Краткий обзор средств поддержки серверных сценариев

ГЛАВА

2

В этой главе...

Настоящая глава посвящена описанию проблематики создания серверных сценариев и ее связи со статическими HTML-страницами и наиболее распространенными клиентскими технологиями. Кроме того, эта глава дает возможность приобрести четкое понимание того, каких целей позволяет и не позволяет достичь язык PHP, а также ознакомиться с общими сведениями о том, как можно организовать взаимодействие кода PHP с клиентским кодом (со сценариями JavaScript, аплетами Java, объектами Flash, таблицами стилей и т.д.).

Статические HTML-страницы

Веб-страницы наиболее простого типа представляют собой статические и текстовые страницы, написанные полностью на языке HTML. Рассмотрим в качестве примера простую страницу HTML, включающую только код HTML, которая показана на рис. 2.1.

Исходный код веб-страницы, показанной на рис. 2.1, приведен ниже.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01
//EN" "http://www.w3.org/TR/html4/strict.dtd">
<html>
<head>
<title>Selected Constellations</title>
</head>
<body>
<h1>Constellations</h1>
<ul>
<li><a href="Aquila.html">Aquila</a></li>
```

```
<li><a href="Bootes.html">Bootes</a></li>
<li><a href="Cassiopeia.html">Cassiopeia</a></li>
<li><a href="Cygnus.html">Cygnus</a></li>
<li><a href="Deneb.html">Deneb</a></li>
<li><a href="Draco.html">Draco</a></li>
<li><a href="Gemini.html">Gemini</a></li>
<li><a href="Leo.html">Leo</a></li>
<li><a href="Libra.html">Libra</a></li>
<li><a href="Lynx.html">Lynx</a></li>
<li><a href="Orion.html">Orion</a></li>
<li><a href="Pegasus.html">Pegasus</a></li>
<li><a href="Perseus.html">Perseus</a></li>
<li><a href="Pisces.html">Pisces</a></li>
<li><a href="Taurus.html">Taurus</a></li>
<li><a href="Ursa_Major.html">Ursa Major</a></li>
<li><a href="Ursa_Minor.html">Ursa Minor</a></li>
<li><a href="Vega.html">Vega</a></li>
</ul>
</body>
</html>
```

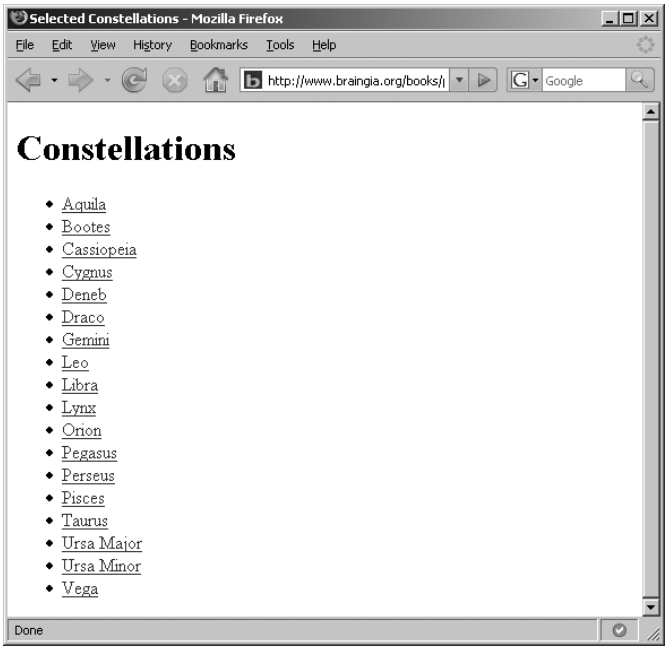

Рис. 2.1. Пример статической HTML-страницы

Клиентские технологии

Наиболее часто дополнения к простому коду HTML реализуются после поступления страницы в клиентское приложение. К этим дополнениям относятся такие расширения, обеспечивающие лучшее форматирование, как каскадные таблицы стилей и средства Dynamic HTML, а также клиентские языки сценариев, такие как JavaScript и VBScript, аплеты Java и объекты Flash. Средства поддержки всех этих технологий должны быть встроены в программное

