

Глава 1

Что такое язык PHP

Аббревиатурой PHP называют один из наиболее популярных языков, используемых для создания веб-приложений. Первоначально PHP означало *Personal Home Page* — *персональная домашняя страница*, но потом, с расширением области применения языка, аббревиатура PHP стала расшифровываться как *PHP: Hypertext Preprocessor* — *PHP: гипертекстовый препроцессор*. Текущая версия языка PHP предназначена для создания веб-приложений самого широкого назначения, например, на его основе создана небезызвестная социальная сеть Facebook. В настоящее время PHP поддерживается подавляющим большинством хостинг-провайдеров, его используют сотни тысяч программистов.

Язык PHP разрабатывается группой энтузиастов в рамках проекта с открытым кодом. Проект распространяется под собственной лицензией, несовместимой с GNU GPL. Официальный сайт языка находится по адресу www.php.net. На момент написания книги последним стабильным релизом языка являлся PHP 5.4.12 и 5.3.22. Несколько лет назад были предприняты попытки создания версии PHP 6, но сейчас этот проект по некоторым соображениям признан неперспективным. Поэтому при знакомстве с PHP мы будем основываться на версиях 5.3, 5.4, имея в виду, что при практическом применении своих программ, то есть размещении их в Сети, вы всегда будете ограничены возможностями, предлагаемыми хостинг-провайдерами. Опыт же показывает, что провайдерам свойственен вполне понятный консерватизм, и ныне наиболее распространена поддержка версий 5.2–5.4, причем не самых последних релизов. Так что новые и новейшие возможности языка PHP следует использовать осторожно.

1.1. Работа сценария PHP

Как работает приложение, созданное на основе языка PHP? Принцип очень похож на тот, что используют обычные, статические веб-страницы, созданные на основе языка HTML. В последнем случае на веб-сервере создается сайт, состоящий из нескольких страниц с кодом HTML. Посетитель, зашедший на сайт, загружает их в браузере, щелкая на гиперссылках, представленных на этих страницах.

Щелчок на ссылке инициирует цепочку операций взаимодействия между компьютером пользователя и веб-сервером. Когда вы щелкаете на ссылке какой-то веб-страницы, ваш браузер отправляет запрос на веб-сервер, который считывает HTML-код запрашиваемой страницы, формирует ответ и отправляет его обратно браузеру. В результате содержимое страницы, отображаемой в ответ на запрос пользователя, будет всегда одним и тем же, то есть эта веб-страница *статична*. Вы не можете никаким образом изменить ее, уточнить отображаемый контент, передав в запросе серверу определенные данные.

Например, если вы посетите сайт со статическими веб-страницами, предлагающими данные о прогнозах погоды по всей стране, то *не сможете* указать веб-серверу, что вас интересует погода в определенном месте, скажем в Москве, введя название города в специальное поле ввода и щелкнув на ссылке для отображения данных. Чтобы реализовать такую функцию на статическом сайте, вам пришлось бы создать огромный список городов и для каждого из них оформить ссылку на статическую страницу с нужной информацией. Это крайне неудобно и совершенно нереализуемо, когда список выбора достаточно велик или запрашиваемая информация непрерывно изменяется, как на сайтах погоды.

Все это легко решается средствами языка PHP, позволяющего создавать *динамические* сайты! С помощью PHP при создании веб-страниц вы сможете включать в них специальный сценарий и передавать сценарию данные, которые он должен использовать при обработке запроса вашего браузера. Например, для предыдущего примера сайта погоды вы сможете на одной из его веб-страниц с помощью HTML-формы выбрать нужный город и нажать кнопку подтверждения выбора. Браузер автоматически запросит сервер, указав ему адрес той веб-страницы, которая должна отобразиться в ответ на ваш запрос, и передаст ей всю информацию, введенную вами в форму, в данном случае название выбранного города. Далее сценарий, размещенный внутри запрашиваемой страницы, получит эти данные, обработает их и передаст вам в ответ именно то, что требуется, — прогноз по Москве.

Такие веб-страницы, «начиненные» кодом сценария PHP (или другого аналогичного языка), называются динамическими, поскольку могут реагировать на действия посетителя сайта, передавая ему в ответ на сделанный выбор именно ту информацию, которая нужна. В отличие от статического сайта на динамическом она может меняться с течением времени. Это важно, поскольку, например, погода — вещь переменчивая, а сайт с прогнозом погоды должен показывать самую свежую информацию. Для этого сценарий PHP постоянно обновляет свои данные, обращаясь к нужным источникам: базам данных, показаниям приборов и т. д. Все эти функции легко реализуются средствами PHP, достаточно обширными и развитыми для решения огромного круга задач.

