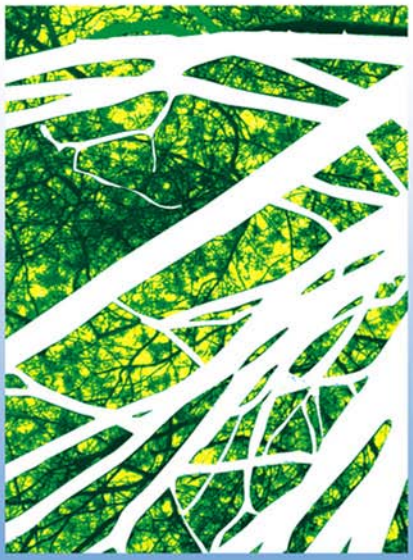
Владимир Дронов

bhv

JavaScript и AJAX в Web-дизайне



2-е издание

- HTML, CSS, JavaScript и AJAX
- Web-сценарии и события
- Управление Web-обозревателем и содержимым Web-страницы
- Графика и анимация
- Web-формы и базы данных
- Фильтры и преобразования

Наиболее полное руководство

В ПОДЛИННИКЕ®

Владимир Дронов

JavaScript и AJAX в Web-дизайне

2-е издание

Санкт-Петербург

«БХВ-Петербург»

2008

УДК 681.3.06
ББК 32.973.26-018.2
Д75

Дронов В. А.

Д75 JavaScript и AJAX в Web-дизайне: 2-е изд., перераб. и доп. — СПб.: БХВ-Петербург, 2008. — 736 с.: ил. — (В подлиннике)

ISBN 978-5-9775-0251-1

В книге описывается все, что должен знать Web-дизайнер: принципы создания Web-страниц, язык JavaScript, основы написания Web-сценариев, работа с содержимым Web-страницы, обработка данных, введенных в Web-форму, особенности различных Web-обозревателей, использование баз данных, фильтров и преобразований, графика, анимация и пр. Изложение сопровождается большим количеством подробно разобранных примеров и полезных советов. Особое внимание уделено вопросам совместимости Web-сценариев с различными Web-обозревателями.

Второе издание книги, ранее выходившей под названием "JavaScript в Web-дизайне", полностью переработано и дополнено с учетом современных технологий, дан вводный курс AJAX.

Для Web-дизайнеров

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	*Екатерина Капалыгина*
Компьютерная верстка	*Натальи Караваевой*
Корректор	*Виктория Пиотровская*
Дизайн серии	*Инны Тачиной*
Оформление обложки	*Елены Беляевой*
Зав. производством	*Николай Тверских*

Лицензия ИД № 02429 от 24.07.00. Подписано в печать 27.06.08.
Формат $70\times100^{1}/_{16}$. Печать офсетная. Усл. печ. л. 59,36.
Тираж 2500 экз. Заказ №
"БХВ-Петербург", 194354, Санкт-Петербург, ул. Есенина, 5Б.

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

ISBN 978-5-9775-0251-1

Оглавление

Введение

В далеком уже 2001 году автор написал книгу о языке JavaScript и его использовании в Web-дизайне. Эта книга стала первой и одной из самых активно продаваемых в карьере автора.

Но время идет, интернет-стандарты меняются, интернет-технологии развиваются, выходят новые версии Web-обозревателей, имеющие новые возможности, книги устаревают. Давно устарела и первая книга "JavaScript в Web-дизайне". Вот поэтому автор снова, образно говоря, взялся за перо.

Новая книга — "JavaScript и AJAX в Web-дизайне" — это обновленное издание, включающее описание некоторых новых возможностей и технологий и очищенное от уже устаревшего материала (описывающего, в основном, давно почивший в бозе Netscape Navigator). Вооружившись самой "свежей" документацией и самыми последними версиями Web-обозревателей Internet Explorer, Opera и Firefox, автор как следует в них покопался и теперь представляет на суд читателей результат этих "раскопок".

Грамотные интернетчики могут пропустить следующую главу. Остальным лучше ее прочитать, чтобы узнать, о чем же эта книга.

О чем эта книга?

Что мы чаще всего делаем в Интернете? Правильно, смотрим Web-страницы. А что такое Web-страница? Это особым образом подготовленный документ, содержащий текст, изображения, таблицы и пр. Мы вводим интернет-адрес страницы в специальное поле ввода Web-обозревателя и наблюдаем все это в его окне.