обеспечение веб-браузера (но в некоторых версиях указанного программного обеспечения подобные средства поддержки отсутствуют). Клиентские технологии применяются для выполнения задач, перечисленных в табл. 2.1; очевидно, что области использования некоторых технологий перекрываются.

Таблица 2.1. Клиентские расширения HTML

Клиентская технология	Основные области применения	Примеры эффектов
Каскадные таблицы стилей (Cascading Style Sheet — CSS), Dynamic HTML	Форматирование страниц: управление размером, цветом, размещением, компоновкой и временем показа элементов	Перекрытие, шрифты с изменяющимися цветами и размерами. Слои, точное позиционирование
Создание клиентских сценариев (JavaScript, VBScript)	Обработка событий: управление действиями, следующими за возникновением определенных событий	Ссылка, цвет которой изменяется при перемещении указателя мыши. Калькулятор выплат по ипотеке
Аплеты Java	Предоставление в распоряжение пользователя небольших автономных приложений	Перемещающийся логотип. Кроссворд
Средства создания анимационных изображений Flash	Анимация	Короткий анимационный фильм

Страница, показанная на рис. 2.2, основана на том же содержимом, что и страница на рис. 2.1. Но в этом примере дополнительно введено некоторое оформление в виде стилей с помощью встроенного кода CSS, как показано в приведенном ниже исходном коде.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
<html>
<head>
<STYLE TYPE="text/css">
BODY, P {color: black; font-family: verdana; font-size: 10 pt}
H1 {margin-top: 10; color: black; font-family: arial; font-size: 12 pt}
H2 {margin-bottom: -10; color: black; font-family: verdana; font-size: 18 pt}
A:link, A:visited {color: #000080; text-decoration: none}
</STYLE>
<title>Selected Constellations</title>
</head>
<body>
<h1>Constellations</h1>
<ul>
<li><a href="Aquila.html">Aquila</a></li>
<li><a href="Bootes.html">Bootes</a></li>
<li><a href="Cassiopeia.html">Cassiopeia</a></li>
<li><a href="Cygnus.html">Cygnus</a></li>
<li><a href="Deneb.html">Deneb</a></li>
<li><a href="Draco.html">Draco</a></li>
<li><a href="Gemini.html">Gemini</a></li>
<li><a href="Leo.html">Leo</a></li>
<li><a href="Libra.html">Libra</a></li>
<li><a href="Lynx.html">Lynx</a></li>
<li><a href="Orion.html">Orion</a></li>
<li><a href="Pegasus.html">Pegasus</a></li>
<li><a href="Perseus.html">Perseus</a></li>
<li><a href="Pisces.html">Pisces</a></li>
<li><a href="Taurus.html">Taurus</a></li>
<li><a href="Ursa_Major.html">Ursa Major</a></li>
```

```
<li><a href="Ursa_Minor.html">Ursa Minor</a></li>
<li><a href="Vega.html">Vega</a></li>
</ul>
</body>
</html>
```