Для исполнения функций, реализуемых сценариями PHP, на веб-сервере должны содержаться специальные средства обработки кода сценария. Их и используют средства PHP, которые играют роль *препроцессора*, то есть предварительного обработчика запроса. Именно они исполняют код сценария внутри запрошенной страницы, передают ему данные из запроса и отправляют ответ браузеру, пользуясь возможностями веб-сервера. Таким образом, средства языка PHP представляют собой дополнение веб-сервера, которое делает возможным интерактивное взаимодействие посетителя сайта с его содержимым.

Итак, чтобы ваш PHP-сценарий смог работать, ему нужны следующие компоненты: браузер, веб-сервер и средства самого препроцессора PHP, входящие в состав веб-сервера. Эти средства предназначены для обработки кода сценария внутри страниц динамического сайта, отображаемых по запросам посетителей в ответ на их действия на сайте.

Кроме того, для создания полнофункциональных PHP-сценариев требуется обеспечить сохранение данных на веб-сервере. Это можно реализовать сохранением информации в файле, и для простых сценариев такой метод вполне приемлем. Но для сложных, профессиональных сценариев применяются более развитые средства — базы данных, обеспечивающие хранение и извлечение данных любого типа по специальным запросам. Эти запросы формируются на специальном языке, называемом SQL (*Structured Query Language* — язык структурированных запросов). SQL — это международный стандартный язык для работы с базами данных, и мы познакомимся с ним далее в этой книге.

Для исполнения запросов на веб-сервере с поддержкой PHP должен быть установлен еще один сервер, который будет воспринимать запросы к базе данных, поступающие из сценария, и исполнять их. Таких серверов множество, но в среде разработчиков PHP наиболее популярен MySQL, поддерживаемый и бесплатно распространяемый на сайте mysql.com. Его мы и выберем для изучения.

Подведем итог: для практической работы по созданию и исполнению сценариев PHP нам нужен браузер и веб-сервер, средства самого языка PHP и сервер MySQL. Любой провайдер, обеспечивающий работу с PHP, предоставляет эти инструменты для развертывания сайта на хостинге. Однако для создания своих профессиональных сценариев лучше будет установить инструменты работы с PHP на рабочем компьютере, локально, что даст вам более гибкий и удобный доступ к PHP-сценарию и средствам его отладки. Поэтому для изучения материала этой книги вам нужно обзавестись всеми этими компонентами, установив их на свой компьютер. Это можно сделать следующим образом.

- ❑ Вы можете загрузить на свой компьютер установочный пакет PHP с сайта www.php.net и установить его самостоятельно, пользуясь инструкцией на сайте.
- ❑ Можете установить на свой компьютер сервер MySQL, загрузив с сайта mysql.com соответствующий дистрибутив.
- ❑ Если на вашем компьютере отсутствует веб-сервер (IIS, Apache и др.), поддерживающий PHP, вы должны запустить его. Для системы Windows вам нужно только ввести в действие сервер IIS, воспользовавшись инструментом добавления компонентов Windows на Панели управления. Другой вариант — загрузка установочного пакета сервера Apache с сайта производителя www.apache.org с последующей установкой и настройкой.
- ❑ Воспользоваться специальными средствами для создания среды разработки PHP-приложений — так называемыми интегрированными пакетами работы с PHP. В их состав входит все, что нужно для разработки сценариев PHP: веб-сервер, SQL-сервер, средства PHP, дополнительные полезные инструменты, причем в интегрированном виде. Вам не придется самостоятельно устанавливать веб- и SQL-сервер и подключать к нему PHP (тем более что это требует достаточно глубоких познаний в этой сфере) — за вас все сделает инсталлятор пакета.
- ❑ Прибегнуть к интегрированным средам разработки (IDE), включающим, помимо компонентов среды для работы с PHP, средства отладки сценариев PHP. Это очень полезные возможности, которые значительно облегчают разработку сценариев, позволяя их пошагово исполнять и контролировать результаты работы кода.

Рекомендую использовать две последние возможности как самые доступные для начинающего программиста, чтобы не усложнять себе освоение самого языка установкой и настройкой установочных пакетов PHP и MySQL. В качестве интегрированного пакета средств PHP мы выберем XAMPP, дистрибутив которого можно бесплатно загрузить на сайте производителя www.apachefriends.org. Он включает в себя веб-сервер Apache, сервер MySQL, средства PHP и множество

других полезных вещей, например почтовый сервис, сервер FTP. Пакет XAMPP устанавливается на компьютерах Windows XP/Vista/7 и Linux, в процессе установки инсталлятор потребует от вас лишь ответить на несколько вопросов, уточняющих конфигурацию среды, а в результате вы получите полный набор средств разработки PHP. Это избавит вас от непростой установки и настройки средств PHP из отдельных компонентов.