Web-страницы создаются с помощью особых языков — HTML и CSS. Первый язык описывает собственно содержимое страницы (какой текст, какие изображения и какие таблицы на ней будут присутствовать), второй — ее оформление (размер шрифта, цвет текста и фона, толщину рамки и пр.). Если короче, то языки HTML и CSS задают представление страницы.

Но ни HTML, ни CSS не позволяют задать поведение страницы. Например, мы не сможем только с их помощью сделать так, чтобы при наведении курсора мыши на изображение оно менялось или чтобы какой-то из элементов страницы скрывался и вновь показывался в ответ на нажатие кнопки. И уж тем более мы не сможем сделать какой-то из элементов страницы анимированным. Страница статична — загрузившись, она остается неизменной, что бы мы с ней не делали.

Почему? А потому, что языки HTML и CSS не предусматривают никаких средств, чтобы описать поведение элемента страницы. Они, как было сказано ранее, описывают только ее представление.

Но ведь мы часто встречаем страницы с меняющимися в ответ на движения мыши изображениями, скрывающимися и показывающимися элементами и даже анимацией! Как все это делается?

С помощью Web-сценариев — особых программ, вставляющихся прямо в страницу и выполняющихся Web-обозревателем. Они как раз и реализуют все эти чудеса.

Web-сценарии пишутся, как правило, на языке JavaScript. (Иногда для этого используются и другие языки, но крайне редко.) Его мы и будем изучать на протяжении всей этой книги.

Что мы можем сделать с помощью Web-сценариев? О, довольно много…

- Управлять Web-обозревателем (открывать и закрывать его окна, менять их размеры и местоположение на экране и др.).
- Управлять содержимым Web-страницы (добавлять новые элементы и удалять ненужные, изменять содержимое элементов и их параметры).
- Работать с произвольными данными (сохранять и пересылать другим страницам).
- Работать с данными, введенными посетителем в Web-форму.
- Реализовывать специальные эффекты (анимация, создание "красивостей" наподобие тени, эффекта размытия и пр.).

И все это мы изучим благодаря данной книге.

Какие программы будут использоваться в этой книге?

Вопрос далеко не праздный, учитывая то, что за программы сегодня приходится платить… Так что же за программы использовал автор?

Только бесплатные!

- Блокнот — простейший текстовый редактор, стандартно поставляемый в составе Windows.
- Internet Explorer 7 — Web-обозреватель, самый популярный на данный момент.
- Opera 9.26 — Web-обозреватель.
- Firefox 2.0.0.11 — Web-обозреватель.

Все примеры тестировались на перечисленных ранее Web-обозревателях. В некоторых случаях примеры также тестировались на более старых Web-обозревателях, в частности, на Netscape Navigator 7 preview 1.

Другие программы автором в работе практически не использовались.

Типографские соглашения

В этой книге часто приводятся форматы написания различных тегов HTML, атрибутов стилей CSS и выражений JavaScript. Нам необходимо выучить типографские соглашения, используемые для их написания.

> ***Внимание!***
>
> Все эти типографские соглашения применяются автором только в форматах написания тегов HTML, атрибутов стилей CSS и выражений JavaScript. В коде примеров они не имеют смысла.

В угловые скобки (<>) заключаются названия значений атрибутов, параметров или фрагментов кода, которые, в свою очередь, набраны курсивом. В код реального сценария, разумеется, должны быть подставлены реальное значение, реальный параметр или реальный код. Например:

```
<MAP NAME="<имя карты>">
  <набор описаний горячих областей>
</MAP>
```

Здесь вместо подстроки *<имя карты>* должно быть подставлено реальное имя карты, а вместо подстроки *<набор описаний горячих областей>* — реальный код, описывающий горячие области.

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

```
USEMAP=[<интернет-адрес страницы>]#<имя карты>
```

Здесь *<интернет-адрес страницы>* может присутствовать, а может и отсутствовать.

Символом вертикальной черты (|) разделяются фрагменты кода, из которых в данном месте должен присутствовать только один.

```
SHAPE="rect|circle|poly"
```

Здесь в качестве значения атрибута `SHAPE` должна присутствовать только одна из доступных строк: `rect`, `circle` или `poly`.

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

```
var langDataObj = xhr2Obj.responseXML.
↳getElementsByTagName("langdata")[0];
```

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

> ***ЕЩЕ РАЗ ВНИМАНИЕ!***
>
> Все приведенные ранее типографские соглашения имеют смысл только в форматах написания тегов HTML, атрибутов стилей CSS и выражений JavaScript. В коде примеров 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.
- 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, кого я забыл здесь перечислить, — за все хорошее.

