Самоучитель

Максим Кузнецов, Игорь Симдянов

PHP 5/6

Третье издание

Объектно-ориентированное программирование

Работа с базами данных

Защита приложений от взлома

Актуальные примеры

Новое в PHP 5/6

bhv

Максим Кузнецов
Игорь Симдянов

Самоучитель PHP 5/6

Третье издание

Санкт-Петербург
«БХВ-Петербург»
2009

УДК 681.3.06
ББК 32.973.26-018.2
К89

Кузнецов, М. В.

К89 Самоучитель PHP 5/6 / М. В. Кузнецов, И. В. Симдянов —
3-е изд., перераб. и доп. — СПб.: БХВ-Петербург, 2008. — 672 с.: ил.

ISBN 978-5-9775-0409-6

Описаны самые последние версии языка разработки серверных сценариев PHP — 5.3 и 6.0. Рассмотрены основы языка, вопросы объектно-ориентированного программирования на PHP, обработки исключительных ситуаций, взаимодействия с MySQL, регулярные выражения, работа с электронной почтой. Книга содержит множество примеров, взятых из реальной практики разработки динамических Web-сайтов.

Третье издание книги, ранее выходившей под названием "Самоучитель PHP 5", существенно переработано, дополнено и будет интересно не только программистам, впервые знакомящимся с языком, но и читателям предыдущих изданий книги и профессионалам.

Для программистов и Web-разработчиков

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	*Анна Кузьмина*
Компьютерная верстка	*Натальи Смирновой*
Корректор	*Наталия Першакова*
Оформление обложки	*Елены Беляевой*
Зав. производством	*Николай Тверских*

Лицензия ИД № 02429 от 24.07.00. Подписано в печать 03.12.08.
Формат 70×100 1/16. Печать офсетная. Усл. печ. л. 54,18.
Тираж 2500 экз. Заказ №
"БХВ-Петербург", 190005, Санкт-Петербург, Измайловский пр., 29.

Санитарно-эпидемиологическое заключение на продукцию № 77.99.60.953.Д.002108.02.07
от 28.02.2007 г. выдано Федеральной службой по надзору
в сфере защиты прав потребителей и благополучия человека.

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

ISBN 978-5-9775-0409-6

Оглавление

Введение

PHP является молодым и динамично развивающимся языком программирования, который используется главным образом для создания Web-приложений. Быстрая динамика развития, с одной стороны, предполагает регулярное введение новых конструкций, функций, библиотек, директив, а, с другой стороны, — изъятие старых конструкций, использование которых оказалось неудобным или провоцировало создание небезопасного кода. Стадию динамичного развития проходит любой язык программирования, определить ее очень просто по быстрой смене версий и количеству нововведений в этих версиях. Со временем интервал между версиями увеличивается, каждая новая версия содержит все меньше и меньше революционных нововведений. Современный PHP пока находится в стадии быстрого развития, однако в ближайшие несколько лет прогнозируется переход к плавному изменению, характерному для состоявшихся языков.

Анонсируя PHP 6, разработчики заранее объявили о нововведениях, которые будут в новой версии. Однако при разработке ими было принято решение не создавать дистрибутив с нуля, а постепенно вносить изменения в текущую версию PHP 5. Если раньше для того, чтобы попробовать новинки языка, требовалось загрузить нестабильный дистрибутив, предназначенный для разработчиков, то теперь особенности PHP 6 постепенно и плавно появляются в версии PHP 5. Таким образом, к тому моменту, когда все обновления будут внедрены в язык, разработчикам останется только поменять цифру.

Что-то из заявленного в PHP 6 в настоящий момент реализовано, что-то еще ожидает своего часа. Тем не менее, язык меняется и книги, написанные ранее, устаревают. Поэтому нами было принято решение выпустить третье издание самоучителя PHP, которое описывает современное состояние дел.

Книга будет интересна как начинающим разработчикам, которым необходимо последовательное изложение языка, так и опытным программистам, желающим познакомиться с нововведениями языка, появившимися в последнее время. Текущее издание будет также интересно читателям предыдущих изданий самоучителя, т. к. нами было принято решение о полной переработке всех глав: по сути, они написаны заново. Связано это с тем, что структура и организация языка значительно изменились за последние годы, поменялась и аудитория PHP-разработчиков.