В качестве среды IDE рекомендую использовать программу PHPEdit, условно бесплатно распространяемую на сайте разработчика www.phpedit.com. Лицензия предоставляется на месяц, но использовать бесплатно программу вы сможете два месяца, после чего решите, нужна ли она вам настолько, что следует купить ее (около \$100). Для сложных сценариев PHP такие средства отладки, которые предлагает PHPEdit, весьма полезны, кроме того, программа помогает вводить код, отображая подсказки и подсвечивая различные компоненты PHP-кода разным цветом.

1.2. Основы синтаксиса языка PHP

Синтаксис языка PHP достаточно прост для изучения, а его корни лежат в языках Perl, Java и C. Однако в отличие от этих языков PHP изначально был разработан как язык программирования, специально предназначенный для написания веб-приложений (сценариев), исполняющихся на веб-сервере. В этом состоит его преимущество, поскольку его средства позволяют быстро и эффективно создавать полнофункциональные веб-приложения.

Другое достоинство PHP — предоставляемая им возможность внедрять свои сценарии в HTML-код веб-страниц, что значительно упрощает задачу создания динамических сайтов. Благодаря PHP разработчики могут динамически изменять HTML-код страниц в зависимости от действий посетителя сайта. Например, при вводе им данных в поля формы, или установке переключателей, флажков, или выборе определенного элемента списка сервер учтет все эти данные и динамически перестроит всю информацию, передаваемую браузеру.

Вместе с тем язык PHP весьма прост и легок для освоения. Даже начинающие программисты смогут быстро научиться создавать на нем достаточно сложные сценарии, реализующие функции и интерфейс на уровне профессиональных сайтов.

Итак, приступим к изучению языка PHP. Надеюсь, что вы прочли предыдущий раздел этой книги и у вас на компьютере установлены серверы Apache и MySQL, чтобы вы могли повторять вслед за изложением примеры конструкции языка PHP.

Если вы этого не сделали — настоятельно рекомендую вернуться назад и выполнить установку этих средств.

Сначала мы займемся изучением синтаксиса языка PHP. Мы познакомимся с типами данных, используемых в PHP, базовыми конструкциями по управлению ходом работы программы, функциями, способами передачи данных в сценарии, работы с базами данных MySQL, открытия сеансов работы со сценарием и записью cookie-файлов. По мере изложения материала применение этих средств будет демонстрироваться на примерах сценариев, исполняемых на локальном сервере пользовательского компьютера. Отсюда видна важность установки этих средств.

1.3. Создание и запуск первого сценария PHP

Первый сценарий будет простым, так как он нужен только для демонстрации общей структуры PHP-кода и его совместного использования с кодом HTML. Откройте какой-либо текстовый редактор, например Блокнот системы Windows, и запишите в него такой PHP-код:

```
<?php
    phpinfo();
?>
```

Далее откройте корневую папку `htdocs` вашего сервера Apache, в котором сохраняются исполняемые сценарии, и создайте там подпапку с названием `examples`. Мы будем сохранять в ней файлы PHP-сценариев, создаваемых в процессе изучения. Назовите файл `start.php` и сохраните его в этой папке.



ПРИМЕЧАНИЕ

Если вы используете пакет XAMPP, то папка `htdocs` находится внутри папки, в которую вы установили пакет. Обычно эта папка называется так: *имя_диска*:\XAMPP. При работе с сервером Apache напрямую этот каталог должен быть указан в директиве `DocumentRoot` в файле `httpd.conf`. Если же вы предпочитаете работать с сервером IIS, то должны помещать свои сценарии в папку `inetpub\wwwroot`.

Теперь все готово для исполнения первого сценария. Откройте свой браузер и в адресной строке введите `http://localhost/examples/start.php`. В окне браузера отобразятся сведения об установленной версии и конфигурации установленного языка PHP (рис. 1.1).

