

Введение

Если вы читаете эту книгу, то, скорее всего, уже создали веб-страницу в HTML. Вы даже придали ей стиливое оформление с помощью каскадных таблиц стилей — Cascading Style Sheets (CSS) и, возможно, написали небольшой код JavaScript для проверки допустимости данных, введенных в созданные вами веб-формы. Но этого было недостаточно, поэтому вы углубились в изучение JavaScript, включили в код библиотеку jQuery и сконструировали множество веб-страниц. Возможно, даже переместили свой код JavaScript во внешние файлы, сделали таблицы CSS общими для всего сайта и привели HTML в соответствие последним стандартам.

Но теперь вам захотелось большего.

Вполне возможно, что вас перестало устраивать то, что ваш сайт не в состоянии хранить пользовательскую информацию в чем-нибудь, кроме cookie-файлов. Возможно, захотелось получить полнофункциональный интернет-магазин, поддерживающий оплату с помощью PayPal и содержащий подробный список того, что имеется в наличии. А возможно, вы просто обнаружили ошибку в программе и захотели выйти за рамки того, что могут дать HTML, CSS и JavaScript.

В любом случае изучение PHP и MySQL станет отличным способом сделать гигантский шаг вперед в программировании. Даже если вы никогда не слышали о PHP, вы поймете, что это наилучшая возможность перейти от создания веб-страниц к разработке полноценных веб-приложений, хранящих всевозможную информацию в базах данных. Как это делается, показано в данной книге.

Что могут PHP и MySQL

PHP может обрабатывать платежи сам или обращаться к сервисам наподобие PayPal и Google Checkout. PHP может сохранять и загружать изображения из базы данных или файловой системы. Он позволяет протоколировать, как пользователи входили в приложение и покидали его, равно как и то, что они в нем просматривали.

Добавьте MySQL, и вы сможете хранить имена пользователей, их адреса, платежи по счетам и даже то, какого цвета страницы они предпочитают. MySQL умеет хранить как несколько битов данных, так и несколько тысяч строк или все обращения всех пользователей, заходивших в приложение, ко всем страницам, которые они запрашивали.

И конечно, PHP легко и просто взаимодействует с MySQL. PHP умеет делать все, начиная с извлечения имен пользователей по их идентификаторам (ID) до записи подробных сведений о финансовых транзакциях для создания и реструктурирования

таблиц, а MySQL обеспечивает весь нижний уровень этой обработки и хранение данных. Это и есть то, что составляет веб-приложение, или то, чем веб-приложение является.

Конечно, веб-приложения не так уж просты. Есть множество сложностей, и все они должны быть упрощены, чтобы в конце концов веб-приложение стало удобным и отзывчивым, вы могли сопровождать его, а пользователи — наслаждаться им. И это то, о чем данная книга: создание веб-приложений при полном понимании того, что вы делаете и почему.

Что такое PHP

PHP начинался как набор средств для выполнения простых веб-ориентированных задач. Он появился на сцене еще в 1994 году и первоначально не умел ничего большего, чем отслеживание посещения отдельных веб-страниц (онлайн-аннотация Rasmus Lerdorf – создателя PHP). В дальнейшем он был дополнен средствами взаимодействия с базами данных и обработки гостевых книг и HTML-форм и стал чрезвычайно популярен как альтернатива языкам, менее дружелюбным к веб-задачам, например C.

Начали выходить новые версии PHP, и все большее количество веб-программистов выбирали их для решения своих задач. PHP версий 3, 4 и 5 лежит в основе Сети. PHP оказался быстрым в силу своей легковесности. А простота взаимодействия с базами данных, такими как MySQL, остается одним из наиболее привлекательных его свойств.

На что похож PHP

PHP — это язык программирования. Он похож на JavaScript тем, что основную часть времени программист проводит за работой со значениями и принимает решения о том, по какому пути в коде нужно проследовать в каждый отдельно взятый момент времени. На HTML он похож тем, что программист работает с выводом — тегами, которые посетители сайта просматривают через призму своих браузеров. По сути, PHP в контексте веб-программирования проявляет универсальный характер: он неплохо справляется с решением многих вопросов, вместо того чтобы заниматься одним конкретным делом. (И если вы все еще удивляетесь, почему он называется PHP, смотрите пояснение далее.)

