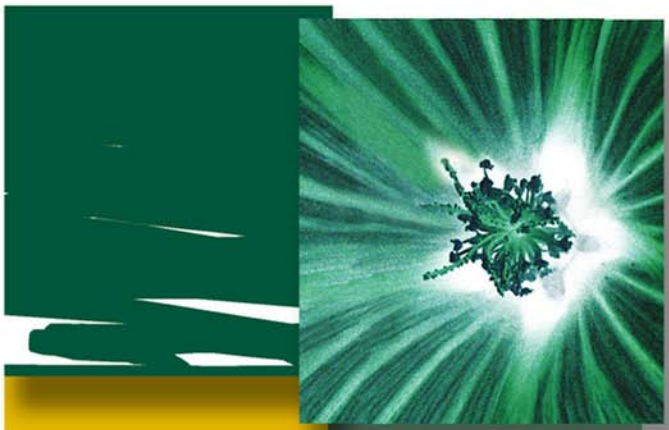
ВЛАДИМИР ДРОНОВ

bhv®

PHP, MySQL и Dreamweaver

разработка интерактивных Web-сайтов

+ ВИДЕОКУРС



НАЧАЛА WEB-ДИЗАЙНА

ВВЕДЕНИЕ В БАЗЫ ДАННЫХ

НАПИСАНИЕ WEB-ПРИЛОЖЕНИЙ НА ЯЗЫКЕ PHP

СОЗДАНИЕ СЕРВЕРНЫХ СТРАНИЦ СРЕДСТВАМИ DREAMWEAVER

ОБУЧЕНИЕ WEB-ПРОГРАММИРОВАНИЮ НА ПРИМЕРАХ

PRO

ПРОФЕССИОНАЛЬНОЕ ПРОГРАММИРОВАНИЕ

+CD

Владимир Дронов

PHP, MySQL и Dreamweaver

разработка интерактивных Web-сайтов

Санкт-Петербург
«БХВ-Петербург»
2007

УДК 681.3.06
ББК 32.973.26-018.2
Д75

Дронов В. А.

Д75 PHP, MySQL и Dreamweaver. Разработка интерактивных Web-сайтов. — СПб.: БХВ-Петербург, 2007. — 480 с.: ил. + Видеокурс (на CD-ROM) — (Профессиональное программирование)

ISBN 978-5-9775-0125-5

Рассмотрены приемы разработки на языке PHP интерактивных Web-сайтов, извлекающих данные из баз MySQL. Для написания простейших страниц используется популярный программный пакет визуального Web-редактора Dreamweaver, попутно дается краткое описание языка HTML. Приводится сжатое описание языка PHP, а также подробно разбираются все сценарии PHP, созданные Dreamweaver, и анализируется их работа. Изложены принципы написания специализированных Web-страниц без использования Dreamweaver. В результате читатель создает полностью работоспособный и весьма развитый блог — интернет-дневник. Прилагаемый CD содержит видеокурс по основам работы в Adobe Dreamweaver CS3.

Для Web-программистов

УДК 681.3.06
ББК 32.973.26-018.2

Группа подготовки издания:

Главный редактор	*Екатерина Кондукова*
Зам. главного редактора	*Евгений Рыбаков*
Зав. редакцией	*Григорий Добин*
Редактор	*Ирина Иноземцева*
Компьютерная верстка	*Ольги Сергиенко*
Корректор	*Зинаида Дмитриева*
Дизайн серии	*Инны Тачиной*
Оформление обложки	*Елены Беляевой*
Зав. производством	*Николай Тверских*

Лицензия ИД № 02429 от 24.07.00. Подписано в печать 28.08.07.
Формат $70\times100^{1}/_{16}$. Печать офсетная. Усл. печ. л. 38,7.
Тираж 3000 экз. Заказ №
"БХВ-Петербург", 194354, Санкт-Петербург, ул. Есенина, 5Б.

Санитарно-эпидемиологическое заключение на продукцию
№ 77.99.02.953.Д.006421.11.04 от 11.11.2004 г. выдано Федеральной службой
по надзору в сфере защиты прав потребителей и благополучия человека.

Отпечатано с готовых диапозитивов
в ГУП "Типография "Наука"
199034, Санкт-Петербург, 9 линия, 12

ISBN 978-5-9775-0125-5

Оглавление

Введение

"Вот еще одна книга о PHP и MySQL... Боже, сколько их уже издано! Так нет, нужно опять тратить бумагу, чтобы рассказать о том же самом".

Так может сказать какой-нибудь знаток компьютерных и интернет-технологий, разглядывая разноцветные книжные обложки на прилавке магазина. Действительно, книг о создании Web-сайтов с использованием серверных Web-страниц PHP и баз данных MySQL сейчас написано очень (возможно, даже слишком) много. "PHP!", "MySQL!", "PHP и MySQL!", "MySQL и PHP!!!" — кричат обложки. Стоит ли писать еще одну книгу на ту же самую тему?

Вопрос не в том, *стоит ли* писать. И даже не в том, *о чем* писать. А *как*! Да-да, именно как писать подобные книги!

Но стоп, давайте обо всем по порядку. Потенциальный читатель наверняка приготовил уйму вопросов, и автору придется на них отвечать. А отвечать лучше всего не торопясь, спокойно, без лишнего шума. Итак...

О чем вообще идет речь?

Действительно, что это за штуки такие — PHP, MySQL и Dreamweaver? И с чем их едят?

Начнем с самого начала.

MySQL — это сервер данных. Особая программа, позволяющая хранить упорядоченные данные в особых структурах, называемых базами данных. Например, в такую базу можно записать инвентарную книгу, список работников, экзаменационные ведомости, перечень книг, хранящихся в библиотеке, и многое другое. И не просто записать и сохранить, а еще и получать эти данные по запросу. И не просто получать, а быстро, без особого труда и только те данные, которые действительно нужны.

PHP — это платформа для создания серверных Web-страниц. Грубо говоря, с ее помощью можно писать программы, которые по запросу посетителя сайта создают и отправляют ему Web-страницы. Эти страницы могут содержать данные, хранящиеся в базе данных, в том числе и MySQL.

MySQL и PHP впечатляют даже по отдельности. А уж вместе — это просто мечта! Например, серверная программа, написанная на PHP и являющаяся частью сайта интернет-магазина, может извлекать из базы данных MySQL список товаров и формировать из него красивую Web-страницу. А другая серверная программа (другая часть того же интернет-магазина) спрашивает посетителя, какой товар он хочет заказать и по какому адресу этот товар нужно выслать, и сама записывает в ту же базу данных сведения о новом заказе.

Неудивительно, что PHP и MySQL так популярны. Мало того, что это весьма мощные программы, с помощью которых можно реализовать практически все, что угодно. Они еще прекрасно работают в связке и — внимание, сторонники использования легальных программ! — абсолютно бесплатны. Находка для создателя некоммерческого Web-сайта с большим информационным наполнением. Да и вполне коммерческий интернет-магазин они тоже "потянут".

Dreamweaver — мощнейший пакет для создания Web-страниц, обычных (статичных) и серверных, в том числе и написанных на языке PHP. С его помощью мы можем создавать простейшие Web-страницы PHP, вообще не зная этого языка и принципов программирования на нем. А возможностей у Dreamweaver столько, что о нем можно написать несколько толстенных книг, и все равно какие-то секреты наверняка останутся нераскрытыми.

Dreamweaver был создан фирмой Macromedia, ныне принадлежит корпорации Adobe и — увы!.. — является платным. Однако на официальном сайте Adobe (**http://www.adobe.com**) доступна демонстрационная версия, работающая 30 дней. Так что все желающие могут попробовать его в действии, так сказать, "повертеть в руках".

Но знаток, собаку съевший на PHP и MySQL, недовольно кривит губы. "Прекрасно..." — бормочет он. — "PHP и MySQL — это понятно. Но при чем тут Dreamweaver?"

Dreamweaver: "Делай, как я!"

А вот здесь мы вплотную подошли к ответу на вопрос — как писать. Вообще, это главный вопрос, на который автор любой книги (и не только по компьютерной тематике) всегда должен дать ответ.

Писать книги о PHP и MySQL можно по-разному. Можно начать с описания языка, продолжить примерами простейших страничек, а закончить описанием работы с базами данных MySQL. Можно сразу начать с описания работы с базами данных, затем обсудить тонкости составления запросов к ним, а под конец забраться в такие дебри, что читателей дрожь пробьет. А еще можно почти полностью сосредоточиться на MySQL, а PHP описать поверхностно, только "чтобы было".

А можно и совсем по-другому — вот так: "Значит, при чем здесь Dreamweaver? А вот при чем..."

Ранее упоминалось, что Dreamweaver может сам создавать простейшие серверные Web-страницы PHP, извлекающие данные из баз MySQL. Так вот, мы начнем с того, что будем пользоваться для создания своих первых страничек услугами Dreamweaver и внимательно разбирать код PHP, который он создаст. В этом нам поможет руководство по PHP, которое можно найти на официальном сайте этой платформы — **http://www.php.net**.

Когда мы приобретем кое-какой опыт, то сами начнем писать серверные страницы. В конце концов, возможности Dreamweaver в PHP-программировании не очень велики, многие нужные вещи он просто не сможет для нас сделать. А мы сможем!

Фактически мы будем учиться писать серверные страницы PHP на готовых примерах. Это, пожалуй, лучший способ обучения программированию. Автор сам точно так же изучал в свое время PHP — читая код, созданный Dreamweaver. "Делай, как я!" — командует Dreamweaver. — "Учись у меня!" И этому призыву стоит последовать. Хотя бы первое время.

Знаток компьютеров пожимает плечами: "Программирование — это, прежде всего, искусство, и искусству компьютеры не обучены". Что ж, спору нет. Но не будем при этом забывать, что программы создают люди. А ведь именно люди создали искусство...

