самоучитель + желтые страницы русского Интернета. М.: Изд-во «Рипол Классик», 2001.
5. Кент П. World Wide Web / Пер с англ. В.Л. Григорьева. М.: Компьютер, ЮНИТИ, 1996.
6. Копыл В. Знакомьтесь: Интернет! М.: Изд-во Харвест, 2003.
7. Крейнак Дж., Хойброкен Дж. Интернет: Энциклопедия. СПб.: Питер, 2000.
8. Новейший самоучитель по работе в Интернете. Под ред. Симоновича С. М.: Десс; Инфорком-Пресс, 2000.
9. Симонович С., Евсеев Г., Мураховский В. Интернет: лаборатория мастера. Практическое руководство по эффективной работе в Интернете. М.: АСТ-ПРЕСС; Инфорком-Пресс, 2000.
10. Тихомиров В. П., Морозов В. П., Хрусталева Е. Ю. Основы гипертекстовой информационной технологии. М.: МЭСИ, 1993.
11. Тихонов А. И. Публикация данных в Internet. / Под ред. В. А. Филикова. М.: Изд-во МЭИ, 2000.
12. Шафран Э. Создание Web-страниц: Самоучитель. СПб.: Питер, 1999.

# Алфавитный указатель

## A

Access.conf 118  
Adobe PhotoShop 76  
Adobe ImageReady 108  
Apache 21  
Archie 13  
ARPAnet 7

## C

CSLIP 9

## D

Dial-Up Access 9  
Directory 30  
DirectoryIndex 30  
DNS 11  
DocumentRoot 30

## E

E-mail 13  
ErrorLog 30

## F

FAR 104  
FTP 12

## G

GIF 71  
Gopher 13

## H

HTML 63  
HTTP 12  
Httpd.conf 118

## I

IAB 8  
IETF 8

IIS 20

IP (Internet Protocol) 7  
IP-адрес 28  
ISOC 8

## J

JPEG 71

## K

Koi-8r 57

## L

Localhost 28

## M

Macromedia Dreamweaver 60  
Microsoft FrontPage 60  
Microsoft Office 60  
Mime.conf 118

## N

Notepad 63

## P

PNG 71  
Port 29  
PPP 9

## R

RGB 67

## S

ServerAdmin 29  
ServerName 29  
ServerRoot 29  
SLIP 9  
Srm.conf 118

**T**

Tag 63  
TCP 10  
TCP/IP 10

**U**

Unix 57  
URL 11

---

**W**

Wais 14  
Windows-1251 57  
WWW 14

**A**

Анимация 107  
«Архивные» сайты 38

**B**

Баннер 107  
Блок BODY 64  
Блок HEAD 64  
Блок HTML 64  
Браузер 14  
Бюллетень CIAC 117

**B**

Веб-мастер 44  
Веб-сервер 19  
Верстка 81  
Вкладка 99

**G**

Гиперссылка 16  
Глубина цвета 73

**D**

Доска объявлений 13  
Доступ через другие сети 10

**Z**

Закладка 96  
Защита сайта 116

**I**

Имена файлов 56

**Интернет 7**

Интернет-магазин 36

**K**

Карта сайта 47  
Контейнер 64  
Корпоративный сайт 38

**H**

Навигация по сайту 46  
Непосредственный доступ 9

**O**

Онлайновый (on-line) браузер 15  
Оптимизация 112  
Оффлайновый (off-line) браузер  
15

**P**

Параметры тегов 67  
Персональный сайт 32  
Поиск людей 13  
Поисковая система 35  
Порт 116  
Протокол 10

**P**

Размер изображения 72  
Разрешение изображения 72

**C**

Сайт 4



Сайт — сборник документов 35	<b>Ф</b>
Сервис 116	Файловая структура 55
Сеть 7	Файловый менеджер FAR 104
Служба новостей 33	Форматирование текста 66
Служба общения 14	Фрейм 88
Стартовая страница 43	
Структура фреймов 88	<b>Х</b>
	Хакерская атака 116
<b>Т</b>	
Таблица 82	<b>Э</b>
Тестирование сайта 105	Электронная почта 13
<b>У</b>	<b>Я</b>
Удаленный доступ 12	«Якорь» 95

Учебное издание

Монахов Михаил Юрьевич,  
Воронин Алексей Александрович

## СОЗДАЕМ ШКОЛЬНЫЙ САЙТ В ИНТЕРНЕТЕ

Элективный курс

Учебное пособие

Редактор Д. Усенков  
Художник Ф. Инфантэ  
Художественный редактор О. Ланко  
Технический редактор Т. Блэнцева  
Корректор Н. Савельева  
Компьютерная верстка Л. Катуркина

Подписано в печать 29.03.05. Бумага офсетная. Формат 60x90  $\frac{1}{16}$ .  
Гарнитура Школьная. Усл. печ. л. 8. Печать офсетная. Тираж 5000 экз. Заказ 2630

Издательство «БИНОМ. Лаборатория знаний»  
Адрес для переписки: 119071, Москва а/я 32.  
Телефон: (095) 955-0398. E-mail: Lbz@aha.ru  
<http://www.Lbz.ru>

Отпечатано с готовых диапозитивов в полиграфической фирме «Полиграфист»  
160001, г. Вологда, ул. Челюскинцев, 3.

Победитель конкурса по созданию учебной литературы нового поколения для средней школы, проводимого НФПК - Национальным фондом подготовки кадров и Министерством образования Российской Федерации



Это учебное пособие поможет вам узнать, как

- проектировать Web-сайт
- создавать собственный Web-сервер
- создавать Web-страницы
- собирать и устанавливать Web-сайт
- защищать свою информацию в Интернете

Вы приобретете способность к

- творческому и инициативному решению проблем
- эффективной работе в неоднородных командах
- профессиональному самоопределению



Компакт-диск содержит результаты выполнения упражнений, демонстрационные плакаты, спецификацию языка HTML

Для информационно-технологического, физико-математического, естественно-научного, гуманитарного профилей

ISBN 5-94774-158-X



9 785947 741582