Нововведения PHP 6

Как уже отмечалось, не все нововведения пока реализованы: на данный момент нет сечений массивов и строк, отсутствует (в стабильных версиях) поддержка Unicode на уроне ядра, в расширении для работы с графикой не поддерживается работа с анимированными GIF-файлами. Список этот можно продолжить, однако интереснее отметить то, что уже реализовано:

- исправлены многочисленные непоследовательности объектно-ориентированной модели;
- значение `E_ALL` директивы `error_reporting` конфигурационного файла php.ini теперь включает `E_STRICT` *(см. разд. 3.8)*;
- средний параметр `y` в условном операторе `x ? y : z`, начиная с PHP 6.0, не является обязательным *(см. разд. 5.8)*;
- ускорена работа оператора `@` *(см. главу 5)*;
- функциональность цикла `foreach` расширена для работы с многомерными массивами;
- в операторе `break` удалена поддержка динамических меток;
- регулярные выражения POSIX исключены из ядра PHP и доступны лишь как расширение;
- введены новые предопределенные массивы для работы с датой и временем;
- изменен синтаксис множества функций (главным образом за счет увеличения числа необязательных параметров).

Благодарности

Авторы выражают огромную благодарность сотрудникам издательства "БХВ-Петербург", стараниями которых наша рукопись увидела свет.

Мы также очень признательны посетителям наших форумов за интересные вопросы и конструктивное обсуждение.

Глава 1

Что представляет собой PHP?

Язык программирования PHP является универсальным, т. е. может использоваться для создания практически любых программ. Однако наибольшее распространение он получил в области Web-разработки. Львиная доля приложений, созданных с применением PHP, обслуживает Web-сайты.

Язык PHP является интерпретируемым языком программирования. Это означает, что программа выполняется строчка за строчкой, а не компилируется в исполняемый модуль. С одной стороны это приводит к тому, что программы выполняются более медленно по сравнению с компилируемыми языками, с другой стороны исходный код приложения может быть отредактирован в любой удобный момент.

В этой главе обсуждается эволюция языка, мы посмотрим, что представляет собой PHP на сегодняшний день, а также взаимодействие PHP с другими технологиями и языками программирования.

1.1. История PHP

Датой создания языка PHP считается осень 1994 года, а автором языка — программист Расмус Лердорф (Rasmus Lerdorf). Сначала Лердорф собрался написать простой движок для своей персональной странички и завершил эту работу к началу 1995 года. Движок был написан на языке Perl и умел делать очень немного, т. к. создавался только для подсчета количества посетителей домашней странички Расмуса, на которой было размещено его резюме. Этот движок был назван Personal Home Page Tools (PHPT), а позже название трансформировалось в аббревиатуру PHP, что следует понимать как рекурсивный акроним Hypertext Preprocessor (такое решение о переименовании было принято в 1997 году).

Стоит отметить, что в 1994 году специальных инструментов для создания Web-приложений не существовало, да и сам Web только начинал свое развитие. Поэтому движок, разработанный Расмусом, вызвал живейший интерес разработчиков, и к нему хлынул поток писем с просьбами предоставить свой инструментарий. Такой

успех PHP привел к тому, что Лердорф приступил к разработке различных расширений языка. В том же 1994 году Расмус разработал пакет, предназначенный для обработки HTML-форм, который назывался FI (Form Interpretator). А к середине 1995 года, объединив движок для своей странички с интерпретатором форм, Лердорф выпустил вторую версию языка, которая называлась PHP/FI. К этому времени при разработке языка Расмус перешел на компилируемый язык C (Perl является интерпретируемым и значительно уступает по производительности C, одному из самых быстрых языков программирования). Тогда же в PHP была добавлена поддержка основных баз данных, что еще более усилило популярность вновь созданного языка.

> ***Замечание***
>
> И по сегодняшний день расширения являются одним из основных средств увеличения функциональности PHP. Существуют сотни расширений (для работы с базами данных, графикой, PDF-файлами и т. п.), и любой разработчик может самостоятельно создать собственное расширение для своих нужд. В стандартной поставке PHP содержит лишь наиболее популярные и проверенные расширения.