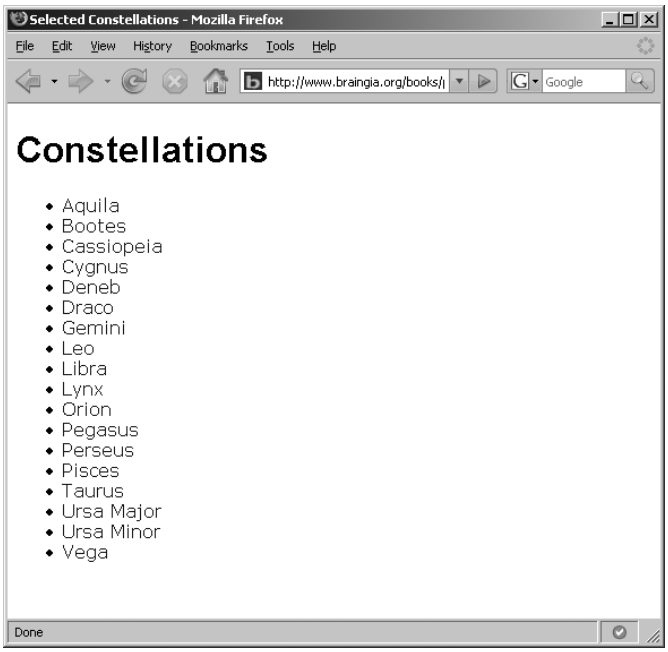

Рис. 2.2. Пример совместного применения кода HTML и каскадных таблиц стилей (CSS)

Как описано выше, клиентские технологии обладают значительными преимуществами, но, к сожалению, из этих преимуществ следует также их серьезный недостаток — поддержка этих технологий полностью зависит от браузера. Браузеры различных типов во многом отличаются друг от друга по своим возможностям, и такие различия наблюдаются даже между разными версиями браузеров одного и того же типа. Кроме того, отдельные пользователи способны также выбирать такие способы настройки конфигурации своих браузеров, которые исключают всю возможность дальнейшей поддержки определенных клиентских технологий. Например, иногда пользователи по соображениям безопасности запрещают работу с языком JavaScript, что исключает для них возможность просматривать сайты, в которых язык JavaScript используется не совсем правильно или недостаточно продуманно.

Предусмотрительный разработчик программ для веб должен учитывать не только эти факторы, но и принимать во внимание то, что стал значительно шире круг устройств, применяемых для просмотра веб-страниц, доступ к веб предоставляется буквально по всему миру, а количество пользователей достигло глобальных масштабов. Кроме того, настоящим бедствием для разработчиков клиентского кода является упорное нежелание широкой публики переходить на новые программные и аппаратные средства; из-за этого остаются нереализованными многие замечательные замыслы. Несмотря на то что уже почти десять лет происходит буквально взрывообразное развитие веб, единственное, в чем разработчик может быть полностью и абсолютно уверен, — что пользователи всегда смогут просматривать простой текстовый код HTML (или, скорее, код, состоящий из конструкций языка HTML, которые широко поддерживаются, выдержали проверку временем и доказали свою полезность).

Средства поддержки серверных сценариев

Клиентские средства поддержки сценариев относятся к самой заметной и привлекательной части разработок для веб. С другой стороны, работа создателей серверных сценариев незаметна для пользователей. Составители сценариев для серверов трудятся в полной безвестности, пребывая на ничейной земле между веб-сервером и базой данных, в то время как их собратья-программисты, разрабатывающие клиентский код, непринужденно демонстрируют свои достижения перед восхищенными пользователями веб-сайтов.

Средства поддержки серверных сценариев для веб в основном применяются для подключения веб-сайтов к таким вспомогательным серверам, как серверы баз данных, обработки данных и управления поведением средств поддержки языков более высокого уровня, таких как HTML и CSS. Благодаря использованию этих средств появляется возможность обеспечивать двухстороннюю связь описанных ниже типов.

- **От сервера к клиенту.** Сборка веб-страниц может осуществляться на основе выходных данных, полученных из вспомогательного сервера.
- **От клиента к серверу.** На сервере может осуществляться обработка информации, введенной пользователем.

К числу распространенных примеров взаимодействия клиента и сервера относятся оперативные формы с заранее заданными раскрывающимися списками (обычно такими, которые требуют щелчка мышью), которые формируются динамически на сервере с помощью сценариев.

Программные продукты поддержки серверных сценариев состоят из двух основных компонентов — языка и обработчика сценариев (который может быть встроен или не встроен в программное обеспечение веб-сервера). Такой обработчик обеспечивает синтаксический анализ и интерпретацию страниц, написанных на определенном языке.

