

1

Введение в динамическое содержимое веб-страницы

Всемирная паутина — это непрерывно развивающаяся сеть, ушедшая далеко вперед от своей концепции ранних 1990-х, когда ее создание было обусловлено решением конкретных задач. Высокотехнологичные эксперименты в ЦЕРНе (Европейском центре физики высоких энергий, известном в наши дни в качестве обладателя Большого адронного коллайдера) выдавали невероятно большой объем данных, который был слишком велик для распространения среди участвующих в экспериментах ученых, разбросанных по всему миру.

К тому времени Интернет уже существовал и к нему было подключено несколько сотен тысяч компьютеров, поэтому Тим Бернерс-Ли (специалист ЦЕРНа) придумал способ навигации между ними с использованием среды гиперссылок — так называемого протокола передачи гиперссылок (Hyper Text Transfer Protocol, HTTP). Он также создал специальный язык разметки, названный языком гипертекстовой разметки (Hyper Text Markup Language, HTML). Для того чтобы собрать все это воедино, он создал первые браузер и веб-сервер.

Теперь эти средства воспринимаются нами как должное, но в то время концепция их применения носила революционный характер. До этого основной объем соединений приходился на пользователей домашних модемов, дозванивавшихся и подключававшихся к электронным доскам объявлений, которые базировались на отдельном компьютере и позволяли общаться и обмениваться данными только с другими пользователями этой службы. Следовательно, для эффективного электронного общения с коллегами и друзьями нужно было становиться участником многих электронных досок объявлений.

Но Бернерс-Ли изменил все это одним махом, и к середине 1990-х годов уже существовали три основных конкурирующих друг с другом графических браузера, пользовавшихся вниманием 5 млн посетителей. Однако вскоре стало очевидно,

что кое-что было упущено. Конечно, текстовые и графические страницы, имеющие гиперссылки для перехода на другие страницы, были блестящей концепцией, но результаты не отражали текущий потенциал компьютеров и Интернета по удовлетворению насущных потребностей пользователей в динамическом изменении контекста. Всемирная паутина оставляла весьма невыразительное впечатление, даже при наличии прокрутки текста и анимированных GIF-картинок. Корзины покупателей, поисковые машины и социальные сети внесли существенные коррективы в порядок использования Всемирной паутины. В этой главе будет дан краткий обзор различных компонентов, формирующих ее облик, и программного обеспечения, способствующего обогащению и оживлению наших впечатлений от ее использования.



Пришло время воспользоваться аббревиатурами. Прежде чем делать это, я старался дать им четкое объяснение. Но если сразу не удастся разобраться, какое именно понятие они замещают или что означают, переживать не стоит, поскольку все подробности прояснятся по мере чтения книги.

HTTP и HTML: основы, заложенные Бернерсом-Ли

HTTP представляет собой стандарт взаимодействия, регулирующий порядок направления запросов и получения ответов — процесса, происходящего между браузером, запущенным на компьютере конечного пользователя, и веб-сервером. Задача сервера состоит в том, чтобы принять запрос от клиента и попытаться дать на него содержательный ответ, обычно передавая ему запрошенную веб-страницу. Именно поэтому и используется термин «*сервер*» («обслуживающий»). Партнером, взаимодействующим с сервером, является *клиент*, поэтому данное понятие применяется как к браузеру, так и к компьютеру, на котором он работает.

Между клиентом и сервером может располагаться ряд других устройств, например маршрутизаторы, модули доступа, шлюзы и т. д. Они выполняют различные задачи по обеспечению безошибочного перемещения запросов и ответов между клиентом и сервером. Как правило, для отправки этой информации используется Интернет. Некоторые из этих промежуточных устройств могут также ускорить Интернет путем локального сохранения страниц или информации в так называемом кэше, обслуживая затем данное содержимое для клиентов непосредственно из кэша без постоянного извлечения его из сервера-источника.

Обычно веб-сервер может обрабатывать сразу несколько подключений, а при отсутствии связи с клиентом он находится в режиме ожидания входящего подключения. При поступлении запроса на подключение сервер подтверждает его получение отправкой ответа.

Процедура «запрос — ответ»

В наиболее общем виде процесс «запрос — ответ» состоит из просьбы браузера к веб-серверу отправить ему веб-страницу и выполнения браузером данной просьбы. После этого браузер занимается отображением страницы (рис. 1.1).