К концу 1997 года два программиста — Зив Сураски (Zeev Suraski) и Энди Гутманс (Andi Gutmans) — переписали первоначальный лексический анализатор, и к лету 1998 года в полной мере увидела свет третья версия языка — PHP 3. Развитие PHP стремительно продолжалось, в язык сотнями добавлялись новые функции, и в 1999 году количество разработчиков, использующих PHP, превысило 1 миллион, что сделало PHP одним из самых популярных языков для разработки Web-приложений. К этому времени к разработке языка подключилось большое количество программистов со всего мира. Стало понятно, что вследствие все более растущей популярности языка необходима его адаптация для разработки крупномасштабных приложений. Дело в том, что поскольку PHP изначально проектировался для разработки несложных приложений, лексический анализ и компиляция кода в нем происходили одновременно. Этим достигалась быстрота выполнения простых сценариев за счет сокращения времени от запуска до начала выполнения. Однако при выполнении более сложных сценариев все происходило с точностью до наоборот по причине многократного повторения лексического анализа кода. Стало очевидным, что необходимо модифицировать основное ядро, что и было быстро и успешно осуществлено фирмой Zend Technologies Ltd. Был создан более устойчивый лексический анализатор, на базе которого уже можно было строить полномасштабные приложения, и в 2000 году появилась четвертая версия языка — PHP 4.0. В этой версии уже существовала возможность создания объектно-ориентированных приложений, что было с радостью воспринято многими пользователями. Однако объектно-ориентированная парадигма находилась в PHP 4.0 в зачаточном состоянии, поскольку в языке отсутствовала реализация многих концепций, свойственных объектно-ориентированной технологии. В 2004 году выходит новая версия языка — PHP 5.0 на базе машины Zend Engine 2. Основные изменения в новой версии как

раз коснулись реализации объектно-ориентированного подхода. Практически сразу после выхода версии PHP 5.0 разработчики приняли план по созданию PHP 6.0. Изменения было решено вводить не революционно, а постепенно добавляя их в новые релизы PHP: PHP 5.1, PHP 5.2 и PHP 5.3. Одновременно изменения добавляются в версию PHP 6.0, которая пока не стабильна и предназначена лишь для тестирования. Таким образом, когда все изменения будут внедрены в движок языка и отлажены, будет выпущена конечная версия PHP 6.0.

Итак, что представляет собой PHP на сегодняшний день? PHP сейчас это:

- поддержка платформ Win32 (9*x*/NT/2000/XP), UNIX, OS/2, QNX, MacOS, BeOS, OCX;
- совместимость с серверами: Apache (Win32, UNIX), phttpd, fhttpd, thttpd, ISAPI (Zeus, IIS), NSAPI, модулем Roxen/Caudium, AOLServer;
- поддержка технологий COM, XML, Java, CORBA, WDDX, Macromedia Flash;
- развитая функциональность для работы с сетевыми соединениями;
- поддержка свыше 20 баз данных и развитая функциональность для работы с ними;
- возможность создания полноценных объектно-ориентированных приложений;
- сравнительно простой синтаксис и удобство в практическом использовании;
- бесплатность;
- открытость кода, благодаря которой вы можете создавать собственные расширения языка.

1.2. Место и роль PHP в Интернете

На сегодняшний день PHP используют 20 млн сайтов. При помощи PHP можно разрабатывать любые программы, использующие протоколы прикладного уровня, будь то Web-приложения, программы для отправки или получения почтовых сообщений, взаимодействие с FTP-сервером и т. п.

Web-программирование — это технология, включающая в себя множество компонентов, одним из которых выступает PHP. Прежде чем приступить к его подробному изучению, определим его место и роль в процессе обмена данными между клиентом и сервером. На рис. 1.1 представлена схема, демонстрирующая место и время действия компонентов типичной среды Web-разработки.

Как видно из схемы, помимо PHP в работе Web-приложений работает множество дополнительных программ и технологий. Все они в той или иной степени будут

затрагиваться в данной книге, т. к. PHP — всего лишь один из компонентов Web-технологии; для освоения всего процесса разработки Web-приложения необходимо понимать принципы работы всех компонентов, участвующих в схеме, представленной на рис. 1.1.