В качестве примера мы создадим блог — что-то вроде интернет-дневника. Да, собственно, многие знают, что такое блог; классический пример блога — LiveJournal (**http://www.livejournal.com**). Заметки, опубликованные на этом блоге, и комментарии к ним, оставленные посетителями, будем хранить в базе данных MySQL. А извлекать их оттуда будут серверные страницы PHP, которые нам поможет создать Dreamweaver.

Типографские соглашения

В этой книге часто приводятся форматы использования различных выражений PHP и вызова функций. Нам необходимо выучить типографские соглашения, используемые для их написания.

Внимание!

Все эти типографские соглашения применяются автором только при описании формата выражений и функций PHP. В коде примеров они не имеют смысла.

Так, в угловые скобки (<>) заключаются названия параметров или фрагментов кода, которые, в свою очередь, набраны курсивом. В код реального сценария, разумеется, должен быть подставлен реальный параметр или реальный код. Например:

```
if (<условие>) {
```

Здесь вместо подстроки *<условие>* должно быть подставлено реальное условное выражение.

В квадратные скобки ([]) заключаются необязательные фрагменты кода. Например:

```
function <имя функции>([<список формальных параметров>])
```

Здесь *список формальных параметров* может присутствовать, а может и отсутствовать.

Слишком длинные, не помещающиеся на одной строке выражения PHP автор разрывает на несколько строк и в местах разрывов ставит знаки ↳. Например:

```
$days = array("понедельник", "вторник", "среда", "четверг", "пятница",
↳"суббота", "воскресенье");
```

Приведенный выше код разбит на две строки, но должен быть набран в одну. Знаки ↳ при этом должны быть удалены.

Внимание!

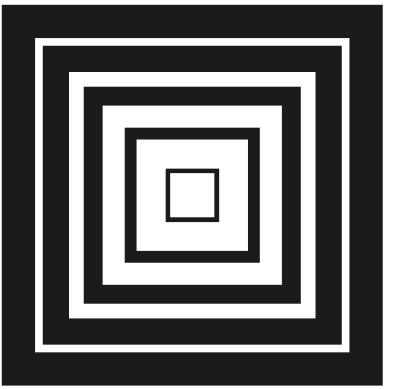
Все приведенные выше типографские соглашения имеют смысл только при описании формата использования выражений и вызова функций PHP. В реальном PHP-коде, сгенерированном Dreamweaver и написанном вручную, используется только знак ↳.

Благодарности

Автор приносит благодарности своим родителям, знакомым и коллегам по работе.

- Губине Наталье Анатольевне, начальнику отдела АСУ Волжского гуманитарного института (г. Волжский Волгоградской обл.), где работает автор, — за понимание и поддержку.

- Всем работникам отдела АСУ — за понимание и поддержку.
- Родителям — за терпение, понимание и поддержку.
- Архангельскому Дмитрию Борисовичу — за дружеское участие.
- Шапошникову Игорю Владимировичу — за содействие.
- Евгению из Волгограда — за фильмы ужасов — лучшее средство для развития чувства юмора.
- Рыбакову Евгению Евгеньевичу, заместителю главного редактора издательства "БХВ-Петербург", — за неоднократные побуждения к работе, без которых автор давно бы обленился.
- Издательству "БХВ-Петербург" — за издание моих книг.
- Всем своим читателям и почитателям — за прекрасные отзывы о моих книгах.
- Всем российским программистам, занятым в разработке MySQL и PHP, — за прекрасные программные продукты.
- Всем, кого я забыл здесь перечислить, — за все хорошее.

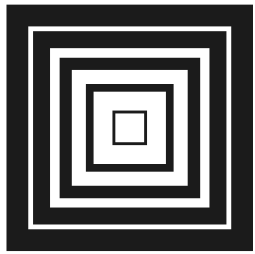


ЧАСТЬ I

Основы Web-дизайна

В первой части мы займемся тем, что создадим свой самый первый Web-сайт, даже не сайт, а его *прототип* — своего рода набросок с минимальной функциональностью. Глядя на этот "набросок", мы сможем оценить внешний вид нашего сайта, решить, подходит он нам или нет, и легко исправить. Ведь — согласитесь! — переделывать небольшой прототип проще, чем уже готовый и работающий сайт.

Но все это будет гораздо позже. А сначала мы пройдем вводный курс интернет-технологий и выясним, как создаются Web-страницы. В конце концов, нам еще нужно познакомиться с самим пакетом Adobe Dreamweaver, с которым мы и будем работать. Разве правильно сразу же приступать к практике, не познакомившись с теорией?



ГЛАВА 1

Современные интернет-технологии

Да, без теории не обойтись, ведь прежде чем приниматься что-то делать, нужно выяснить, зачем, как и почему именно так это "что-то" делается. А иначе у нас ничего толкового не выйдет. Так что давайте выключим компьютер — пусть отдохнет! — и почитаем.

Принципы работы Интернета

Сначала мы поговорим о том, что такое Интернет и как он работает — рассмотрим некоторые общие вопросы.

Что такое Интернет. Сервисы Интернета

И первый же вопрос, на который нам нужно получить ответ, — это что, собственно, такое Интернет и что он может нам дать. А разбором принципов его работы мы займемся потом.

Итак, *Интернет* — это Всемирная компьютерная сеть. (Конечно, это всем известно, но ведь автор должен дать определение.) Ее, кстати, так часто и называют: Всемирная сеть, или даже просто Сеть с большой буквы. Протянутая по всему земному шару паутина медных проводов, волоконно-оптических линий и радиоканалов, связывающих друг с другом многочисленные компьютеры, — вот что такое Интернет. Разумеется, все это подчиняется общим стандартам (о которых мы поговорим далее), а иначе эта суперсеть просто не будет работать.

Если же быть совсем точным, то Интернет — это не единая сеть, а совокупность более мелких сетей, связанных друг с другом общими каналами и стандартами. Таких сетей превеликое множество: огромные территориальные сети, раскинувшиеся на целые области, штаты и государства, и ведомственные

сети, объединяющие родственные организации, и локальные компьютерные сети отдельных организаций, и так называемые кампусные сети — сети, объединяющие компьютеры одного или нескольких близлежащих районов города. Благодаря проложенным между ними каналам высокоскоростной связи, они составляют единое целое, имя которому Интернет.

Даже частные пользователи, подключающиеся к Интернету через аналоговый или цифровой модем, по выделенной линии или поддерживающему такую возможность сотовому телефону, тоже по сути дела являются частью Сети. Так что когда мы включаем наш модем и дозваниваемся до нашего *интернет-провайдера* (организации, предоставляющей доступ в Интернет), то приобщаемся к единому целому. А что, разве это не повод для законной гордости?

Сеть Интернет имеет одну замечательную особенность — она очень устойчива к сбоям. Так, если где-то порвется провод, соединяющий два участка (или, как говорят профессионалы-сетевики, сегмента) Сети, мы этого не заметим. А все потому, что данные, которые мы запрашиваем, пойдут в этом случае по другому каналу. Специалисты говорят, что Интернет децентрализован — он не имеет единого центра, из которого ведется управление пересылкой данных, поэтому в случае аварии автоматически переконфигурируется и продолжит нормально работать.

Еще одна замечательная особенность Интернета — его, так сказать, глобальность. Не вставая из-за компьютера, мы можем совершить путешествие по всему миру: побывать в США, Австралии, Германии, Зимбабве, Огненной Земле и даже в Антарктиде (да, и туда протянулись вездесущие провода!). Для этого нужно всего лишь набрать нужный нам интернет-адрес.

Итак, что такое Интернет, мы выяснили. Теперь совершим небольшое путешествие в прошлое и посмотрим, как все начиналось.

Интернет имеет достаточно долгую историю. Он появился еще в первой половине 70-х годов XX века, когда Министерство обороны США финансировало проект создания компьютерной сети, устойчивой к сбоям. Разумеется, создавалась эта сеть для нужд обороны, да и название имела другое — *ARPANET*. Позднее, в начале 80-х, эта сеть отошла в ведение ученых, а военные приступили к созданию другой сети, которой пользуются до сих пор. И в то же самое время ARPANET был переименован в Internet, или, по-русски, Интернет.

Первоначально, еще во времена ARPANET, эта сеть использовалась для пересылки электронной почты и файлов. Web-странички, ради которых мы, в основном, и путешествуем по Сети, появились только в конце 80-х. Именно тогда Интернет и "пошел в народ", перестав быть сетью ученых и превратившись в сеть для всех.

В Россию Интернет официально пришел в 1991 году, но популярность среди широких масс компьютерщиков приобрел только в середине 90-х годов XX века. В настоящее время в России количество пользователей Интернета исчисляется десятками миллионов.

Раз уж мы заговорили об услугах, предоставляемых Интернетом, или, как говорят профессионалы, *сервисах* Интернета, то давайте узнаем о них побольше. В конце концов, нам ими пользоваться...

Итак, самый старый и самый популярный до сих пор сервис Интернета — это электронная почта (e-mail). Ежедневно в мире отправляются и принимаются сотни миллионов электронных писем, и это количество в будущем будет только увеличиваться. В самом деле, электронная почта доступна, удобна, быстра и бесплатна, в отличие от почты "бумажной", которую пользователи Интернета уже успели презрительно прозвать "улиточной" (от англ. *"snail mail"*). Конечно, доступность, удобство, быстрота и бесплатность имеют оборотную сторону, вроде "спама" — несанкционированных рекламных рассылок, но с ними вполне можно бороться.

Еще один сервис Интернета, почти такой же старый, как почта, — это пересылка файлов. Он называется *FTP* (File Transfer Protocol, протокол передачи файлов). Сейчас FTP отошел на второй план, уступив место более новым сервисам, но все еще весьма активно используется.