ЧАСТО ЗАДАВАЕМЫЙ ВОПРОС

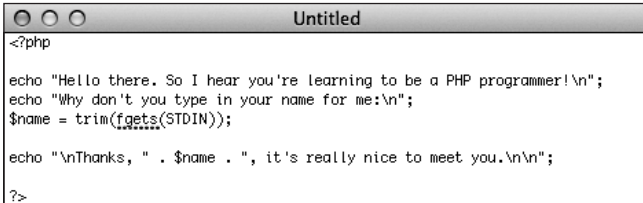
Personal Home Pages как он есть. Что означает PHP?

PHP — это акроним. Изначально PHP расшифровывалось как Personal Home Page (персональная домашняя страница), поскольку множество программистов использовали его для создания своих сайтов, применяя гораздо чаще, чем HTML, CSS и JavaScript. Однако в последние несколько лет фразой «Personal Home Page» можно описать скорее какой-то сайт, размещенный на дешевом хостинге, а не высокоэффективный язык программирования.

Поэтому теперь PHP означает PHP: Hypertext Preprocessor (PHP: гипертекстовый препроцессор). Возможно, это звучит несколько странно, как программистская шутка: акроним PHP означает нечто, что само содержит в себе этот акроним. Этаким рекурсивный акроним, или акроним, ссылающийся на самого себя. Вам не нужно знать, что такое рекурсивный акроним, поскольку здесь не викторина. Просто помните, что рекурсивный акроним PHP — не последняя странная и забавная вещь, встретившаяся в языке PHP.

PHP ориентирован на работу в Сети

Если вы взяли в руки эту книгу с целью заняться веб-программированием, то вы на правильном пути. Хотя на PHP можно писать программы, запускаемые из командной строки (посмотрите на пример, показанный на рис. 0.1), его истинная ценность заключается не в этом. Ваши PHP-программы работают изнутри сайтов как неотъемлемая часть HTML-форм, веб-сессий и cookie-файлов браузеров. Например, PHP отлично встроится в существующую на вашем сайте систему аутентификации или же позволит вам создать собственную систему.



```
<?php
echo "Hello there. So I hear you're learning to be a PHP programmer!\n";
echo "Why don't you type in your name for me:\n";
$name = trim(fgets(STDIN));

echo "\nThanks, " . $name . ", it's really nice to meet you.\n\n";

?>
```

Рис. 0.1. PHP-программы можно запускать из окна терминала или из командной оболочки Windows

Зачастую вы будете не только передавать управление HTML-странице, но и вписывать хорошо знакомый код HTML непосредственно в сценарий PHP. Вам часто придется набирать код PHP, а затем код HTML, и все это в одном и том же PHP-файле, как показано в следующем примере:

```
<?php
require '../scripts/database_connection.php';
// Получение идентификатора пользователя, чьи данные нужно показать
$user_id = $_REQUEST['user_id'];

// Создание инструкции SELECT
$select_query = "SELECT * FROM users WHERE user_id = " . $user_id;

// Выполнение запроса
$result = mysql_query($select_query);

// Присваивание значений переменным
```

?>

```
<html>
  <!-- Весь код HTML со вставками из кода PHP -->
  <head>
    <link href="../css/phpMM.css" rel="stylesheet" type="text/css" />
  </head>
  <body>
    <div id="header"><h1>PHP & MySQL: The Missing Manual</h1></div>
    <div id="example">User Profile</div>
    <div id="content">
      <div class="user_profile">
        <h1><?php echo "{$first_name} {$last_name}"; ?></h1>
        <p>
        <?php echo $bio; ?></p>
        <p class="contact_info">Get in touch with <?php echo $first_name; ?></p>
      </div>
    </div>
  <!-- И так далее - много, много HTML-кода... -->
</html>
```

Что получается в результате? Страницы, наполненные как HTML, так и динамическим контентом (рис. 0.2).



Рис. 0.2. Страница с HTML и динамическим контентом

На этой странице используется не меньше PHP, чем HTML. Она ищет имя посетителя в базе данных и отображает сведения о нем в динамическом режиме.

В меню создается пункт **Мой профиль**, характерный для этого пользователя. Но здесь все еще очень много кода HTML. Тут PHP проявляется во всей своей красе: на странице уже известный вам HTML и даже JavaScript сочетаются с PHP, который вы собираетесь изучать.