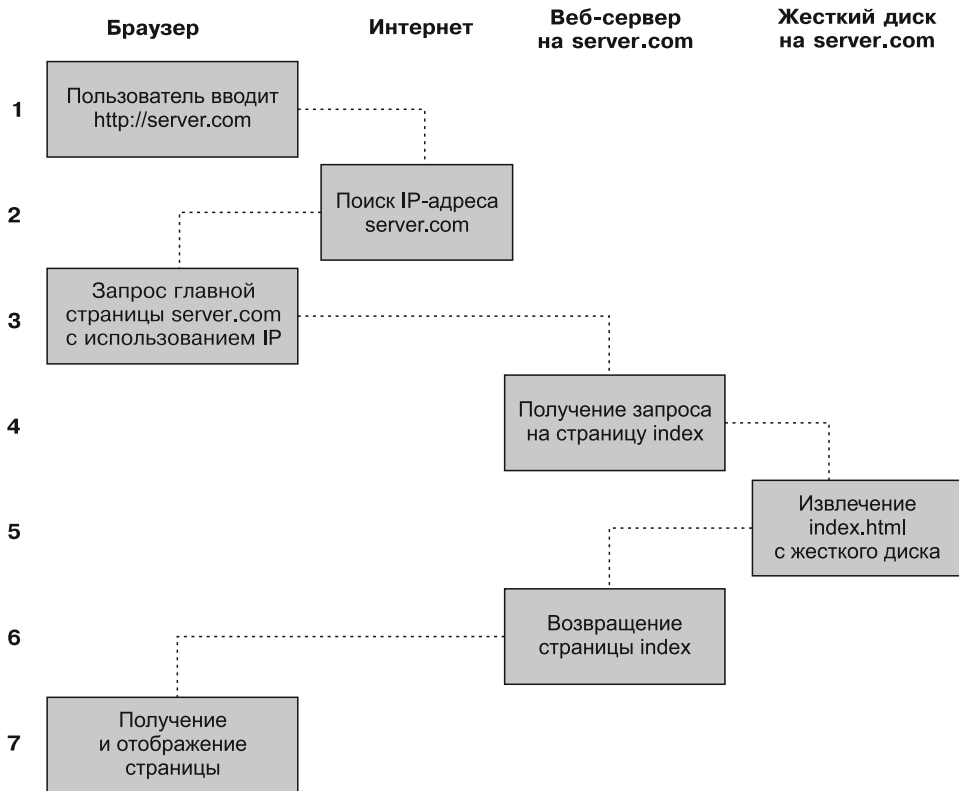


Рис. 1.1. Основная последовательность процесса «запрос — ответ» между клиентом и сервером

При этом соблюдается такая последовательность действий.

1. Вы вводите в адресную строку браузера `http://server.com`.
2. Ваш браузер ищет IP-адрес, соответствующий доменному имени `server.com`.
3. Браузер посылает запрос на главную страницу `server.com`.
4. Запрос проходит по Интернету и поступает на веб-сервер `server.com`.
5. Веб-сервер, получивший запрос, ищет веб-страницу на своем жестком диске. Сервер извлекает веб-страницу и отправляет ее по обратному маршруту в адрес браузера.
6. Браузер отображает веб-страницу.

При передаче типовой веб-страницы этот процесс также осуществляется для каждого имеющегося на ней объекта: элемента графики, встроенного видео- или Flash-ролика и даже шаблона CSS.

Обратите внимание на то, что на шаге 2 браузер ищет IP-адрес, принадлежащий доменному имени `server.com`. У каждой машины, подключенной к Интернету, включая и ваш компьютер, есть свой IP-адрес. Но, как правило, доступ к веб-серверам осуществляется по именам, таким как `google.com`. Вам должно быть известно, что браузер обращается к вспомогательной интернет-службе, так называемой службе доменных имен (Domain Name Service, DNS), чтобы найти связанный с сервером IP-адрес, а затем воспользоваться им для связи с компьютером.

При передаче динамических веб-страниц процедура состоит из большего количества действий, поскольку к ней могут привлекаться как PHP, так и MySQL. Например, можно щелкнуть кнопкой мыши на картинке с изображением плаща. После этого PHP составит запрос, используя стандартный язык базы данных, SQL — множество используемых для этого команд будет рассмотрено в книге, — и отправит запрос в адрес MySQL-сервера. Этот сервер возвратит информацию о выбранном вами плаще, и PHP-код заключит ее в некий код HTML, который будет отправлен сервером в адрес вашего браузера (рис. 1.2).