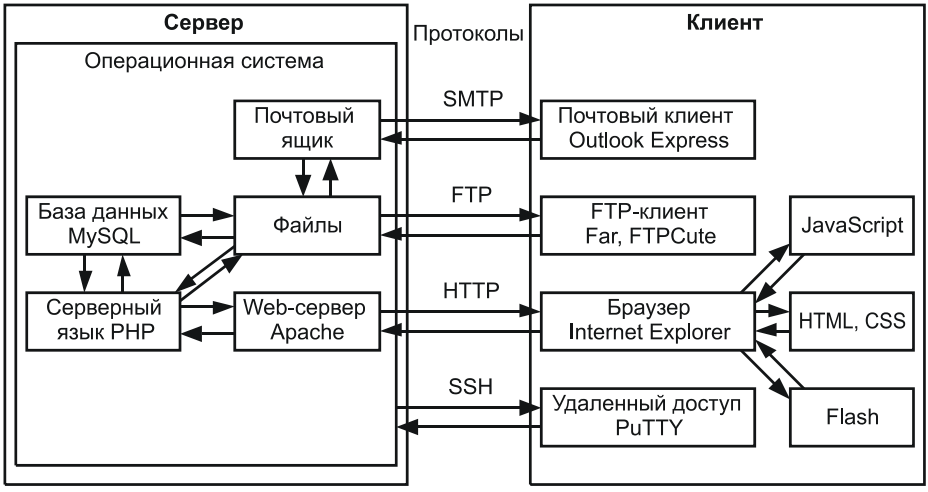

Рис. 1.1. Клиент-серверное взаимодействие

Схему взаимодействия клиента и сервера можно условно поделить на три части:

- серверные технологии — эти компоненты работают на сервере, вы не сможете запускать и использовать их на стороне клиента, если только последний сам не выступает в качестве сервера;
- протоколы — при помощи протоколов клиенты и серверы обмениваются информацией;
- клиентские технологии — Web-приложения по своей природе распределенные, часть работы выполняется на сервере, часть — лишь после того, как страница загружена по протоколу HTTP на сервер.

1.2.1. Серверные технологии

UNIX-подобная операционная система

Все программы, на сервере или на клиенте, работают под управлением операционной системы. В Интернете на серверах в качестве серверной операционной системы используется, как правило, UNIX-подобная операционная система (Linux, FreeBSD, Solaris и т. д.). Большинство пользователей Интернета, по крайней мере, в

Российской Федерации в качестве операционной системы используют Windows, поэтому при изучении Web-технологий основы UNIX будут крайне полезны, т. к. порядок работы в этих двух операционных системах различается достаточно сильно. Знания UNIX-подобной операционной системы не требуются на начальном этапе обучения, многие обходятся без основ UNIX, уже профессионально занимаясь Web-разработкой. Однако изучение этой операционной системы позволяет более уверенно чувствовать себя в мире Web-разработки, т. к., несмотря на кросс-платформенную независимость языка PHP, различие операционных систем Windows и UNIX дает о себе знать (особенно при работе с файловой системой). Кроме того, для многих параметров сервера не предусматривается Web-интерфейс, и ими можно управлять только через удаленный доступ (SSH), при работе через который необходимы глубокие знания команд операционной системы.

Web-сервер

За формирование Web-страниц на сервере несет ответственность Web-сервер. Существуют несколько различных Web-серверов, наиболее известные: IIS-сервер операционной системы Windows и Apache, применяющийся главным образом на UNIX-серверах (впрочем, под управлением Windows Apache также прекрасно работает). При использовании PHP мы будем ориентироваться на Web-сервер Apache, как наиболее распространенный. Помимо того, что он реализует протокол HTTP и позволяет обрабатывать обращения клиентов, имеет множество модулей, позволяющих преобразовывать URL-адреса, осуществлять переадресацию, подключать языки программирования (PHP, Perl и т. п.), защищать папки и файлы паролем и многое другое. Сервер Apache и его модули имеют множество директив, которые могут применяться как в основном конфигурационном файле, так и в специальном файле .htaccess, позволяющем настраивать работу файлов в отдельных папках. Изучение принципов работы и директив Web-сервера Apache позволят создавать более эффективные Web-приложения, а также решать множество задач в области управления сайтами, управление которыми возможно только на уровне Web-сервера.

Серверный язык

Следующим компонентом выступает серверный язык, позволяющий формировать динамические страницы. Практически любой серверный язык можно использовать для создания динамических страниц.

> ***ЗАМЕЧАНИЕ***
>
> На заре Интернета, когда язык программирования PHP еще не существовал, авторы для создания сайтов успешно использовали C++ и даже Fortran.

Однако наибольшее распространение получили специализированные языки, которые либо позволяли легко обрабатывать текст (Perl), либо имели встроенные средства для работы с Web-средой (ASP, PHP, Java). В нашей стране наибольшую популярность приобрел язык программирования PHP, рассматривающийся в данной книге.