PHP не допускает столько вольностей, сколько JavaScript

Если вы уже создавали программы на JavaScript, то знаете, что JavaScript позволяет безнаказанно делать практически все что угодно. Вы можете случайно не поставить точку с запятой, использовать или не использовать скобки, применять или не применять ключевое слово `var`. Такая вольность, с одной стороны, ускоряет работу, а с другой — может принести неприятности. Порой она затрудняет поиск ошибок и превращает работу в разных браузерах в сплошной кошмар.

PHP не допускает столько вольностей, сколько JavaScript, поэтому он требует дополнительного изучения структуры и понимания того, как построена и как работает программа. Это положительный момент, поскольку в итоге вы подтянете и свое мастерство работы с JavaScript. Кроме того, строгое постоянство PHP упрощает его изучение. Вы получаете четкие правила, которых нужно придерживаться, а не массу фраз наподобие «Вы можете сделать это... или это... или это».

Итак, приготовьтесь. Вам нужно многое усвоить, но все изученное даст надежный фундамент для создания программ. И PHP сразу же сообщит, если возникнет какая-нибудь проблема. Вам не придется раскрывать консоль ошибки или отводить взгляд от небольшого желтого треугольника с предупреждением Internet Explorer, как это случалось при работе с JavaScript. Чаще вы будете получать кричащую ошибку, которая сразу же вас затормозит: «Исправь меня!» И через пару сотен страниц вы уже сможете сделать это — решить проблемы, встретившиеся в типичных PHP-программах, независимо от того, написаны они вами или кем-то другим.

PHP — это интерпретатор

PHP-код существует в виде сценариев — простых текстовых файлов, которые вы создаете и заполняете кодом. В то время как HTML содержит множество угловых скобок и ключевых слов наподобие `html`, `head` или `ul`, PHP использует множество символов доллара (\$) и таких ключевых слов, как `mysql_query` и `echo`. Поэтому HTML и PHP совсем не кажутся похожими. Однако форматы их кода подобны в своей основе: и тот и другой являются обычным текстом. HTML-документ можно открывать не только в браузере, но и в Блокноте или интегрированной среде разработки (IDE) наподобие Eclipse и даже в диалоговых редакторах, например `vi` или `emacs`. То же верно и для PHP — это обычный текст. Поэтому приготовьтесь: на протяжении всей книги мы будем печатать слова — пусть и довольно странные, с множеством символов подчеркивания — и сохранять эти слова в текстовых файлах, называемых сценариями.

Как только сценарий получен, можно передать его для интерпретирования программе PHP. Интерпретатор PHP — это та часть программного обеспечения (ПО) вашего веб-сервера, которая читает этот файл, разбирается в нем, а затем возвращает веб-серверу HTML-ответ и указывает направление дальнейших действий или порядок интерпретации записей пользовательской формы. Ваш текстовый файл — напомним, это обычный текст — интерпретируется построчно при каждом доступе к файлу.

Эта схема отличается от работы с компилируемыми языками, такими как Java или C++. При работе с этими языками вы пишете текстовые файлы, но затем запускаете команду, которая превращает их в нечто иное: файлы классов, двоичные файлы, куски нечитаемого кода, используемого компьютером.

Прелесть интерпретирующих языков, таких как PHP и JavaScript, состоит в том, что вы пишете код и тут же его запускаете. Вам не нужны пакеты инструментов или какие-то этапы обработки. Вы создаете код PHP, тестируете его в браузере и затем снова пишете. Это делается быстро и обычно означает, что работа приносит удовольствие.

PHP выполняется не внутри браузера

Существует одно большое отличие PHP от HTML, CSS и JavaScript, которые вы, возможно, использовали. Это отличие не просто большое — оно влияет буквально на все, что происходит при написании PHP-сценариев, запуске их на выполнение или тестировании в браузере.

Что же это за отличие? PHP, в отличие от HTML, CSS или JavaScript, не выполняется внутри браузера.

Что это означает? Глава 1 начнет погружать вас в подробности, но пока просто необходимо знать, что HTML, JavaScript и CSS обрабатываются внутри вашего браузера. Неважно, используете ли вы Internet Explorer, Apple Safari, Google Chrome, Mozilla Firefox или Opera, но если у вас есть браузер, то и все необходимое уже есть. Вот почему можно создать HTML-документ, сохранить его с расширением `html`, дважды щелкнуть на нем — и вот открылся браузер, а в нем виден HTML. В этом HTML можно обращаться к CSS, равно как и к JavaScript, и все будет так же. Замечательно простая штука!