В следующем коде показан простой пример применения серверных сценариев для динамического формирования страницы на основе данных, полученных из базы данных. В код этой страницы включены вызовы функций доступа к базе данных (описание которых будет приведено в части II), к тому же некоторые включаемые файлы остаются нераскрытыми, поскольку данный пример предназначен для демонстрации в качестве конечного продукта применения языка PHP, а не практически применимого фрагмента рабочего кода.

Ниже приведен исходный код PHP страницы, который обрабатывается на сервере.

```
<?php
require_once('db-config.inc.');
$dbh = mysql_connect(DB_HOST,DB_USER,DB_PASSWORD) or die("Unable to
connect to database.");
mysql_select_db('webdb') or die("Cannot access database.");
$query = "SELECT pagetitle FROM sitepages
    WHERE site = 'braingia.org'
    AND page_id = '1'";
$qresult = mysql_query($query) or die("Unable to query database.");
$title = mysql_fetch_array($qresult);
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
<html>
<head>
<STYLE TYPE="text/css">
BODY, P {color: black; font-family: verdana; font-size: 10 pt}
H1 {margin-top: 10; color: black; font-family: arial; font-size: 12 pt}
H2 {margin-bottom: -10; color: black; font-family: verdana; font-size: 18 pt}
```

```
A:link, A:visited {color: #000080; text-decoration: none}
</STYLE>
<title><?php echo $title[0] ?></title>
</head>
<body>
<h1>$title[0]</h1>
<ul>
<?php
$linksQuery = "SELECT description,href FROM sitepagedata
    WHERE site = 'braingia.org'
    AND pagetitle = '{$title}'";
$linksResult = mysql_query($linksQuery) or die("Unable to query database.");
while ($row = mysql_fetch_array($linksResult)) {
  print "<li><a href=\"{$row[1]\">$row[0]</a></li>\n";
}
?>
</ul>
</body>
</html>
```

Данную конкретную страницу нельзя назвать намного более выразительной по сравнению с простой версией страницы, показанной на рис. 2.2.

Но любопытно сравнить версию, полученную с помощью кода PHP, с вариантами на языке HTML, приведенными ранее в этой главе. Исходный код, в котором используется язык PHP, короче, поскольку в нем происходит получение информации из базы данных. Но этот серверный код не поступает к конечным пользователям в непосредственном виде. Дело в том, что в конечном итоге в окне браузера будет сформировано точно такое же изображение, как и в предыдущем примере, основанном на применении языка HTML. Единственным свидетельством того, что страница в окне браузера сформирована с применением файла на языке PHP, является расширение имени файла, `.php`. Все сложные операции обработки данных завершаются еще до того, как сформированный в итоге код передается в клиентский браузер. После выхода за пределы веб-сервера код попадает к клиенту и отображается в клиентской программе в виде обычного кода HTML и JavaScript. Это также означает, что пользователь не имеет возможности определить, какой серверный язык сценариев использовался для формирования страницы, если на этот счет нет какой-либо информации в заголовке страницы или в URL (а обычно такая информация действительно имеется, поскольку имя файла сценария формирования запрашиваемой страницы часто оканчивается таким расширением, как `.jsp` или `.php`). Итак, приведенные выше сценарии демонстрируют некоторые возможности кода, написанного на языке PHP, в котором в качестве источника данных используется база данных MySQL; все необходимые сведения, позволяющие использовать данные технологии, приведены в части II.

Сопоставление возможностей средств поддержки серверных и клиентских сценариев

Для решения многих задач могут в равной степени применяться и клиентские, и серверные методы. Например, клиентское программное обеспечение при отправке электронной почты позволяет открыть в почтовой клиентской программе пустое почтовое сообщение с заранее заполненным полем адреса, после того как пользователь щелкнет на ссылке `MAILTO`. А серверный метод предусматривает заполнение пользователем формы и последующее форматирование полученных текстовых данных в виде электронного письма, передаваемого с помощью сервера SMTP (который вполне может находиться на том же компьютере, где эксплуатируется данный серверный сценарий). Кроме того, существуют разные клиентские и серверные методы распознавания версии браузера, проверки правильности формы, создания раскрывающихся списков и выполнения арифметических вычислений. Иногда при

этом обнаруживается тонкие, но значимые различия в функциональных возможностях (например, сборка раскрывающихся списков в серверном программном обеспечении может выполняться динамически, а в клиентском программном обеспечении это невозможно), но такие различия возникают не всегда.