Выполняется такая последовательность действий.

1. Вы вводите в адресную строку браузера `http://server.com`.
2. Ваш браузер ищет IP-адрес, соответствующий доменному имени `server.com`.
3. Браузер посылает запрос на главную страницу `server.com`. Запрос проходит по Сети и поступает на веб-сервер `server.com`.
4. Веб-сервер, получивший запрос, ищет веб-страницу на своем жестком диске.
5. Теперь, когда главная страница размещена в его памяти, веб-сервер замечает, что она представлена файлом, включающим в себя PHP-сценарии, и передает страницу интерпретатору PHP.
6. Интерпретатор PHP выполняет PHP-код.
7. Кое-какие фрагменты кода PHP содержат MySQL-инструкции, которые интерпретатор PHP, в свою очередь, передает процессору базы данных MySQL.
8. База данных MySQL возвращает результаты выполнения инструкции интерпретатору PHP.
9. Интерпретатор PHP возвращает веб-серверу результаты выполнения кода PHP, а также результаты, полученные от базы данных MySQL.
10. Веб-сервер возвращает страницу выдавшему запрос клиенту, который отображает эту страницу на экране.

Конечно, ознакомиться с этим процессом и узнать о совместной работе трех элементов не помешает, но на практике эти подробности не понадобятся, поскольку все происходит в автоматическом режиме.

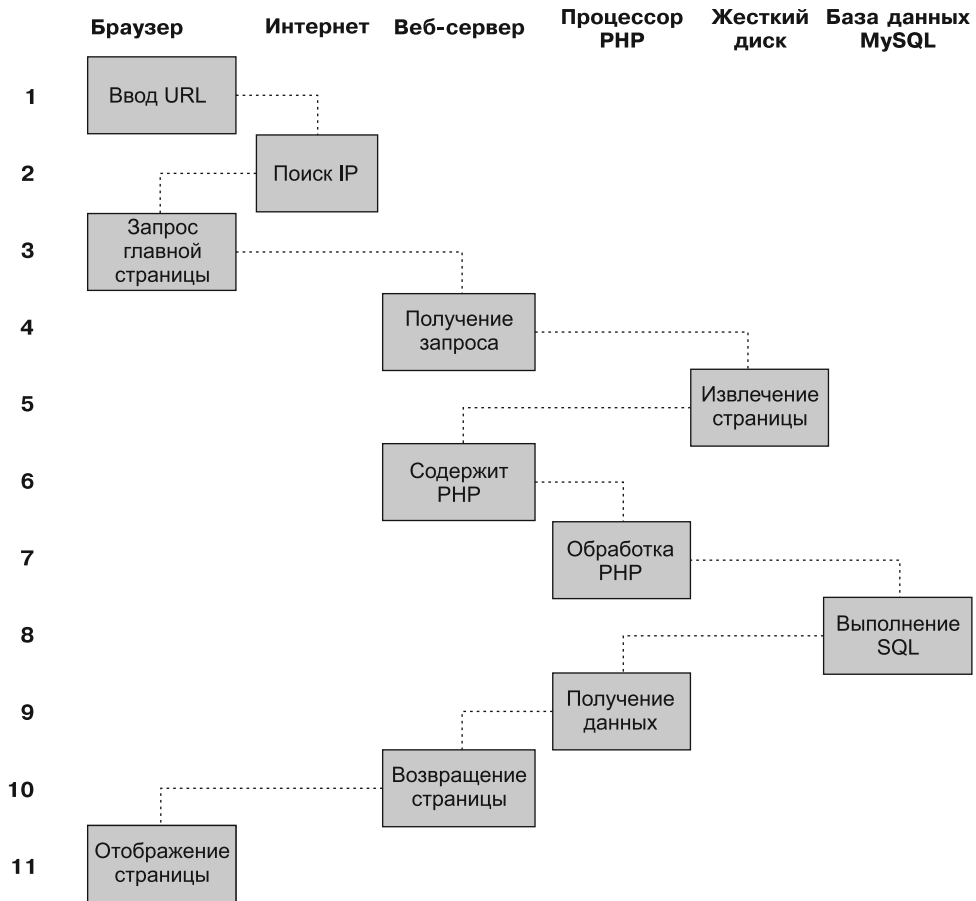


Рис. 1.2. Динамическая последовательность процесса «запрос — ответ», выполняемого клиентом и сервером

В каждом из примеров возвращенные браузеру HTML-страницы могут содержать также код JavaScript, интерпретируемый локально на машине клиента. Этот код может инициировать еще один запрос, точно так же запрос может быть инициирован встроенными объектами, например изображениями.