PHP требует немного большего. Интерпретатор PHP взаимодействует с браузером, но не запускает его автоматически. Иначе говоря, нельзя просто щелкнуть на PHP-сценарии и ждать, что браузер всплывет и сделает свое дело. HTML-формы, иницирующие PHP-сценарии, не могут «просто работать», подобно HTML и JavaScript.

Уже сейчас вам нужно знать еще две вещи.

- Для того чтобы PHP-программы заработали, нужно немного больше операций. Недостаточно просто написать PHP-сценарий, сохранить его и потом открыть, как HTML. Не волнуйтесь: вы научитесь выполнять PHP как локально — на своем компьютере, так и удаленно — на сервере веб-хостинга. Но это потребует чуть больше усилий.

- Установка всего необходимого для выполнения PHP-программ — немного нетривиальная задача, особенно если требуется еще и MySQL (тем более если надо это сделать быстро). Вот для этого-то и нужны провайдеры интернет-сервисов! Они берут на себя заботу о подобных вещах. Поэтому, хотя можно программировать на PHP и на собственном компьютере, гораздо чаще написанные сценарии отправляют на удаленный веб-сервер. Звучит пугающе? Это не страшно... но это важно. Работая с этой книгой, вы часто будете писать код и выкладывать его на сервер.

PHP отличается от JavaScript и HTML по нескольким важным пунктам. Вам еще предстоит привыкнуть к этим различиям, но вы уже не будете так удивлены и смущены, узнавая, что с PHP некоторые вещи нужно делать иначе.

Что такое MySQL

MySQL — это база данных. В ней хранится ваша информация, данные ваших пользователей и другие нужные сведения. Из-за способности хранить информацию MySQL очень популярна. Это наиболее распространенная открытая СУБД (система управления базой данных) в мире. Миллионы пользователей работают с ней, отыскивая и описывая ее проблемы и проверяя границы ее возможностей, и тысячи разработчиков помогали совершенствовать ее код.

Если кратко, MySQL — это хранилище, куда можно класть вещи, которые позже можно будет там найти. Кроме того, MySQL предоставляет по-настоящему быстрый механизм для поиска в этом хранилище всего, что туда помещено, как только это понадобится. За время, проведенное за чтением этой книги, вы полюбите MySQL. Она будет выполнять работу, с которой вы сами никогда не справились бы, и будет делать это быстро и неумолимо.

Кроме того, она является прекрасным компаньоном для PHP. Она легко устанавливается в любой системе и не требует таких огромных ресурсов, как более крупные коммерческие СУБД, например продукты Oracle и IBM, и с ней легко взаимодействовать. Действительно, вы обнаружите, что PHP и MySQL отлично подходят друг другу, обладая множеством функций, простых в использовании и позволяющих PHP-сценариям делать в базах данных MySQL практически все, что вы можете себе вообразить.

ПРИМЕЧАНИЕ

Конечно, в MySQL и SQL (это язык, на котором осуществляется взаимодействие с MySQL) есть множество тонкостей, но лучше приберечь их для гл. 4, когда у вас в запасе уже будут некоторые знания по PHP.

Об этой книге

PHP — это веб-язык, а не программа, поставляемая в коробке. Десятки (может быть, даже сотни) тысяч сайтов предлагают учебники и руководства по PHP. Впечатляет, не так ли? Но не все они актуальны. Одни полны ошибок. На других бурные

обсуждения полны перебранок, недовольства и ругани программистов. Так что найти среди них действительно подходящий не так-то просто.

Цель этой книги — послужить руководством, которое должно быть под рукой уже в тот момент, когда вы загружаете PHP. Если хотите, это тот PDF (или электронная книга, если у вас есть Kindle, или Nook, или iPad), которого вам не доставало. На страницах издания вы найдете пошаговые инструкции для получения работоспособного PHP-интерпретатора, написания первой программы... и второй программы... и в конечном счете создания с нуля целого веб-приложения. Кроме того, тут вы обнаружите четкий анализ самых важных компонентов PHP, которые будете ежедневно использовать при создании персонального блога или внутренней корпоративной сети.

ПРИМЕЧАНИЕ

В этой книге периодически рекомендуются другие издания на темы, слишком специализированные или уходящие далеко в сторону от руководства по PHP и MySQL. Внимательные читатели могут заметить, что не все эти издания выпущены компанией O'Reilly Media. Но хорошая книга заслуживает того, чтобы о ней знать, даже если она издана не в O'Reilly.