Третий сервис Интернета, который мы подробно рассмотрим, — это Всемирная паутина, или *WWW* (World Wide Web), или просто *Web*. Это и есть те самые Web-страницы и Web-сайты, которые мы просматриваем в *Web-обозревателе* (программе для просмотра Web-страниц). Пожалуй, это самый впечатляющий и самый востребованный сервис, собственно, и приведший к тому, что Интернет и "пошел в народ". Вот о нем-то и пойдет речь в этой книге.

Осталось упомянуть о нескольких сервисах Интернета, уже (или еще) не имеющих такой широкой популярности.

- *Новости*, или *USENET*. Чем-то этот сервис похож на электронную почту: пользователь пишет письмо в так называемую *группу новостей* — своего рода электронную доску объявлений. Другие пользователи читают это письмо и пишут ответы, которые помещаются на эту же самую "доску". Совокупность первоначального письма и ответов на него образуют *обсуждение* (по-англ. — *thread*), участие в котором может принять любой пользователь, подписавшийся на эту группу новостей. Когда-то сервис новостей был весьма популярен, но сейчас, после появления различных Web-форумов, прозябает на задворках Интернета.
- *Потоковое вещание*. Это своего рода теле- и радиовещание через Интернет появилось совсем недавно, несколько лет назад, но быстро набирает

популярность. К сожалению, чтобы слушать интернет-радио, и в особенности смотреть интернет-телевидение, нужен достаточно быстрый канал связи, но такие каналы сейчас есть у многих, даже домашних пользователей Интернета.

- *Интернет-пейджеры*. Этот сервис также похож на электронную почту: пользователи пересылают друг другу короткие "записки" по аналогии с обычным пейджером. Интернет-пейджеры работают, как правило, быстрее, чем обычная электронная почта, и временами создают иллюзию непосредственного общения. В качестве примера можно вспомнить популярнейший ICQ и его менее известных "коллег": Miranda, Odigo и пр.
- *Чаты* (от англ. *chat* — "болтовня"). Это своего рода "разговор" через Интернет, еще более напоминающий непосредственное общение. Пользователь набирает на клавиатуре текст, который в мгновение ока пересылается его собеседнику или собеседникам. По популярности чаты превосходят интернет-пейджеры и приближаются к WWW.

Ну, вот и все. Совсем уж малоизвестные и узкоспециализированные сервисы Интернета мы рассматривать не будем. В конце концов, сведения о них (как и о многом другом) можно найти в том же самом Интернете. А тема этой книги совсем иная.

Клиенты и серверы

Продолжим наше путешествие в электронные дебри Интернета. На этот раз речь пойдет о двух разновидностях программ, с помощью которых предоставляются интернет-услуги.

В самом деле, каким образом мы пользуется всем тем богатством, что дает нам Всемирная сеть? С помощью особых программ! Такие программы делятся на две принципиально разные категории, и мы сейчас о них поговорим.

Программы, относящиеся к первой категории, взаимодействуют непосредственно с пользователями Интернета и помогают им получать различную информацию: электронные письма, Web-страницы, сообщения интернет-пейджеров, чатов и пр. Это Web-обозреватели, клиенты электронной почты, чатов, интернет-пейджеры — все те, с которыми мы имеем дело на своих компьютерах. Такие программы называются программами-*клиентами* (а компьютеры, на которых они работают, — наши компьютеры! — *клиентскими*).

Информация, с которой мы работаем посредством программ-клиентов, все эти Web-сайты, письма, звуковые и видеофайлы, хранится на других компьютерах — *серверных*. За выдачу ее клиентским программам, а значит, и нам, отвечают программы, относящиеся ко второй категории, — *серверы*. Для каждого сервиса (и для каждой соответствующей им разновидности программ-

клиентов) Интернета существует свой класс серверов: Web-серверы, серверы электронной почты, чата, интернет-пейджеров, потокового вещания и пр.

Замечание

Очень часто понятие "сервер" распространяется и на серверный компьютер, и на саму программу-сервер. Это, вообще-то, не совсем правильно, т. к. на одном серверном компьютере может быть установлено несколько разных программ-серверов, но вошло в практику.

Теперь поговорим подробнее о том, как же клиенты взаимодействуют с серверами. Причем процессы приема и отправки данных мы рассмотрим отдельно.

Процесс получения информации от сервера клиентами состоит из пяти шагов.

1. Пользователь запрашивает с помощью программы-клиента некую информацию.
2. Клиент устанавливает *соединение* (своего рода воображаемый канал связи) с сервером и посылает тому особый информационный блок, называемый *клиентским запросом*. Структура этого запроса жестко стандартизирована, чтобы сервер его понял.
3. Сервер принимает запрос и расшифровывает его.
4. Сервер извлекает нужный клиенту файл или фрагмент данных, записанных в файле, и посылает его клиенту в виде другого информационного блока — *серверного ответа*. Если же нужных данных нет, или сервер почему-то не смог обработать клиентский запрос, он возвращает в составе ответа *сообщение об ошибке* — особый информационный блок, содержащий описание возникшей ошибки. Разумеется, и серверный ответ, и сообщение об ошибке также жестко стандартизированы.
5. Клиент получает ответ от сервера, расшифровывает его и выдает полученную информацию пользователю. Если получено сообщение об ошибке, клиент уведомляет об этом пользователя либо предпринимает какие-то действия самостоятельно. После принятия ответа от сервера клиент разрывает соединение с ним.

Процесс отправки клиентом данных серверу также состоит из пяти шагов.

1. Пользователь каким-то образом передает программе-клиенту отправляемую информацию.
2. Клиент устанавливает соединение с сервером и посылает тому отправляемую информацию в составе клиентского запроса. При этом отправляемая информация, как правило, особым образом шифруется.

3. Сервер принимает запрос, расшифровывает его и извлекает отправленную информацию.
4. Сервер записывает отправленную клиентом информацию в файл или каким-то образом ее обрабатывает. После этого в случае успешной записи он отправляет клиенту в составе ответа так называемое *подтверждение* — особый информационный блок, сообщающий о том, что все прошло нормально. В случае неудачи отправляется сообщение об ошибке.
5. Клиент получает ответ от сервера, расшифровывает его и уведомляет пользователя об успешной или неуспешной отправке данных либо предпринимает какие-то действия самостоятельно. После принятия ответа от сервера клиент разрывает соединение с ним.

Весь процесс "общения" клиента и сервера, начиная с отправки клиентом запроса и заканчивая принятием им ответа от сервера, называется *сеансом*.

Ранее было сказано, что перед отправкой клиентом запроса серверу, т. е. перед началом сеанса, клиент должен установить соединение с сервером. Так вот, это соединение существует ровно столько времени, сколько требуется для сеанса, и поэтому называется *сеансовым*, или *временным*.

Каждое соединение требует компьютерных ресурсов. Серверу нужно хранить в памяти сведения о клиенте, установившем это соединение, а при большом количестве таких соединений (а на сильно загруженных корпоративных серверах так часто и бывает) памяти расходуется очень много. Поэтому сеансовые соединения имеют большое преимущество — они длятся очень недолго, ровно столько, сколько нужно для отправки серверного ответа, после чего разрываются, и отведенная им компьютерная память автоматически освобождается.

Но сеансовые соединения имеют огромнейший недостаток — с их помощью начать сеанс обмена данными может только клиент. А в случае, например, чата или интернет-пейджера серверу самому может понадобиться начать сеанс, чтобы отправить новое сообщение "отдыхающему" клиенту. В этом случае используется *постоянное*, или *многосеансовое*, подключение с другим сценарием работы.

1. Клиент устанавливает постоянное соединение с сервером.
2. Клиент и сервер ожидают прихода отправляемых данных.
3. Если данные пришли, клиент или сервер выполняют сеанс обмена.
4. Если клиент еще не разорвал соединение, выполняется переход к пункту 2.

Здесь нужно обратить внимание на то, что соединение устанавливается и завершается только клиентом. Сервер установить соединение с клиентом не может.

В отличие от клиента, "имеющего дело" с одним-единственным пользователем, сервер работает сразу с множеством пользователей, причем одновременно. Данные о соединениях, данные, пересылаемые клиентам и принимаемые от клиентов, — все это активно потребляет системные ресурсы компьютера, и чем больше соединений и данных проходят через сервер, тем больше требуется ресурсов. Поэтому на серверных компьютерах, как правило, не экономят.

Серверные компьютеры — настоящие монстры, содержащие несколько процессоров, дисковые массивы впечатляющей емкости, быстрые каналы связи с Интернетом и специальное программное обеспечение, у которого достаточно "сил", чтобы управлять всей этой мощью. Все в них нацелено на то, чтобы обслужить как можно больше клиентов, обработать как можно больше запросов, чтобы пользователи получили запрошенную информацию за приемлемое время. Но часто, если клиентов и запросов оказывается слишком много, ресурсов серверного компьютера не хватает, и начинаются проблемы. Они могут проявляться в том, что сервер просто отказывается обслужить "лишних" клиентов, предлагая им подождать немного, когда нагрузка немного снизится, а то и в том, что могучий серверный компьютер просто-напросто "зависает". Такое тоже случается, и не так уж редко.

Но не будем о грустном! Не стоит начинать знакомство с таким притягательным миром интернет-технологий со столь печальных вещей, как системные сбои. Чем их меньше, и чем реже они случаются, тем лучше для всех нас.

Итак, мы только что познакомились с особой *архитектурой* (принципом построения компьютерных систем), называемой *двухзвенной*, или архитектурой *"клиент-сервер"*, разделяющей все интернет-программы на клиенты и серверы. Эта архитектура используется для реализации практически всех современных интернет-сервисов и пока что себя оправдывает.