Часть I

Введение в Web-дизайн и Web-программирование

Глава 1

Что такое Интернет и как он работает

Первая часть этой книги будет посвящена рассмотрению всех интернет-технологий, используемых для создания Web-страниц. Ведь согласитесь — глупо начинать с Web-программирования, не научившись делать сами Web-страницы.

Также глупо начинать с создания Web-страниц, не выяснив, что такое Интернет и как он работает. Так что первая глава книги будет посвящена именно Интернету и основным принципам его работы.

Основные принципы работы Интернета

С принципов работы Интернета мы и начнем. И первым же делом выясним, что же такое Интернет.

Что такое Интернет

В самом деле, что такое *Интернет*? Всемирная компьютерная сеть. (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 кабелей и радиоканалов, связывающих друг с другом многочисленные компьютеры, — вот что такое Интернет. Разумеется, все здесь подчиняется общим стандартам (о которых мы поговорим далее) — 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, ведомственные сети, объединяющие родственные организации, локальные компьютерные

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

Даже частные пользователи, подключающиеся к Интернету по модему, выделенной линии, радиоканалу 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, мы этого не заметим. А все потому, что данные, которые мы запрашиваем, пойдут в этом случае по другому проводу. Специалисты говорят, что Интернет децентрализован — он не имеет единого центра, из которого ведется управление пересылкой данных, поэтому в случае аварии автоматически переконфигурируется и продолжает нормально работать.

Еще одна замечательная особенность Интернета — его глобальность, всемирность. Не вставая из-за компьютера, мы можем совершить путешествие по всему миру, побывать в США, Австралии, Германии, Зимбабве, на Огненной Земле и даже в Антарктиде! Для этого нужно всего лишь набрать нужный нам интернет-адрес.

Интернет имеет достаточно долгую и бурную историю. Он появился еще в конце 60-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 к ученым, а военные приступили к созданию другой сети, которой пользуются до сих пор. И в то же самое время ARPANET был переименован в Internet, или, если по-русски, Интернет.

Первоначально, еще во времена ARPANET, эта сеть использовалась для пересылки электронной почты и обмена файлами. 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, точнее, в СССР, Интернет официально пришел в 1991 году, но популярность среди широких масс компьютерщиков приобрел только в середине 90-х. В настоящее же время в России, наверное, и не найти человека, не слышавшего об Интернете. Вы такого встречали? Автор — еще нет.

НА ЗАМЕТКУ

Говорят, в первой польской энциклопедии, изданной, кажется, в XVII столетии, термин "лошадь" описывался так: "что такое лошадь, знают все". То же самое можно сейчас сказать об Интернете. (Вот только можно ли сейчас сказать то же самое о лошади?..)

Сервисы Интернета

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…

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" (по-английски — snail mail). Конечно, эти доступность, удобство, быстрота и бесплатность имеют и некоторые недостатки, вроде "спама" — несанкционированных рекламных рассылок, но эти недостатки вполне можно стерпеть.

Еще один сервис Интернета, почти такой же старый, как почта, — это пересылка файлов. Пользователи Интернета называют его *FTP* (File Transfer Protocol, протокол передачи файлов; почему так — мы узнаем чуть позже). Сейчас FTP уже не имеет той популярности, как на заре существования Интернета, но все еще довольно часто используется. Так, все крупные корпорации — Intel, Microsoft и др. — помимо Web-сайта, имеют и сервер FTP.

Третий сервис Интернета — это Всемирная паутина, или *WWW* (World Wide Web, повсеместно протянутая паутина), или просто *Web*, те самые Web-страницы и Web-сайты, которые мы просматриваем в Web-обозревателе. Появившийся значительно позже электронной почты и FTP, WWW стала самым популярным сервисом и, собственно, превратила Интернет из сети ученых в сеть для всех.

Об остальных сервисах Интернета (а их немало) мы только упомянем. Это новости UseNet, потоковое вещание, интернет-пейджеры, чаты, нашумевшие в последнее несколько лет файлообменные сети и некоторые другие, менее известные сервисы.

Клиенты и серверы

Но каким образом мы пользуемся всем тем богатством, что дает нам Всемирная сеть? С помощью особых программ! Это Web-обозреватель, клиент электронной почты, программа просмотра интернет-телевидения и прослушива-

ния интернет-радио, интернет-пейджер и "чатилка". Все они очень хорошо нам знакомы.