Преимущества использования PHP, MySQL, JavaScript и CSS

В начале этой главы был представлен мир технологии Web 1.0, но рывок к созданию технологии Web 1.1, вместе с которой были разработаны такие браузерные расширения, как Java, JavaScript, JScript (несколько иной вариант JavaScript от корпорации Microsoft) и ActiveX, не заставил себя долго ждать. На серверной стороне

прогресс был обеспечен за счет общего шлюзового интерфейса (Common Gateway Interface, CGI), использования таких языков сценариев, как Perl (альтернатива языку PHP), и выполнения *сценариев на стороне сервера* — динамической вставки содержимого одного файла (или выходных данных выполняемой локальной программы) в другой файл.

Когда ситуация окончательно прояснилась, на передовых позициях остались три основные технологии. Несмотря на то что язык сценариев Perl силами своих стойких приверженцев сохранил популярность, простота PHP и допустимость использования в нем встроенных ссылок на программу базы данных MySQL обеспечили этому языку более чем двойное превосходство по количеству пользователей. А JavaScript, ставший важнейшей составной частью уравнения, используемого для динамического манипулирования каскадными таблицами стилей (Cascading Style Sheets, CSS) и HTML, в настоящее время берет на себя наиболее трудоемкие задачи асинхронного обмена данными (осуществляемого между клиентом и сервером после загрузки веб-страницы). Используя асинхронный обмен данными, веб-страницы обрабатывают данные и отправляют запросы веб-серверу в фоновом режиме, не оповещая пользователя о происходящем.

Несомненно, своеобразный симбиоз PHP и MySQL способствует их продвижению, но что привлекает к ним разработчиков в первую очередь? На это следует дать простой ответ: та легкость, с которой эти технологии можно использовать для быстрого создания на сайтах динамических элементов. MySQL является быстродействующей и мощной, но при этом простой в использовании системой базы данных, предлагающей сайту практически все необходимое для поиска и обработки данных, которые предназначены для браузеров. Когда PHP для хранения и извлечения этих данных выступает в союзе с MySQL, вы получаете основные составляющие, необходимые для разработки сайтов социальных сетей и для перехода к технологии Web 2.0.

И когда вы также соедините вместе JavaScript и CSS, у вас появится рецепт для создания высокодинамичных и интерактивных сайтов, особенно в современных условиях доступности множества сложных функциональных сред JavaScript, вызов которых действительно позволяет ускорить веб-разработку. Достаточно упомянуть широко известную библиотеку JQuery, применение которой в настоящее время, наверное, является наиболее востребованным способом доступа программистов к функциям асинхронного обмена данными.

MariaDB: клон MySQL

После того как компания Oracle приобрела Sun Microsystems (владельца MySQL), возникли опасения, что полностью открытым код MySQL может не остаться, и в итоге от этой СУБД отпочковалась MariaDB, дабы код оставался открытым в соответствии с положениями лицензии GNU GPL. Разработка MariaDB шла под

руководством ряда первоначальных создателей MySQL, и эта СУБД сохранила максимальную совместимость с MySQL. Поэтому вероятность встречи на некоторых серверах MariaDB вместо MySQL весьма высока, что не должно вызывать никаких опасений, поскольку все показанное в данной книге одинаково успешно работает как под MySQL, так и под MariaDB, имеющей ту же кодовую базу, что и MySQL Server 5.5. В любом случае одна СУБД может заменяться другой, и вы при этом не заметите никакой разницы.

Впрочем вышло так, что многие возникшие поначалу опасения были напрасны, поскольку код MySQL остался открытым, а компания Oracle просто сделала платной приобретение и поддержку тех выпусков, которые предоставляют дополнительные функциональные возможности, включающие георепликацию и автоматическое масштабирование. Тем не менее, в отличие от MariaDB, MySQL больше не поддерживается сообществом, но осознание того, что MariaDB никогда не лишится этой поддержки, позволит многим разработчикам спать спокойно и, вероятно, даст гарантии того, что код самой MySQL останется открытым.