Замечание

Некоторые интернет-сервисы, в частности, так называемые *файлообменные сети* (Napster, Gnutella, Kazaa и пр.), используют принципиально другую архитектуру — *однозвенную*. Здесь все компьютеры, подключенные к Интернету и реализующие этот сервис, фактически равны между собой; любой из них может выступать в роли как клиентского (запрашивать информацию у других компьютеров), так и серверного (предоставлять хранящуюся на нем информацию другим компьютерам). Само собой, здесь используется особое программное обеспечение, реализующее функции и клиента, и сервера.

Протоколы

Люди, чтобы понимать друг друга, должны разговаривать на одном языке. Точно так и с компьютерами, подключенными к сети, неважно, какой — все-

мирной или локальной. Обмен данными по этим сетям должен проходить по единым стандартам, иначе случится новое вавилонское столпотворение.

Стандарт, по которому кодируются данные для отправки по сети, называется *протоколом*. В Интернете для обмена данными используются несколько протоколов, которые мы здесь вкратце рассмотрим.

Самый фундаментальный, если так можно сказать, протокол Интернета — это *TCP/IP* (Transfer Control Protocol/Internet Protocol, протокол управления передачей/протокол Интернета). Это так называемый *протокол низкого уровня*, определяющий только самые основные параметры передаваемых данных: длину отдельных порций *(пакетов)* данных, формат указания адресов получателя и отправителя, а также простейшие средства защиты от ошибок. Можно сказать, что TCP/IP занимается исключительно передачей данных по каналам Интернета, не вникая, что же именно он передает.

На протоколе TCP/IP базируются другие протоколы, уже *высокого уровня*. Эти протоколы описывают способы построения клиентских запросов и серверных ответов, особые команды, пересылаемые клиентом серверу при запросе или передаче данных, и способы кодирования передаваемой информации. Сам процесс передачи данных они не затрагивают — для этого существует "чернорабочий" TCP/IP.

Замечание

Строго говоря, существуют еще *протоколы физического уровня*, располагающиеся "ниже" TCP/IP. Они определяют электрические параметры сигналов, кабелей, разъемов и пр.

Каждый сервис Интернета использует свой собственный протокол высокого уровня (а то и сразу несколько, предназначенных для разных задач или разработанных конкурирующими организациями). Давайте рассмотрим протоколы, с которыми мы столкнемся в будущем.

Начнем мы, конечно, с WWW. Для передачи данных Всемирная паутина использует протокол *HTTP* (HyperText Transfer Protocol, протокол передачи гипертекста). Он задает набор команд для запроса и отправки данных, пересылаемых клиентом (Web-обозревателем) Web-серверу, и способы представления пересылаемых в обе стороны данных. Пожалуй, это самый широкоизвестный протокол Интернета (конечно, после TCP/IP).

Замечание

Протокол HTTP для управления обменом данных предусматривает всего три команды: загрузки файла, отправки файла и получения сведений о файле.

Сервис пересылки файлов FTP использует протокол, который так и называется — *FTP*. Он также определяет набор команд для управления файлами на

сервере (загрузка с сервера, отправка на сервер, копирование, перемещение, удаление и т. д.) и способы кодирования файлов для пересылки по каналам связи. В этом смысле протоколы HTTP и FTP — "родственники".

А вот электронная почта использует целых два протокола. Первый протокол — *SMTP* (Simple Mail Transfer Protocol, простой протокол пересылки почты) — используется для пересылки почты клиентом серверу. При получении же почты от сервера клиент общается с ним по протоколу *POP3* (Post-Office Protocol, протокол почты).

Замечание

Нужно также упомянуть протокол *IMAP* (Internet Message Access Protocol, протокол доступа к почте Интернета), применяемый также для получения клиентом почты от сервера. По сравнению с более старым POP3 он предоставляет больше возможностей, но распространен не так широко.

Сервис новостей использует для работы протокол *NNTP* (Network News Transfer Protocol, протокол передачи сетевых новостей). Остальные сервисы используют свои протоколы. Но мы не будем на них останавливаться.

Мы уже знаем, что перед началом сеанса связи клиент должен установить соединение с сервером и что соединение это можно представить как воображаемый канал связи, существующий "внутри" физического канала (кабеля, волоконно-оптической линии или радиоканала). Так вот, протокол TCP/IP позволяет создать всего 65 535 таких воображаемых каналов, и называются эти каналы *портами* TCP/IP. Соединение с сервером может быть установлено через любой из этих портов; соединения, установленные через разные порты, действуют независимо, не "мешая" друг другу. Это позволяет сразу нескольким программам (например, Web-обозревателю, почтовому клиенту и интернет-пейджеру) обмениваться данными с соответствующими серверами через один физический канал (модем, сетевой кабель, радиоканал и пр.).

Каждый существующий протокол высокого уровня использует для передачи данных свой собственный порт TCP/IP (так называемый *порт по умолчанию*). В табл. 1.1 перечислены некоторые протоколы и используемые ими порты по умолчанию.

Таблица 1.1. *Порты TCP/IP, используемые по умолчанию для передачи данных некоторых протоколов высокого уровня*

Протокол	Порт по умолчанию
HTTP	80
FTP	21
SMTP	25
POP3	110

Но почему такое странное название — "порт по умолчанию"? Давайте разберемся.

Дело в том, что все более-менее серьезные серверы предоставляют возможность изменить порт, используемый протоколом, которые они обслуживают, на другой. Например, Web-сервер может быть настроен так, чтобы использовать для "общения" с клиентами не 80-й порт, а, скажем, 8000-й. (Автору этой книги время от времени встречаются Web-серверы, настроенные таким образом.) Это весьма редко, только в особых случаях, но все же применяется. (Например, если на одном серверном компьютере работают два Web-сервера, один из них настраивают на порт по умолчанию — 80-й, — а другой — да хотя бы и на 8000-й.)

Замечание

Вообще, специально настраивать сервер для использования порта по умолчанию не нужно — он уже настроен соответствующим образом.

Интернет-адреса

Теперь давайте поговорим о том, каким образом идентифицируются компьютеры, подключенные к Интернету. А именно — об интернет-адресах.

Интернет-адрес — это уникальное числовое или строковое значение, позволяющее точно идентифицировать компьютер в Сети. Именно по интернет-адресу клиент находит нужный ему сервер. Именно по интернет-адресу происходит отправка данных. Интернет-адрес — это своего рода "имя" сервера.

Изначально, на заре эпохи Интернета, в качестве интернет-адреса использовался *IP-адрес* — числовое значение, идентифицирующее компьютер для протокола TCP/IP. Как мы помним, TCP/IP разбивает передаваемую информацию на пакеты. Так вот: в каждом таком пакете содержатся IP-адреса компьютера-отправителя и компьютера-получателя.

IP-адрес замечательно подходит для компьютеров, но очень плохо — для людей. Он имеет такой вид:

```
192.168.1.10
```

Не очень-то наглядно, правда? Именно поэтому с расширением Интернета была введена в строй новая система интернет-адресов, которой мы пользуемся до сих пор. Это так называемые доменные адреса, о которых стоит поговорить подробно.

Но прежде чем мы начнем разговор о доменных адресах, давайте выясним, что такое домен. *Домен*, или *доменная зона*, — это участок Интернета, выделенный по какому-либо принципу, например, территориальному. Домен мо-

жет быть крупным, или мелким, или вообще состоять из одного компьютера. Он обозначается особой строкой текста.

Внимание!

В обозначении домена могут присутствовать только латинские буквы, цифры, знаки "минус" и подчеркивания.

Структура доменов похожа на матрешку: мелкие домены "вложены" внутрь крупных, а крупные, в свою очередь, — внутрь гигантских. Гигантские домены называются *доменами верхнего уровня*, а вложенные в них более мелкие — *доменами нижнего уровня*.

Домены верхнего уровня бывают интернациональными и национальными. *Интернациональные домены* объединяют компьютеры по какому-то нетерриториальному признаку; к ним относятся домены com (коммерческие серверы), edu (образовательные), mil (военные), org (организации, не занимающиеся компьютерами и Интернетом), net (организации, занимающиеся компьютерами и Интернетом), biz (коммерческие организации) и некоторые другие. *Национальные домены* объединяют компьютеры по территориальному признаку и выдаются целым странам; это домены us (США), uk (Великобритания), fr (Франция), de (Германия), ru (Россия) и др.

Что касается доменов нижнего уровня, то они выдаются, как правило, отдельным организациям или, опять же, по территориальному признаку. Их текстовое обозначение часто совпадает с названием организации или района — владельца домена.

Если теперь записать обозначения всех доменов, в которых находится нужный нам компьютер, в порядке от более мелких к более крупным, разделив их точками, мы получим *доменное имя* этого компьютера. Так, если у нас сам компьютер имеет имя comp45, отдел, в котором он стоит, — buh (бухгалтерия), организация, включающая этот отдел, — department, а страна — ru (Россия), то мы получим такое доменное имя:

```
comp45.buh.department.ru
```

Согласитесь — запомнить это гораздо проще, чем невразумительный IP-адрес.

Да, но проблема в том, что протокол TCP/IP не понимает доменные имена! Что делать? Как преобразовать доменное имя в понятный ему IP-адрес?

Для этого используются особые программы, называемые серверами *DNS* (Domain Name System, система доменных имен). Занимаются они тем, что принимают от компьютеров, которым нужно куда-то отправить данные по протоколу TCP/IP, доменные имена и возвращают соответствующие этим именам IP-адреса. Такие серверы DNS имеются в каждом домене; кроме того,

несколько самых мощных в мире серверов DNS находятся как бы "выше" всех доменов, даже доменов верхнего уровня, — это глобальные серверы DNS.