Но программ, используемых для предоставления нам сервисов Интернета, гораздо больше. И очень многие из них нам, если так можно сказать, "не видны", то есть мы не общаемся с ними напрямую. Вообще, существуют два совершенно разных вида интернет-программ. И сейчас мы о них поговорим.

Программы, относящиеся к первому виду, — это Web-обозреватели, клиенты электронной почты, чатов, интернет-пейджеры, в общем, все те, с которыми мы имеем дело непосредственно. Мы получаем с их помощью различную информацию из Сети и работаем с ней. Такие программы называются программами-*клиентами*, а компьютеры, на которых они работают, — наши с вами компьютеры! — *клиентскими*.

Да, но как программы-клиенты получают из Сети нужную нам информацию (Web-страницы, файлы, письма и пр.)? Очень просто — для этого они обращаются к другим программам, относящимся ко второму виду. Это программы-*серверы*, работающие на *серверных* компьютерах, где также хранится и запрашиваемая клиентами информация. Существуют Web-, FTP-серверы, серверы электронной почты, чата, интернет-пейджеров, потокового вещания и пр.

> ***НА ЗАМЕТКУ***
>
> Очень часто понятие "сервер" распространяется и на серверный компьютер, и на саму программу-сервер. Это, вообще-то, неправильно, так как на одном серверном компьютере может быть установлено несколько программ-серверов, но вошло в практику.

Процесс получения информации клиентами от сервера включает шесть шагов.

1. Пользователь запрашивает с помощью программы-клиента некую информацию, введя в программу интернет-адрес сервера. (Об интернет-адресах мы поговорим потом, а пока будем знать, что это особый адрес, однозначно идентифицирующий нужный нам компьютер в Интернете и работающую на нем серверную программу.)
2. Клиент устанавливает *соединение* (своего рода линию связи) с сервером и посылает тому особый информационный блок, называемый *клиентским запросом*. Этот запрос должен быть определенным образом оформлен, чтобы сервер его понял.
3. Сервер принимает запрос и расшифровывает его.
4. Сервер извлекает запрошенный файл или фрагмент данных, записанных в файле, и посылает его клиенту в составе другого информационного блока — *серверного ответа*. Разумеется, этот ответ также должен быть составлен определенным образом. Если же запрашиваемых клиентом дан-

ных не найдено или сервер почему-то не смог понять клиентский запрос, он возвращает *сообщение об ошибке* — информационный блок, содержащий *код* (числовой номер) и, возможно, текстовое описание возникшей ошибки.

5. Клиент получает ответ от сервера, расшифровывает его и выдает полученную информацию пользователю. Если получено сообщение об ошибке, клиент сообщает об этом пользователю либо предпринимает какие-то действия самостоятельно.
6. Клиент разрывает соединение с сервером.

Процесс отправки клиентом данных серверу также включает шесть шагов.

1. Пользователь вводит в программу-клиент информацию и интернет-адрес сервера, которому она должна быть отправлена.
2. Клиент устанавливает соединение с сервером и посылает тому отправляемую информацию в составе клиентского запроса. При этом отправляемая информация, как правило, особым образом кодируется.
3. Сервер принимает запрос, расшифровывает его и извлекает отправленную информацию.
4. Сервер записывает отправленную клиентом информацию в файл или обрабатывает каким-то образом. В случае успешной записи или обработки он отправляет клиенту так называемое *подтверждение* — информационный блок, сообщающий о том, что все прошло нормально. Если у сервера возникли проблемы с приемом информации, он отправляет 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.
6. Клиент разрывает соединение с сервером.

Весь процесс "общения" клиента и сервера, начиная с отправки клиентом запроса и заканчивая принятием им ответа от сервера, называется *сеансом*. А соединение между клиентом и сервером, устанавливаемое на время этого сеанса и разрываемое после его окончания, называется *сеансовым*, или *временным*.

Любое соединение между клиентом и сервером устанавливается только клиентом. Сервер установить соединение с клиентом не может. Можно сказать, что серверу здесь отведена подчиненная роль.

Мы только что познакомились с особой *архитектурой* (принципом построения компьютерных систем), называемой *двухзвенной*, или архитектурой

"клиент-сервер". Эта архитектура использует два вида программ — клиенты и серверы, — выполняющие разные роли. Она применяется для реализации почти всех современных интернет-сервисов и пока что себя оправдывает.