Чтобы успешно выбрать наиболее подходящий метод, необходимо прежде всего изучить состав пользователей. Серверные методы являются, как правило, менее быстродействующими на этапе прогона, поскольку требуют выполнения дополнительных операций обмена данными по сети, но при их использовании не нужно строить предположения о том, обеспечивает ли браузер посетителя поддержку тех или иных возможностей; кроме того, разработчик может затрачивать меньше времени на сопровождение страниц.

Области применения средств поддержки серверных сценариев

С другой стороны, языки серверных сценариев, такие как PHP, являются идеальным средством осуществления в веб многих полезных функций, а также позволяют развертывать в веб чрезвычайно важные сайты и системы, в том числе перечисленные ниже.

- Сайты с тематическим содержимым (производственные и ознакомительные).
- Системы, предоставляющие возможности общения сообществу пользователей (форумы, доски объявлений и т.д.).
- Системы электронной почты (почта веб, маршрутизация почты и передача почты из приложений веб).
- Системы поддержки заказчиков и технической поддержки.
- Сети рекламных агентств.
- Системы предоставления доступа к деловым приложениям по веб.
- Системы доступа к справочникам и спискам членов различных организаций.
- Средства публикации обзоров, результатов опросов и тестов.
- Средства заполнения и передачи форм в оперативном режиме.
- Средства реализации технологий персонализации (позволяющих учитывать личные требования).
- Средства предоставления программного обеспечения для коллективной работы.
- Сайты с каталогами, рекламными проспектами и информацией о товарах.
- Сайты для ведения игр (например, шахмат), которые обеспечивают быстрое выполнение большого объема вычислений, но имеют простой (статический) графический интерфейс.
- Средства поддержки любых других приложений, которое требуют подключения к вспомогательному серверу (серверу базы данных, серверу LDAP и т.д.).

Система PHP позволяет решать не только эти, но и многие другие важные задачи.

В настоящей главе приведены основные сведения, позволяющие понять различия между клиентскими и серверными технологиями и перейти к усвоению практических сведений. В главе 3 показано, как получить, инсталлировать и настроить конфигурацию системы PHP самостоятельно (или с привлечением сторонних лиц).

Резюме

Для того чтобы иметь представление о том, какие возможности предоставляет система PHP (или любая другая технология поддержки серверных сценариев), необходимо твердо знать, как происходит разделение труда между клиентом и сервером. В настоящей главе вначале рассматриваются примеры простого, статического кода HTML, затем демонстрируется код HTML, в котором введены фрагменты такого клиентского кода, как сценарии JavaScript и определения каскадных таблиц стилей, наконец, приведено описание веб-страниц, создаваемых с применением кода PHP, с точки зрения того, как эти страницы выглядят и в серверной, и в клиентской программах.

Клиентские сценарии обеспечивают создание внешне привлекательного интерфейса, способного быстро реагировать на ввод данных пользователем, но возможность применения в таких сценариях любых конструкций, не относящихся к категории самых основных дескрипторов HTML, полностью зависит от того, поддерживаются ли эти конструкции в применяемой версии браузера. Кроме того, для сопровождения и обновления статических клиентских сценариев требуются большие затраты труда разработчика, поскольку сценарии такого типа не обеспечивают динамическое создание страниц на основе данных, поступающих из непрерывно изменяющегося хранилища данных. С другой стороны, серверные средства программирования и серверные языки сценариев, такие как PHP, позволяют применять для формирования и обновления веб-страниц не только системы управления базами данных, но и серверы других типов.