PHP

Использование PHP существенно упрощает встраивание средств, придающих веб-страницам динамические свойства. Когда страницам присваивается расширение PHP, у них появляется прямой доступ к языку сценариев. Разработчику нужно лишь написать код, похожий на этот:

```
<?php
echo " Today is " . date('T') . ". ";
?>
Here's the latest news.
```

Открывающий тег `<?php` дает веб-серверу разрешение на интерпретацию всего последующего кода вплоть до тега `?>`. Все, что находится за пределами этой конструкции, отправляется клиенту в виде простого HTML. Поэтому текст `Here's the latest news` просто выводится в браузер. А внутри PHP-тегов встроенная функция `date` отображает текущий день недели, соответствующий системному времени сервера.

В итоге на выходе из этих двух частей получается примерно следующее:

Today is Wednesday. Here's the latest news.

PHP — довольно гибкий язык, и некоторые разработчики предпочитают помещать PHP-конструкцию непосредственно рядом с кодом PHP, как в этом примере:

```
Today is <?php echo date("l"); ?>. Here's the latest news.
```

Существуют также другие способы форматирования и вывода информации, которые будут рассмотрены в главах, посвященных PHP. Важно усвоить то,

что, используя PHP, веб-разработчики получают язык сценариев, который хотя и не обладает быстротой кода, скомпилированного на C или ему подобных языках, но все же работает невероятно быстро и к тому же очень хорошо вписывается в разметку HTML.



Если вы собираетесь набирать встречающиеся в этой книге примеры на PHP в программе-редакторе, чтобы работать параллельно с моим повествованием, не забывайте предварять их тегом `<?php`, а в конце ставить тег `?>`, для того чтобы обеспечить их обработку интерпретатором PHP. Для упрощения этой задачи можно заранее подготовить файл `example.php`, содержащий эти теги.

Используя PHP, вы получаете средство управления своим веб-сервером с неограниченными возможностями. Если понадобится на лету внести изменения в HTML, обработать данные кредитной карты, добавить сведения о пользователе в базу данных или извлечь информацию из стороннего сайта, все это можно будет сделать из тех же самых PHP-файлов, в которых находится и сам код HTML.

MySQL

Разумеется, без средств отслеживания информации, предоставляемой пользователем в ходе работы с вашим сайтом, нельзя в полной мере говорить о возможностях динамического изменения выходного кода HTML. На заре создания Всемирной паутины многие сайты использовали неструктурированные текстовые файлы для хранения таких данных, как имена пользователей и пароли. Но подобный подход мог вызвать ряд проблем, если файл не был надежно заблокирован от повреждений, возникающих при одновременном доступе к нему множества пользователей. К тому же неструктурированный файл мог разрастаться до таких размеров, что с ним непросто было работать, не говоря уже о трудностях, связанных с попытками объединения файлов и осуществления в них сложных поисковых операций за какое-нибудь мало-мальски приемлемое время.

Именно в таких случаях большое значение приобретает использование реляционных баз данных со структурированной системой запросов. И MySQL, будучи совершенно бесплатной и установленной на огромном количестве веб-серверов системой, оказывается как нельзя кстати. Она представляет собой надежную и исключительно быстродействующую систему управления базами данных, использующую команды, похожие на простые английские слова.

Высшим уровнем структуры MySQL является база данных, внутри которой можно иметь одну или несколько таблиц, содержащих ваши данные. Предположим, вы работаете над таблицей под названием `users` (пользователи), внутри которой были созданы столбцы для фамилий — `surname`, имен — `firstname` и адресов электронной

почты — `email`, и теперь нужно добавить еще одного пользователя. Одна из команд, которую можно применить для этого, выглядит следующим образом:

```
INSERT INTO users VALUES('Smith', 'John', 'jsmith@mysite.com');
```

Для создания базы данных и таблицы, а также настройки всех нужных полей понадобится сначала выдать и другие команды, но используемая здесь SQL-команда `INSERT` демонстрирует простоту добавления в базу данных новой информации. `Structured Query Language (SQL)` является языком, разработанным в начале 1970-х годов и напоминающим один из старейших языков программирования — `COBOL`. Тем не менее он хорошо подходит для запросов к базе данных, что и предопределило его использование в течение столь длительного времени.