Эта книга рассчитана на читателей с разным уровнем подготовки. Основная часть материала предназначена для тех, кто уже прошел определенный путь от начала изучения веб-технологий, или любителей веб-технологий и программистов со средним уровнем. Я надеюсь, что вы уже разбираетесь в HTML и CSS и, может быть, даже немного знаете JavaScript. Если же все эти веб-компоненты вам незнакомы, специальные врезки под названием «К вашему сведению» представят вводную информацию, необходимую для понимания текущей темы. Во врезках «Курсы повышения квалификации» предлагаются дополнительные технические советы, рассматриваются различные приемы и методы ускоренной работы, предназначенные для более или менее опытных любителей компьютеров.

Macintosh и Windows

Версии PHP и MySQL, предназначенные для Macintosh и Windows, работают почти одинаково. Самым важным является то, что основная часть работы выполняется путем отправки ваших сценариев на веб-сервер и запуска на нем кода базы данных. Это означает, что вопросы, касающиеся операционной системы, должен решать хостинг-провайдер, а вам нужно сконцентрироваться на своем коде и информации.

Прочитав первые главы книги, вы получите систему, настроенную на работу с программным кодом и PHP-сценариями. Но вскоре забудете о том, под управлением какой операционной системы работаете, Macintosh или Windows. Вы просто будете создавать код. Точно так же, как пишете HTML и CSS.

FTP: важная деталь

Обратите внимание, что вам понадобится хороший FTP-клиент. Каким бы впечатляющим ни был ваш опыт написания сценариев — а он должен быть немалым! — вам необходимо загрузить сценарии на веб-сервер. Тут и приходит

на помощь FTP — это то средство, с помощью которого файлы с вашего компьютера перемещаются в нужное место на удаленном сервере.

ПРИМЕЧАНИЕ АВТОРА

Набор текста в редакторе командной строки — одна из особенностей моей работы. Но я в некотором роде динозавр из тех минувших дней, когда приходилось смотреть коммерческие каналы, чтобы увидеть все интересные передачи.

Сейчас для большинства из вас куда более подходящими будут хорошие текстовый редактор и графический FTP-клиент. А если серьезно, я уже давно хочу отказаться от своих причуд.

В гл. 1 будут рассмотрены несколько хороших редакторов, причем некоторые из них имеют встроенный FTP-клиент. Если вы не пользуетесь интегрированной средой, кстати будет специализированная FTP-программа, например Cyberduck (www.cyberduck.ch). Вы можете написать сценарий, отправить его по Сети и протестировать всего лишь несколькими щелчками кнопкой мыши. Итак, загрузите эту FTP-программу, настройте свой веб-сервер (его тоже можно назвать интегрированной средой) и приступайте к работе. Это вам, несомненно, пригодится.

Структура книги

Эта книга состоит из четырех частей, каждая из которых содержит несколько глав, и приложений.

- **Часть 1. Основы PHP и MySQL.** В первых трех главах данной части вы научитесь устанавливать PHP, запустите его на своем компьютере, напишете несколько первых программ на PHP и овладеете некоторыми элементарными вещами, такими как сбор сведений о пользователе через веб-форму и работа с текстом. Вы также установите MySQL и начнете всесторонне знакомиться со структурой баз данных.
- **Часть 2. Динамические веб-страницы.** В этой части содержатся главы, с помощью которых вы приступите к созданию основ настоящего веб-приложения. Вы добавите таблицу, в которой сможете хранить сведения о пользователях, и получите представление о том, как легко можно манипулировать текстом. Для свободного манипулирования буквами, цифрами и слешами в URL-адресах, адресах электронной почты, идентификаторах Twitter и т. д. будут использоваться регулярные выражения и операции со строками.
- **Часть 3. Переход от веб-страниц к веб-приложениям.** Учитывая крепкую базу, полученную в предыдущих частях книги, вы уже будете готовы объединить свои веб-страницы в более заверченный модуль. Вы добавите самостоятельно созданную систему обработки ошибок, чтобы пользователи не пребывали в недоумении при сбое приложения, а также собственную систему отладки, помогающую обнаружить источники проблем. Кроме того, вы научитесь сохранять в базе данных ссылки на изображения пользователей и сами изображения, а также выбирать, какой из этих двух подходов и в каких случаях более предпочтителен.