Но вернемся к доменным именам. Доменное имя идентифицирует сам серверный компьютер, а не выполняющуюся на нем программу-сервер. А таких серверов на одном компьютере может быть несколько: Web, FTP, почта, чат и пр. Чтобы обратиться к нужному серверу, не "беспокоя" остальных, перед доменным именем указывается обозначение протокола, по которому этот сервер "общается" с клиентами; при этом для передачи данных будет задействован порт по умолчанию этого протокола. Вот пара примеров интернет-адресов, указывающих на программы-серверы (обозначение протокола выделено полужирным шрифтом):

```
http://comp45.buh.department.ru
ftp://comp45.buh.department.ru
```

В первом случае мы обращаемся к Web-серверу, а во втором — к серверу FTP, находящимся на одном и том же компьютере comp45.buh.department.ru.

Если же какой-либо сервер использует порт, отличный от порта по умолчанию, то номер нужного порта записывается после доменного имени серверного компьютера и отделяется от него двоеточием. Вот так мы можем обратиться к Web-серверу, использующему порт 8000 (номер порта выделен полужирным шрифтом):

```
http://comp45.buh.department.ru:8000
```

Ну вот, с основными принципами работы Интернета и соответствующими им понятиями и терминами мы ознакомились. Конечно, кое-что новое мы узнаем потом, в процессе чтения книги, но пока что полученных знаний нам хватит. Давайте сосредоточимся на WWW — в основном, именно этим сервисом мы будем пользоваться на протяжении всей книги.

Базовые понятия WWW

Здесь мы выясним все о Web-страницах и Web-сайтах, узнаем, чем сайт отличается от страницы, поговорим о Web-клиентах и Web-серверах и освоим несколько новых понятий.

Web-страницы и Web-сайты

Что такое *Web-страница*? Ответить на этот вопрос могут многие. Это интернет-документ, предназначенный для распространения через Интернет посредством сервиса WWW. А если уж говорить совсем упрощенно, это то, что

показывает в своем окне программа для просмотра Web-страниц — Web-обозреватель.

С технической точки зрения Web-страница — это текстовый файл, содержащий собственно текст наряду со специальными командами, выполняющими форматирование текста и создающими элементы, не относящиеся к тексту (изображения, таблицы и пр.), и сохраненный на жестких дисках серверного компьютера. Получив от Web-обозревателя запрос по протоколу HTTP, *Web-сервер* (серверная программа, обеспечивающая работу сервиса WWW) извлекает этот файл и отправляет его Web-обозревателю.

Файлы, хранящие Web-страницы, должны иметь расширение htm или html. Оба этих расширения вполне равноправны.

Внимание!

Web-страницы, созданные с использованием какой-либо технологии серверных страниц, должны иметь расширение, соответствующее используемой технологии. Так, страницы, созданные с использованием PHP, должны иметь расширение php. (О серверных Web-страницах будет рассказано в *главе 6*.)

А что такое *Web-сайт*? Это набор Web-страниц, подчиненных общей тематике и объединенных в единое целое (как — будет рассказано в *главе 2*). Как видим, чисто технических отличий у Web-страницы и Web-сайта не слишком много.

Замечание

Отдельные Web-страницы и целые Web-сайты также могут быть сохранены на жестком диске клиентского компьютера и открываться в Web-обозревателе прямо с него. В этом случае роль своеобразного Web-сервера выполняет файловая система.

Теперь поговорим о некоторых технических деталях размещения сайта на серверном компьютере. Эти детали очень важны.

Прежде всего, для хранения всех файлов, составляющих Web-сайт, на диске серверного компьютера создается особая папка, называемая *корневой*. Эта папка создается человеком, занимающимся настройкой и обслуживанием программы Web-сервера (или же всего серверного компьютера), — *администратором*. Полный путь к этой папке тот же самый администратор заносит в настройки Web-сервера, чтобы последний смог ее найти.

Внимание!

Все без исключения файлы, являющиеся содержимым Web-сайта, должны находиться в корневой папке. Все файлы, не находящиеся в корневой папке, автоматически исключаются Web-сервером из состава сайта.

Внутри корневой папки могут быть созданы другие папки, хранящие файлы второстепенных Web-страниц, графических изображений, архивов и дистрибутивов программ. Обычно это делается для удобства управления большим Web-сайтом.

Замечание

Вообще-то, все серьезные программы Web-серверов предоставляют возможность создания так называемых *виртуальных папок*. Виртуальная папка — это папка, находящаяся в любом месте файловой системы компьютера, но считаемая Web-сервером частью сайта. Виртуальные папки также создаются администратором Web-сервера.

Одна из страниц Web-сайта должна быть так называемой *страницей по умолчанию*. Она отправляется Web-обозревателю, если мы обратимся напрямую к Web-серверу, отправив запрос на его интернет-адрес, скажем:

```
http://www.somesite.ru
```

Имя файла этой страницы задается администратором Web-сервера в его настройках. Обычно файл страницы по умолчанию называется default.htm[l] или index.htm[l].

Если же нам нужна какая-то конкретная страница сайта, мы отправим Web-серверу интернет-адрес, указывающий на файл этой страницы. Например:

```
http://www.somesite.ru/somepage.html
```

Как мы видим, имя файла нужной нам страницы (в нашем случае — somepage.html) указывается сразу после интернет-адреса Web-сервера. Получив его, Web-сервер ищет этот файл в корневой папке сайта и, если он там есть, отправляет его клиенту.

Если же нам понадобится файл, находящийся не в самой корневой папке сайта, а в одной из вложенных в нее папок, мы отправим Web-серверу запрос вида:

```
http://www.somesite.ru/download/archive.zip
```

В этом случае Web-сервер отправит Web-обозревателю архивный файл archive.zip, находящийся в папке download, вложенной в корневую папку сайта.

Замечание

Для обращения к файлу, находящемуся в виртуальной папке, используется аналогичный запрос:

```
http://www.somesite.ru/pictures/somepicture.jpg
```

Здесь pictures — виртуальная папка.

Пожалуй, на этом знакомство с интернет-адресами можно закончить. Далее, изучая создание Web-страниц и, в частности, гиперссылок в *главе 2*, мы узна-

ем о них кое-что еще. А сейчас временно расстанемся с ними и поговорим о различных программах Web-обозревателей и Web-серверов, имеющих хождение в настоящее время.

Web-обозреватели

Мы уже знаем, что Web-обозреватели — это программы для просмотра Web-страниц и Web-сайтов. Их основная задача — это отправить Web-серверу корректно, в соответствии со всеми стандартами сформированный клиентский запрос, принять серверный ответ и обработать полученные в составе этого ответа данные (скажем, вывести на экран полученную Web-страницу или сохранить на жестком диске архивный файл). Для этого окно Web-обозревателя содержит поле ввода интернет-адреса и область, в которую собственно выводится Web-страница. (Разумеется, оно также содержит заголовок, главное меню и панели инструментов, как и многие окна приложений Windows.)

А теперь — нечто новенькое. После получения от сервера файла Web-страницы (и всех связанных с ней файлов, т. к. страница может состоять из множества файлов; подробнее об этом мы поговорим в *главе 2*) Web-обозреватель сохраняет их на жестком диске клиентского компьютера в особой области, называемой *кэшем*. Этот кэш может иметь вид как обычной папки (кэш Microsoft Internet Explorer и Opera), так и большого файла (кэш Netscape Navigator и Firefox).

Зачем это нужно? Да хотя бы затем, чтобы мы смогли впоследствии просмотреть данную страницу, не подключаясь к Интернету. Все современные Web-обозреватели поддерживают так называемый *автономный режим* (offline mode), когда они отображают только те страницы, что находятся в кэше. (Кстати — исключительно удобная вещь, особенно для тех, кто выходит в Интернет через модем.) Если же мы попытаемся просмотреть страницу, которой нет в кэше, Web-обозреватель предложит нам подключиться к Интернету и загрузить ее.

Теперь познакомимся с программами Web-обозревателей, имеющими в настоящее время наибольшую популярность. Все они, в общем, следуют одним и тем же стандартам и отличаются друг от друга только деталями, не оговоренными в этих стандартах, и удобством для пользователей.

Король виртуальных просторов — это, конечно, Microsoft Internet Explorer. Он имеется на любом компьютере, работающем под управлением Windows (что, как говорят злые языки, и обусловило его популярность). Однако это очень мощная, быстрая, весьма нетребовательная к ресурсам и исключительно удобная программа. Автор данной книги для просмотра Web-страниц пользуется именно Internet Explorer. В настоящее время доступна версия 7.0.

Замечание

Начиная с версии 7.0, Internet Explorer официально называется Windows Internet Explorer. Но автор в дальнейшем будет использовать старое его наименование.

Почетное второе место занимает амбициозный новичок по имени Firefox. Эта программа распространяется бесплатно; более того, ее исходные тексты открыты для изучения и модификации. Она весьма быстра и компактна, поддерживает все Web-стандарты, нетребовательна к системным ресурсам и имеет множество интересных и весьма полезных возможностей, которыми пока не может похвастаться ни один из его конкурентов. Совсем недавно вышла версия 2.0.0.3, а на момент выхода данной книги в свет наверняка будет доступна более новая.

Третье место оккупировал "старший брат" Firefox под названием Mozilla. Он также распространяется бесплатно, исходные тексты его открыты, а по возможностям он примерно аналогичен Firefox. Последняя версия этой программы имеет номер 1.7, и, судя по всему, новые версии уже не появятся, т. к. проект Mozilla уже закрыт. Впрочем, ему на смену должен прийти его "преемник" SeaMonkey, разрабатывающийся той же командой программистов.