Так же просто выполняется и поиск данных. Предположим, что имеется адрес электронной почты пользователя и нужно найти имя его владельца. Для этого можно ввести следующий запрос MySQL:

```
SELECT surname,firstname FROM users WHERE email='jsmith@mysite.com';
```

После этого MySQL вернет `Smith, John` и любые другие пары имен, которые могут быть связаны в базе данных с адресом электронной почты.

Нетрудно предположить, что возможности MySQL простираются значительно дальше выполнения простых команд вставки и выбора — `INSERT` и `SELECT`. Например, можно скомбинировать родственные наборы данных, чтобы собрать вместе взаимосвязанные части информации, запросить результаты, выбрав порядок их выдачи из множества вариантов, найти частичные совпадения, если известна только часть искомой строки, вернуть конкретно заданное количество результатов и сделать многое другое.

При использовании PHP все эти вызовы можно направлять непосредственно к MySQL без необходимости самостоятельного применения ее интерфейса командной строки. Это значит, что для того, чтобы докопаться до нужного вам элемента данных, вы можете сохранять результаты в массивах для их обработки и осуществления множества поисковых операций, каждая из которых зависит от результатов, возвращенных предыдущими операциями.

Далее будет показано, что для придания еще большей мощности прямо в MySQL встроено несколько дополнительных функций, которые можно вызвать с целью эффективного выполнения наиболее часто встречающихся в MySQL операций, избавляясь от необходимости их составления из нескольких PHP-вызовов в адрес MySQL.

JavaScript

Самая старая из трех основных технологий, рассматриваемых в данной книге, — JavaScript — была создана для получения доступа из сценариев ко всем элементам HTML-документа. Иными словами, она предоставляет средства для динамического

взаимодействия с пользователем, например для проверки приемлемости адресов электронной почты в формах ввода данных, отображения подсказок наподобие «Вы действительно подразумевали именно это?» и т. д. (хотя с точки зрения безопасности, которая всегда должна реализовываться на веб-сервере, на эту технологию положиться нельзя).

В сочетании с CSS JavaScript закладывает основу мощности динамических веб-страниц, которые изменяются буквально на глазах, в отличие от новой страницы, возвращаемой сервером.

Тем не менее с использованием JavaScript могут возникнуть осложнения, обусловленные некоторыми существенными различиями в способах реализации этого языка, выбранных разными разработчиками браузеров. В основном эти различия возникают, когда некоторые производители пытаются придать своим браузерам дополнительные функциональные возможности, не обращая внимания на совместимость с продуктами своих конкурентов.

К счастью, разработчики в большинстве своем уже взяли себя за ум, и теперь оптимизация вашего кода для различных браузеров утратила прежнюю актуальность. Но остаются миллионы экземпляров устаревших браузеров, которыми будут пользоваться на протяжении еще многих лет. Тем не менее и для них существуют решения проблем несовместимости, и позже в этой книге будут рассмотрены библиотеки и технологии, позволяющие без каких-либо опасений игнорировать существующие различия.

А сейчас взглянем на то, как можно воспользоваться обычным JavaScript-кодом, воспринимаемым всеми браузерами:

```
<script type="text/javascript">
  document.write("Today is " + Date() );
</script>
```

Этот фрагмент кода предписывает браузеру интерпретировать все, что находится внутри тегов `<script>`, в качестве кода JavaScript, что затем браузер и сделает, записав в текущий документ текст "Today is ", а также дату, полученную за счет использования принадлежащей JavaScript функции `Date`. В результате получится нечто подобное следующему:

Today is Sun Jan 01 2017 01:23:45



Если не требуется указывать конкретную версию JavaScript, то, как правило, можно опустить `type="text/javascript"` и использовать для начала интерпретации JavaScript тег `<script>`.

Ранее было упомянуто, что изначально JavaScript разрабатывался для того, чтобы получить возможность динамического управления различными элементами, нахо-

дящимися внутри HTML-документа, и это его предназначение по-прежнему является основным. Но все чаще JavaScript применяется для осуществления процесса доступа к веб-серверу в фоновом режиме с целью *асинхронного обмена данными*.

Асинхронный обмен данными позволил веб-страницам стать похожими на автономные программы, поскольку для отображения нового содержимого их уже не нужно загружать целиком. Вместо этого асинхронный вызов может использоваться для извлечения и обновления отдельно взятого элемента веб-страницы, например может быть изменена ваша фотография на сайте социальной сети или заменена кнопка, на которой нужно щелкнуть, отвечая на вопрос. Полностью эта тема будет рассмотрена в главе 17.