> ***НА ЗАМЕТКУ***
>
> Некоторые интернет-сервисы, в частности, файлообменные сети (Napster, Gnutella, Kazaa и др.), используют 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, которое может работать и как клиент, и как сервер.

В отличие от клиента, "имеющего дело" с одним-единственным пользователем, сервер работает сразу с множеством пользователей, причем одновременно. Сведения о соединениях, данные, пересылаемые клиентам и принимаемые от клиентов, — все это активно отнимает 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…

Ну да 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.

Протоколы

Люди, чтобы понимать друг друга, должны разговаривать на одном языке. Точно так и с компьютерами, подключенными к сети, неважно какой — всемирной или локальной. Обмен данными по этим сетям должен проходить по единым стандартам, иначе начнется новое вавилонское столпотворение.

Стандарт, согласно которому организуются передаваемые по сети данные и команды, управляющие передачей этих данных, называется *протоколом*. В Интернете для обмена данными используются довольно много протоколов, и некоторые мы здесь вкратце рассмотрим.

Самые, так сказать, фундаментальные протоколы Интернета — *IP* (Internet Protocol, межсетевой протокол) и *TCP* (Transfer Control Protocol, протокол управления передачей). Это так называемые *протоколы низкого уровня*, определяющие самые основные параметры передаваемых данных: длину отдельных порций (*пакетов*) данных, оформление интернет-адресов получателя и отправителя и средства защиты от ошибок. Эти протоколы выполняют самую "грязную" работу по пересылке данных, не вникая, что же именно они передают.

Протокол IP занимается тем, что "упаковывает" особым образом подготовленные данные в пакеты и помещает в каждый пакет интернет-адреса компьютера-отправителя и компьютера-получателя. Протокол TCP, базирующийся на IP, обеспечивает гарантированную отправку данных, то есть следит за тем, чтобы ни один пакет не потерялся в пути, а также занимается подготовкой данных для передачи протоколом IP, в частности, разбивает слишком объемные массивы данных на несколько пакетов и потом собирает их вновь. Эти два протокола настолько взаимосвязаны друг с другом, что часто эту парочку называют одним словом *TCP/IP*, а иногда даже считают за один протокол.

Кстати, протокол IP выполняет еще одну очень важную работу: он делит один реальный, физический, канал связи Интернета (кабель, волоконно-оптическую линию или радиоканал) на несколько воображаемых, виртуальных, "канальчиков", называемых *портами IP*. Всего таких портов предусмотрено 65 535; они идентифицируются по номеру, и любой из них может быть использован для установления соединения и пересылки данных. Более того, разные программы могут одновременно пересылать по одному физическому каналу разные потоки данных, используя разные порты.

TCP/IP применяется другими протоколами, уже *высокого уровня*. Эти протоколы описывают способы оформления клиентских запросов, серверных ответов, подтверждений и сообщений об ошибках, команды, пересылаемые клиентом серверу при запросе или передаче данных, и способ кодирования передаваемой информации. (Собственно передачей этих данных, как мы уже выяснили, занимается "дуэт" TCP/IP.)

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

Каждый сервис Интернета использует свой собственный протокол высокого уровня, а то и 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, отправляемых клиентом (Web-обозревателем) Web-серверу, и способы представления пересылаемых в обе стороны данных. Пожалуй, это самый широкоизвестный протокол Интернета — всем более-менее грамотным интернетчикам знакомы эти четыре буквы.

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, создание папки, копирование, перемещение, удаление файлов и папок и т. д.) и способы кодирования файлов для пересылки по каналам связи. В этом смысле протоколы HTTP и FTP весьма похожи.

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. Для получения же почты от сервера клиент общается с ним по протоколу *POP3* (Post-Office Protocol 3, протокол почты 3).

Существует еще один почтовый протокол — *IMAP* (Internet Message Access Protocol, протокол доступа к почте Интернета). "Коллега" и "наследник" более старого POP3, он предоставляет больше возможностей, но распространен не так широко.

Чуть раньше мы узнали о портах IP. Так вот, каждый существующий протокол высокого уровня использует для передачи данных свой собственный порт (так называемый *порт по умолчанию*). В табл. 1.1 перечислены некоторые протоколы и используемые ими порты по умолчанию.

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

Протокол	Используемый порт IP
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. Это весьма редко, только в особых случаях, но все же применяется. (Например, если на одном серверном компьютере работают два Web-сервера, один из них использует порт по умолчанию — 80-й, — а другой настраивается на использование другого порта — того же 8000-го.)