Некогда властелин WWW Netscape Navigator сейчас непопулярен — ему оказывают предпочтение буквально доли процента пользователей Интернета. Хотя последняя версия Navigator под номером 8.0 выглядит весьма неплохо, поддерживает все современные стандарты Интернета, корректно отображает большинство Web-страниц и не очень требовательна к системным ресурсам. Но все равно Navigator по многим параметрам проигрывает и Internet Explorer, и Opera, к тому же, неизвестно, будут ли выходить новые версии.

Пятое место занято разработкой фирмы Apple, производящей широко известные в узких кругах компьютеры Macintosh, — Safari. В настоящий момент имеет хождение версия Safari 2.0. Утверждается, что это самый быстрый в мире Web-обозреватель. Пока трудно сказать, что в действительности этот Safari собой представляет — у автора нет под рукой компьютера Apple Macintosh, чтобы попробовать эту программу в действии.

На шестом месте отдыхает детище норвежских программистов Opera. Эта достаточно мощная и очень быстрая программа (хотя и уступающая, по слухам, Safari), поддерживающая все официальные Web-стандарты, тем не менее, весьма требовательна к системным ресурсам и не всегда правильно отображает некоторые Web-страницы. Последняя вышедшая в свет версия носит номер 9.0 и, скорее всего, после выхода книги устареет, т. к. новые версии Opera появляются очень часто.

В настоящее время просторы WWW "бороздят" практически только шесть перечисленных выше программ. Существует, однако, еще несколько малоиз-

вестных Web-обозревателей, а также довольно многочисленная когорта программ, построенных на основе программного ядра Internet Explorer. Мы не будем их рассматривать.

Осталось только сказать, что выбор Web-обозревателя — это личное дело каждого. Все они поддерживают одни и те же стандарты (правда, зачастую по-своему) и предоставляют пользователю примерно одинаковый набор возможностей (хотя, не все найдут его удобным). Так что, как в песне поется, "думайте сами, решайте сами"...

Web-серверы

Поскольку мы не только пользователи, но и уже наполовину разработчики, нас будут интересовать не только Web-обозреватели, но и Web-серверы. Давайте поговорим и о них.

"Зоопарк" Web-серверов ничуть не меньше "зоопарка" (или, если учесть, что Web-обозреватели жестко конкурируют друг с другом, "серпентария") Web-обозревателей, так что мы можем подобрать себе программу по вкусу. И, в отличие от Web-обозревателей, среди Web-серверов нет безоговорочного лидера — даже самые распространенные из них занимают менее половины рынка.

Web-сервер *Apache* — пожалуй, самый распространенный. Среди его достоинств: полная бесплатность (более того — его исходные тексты открыты), легкость настройки, довольно высокая производительность, хорошая для бесплатного продукта поддержка. По крайней мере, для Web-сайтов с небольшой загрузкой — это идеальный выбор.

Кстати, именно Web-сервер Apache, как самый популярный, мы и будем использовать для тестирования своих первых Web-сайтов.

Еще один весьма примечательный Web-сервер — Sambar. Он поддерживает такое количество интернет-технологий (многие из них — эксклюзивные, больше нигде не реализованные), что просто оторопь берет — как же всем этим богатством воспользоваться? Недостатка у Sambar всего два: небольшая известность и не очень удобная настройка. (Кстати, собственный Web-сайт автора этой книги, доступный в локальной сети Волжского гуманитарного института, работает именно под этим Web-сервером.)

И, конечно, нельзя не упомянуть о двух разработках фирмы Microsoft: Personal Web Server и Internet Information Server. Они поставляются в составе Microsoft Windows, первая программа — в составе Windows 98 и ME, а вторая — в составе Windows NT, 2000, XP, 2003 и Vista. Со своими обязанностями они справляются очень хорошо, не транжирят системные ресурсы, легко настраиваются, поддерживают множество передовых интернет-технологий

и, как говорят, при надлежащей настройке легко затыкают за пояс конкурентов. Кроме собственно Web-сервера, они содержат и серверы FTP и электронной почты, а также некоторое количество вспомогательных программ.

Менее известные и специализированные Web-серверы мы рассматривать не будем, т. к. их очень много. Поговорим лучше о том, как разместить созданный нами сайт в Интернете.

Публикация Web-сайта в Интернете. Хостинг-провайдеры

Итак, предположим, что мы создали свой Web-сайт (а мы его и создадим, пока будем изучать интернет-технологии по этой книге). Теперь нам нужно сделать так, чтобы все желающие увидеть его собственными глазами, а именно — разместить, или, как говорят опытные разработчики сайтов, *опубликовать* его в Интернете. А значит, нам нужно подключение к Интернету и Web-сервер.

Если наш компьютер подключен к Интернету по скоростному каналу или находится в локальной сети, работающей по тем же стандартам, что и Интернет (так называемый *интранет*), то мы можем просто установить на него Web-сервер и сюда же поместить наш Web-сайт. Это самый простой способ, хотя, конечно, нам придется попутно освоить профессию администратора.

Для тех "счастливчиков", что выходят в Интернет по телефонным каналам (как автор этой книги), существуют три способа донести свое Web-творение до страждущих масс. Давайте перечислим их в порядке от простых к более хлопотным.

Большинство солидных интернет-провайдеров, кроме собственно доступа в Интернет, предоставляют своим клиентам и другие услуги: электронную почту, доступ на собственный сайт с новостями, документацией и файловым архивом и т. д. Есть и такая услуга, как предоставление на жестких дисках серверного компьютера места для размещения Web-сайтов клиентов. В этом и заключается первый способ: выяснить условия публикации сайта на сервере интернет-провайдера и, следуя этим условиям, опубликовать его.

Если же интернет-провайдер жадничает, можно прибегнуть ко второму способу. В Интернете существует довольно много серверов, предоставляющих место для сайтов бесплатно. Процесс и в этом случае очень прост: заходим на сайт такого сервера, регистрируемся, выясняем условия публикации сайта и публикуем.

Всем хороши бесплатные серверы: и денег не берут, и позволяют публиковать сайты. Но бесплатного сыра много не бывает... Как правило, объем пре-

доставляемого под сайт дискового пространства сильно ограничен, значит, большой сайт таким образом не опубликуешь. Вдобавок администратор может ограничить количество посетителей, которые могут одновременно зайти на наш сайт. Да и с поддержкой некоторых технологий, используемых для создания Web-сайтов (да хотя бы того же самого PHP), дело может обстоять не очень хорошо.

Так что если нам нужно нечто большее, чем скромные предложения бесплатных серверов, придется достать кошелек и пойти третьим путем — опубликовать сайт на любом платном сервере. Благо сейчас их довольно много, и услуги их сравнительно дешевы.

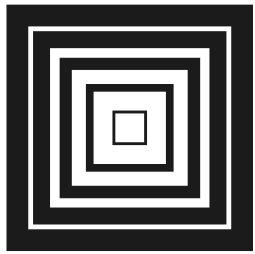
Кстати, организации, предоставляющие место на своих серверах для публикации Web-сайтов, называются *хостинг-провайдерами*.

Что дальше?

Вот и все об интернет-технологиях. Общие вопросы мы рассмотрели, пора приступать к созданию Web-страниц и Web-сайтов.

В следующей главе мы выясним, что же представляют собой Web-страницы и как они создаются. Мы изучим язык написания Web-страниц HTML, узнаем, из чего должна состоять всякая уважающая себя Web-страница, и проясним еще кое-какие вопросы.

А еще мы фактически начнем работу над своим сайтом, разработав его структуру, которой будем следовать в дальнейшем.



ГЛАВА 2

HTML — язык написания Web-страниц

Вот мы и подошли вплотную к тому, чтобы создать нашу первую Web-страницу. Пока что это будет просто небольшой пример, нужный для того, чтобы изучить основные принципы создания Web-документов, которые будут нормально отображаться во всех Web-обозревателях. Что касается прототипа сайта, то мы займемся им позже.

В процессе создания Web-страниц нет ничего особо сложного. Уяснив несколько основных терминов и держа под рукой кое-какие справочники и пару программ, можно лепить Web-страницы, как пирожки. Неудивительно, что сейчас профессия Web-дизайнера так широко распространена.

Другое дело — приличный Web-сайт. Его создание начинается с довольно долгого этапа планирования. Нужно продумать его структуру, собрать все необходимые материалы (которые нужно опубликовать в Интернете — ведь именно ради этого и создается сайт) и привести их к виду, поддерживаемому Web-обозревателями. А после создания всех Web-страниц, входящих в сайт, их нужно связать друг с другом и обязательно проверить, нет ли ошибок в этих связках. Ведь вполне может оказаться так, что на какие-то страницы посетитель сайта вообще не сможет попасть — тот еще сюрпризец...

Теперь давайте включим компьютер — хватит ему отдыхать — и займемся делом. И начнем мы с изучения языка написания Web-страниц HTML.

Введение в язык HTML

HTML (HyperText Markup Language, язык гипертекстовой разметки) — это особый язык, на котором пишутся Web-страницы. Сейчас мы с ним познакомимся, а заодно попробуем силы в написании своих первых, пока еще совсем простых Web-страниц.

Теги HTML. Форматирование текста

Давайте запустим небольшой текстовый редактор Блокнот, поставляемый в составе Windows, и наберем в нем такой текст:

```
Пример Web-страницы
Это простейшая Web-страничка, созданная в стандартном
↳Блокноте и отображенная в Microsoft Internet Explorer.
```

Сохраним этот текст в файле под именем 2.1.txt и откроем его в Web-обозревателе Microsoft Internet Explorer, также поставляемом в составе Windows. (Автор использовал Microsoft Windows Internet Explorer 7.0, поставляемый в составе Windows Vista.) И увидим то, что показано на рис. 2.1.