Точно также как и Web-сервер Apache, PHP имеет ядро, где сосредоточены базовые функции и множество подключаемых модулей. Наиболее популярные и хорошо зарекомендовавшие себя модули поставляются в официальном дистрибутиве PHP, другие модули можно загрузить в дополнительном архиве или найти на многочисленных сайтах в Интернете. В любом случае язык программирования постоянно перестраивается, некоторые модули устаревают и исключаются из ядра и дистрибутива, другие, получившие популярность — включаются. Поэтому язык PHP предоставляет чрезвычайно гибкий инструмент разработки, подстраивающийся под динамично развивающуюся Web-среду.

PHP — не самый быстрый язык программирования, однако один из самых удобных, что позволяет быстро разрабатывать Web-приложения. В условиях, когда среднее время жизни приложения составляет полгода, а сами приложения постоянно подвергаются модификации, скорость разработки приложений становится решающим фактором. Ранее компьютерное время стоило дороже времени разработчиков (да и программы эксплуатировались значительно более длительное время). В настоящий момент ситуация изменилась кардинально — приобрести быстрый сервер проще и дешевле, чем нанимать 10 программистов на C++, разрабатывающих приложение за полгода, вместо одного PHP-программиста, разрабатывающего приложение за один месяц.

> ***Замечание***
>
> Широкая распространенность PHP вовсе не означает, что все остальные серверные языки следует игнорировать — чем больше языков программирования вы освоите, тем более конкурентоспособными вы будете и тем лучше освоите Web-программирование. Помимо PHP, при Web-разработке часто используются Perl, семейство языков ASP.NET (VB, C#, ASP), Java, C/C++, Ruby и Phyton.

Файлы и базы данных

Файлы играют значительную роль в любом языке программирования или технологии, в них хранятся как сами программы, так и данные, которые вводятся пользователями, обрабатываются скриптами и архивируются на сервере. Зачастую к серверу одновременно обращается огромное количество пользователей, что создает определенные трудности при записи информации в один и тот же файл, т. к. сразу несколько потоков стремятся получить доступ к файлам, что при игнорировании этой проблемы может приводить к повреждению записанной в него информации.

Для того чтобы каждый раз при разработке очередного Web-приложения не решать проблемы одновременного доступа к файлу, используются базы данных (или, точнее, системы управления базами данных — СУБД). СУБД разрабатываются десятки лет профессионалами своей области с использованием быстрых компилируемых языков (C, C++). Поэтому СУБД в Web-приложениях всегда быстрее файлового доступа. Кроме того, код с использованием базы данных зачастую в несколько раз меньше по объему (а, следовательно, разрабатывается он в более короткие сроки), чем код, использующий файлы. Чем быстрее вы освоите приемы работы с СУБД, тем более быстродействующим и элегантным будет ваш код, и тем быстрее вы сможете его создавать.

Совместно с PHP может использоваться множество баз данных, начиная с коммерческих Oracle и MS SQL, заканчивая свободно-распространяемыми MySQL, PostgreeSQL и FireBird. Все СУБД объединяет тот факт, что взаимодействие с ними осуществляется при помощи языка структурированных запросов SQL. Взаимодействие с СУБД сводится к выполнению запросов, построенных при помощи специализированного языка SQL. Язык SQL является общим для всех баз данных — освоив одну из баз данных, вы без труда сможете изучить любую другую.

> ***ЗАМЕЧАНИЕ***
>
> Следует все же отметить, что, несмотря на стандартизацию языка SQL, различные базы данных имеют разные SQL-диалекты, иногда достаточно сильно отличающиеся друг от друга.

Одной из самых популярных баз данных, применяемых в Web-разработке, является СУБД MySQL, которая заслужила признательность разработчиков благодаря свободному распространению, высокому быстродействию и надежности, а также богатой функциональности.

Электронная почта

Электронная почта появилась задолго до Web-среды и является в настоящий момент неотъемлемой частью Интернета. Пользователи привыкли отправлять и получать почтовые сообщения. Язык программирования PHP имеет средства как для отправки почтовых сообщений, так и для доступа к почтовым ящикам (локальным или расположенным на удаленном сервере).

1.2.2. Клиентские технологии

Серверные и клиентские технологии разделены в пространстве и времени. PHP выполняется на сервере и формирует страницу, содержащую HTML-разметку, JavaScript-код, Flash-ролики, которые отправляются по протоколу HTTP клиенту.