Затем в главе 21 мы присмотримся к среде jQuery, которую можно использовать, чтобы не изобретать колесо в случае возникновения потребностей в быстродействующем, кросс-браузерном коде для управления веб-страницами. Конечно, доступны и другие подобные среды, но jQuery является самой популярной библиотекой и, исходя из показателей, накопленных за весьма длительный срок ее использования, обладает исключительной надежностью и является основным средством в арсенале многих опытных разработчиков.

CSS

CSS является ключевым дополнением к HTML, обеспечивая соответствующую разметку HTML-текста и встроенных изображений, соотносясь с параметрами экрана пользователя. После появления третьего стандарта (CSS3) CSS предлагает уровень динамической интерактивности, которая прежде поддерживалась только с помощью JavaScript. Например, вы можете не только придать стиль любому элементу HTML, чтобы изменить его размеры, цвета, границы, интервалы, но и, используя всего лишь несколько строк CSS, добавить своим веб-страницам анимированные переходы и преобразования.

Применение CSS может просто заключаться во вставке правил между тегами `<style>` и `</style>`, расположенными в заголовке веб-страницы:

```
<style>
  p{
    text-align:justify;
    font-family:Helvetica;
  }
</style>
```

Эти правила будут изменять исходное выравнивание текста тега `<p>`, чтобы содержащиеся в нем абзацы были полностью выровнены и для них использовался шрифт Helvetica.

В главе 18 вы увидите, что существует множество различных способов задания правил CSS и их также можно включать непосредственно в теги или сохранять во

внешнем файле, предназначенном для отдельной загрузки. Такая гибкость позволяет проводить точную настройку стиля HTML. Вы также увидите, как с помощью CSS можно, например, создать встроенную функцию `hover` для анимирования объектов при проходе над ними указателя мыши. Кроме того, вы научитесь получать доступ ко всем свойствам CSS-элемента из JavaScript и из HTML.

А теперь HTML5

Какими бы ни были полезными все эти дополнения к веб-стандартам, самым амбициозным разработчикам и их было мало. К примеру, так и не был придуман простой способ работы с графикой в браузере, не требующий обращения к таким дополнительным модулям, как Flash. То же самое происходило и в отношении аудио- и видеовставок в веб-страницы. Кроме того, можно отметить множество досадных несоответствий, вкравшихся в HTML в процессе его развития.

Итак, чтобы подчистить все эти шероховатости, перенести Интернет за пределы технологии Web 2.0 в его следующую фазу развития, был создан новый стандарт HTML, устраняющий перечисленные недостатки: *HTML5*. Он начал разрабатываться в далеком 2004 году, когда Mozilla Foundation и Opera Software (разработчики двух популярных браузеров) составили его первый проект. Но его окончательный проект был представлен World Wide Web Consortium (W3C), международной организацией, руководящей веб-стандартами, лишь в начале 2013 года.

На разработку HTML5 ушло несколько лет, но теперь мы располагаем весьма надежной и стабильной версией 5.1 (с 2016 года). Но цикл разработки никогда не заканчивается, и со временем в этот язык несомненно будут встроены дополнительные функциональные возможности. К некоторым наиболее ярким свойствам HTML5, позволяющим управлять медиаресурсами, можно отнести элементы `<audio>`, `<video>` и `<canvas>`, с помощью которых происходит добавление звука, видео и усовершенствованной графики. Все, что нужно знать об этих и всех других аспектах HTML5, весьма подробно изложено в данной книге, начиная с главы 23.



Кроме всего прочего, мне в спецификации HTML5 нравится, что для самозакрывающихся элементов больше не нужен синтаксис XHTML. Раньше перевод на новую строку можно было изобразить с помощью элемента `
`. Затем для обеспечения совместимости в будущем с XHTML (так и не состоявшейся заменой HTML) элемент был изменен на `
` с добавленным символом `/` (поскольку ожидалось, что характерной особенностью закрывающего тега всех элементов станет именно этот символ). Но теперь все вернулось на круги своя и можно использовать любую из версий таких элементов. Итак, для большей лаконичности и меньшего объема набираемого текста в данной книге я вернулся к прежнему стилю: `
`, `<hr>` и т. д.