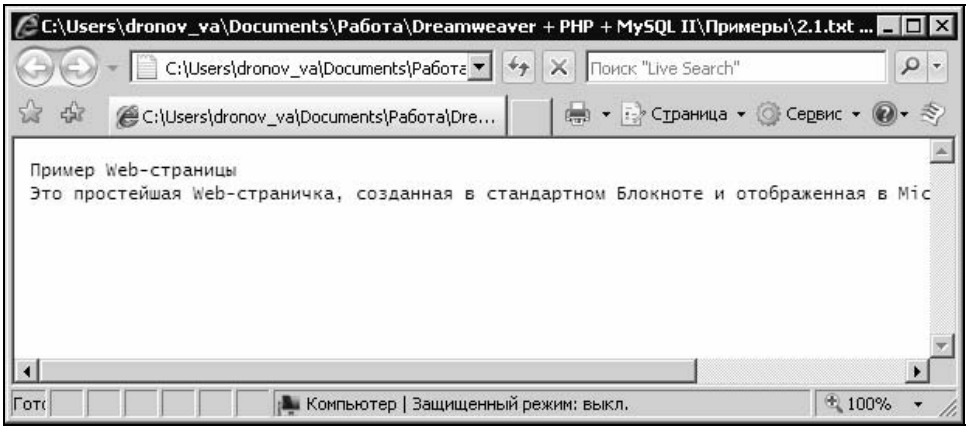


Рис. 2.1. Простой текстовый файл, открытый в Internet Explorer

Да, это еще не Web-страница. Это простой текстовый файл, открытый в Web-обозревателе (Web-обозреватели могут открывать также и простые текстовые файлы). Давайте превратим его в настоящую Web-страницу.

Для начала отредактируем файл 2.1.txt, чтобы его содержимое выглядело так (добавленные фрагменты выделены полужирным шрифтом):

```
<P>Пример Web-страницы</P>
<P>Это простейшая Web-страничка, созданная в стандартном
↳Блокноте и отображенная в Microsoft Internet Explorer.</P>
```

После этого сохраним отредактированный текст в другом файле — 2.2.html. (Только, когда будем вводить имя файла в стандартном окне сохранения, заключим его в кавычки, иначе Блокнот по доброте душевной добавит расширение txt, и наш файл получит имя 2.2.html.txt.) Открыв его в Internet Explorer, увидим то, что изображено на рис. 2.2. Видна разница?

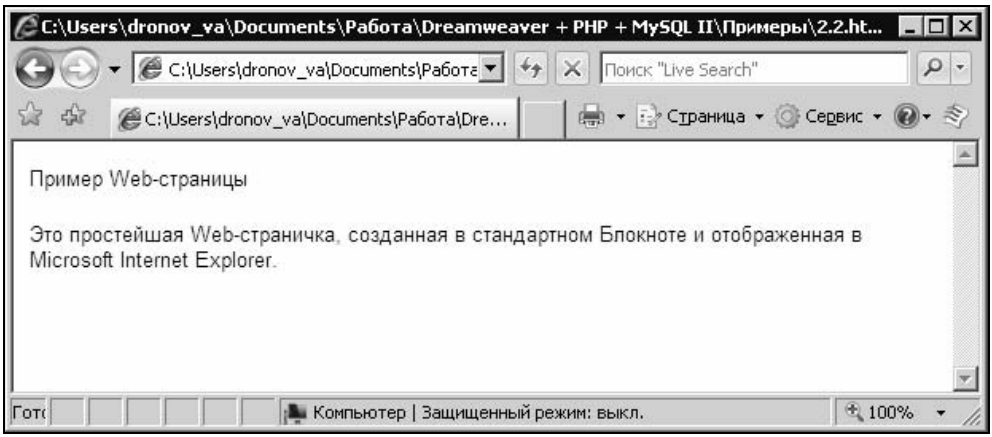


Рис. 2.2. Наша первая Web-страница, открытая в Internet Explorer

Текстовый файл 2.1.txt Internet Explorer вывел "как есть", без всяких изысков. В частности, он не разбил слишком длинный абзац на две строки (хотя это не помешало бы). А все потому, что обычный текст не содержит команд для его форматирования.

Совсем другое дело — код HTML, сохраненный в файле 2.2.html. Internet Explorer сформировал два абзаца — это хорошо заметно на рис. 2.2 — и разбил слишком длинный на строки так, чтобы ширина текста не превышала ширину его окна. А все потому, что он встретил в тексте две знакомые ему команды, описывающие форматирование текста, — так называемые *теги* HTML. Это теги текстового абзаца `<P>` и `</P>`.

В общем случае, теги HTML представляют собой английские слова, заключенные между знаками "меньше" и "больше" (< и >). Первый из тегов (в нашем случае — `<P>`) задает начало фрагмента текста, попадающего под действие тега, и называется *открывающим*. Второй тег (`</P>`) отличается тем, что в нем между знаком < и собственно наименованием тега стоит символ / ("слэш"); он задает конец фрагмента текста, попадающего под действие тега, и называется *закрывающим*. А сам фрагмент текста, на который оказывает влияние тег, называется его *содержимым*.

Абсолютное большинство тегов HTML являются *парными*, т. е. имеют открывающий тег, закрывающий тег и содержимое. Но есть и немногочисленные *одинарные* теги, состоящие только из одного тега вида `<IMG>` и не имеющие содержимого. Мы познакомимся с ними чуть позже.

Манипулируя тегами HTML, мы можем форматировать текст нашей первой странички, как хотим. Давайте, например, выделим названия двух упомянутых в нем программ курсивом. Для этого нам будет нужно использовать

парный тег <EM>...</EM>. Вот так (добавленные теги выделены полужирным шрифтом):

```
<P>Пример Web-страницы</P>
<P>Это простейшая Web-страничка, созданная в стандартном
↳<EM>Блокноте</EM> и отображенная в <EM>Microsoft Internet
↳Explorer</EM>.</P>
```

Результат этих священнодействий показан на рис. 2.3.

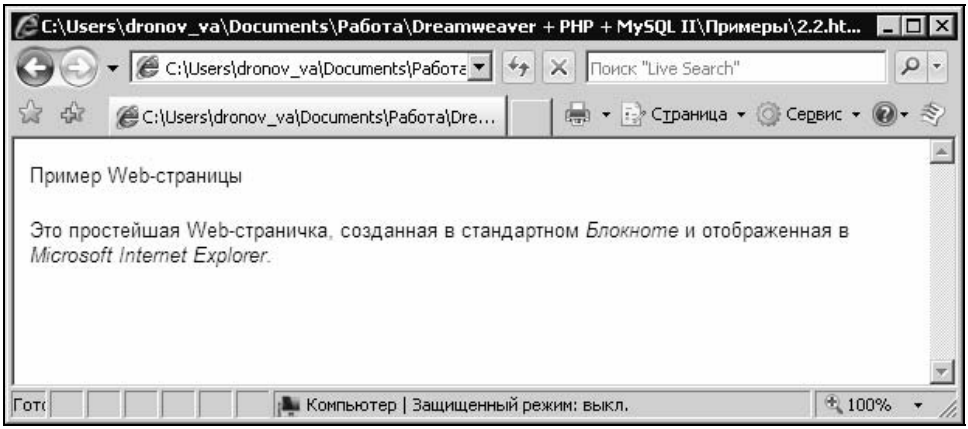


Рис. 2.3. Наша первая Web-страница — названия обеих программ выделены курсивом

Если мы захотим выделить название фирмы Microsoft полужирным шрифтом, то нам будет нужно использовать парный тег <STRONG>...</STRONG>. Вот так (опять же, добавленные теги выделены полужирным шрифтом):

```
<P>Пример Web-страницы</P>
<P>Это простейшая Web-страничка, созданная в стандартном
↳<EM>Блокноте</EM> и отображенная в <EM><STRONG>Microsoft</STRONG>
↳Internet Explorer</EM>.</P>
```

Здесь мы вложили тег <STRONG> внутрь тега <EM>. В результате этого слово "Microsoft" будет набрано полужирным курсивом. Вообще, *вложенность* тегов HTML друг в друга — обычное дело, и нам придется к ней привыкнуть.

Начинающие Web-дизайнеры очень часто допускают такую ошибку: они располагают закрывающие теги в порядке, не строго противоположном порядку открывающих тегов. Так, приведенный ниже код HTML содержит подобную ошибку: теги </EM> и </STRONG> поменялись местами (выделены полужирным шрифтом):

```
<P>Пример Web-страницы</P>
<P>Это простейшая Web-страничка, созданная в стандартном
```

```
↳<EM>Блокноте</EM> и отображенная в <EM><STRONG>Microsoft</EM>
↳Internet Explorer</STRONG>.</P>
```

Поведение Web-обозревателя, встретившего такой код, непредсказуемо (хотя Internet Explorer славится своим умением исправлять мелкие ошибки Web-дизайнера). Так что стоит запомнить на будущее простое правило: закрывающие теги должны повторяться в порядке, обратном порядку соответствующих им открывающих тегов.

И напоследок дадим нашей странице большой "кричащий" заголовок. Для этого в первом абзаце текста заменим тег `<P>...<P>` на `<H1>...<H1>` (измененный тег выделен полужирным шрифтом):

```
<H1>Пример Web-страницы</H1>
<P>Это простейшая Web-страничка, созданная в стандартном
↳<EM>Блокноте</EM> и отображенная в <EM><STRONG>Microsoft</STRONG>
↳Internet Explorer</EM>.</P>
```

Окончательный вариант нашей первой Web-страницы показан на рис. 2.4. Видно, что Web-обозреватель сам вывел содержимое тега `<H1>` очень крупными буквами, чтобы оно стало заметнее.