Такая страница интерпретируется браузером и отображается клиенту. В момент просмотра страницы клиентом PHP, Web-сервер Apache уже отработали и отослали страницу. Заставить их среагировать на какие-то действия пользователя можно, только повторно послав запрос серверу. То же самое касается клиентских технологий, которые не выполняются на сервере — они вступают в действие, только когда страница получена клиентом.

Web-браузеры, HTML

Клиенты Web-серверов используют различные операционные системы и браузеры, программы, предназначенные для просмотра Web-страниц. Наиболее популярные среди них на сегодняшний день: Internet Explorer, FireFox, Opera и Chrome. Одной из особенностей является тот факт, что все браузеры (а также их различные версии) по-разному интерпретируют язык разметки HTML, каскадные таблицы стилей CSS и клиентский язык JavaScript. Это требует от Web-разработчиков тестирования сайтов в нескольких браузерах (разных версий), а также зачастую отказа от нововведений, которые поддерживаются одними браузерами и не поддерживаются другими.

Если в области клиентских технологий среди производителей браузеров существуют разногласия, то взаимодействие с серверной частью все браузеры поддерживают очень хорошо. Взаимодействие между браузерами и Web-серверами осуществляется при помощи протокола HTTP.

> ***ЗАМЕЧАНИЕ***
>
> Существуют две версии протокола HTTP — 1.0 и 1.1. В настоящий момент везде (за исключением некоторых промежуточных прокси-серверов) используется протокол HTTP версии 1.1.

Каскадные таблицы стилей CSS и XML

Изначально HTML разрабатывался как язык разметки, т. е. язык, описывающий документ независимо от его внешнего вида. Однако очень скоро он стал применяться для создания дизайна, его конструкции стали использоваться для того, чтобы добиться того или иного внешнего эффекта. Под давлением разработчиков Web-страниц и браузеров в HTML вносились все новые и новые элементы. Начались браузерные войны, когда производители браузеров намеренно вносили дополнительные элементы, не совместимые с другими браузерами.

Для того чтобы разделить структуру и оформление документов, консорциумом W3C, координирующим развитие Web, были введен язык разметки XML и каскадные таблицы стилей CSS, при помощи которых можно оформить XML-код. Язык

XML настолько гибок, что позволяет самостоятельно определить нужный XML-формат путем разработки своего собственного словаря. В настоящий момент уже разработано огромное число словарей XML, позволяющих описывать любую информацию — от химических реакций (CML, Chemical Markup Language) до финансовых операций (OFX, Open Financial Exchange). По сравнению с HTML, XML является более гибким форматом; повсеместного применения XML еще не получил, но это лишь вопрос времени.

Каскадные таблицы стилей позволяют наложить дизайн на XML-документ. Пока до окончательного разделения структуры и оформления далеко, XML еще не очень популярен в Web в качестве языка разметки. Однако каскадные таблицы стилей применяются очень интенсивно в HTML. Если требуется создание стильных современных дизайнов, без каскадных таблиц стилей CSS не обойтись.

Flash-ролики

Технология Flash позволяет реализовывать на стороне клиента сложные динамические эффекты в виде Flash-роликов. По сути каждый Flash ролик представляет собой мини-сайт, позволяющий демонстрировать сложные графические эффекты, реализовывать игры и т. п. Внешний вид Flash-роликов не зависит от браузера и этим выгодно отличается от связки HTML и CSS. К недостаткам Flash можно отнести значительный размер, необходимость наличия проигрывателя и невозможность индексирования информации роботами поисковых систем.

FTP-клиенты

Протокол HTTP позволяет как загружать файлы на сервер, так и скачивать их с сервера. При помощи серверных языков программирования создаются достаточно сложные Web-интерфейсы, позволяющие работать с файловой системой сервера через браузер. Однако такое взаимодействие не всегда является удобным, зачастую проще получить доступ к папке с файлами сайта на сервере через FTP-протокол и работать с этой папкой так, как если бы она была расположена на клиентской машине (загружать на сервер или скачивать с сервера файлы, создавать и переименовывать папки, выставлять права доступа). В подавляющем большинстве случаев, после того как сайт разработан, файлы, его составляющие, загружаются на удаленный сервер через FTP.