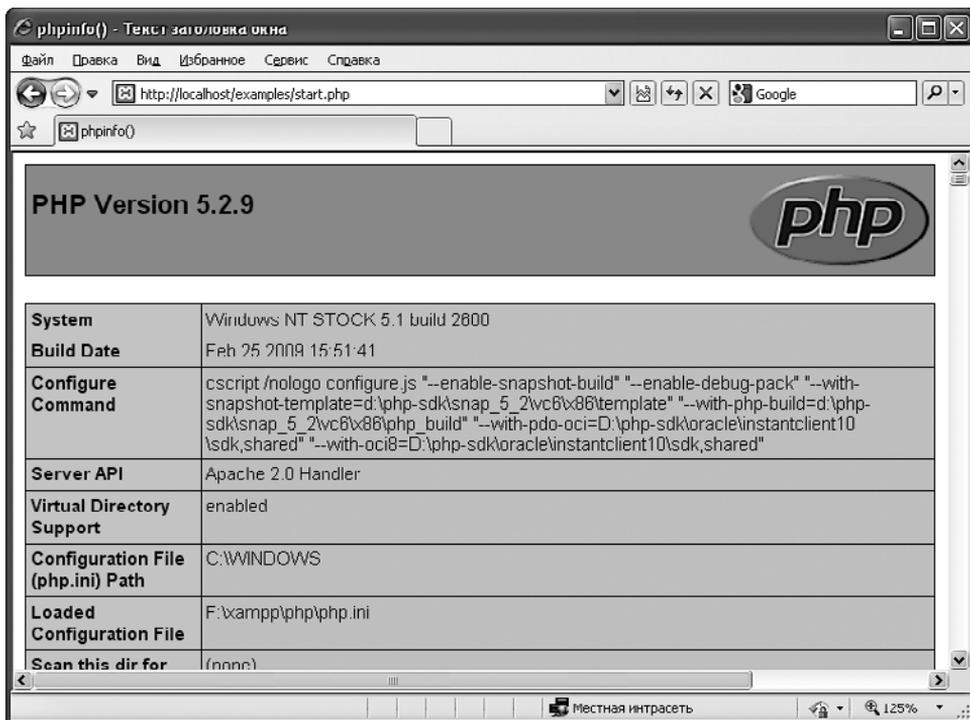


Рис. 1.1. Окно со сведениями об установленной версии языка PHP

Запись `phpinfo()` внутри сценария означает вызов встроенной функции PHP — одной из многих, которые этот язык предлагает разработчикам сценариев. Помните, что в конце любого оператора языка PHP должна стоять точка с запятой (;). Функция `phpinfo()` отображает сведения об установленных компонентах языка PHP и настройках.

Обратите внимание на остальные записи внутри файла. Строки `<?php` и `?>` соответствуют начальному и конечному тегу PHP-кода, внутри которых помещаются сценарии, внедряемые в код HTML веб-страницы. Посмотрим, как это делается на практике. Откройте новый документ текстового процессора и запишите в него такой код (листинг 1.1).

Листинг 1.1. Документ HTML с кодом PHP

```
<html>
  <head>
    <title>
      Эта страничка содержит код PHP
```

```
</title>
<body>
  <h1>
    Внедрение сценария PHP в код HTML
  </h1>
  <?php
    echo "Привет, Мир! Это PHP-сценарий внутри странички Web!";
  ?>
</body>
</html>
```

Сохраните этот код в файле `hello.php` и поместите его в папку `examples`. Введите в адресную строку браузера вызов сохраненного кода `http://localhost/examples/hello.php`, после чего в окне браузера отобразится результат его исполнения (рис. 1.2).

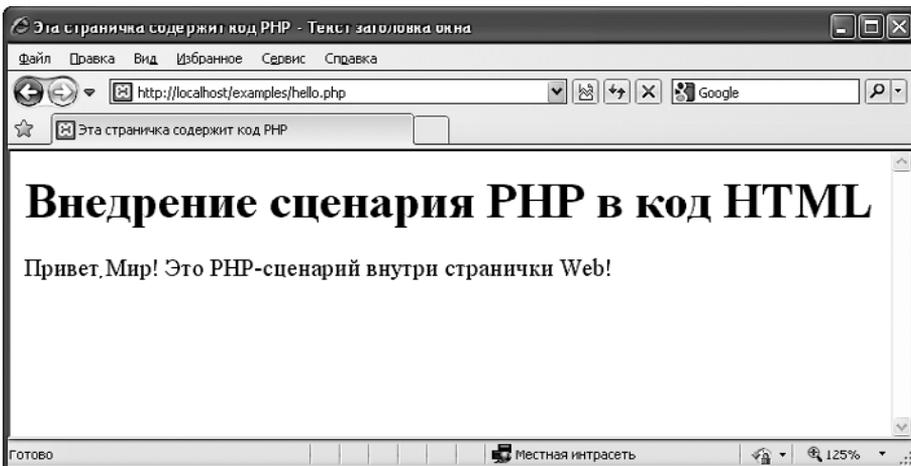


Рис. 1.2. Приветствие PHP-сценария внутри HTML-кода

Рассмотрим содержимое листинга 1.1 внимательнее. Мы видим, что внутри тела документа HTML в содержимое тега `<h1>` помещен заголовок, после которого внутри тегов `<?php` и `?>` расположен оператор `echo` с текстом приветствия. Оператор `echo` — один из основных в языке PHP, он позволяет отображать любой текст внутри веб-страницы. В листинге 1.1 показан простейший пример использования этого оператора:

```
echo "Здесь помещается выводимый текст";
```