Интернет-адреса

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

Интернет-адрес — это числовое или строковое значение, позволяющее точно идентифицировать компьютер в Сети. Именно такой интернет-адрес (точнее, два — отправителя и получателя) подставляется в каждый отправляемый по Сети пакет IP, чтобы он успешно дошел до места назначения.

> ***НА ЗАМЕТКУ***
>
> Существует, правда, возможность дать одному компьютеру сразу несколько интернет-адресов. Но используется это нечасто и в особых случаях. В дальнейшем для простоты мы будем считать, что один интернет-адрес — это один компьютер.

На заре эпохи Интернета в качестве интернет-адреса использовался *IP-адрес* — числовое значение, идентифицирующее компьютер для протокола IP. IP-адрес замечательно подходит для компьютеров, но очень плохо — для людей. Вот пример интернет-адреса:

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 имена, о которых стоит поговорить подробно.

Но прежде чем мы начнем разговор о доменных именах, давайте выясним, что такое домен. *Домен*, или *доменная зона*, — это участок Интернета, созданный для удобства управления им. Такой участок может быть крупным, мелким или вообще состоять из одного компьютера. Каждый домен обозначается строкой текста, состоящей из английских букв.

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 роду деятельности организаций, которым они принадлежат; к ним относятся домены com и biz (коммерческие организации), edu (образовательные), mil (военные), org (организации, не занимающиеся компьютерами и Интернетом), net (организации, занимающиеся компьютерами и Интернетом), travel (туристические организации) и некоторые другие. *Национальные домены* объединяют компьютеры по территориальному признаку и выдаются отдельным странам; это домены us (США), uk (Великобритания), fr (Франция), de (Германия), ru (Россия) и др.

Домены нижнего уровня выдаются, как правило, отдельным организациям или, опять же, по территориальному признаку. Их текстовое обозначение часто совпадает с названием этой организации или территориальной единицы (области, района, штата, города и пр.).

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, — office, а страна — ru (Россия), то мы получим такое доменное имя:

comp45.buh.office.ru

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

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

Для этого используется особый сервис Интернета, называемый *DNS* (Domain Name System, система доменных имен). Клиент отправляет *серверу DNS* запрос, содержащий доменное имя, и получает в виде ответа IP-адрес, соответствующий этому доменному имени. А уж с IP-адресом он знает, что делать.

Такие серверы DNS имеются в каждом домене. А несколько самых мощных в мире серверов DNS (*корневые серверы DNS*) обслуживают домены верхнего уровня.

Так, нужный нам компьютер мы задавать научились. А как задать нужную нам серверную программу? Очень просто: перед доменным именем ставим

название протокола, обслуживаемого этой программой, знак двоеточия и два знака / (обратный слеш), вот так (обозначение протокола подчеркнуто):

http://comp45.buh.office.ru

ftp://comp45.buh.office.ru

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

Также имеется возможность указать номер порта IP, через который производится обмен данными. Номер порта записывается после доменного имени серверного компьютера через двоеточие, вот так (подчеркнут):

http://comp45.buh.office.ru:8000

Здесь мы обращаемся к Web-серверу, работающему на компьютере **comp45.buh.office.ru** и использующему для "общения" с клиентом порт 8000.

Многие серверы (почтовые, FTP и др.) требуют от пользователя ввода его имени и, возможно, пароля. Имя пользователя помещается между названием протокола и самим доменным именем и отделяется от последнего знаком @ ("коммерческое эт"). Вот два примера задания имени пользователя в доменном имени сервера (подчеркнуто):

ftp://user@comp45.buh.office.ru

account@server.ru

Последний пример демонстрирует нам обычный адрес электронной почты. Заметим, что название протокола здесь не указывается — почтовый клиент и почтовый сервер сами знают, какой протокол использовать.

Ну а пароль пользователя помещается между именем и знаком @ и отделяется от имени двоеточием — вот так (подчеркнут):

ftp://user:password@comp45.buh.office.ru

Ну вот, с основными принципами работы Интернета мы ознакомились. Теперь давайте сосредоточимся на WWW — именно этим сервисом мы будем пользоваться на протяжении всей книги.

Основные понятия WWW

Здесь мы узнаем все о Web-страницах и Web-сайтах, выясним, чем сайт отличается от страницы, поговорим о Web-обозревателях и Web-серверах и изучим множество новых терминов.