> ***Замечание***
>
> В качестве FTP-клиента часто выступают файловые менеджеры, такие как Far и Total Commander, однако существуют и специализированные FTP-менеджеры, такие как FTPCute.

Удаленный доступ к серверу. Протокол SSH

Иногда просто получить доступ к папке на удаленном сервере (как это происходит в случае FTP) не достаточно, необходимо иметь возможность выполнять команды и работать на сервере так, как если бы монитор, клавиатура и мышь были подсоединены непосредственно к серверу. В этом случае в действие вступает протокол SSH. В среде Windows доступ к удаленному серверу по SSH осуществляется при помощи утилиты PuTTY.

Глава 2

Быстрый старт

Несмотря на то, что PHP является универсальным языком программирования и может применяться для разработки практически любого программного обеспечения, основная его специализация — Web-разработка. Именно с этой целью он и проектировался, поэтому содержит множество инструментов для работы в Интернете.

> ***Замечание***
>
> В данной и последующих главах будем исходить из того, что интерпретатор PHP и Web-сервер установлены у вас на локальной машине и связаны таким образом, что скрипты, помещенные в файл с расширение php, обрабатываются интерпретатором, и по запросу к Web-серверу в браузере выводится результат работы. Если у вас не настроена связка Web-сервера и PHP, вы можете обратиться к *приложениям 1* и *2*, в которых детально описывается процесс развертывания Web-среды в операционных системах Windows и UNIX.

2.1. Скрипты

Язык программирования PHP считается скриптовым языком, поэтому программы, написанные на нем, называют *скриптами*. Главное отличие традиционных программ от скриптов заключается в том, что скрипты работают только в определенной среде и используют ресурсы данной среды. Программы, будь то компилируемые или интерпретируемые, не зависят от какого-то стороннего программного обеспечения. Например, скриптовый язык программирования JavaScript работает только в Web-браузерах, Visual Basic for Applications только в среде Microsoft Office. С использованием этих языков программирования невозможно создать программу, работающую без соответствующей среды. В случае PHP в качестве такой среды выступает Web-окружение (Web-сервер, сервер базы данных, почтовый сервер и т. п.).

Замечание

Впрочем, язык программирования PHP допускает создание программ, работающих независимо от Web-сервера, однако в такой форме он не получил сколько бы то ни было широкого распространения.

Одной из главных особенностей языка программирования PHP является тот факт, что его код может располагаться вперемешку с HTML-кодом. Для того чтобы интерпретатор PHP различал HTML- и PHP-код, последний заключается в специальные теги `<?php` и `?>`, между которыми располагаются конструкции и операторы языка программирования PHP. В листинге 2.1 приводится классический пример, выводящий в окно браузера при помощи конструкции `echo` фразу "Hello world!" ("Привет мир!"). Содержимое листинга 2.1 следует поместить в файл с расширением php, например, в файл index.php. По умолчанию файлы с другими расширениями не обрабатываются PHP-интерпретатором, и PHP-код остается необработанным. Впрочем, такое поведение Web-сервера можно изменить *(см. приложение 1)*.

Замечание

`echo` — это конструкция языка программирования, позволяющая вывести одну или несколько строк в окно браузера. Более подробно конструкция `echo` обсуждается в *разд. 5.2*, посвященном строковым функциям.

Листинг 2.1. Простейший PHP-скрипт, index.php

```
<html>
  <head>
    <title>Пример</title>
  </head>
  <body>
    <?php
       echo "Hello world!";
    ?>
  </body>
</html>
```

В результате работы скрипта из листинга 2.1 в окно браузера будет выведена фраза "Hello world!" (рис. 2.1).

При работе с серверными языками программирования, такими как PHP, следует помнить, что скрипты, расположенные между тегами `<?php` и `?>`, выполняются на сервере. Клиенту приходит лишь результат работы PHP-кода, в чем можно легко убедиться, просмотрев исходный код HTML-страницы. В листинге 2.2 приводится результат работы PHP-скрипта из листинга 2.1, который браузер получает от сервера.