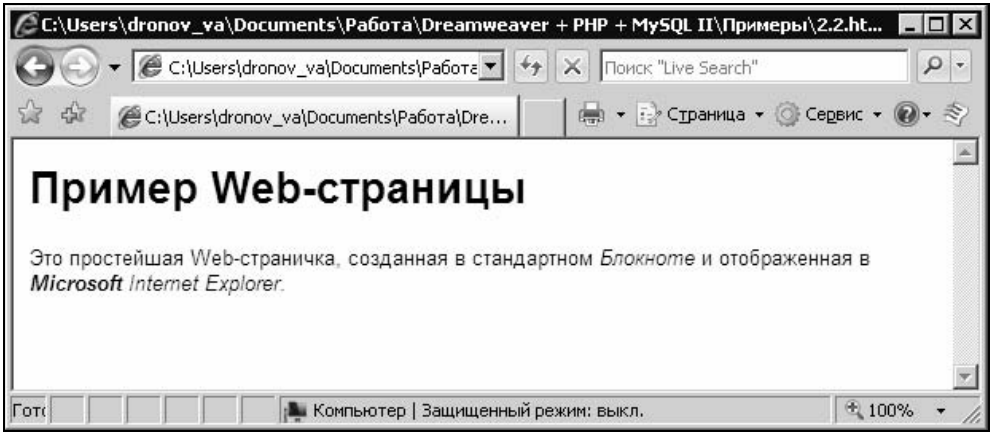


Рис. 2.4. Наша первая Web-страница с заголовком

Кстати, тег `<H1>` так и называется — тегом *заголовка* первого уровня. Почему первого? А потому, что стандарт HTML предусматривает целых шесть уровней заголовков. Эти заголовки задаются с помощью тегов `<H2>` (второй уровень), `<H3>` (третий) и так вплоть до `<H6>` (шестой, самый низкий уровень). Заголовками первого уровня обычно выводятся названия страниц, заголовками второго — названия отдельных частей страниц и т. д.

Кстати, для Web-обозревателя нет никакой разницы, как мы отформатировали HTML-код нашей страницы при его написании. Он следует очень простым правилам, которые перечислены ниже.

- Содержимое тега <P> и <H1> должно быть выведено как отдельный абзац текста. Если ширина этого абзаца превосходит ширину окна Web-обозревателя, абзац разбивается на более короткие строки (что и показано на рис. 2.2).
- Несколько следующих друг за другом пробелов в HTML-коде преобразуются в один пробел. (Последовательные пробелы часто появляются при неаккуратном наборе текста, и хорошо, что Web-обозреватель исправляет эту ошибку.)
- Перевод строки в HTML-коде преобразуется в пробел. (Это значит, что мы можем разбить слишком длинную строку HTML-кода на несколько более коротких.)

Давайте проверим это на практике. Запишем HTML-код страницы 2.2.html вот так:

```
<H1>Пример Web-страницы</H1>
<P>Это простейшая Web-страничка,
созданная в стандартном
<EM>Блокноте</EM> и отображенная в
<EM><STRONG>Microsoft</STRONG>
Internet Explorer</EM>.</P>
```

Мы разбили слишком длинную вторую строку кода на несколько более коротких — так намного удобнее работать с ней. (Кстати, давайте и в дальнейшем так делать.) Если мы теперь откроем отредактированную страницу 2.2.html в Web-обозревателе, увидим, что страница отобразится точно так же, как и до правки, — см. рис. 2.4.

А теперь давайте немного отдохнем и узнаем за время отдыха кое-что новое.

Понятно, что для того, чтобы различные Web-обозреватели отображали одну и ту же Web-страницу одинаково, язык HTML должен быть стандартизирован. Его стандартизацией (а также множеством других стандартов Интернета) занимается особая организация, называемая *World Wide Web Consortium*, или, сокращенно, *WWWC*, или, что встречается чаще, *W3C*. Это название можно перевести как "Комитет Всемирной Паутины".

W3C издает весьма увесистые труды, описывающие различные версии стандарта HTML. Последняя версия этого языка — 4.01 — вышла в конце 90-х годов прошлого века. Все современные версии Web-обозревателей поддерживают именно эту версию HTML.

Замечание

Судя по всему, версия 4.01 HTML станет действительно последней. В дальнейшем язык HTML будет постепенно заменен своим потомком — языком *XHTML* (eXtensible HypetText Markup Language, расширяемый язык гипертекстовой разметки). Фактически XHTML — это более строгий, усовершенствованный и очищенный от устаревших тегов потомок HTML. По крайней мере, по набору тегов языки HTML и XHTML очень похожи. В настоящее время язык XHTML еще не очень распространен и, судя по всему, более-менее значительную популярность получит не скоро.

Графика на Web-страницах. Внедренные элементы

Продолжим наши эксперименты с языком HTML и Web-страницами. И на этот раз поговорим о том, как поместить на нашу страничку графическое изображение.

Но сначала давайте введем еще один термин, который поможет нам в дальнейшей работе. Пусть любой абзац текста, любой его фрагмент, являющийся содержимым парного тега, а также все, что помещается на Web-страницу одинарным тегом, называется *элементом страницы*. То есть элементами страницы в нашей терминологии будут текстовый абзац, его фрагмент, выделенный курсивом, заголовок, графическое изображение и пр.

Ранее мы имели дело только с *текстовыми элементами* Web-страниц. Это пресловутые абзацы текста, заголовки, фрагменты курсива и полужирного текста. Они создаются с помощью уже знакомого нам кода HTML, который пишется в текстовом редакторе и сохраняется в текстовых файлах с расширением htm[l]. А теперь разговор пойдет о тех элементах страниц, которые нельзя закодировать с помощью HTML, — о *внедренных элементах*. Каждый внедренный элемент страницы хранится в отдельном файле, а в самом коде HTML-страницы, которая его содержит, с помощью особого тега проставляется ссылка на этот файл.

Внедренные элементы — это, в частности, графические изображения. В самом деле, графическую информацию невозможно сохранить в текстовом виде; для хранения рисунков, фотографий, схем придуманы свои собственные форматы — GIF, JPEG, BMP, TIFF и др. Для работы с ними текстовые редакторы не подходят — нужны специальные программы графических редакторов.

Предположим, у нас имеется некая картинка, которую мы хотим поместить на нашу Web-страницу. Что для этого нужно сделать?

Прежде всего, нам нужно определить, в какое место Web-страницы будет помещен рисунок, и найти соответствующий участок HTML-кода. После этого

остается вставить туда особый одинарный тег <IMG> (выделен полужирным шрифтом):

```
<H1>Пример Web-страницы</H1>
<P>Это простейшая Web-страничка, созданная в стандартном
<EM>Блокноте</EM> и отображенная в <EM><STRONG>Microsoft</STRONG>
Internet Explorer</EM>.</P>
<IMG SRC="exclam.gif">
```

Встретив этот тег, Web-обозреватель пошлет Web-серверу еще один запрос — на получение файла exclam.gif. Получив этот файл, он выведет его содержимое на Web-странице в том месте, где стоит тег <IMG> (рис. 2.5).

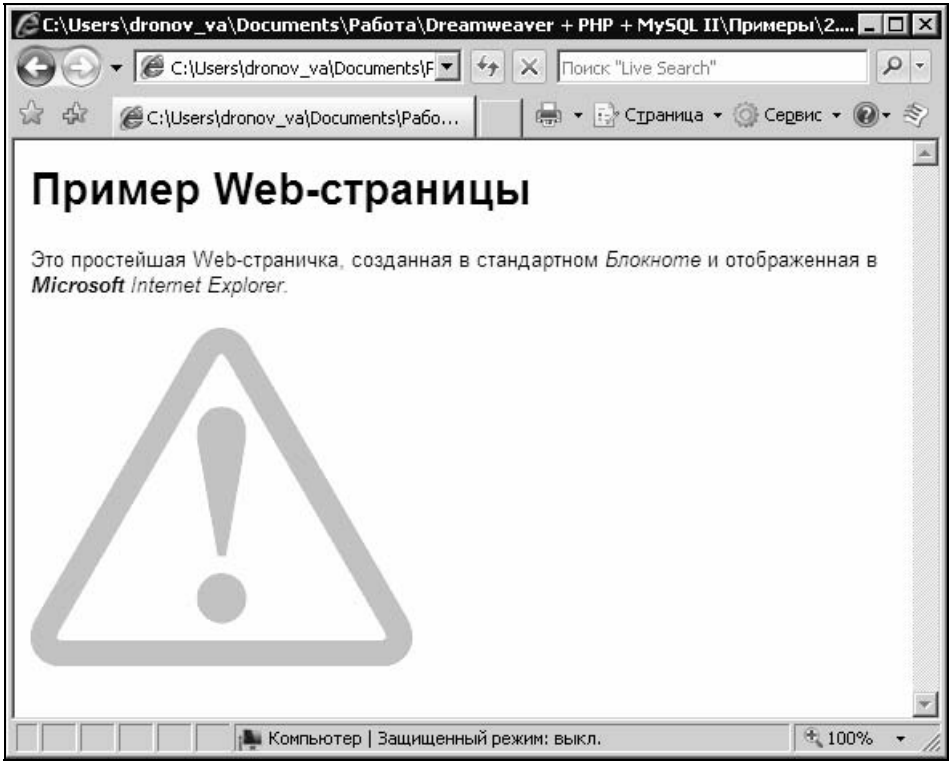


Рис. 2.5. Наша первая Web-страница с графическим изображением

Здесь мы столкнулись с *атрибутами тега* — своего рода параметрами, задающими для тега некоторые дополнительные условия или значения. В нашем случае тег <IMG> содержит атрибут SRC, которому присвоено *значение* "exclam.gif" — имя (а фактически — интернет-адрес) нужного нам файла. Запомним, что все значения атрибутов пишутся в двойных кавычках.