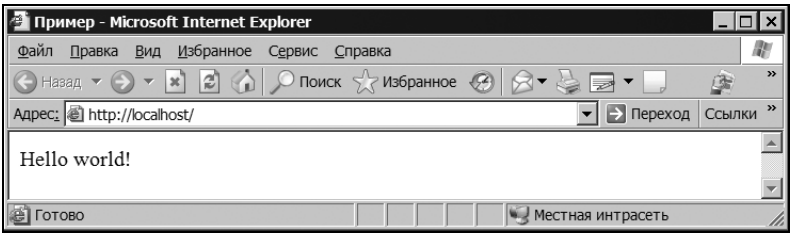

Рис. 2.1. Результат работы скрипта из листинга 2.1

ЗАМЕЧАНИЕ

Для просмотра исходного текста необходимо щелкнуть правой кнопкой мыши по странице и выбрать в контекстном меню соответствующий пункт, название которого зависит от браузера. В Internet Explorer необходимо выбрать пункт **Просмотр HTML-кода**, в Opera — **Исходный текст**, в FireFox — **View Page Source**.

Листинг 2.2. Исходный код HTML-страницы, которую получает браузер

```
<html>
  <head>
    <title>Пример</title>
  </head>
  <body>
    Hello world!  </body>
</html>
```

Если связка PHP и Web-сервера настроена неправильно, или PHP-скрипт был размещен в файле с расширением, отличным от php, например, в файле index.html, то в исходном коде HTML-страницы вместо содержимого листинга 2.2 можно увидеть содержимое листинга 2.1. В этом случае все, что расположено между тегами `<?php` и `?>`, будет рассматриваться браузером как один большой неизвестный ему HTML-тег, а окно браузера будет пустым (рис. 2.2).

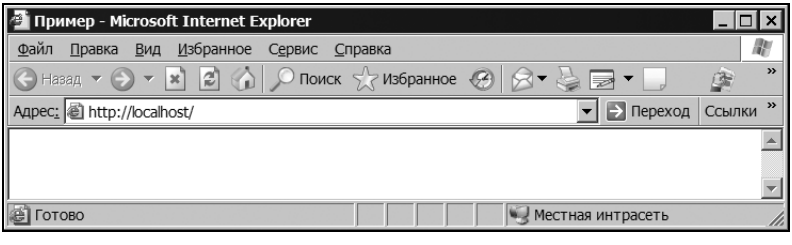

Рис. 2.2. PHP-код не подвергся интерпретации

2.2. Начальные и конечные теги

Как было указано в предыдущем разделе, PHP-скрипт должен быть размещен между начальным тегом `<?php` и конечным тегом `?>` для того, чтобы интерпретатор мог разделить HTML- и PHP-коды. Даже если HTML-код не используется, указание PHP-тегов является обязательным, в противном случае PHP-код будет выведен в окно браузера как есть, без интерпретации. Помимо тегов `<?php` и `?>`, PHP поддерживает еще ряд тегов, представленных в табл. 2.1.

Замечание

До версии PHP 6.0 поддерживались начальные и конечные теги, характерные для ASP-скриптов, `<%` и `%>`, включить которые можно было в конфигурационном файле php.ini, назначив директиве `asp_tags` значение `On`. В версии PHP 6.0 данный тип тегов был исключен.

Таблица 2.1. *Варианты PHP-тегов*

Вариант тегов	Описание
`<?php ... ?>`	Классический вариант PHP-тегов, именно его рекомендуется использовать в повседневной практике
`<? ... ?>`	Краткий вариант PHP-тегов, работает только в том случае, если в конфигурационном файле php.ini включена (т. е. принимает значение `On`) директива `short_open_tag`. По умолчанию эта директива включена, однако все-таки лучше ориентироваться на полный вариант PHP-тегов `<?php ... ?>`. Последнее особенно актуально при использовании PHP совместно с XML-кодом, который использует теги `<?xml ... ?>`. При использовании коротких тегов может возникнуть неоднозначность в интерпретации

`<script language="php">`
`...`
`</script>`
	Расширенный вариант PHP-тегов, так же как и вариант `<?php ... ?>`, доступен в любой момент без дополнительных настроек конфигурационного файла php.ini. Данный вариант несколько громоздок и распознается не всеми PHP-редакторами, тем не менее, следует быть готовыми встретить и такой вариант в PHP-скриптах

`<?= ... ?>`	Специальный вид тегов, предназначенный для вывода простого выражения в окно браузера. Конструкция `<?= ... ?>` эквивалентна `<?php echo ... ?>`

Отдельно следует отметить конструкцию `<?= ... ?>`, которую удобно использовать, когда скрипт состоит из одного выражения.

Например, скрипт из листинга 2.1 можно было бы переписать так, как это продемонстрировано в листинге 2.3.

Листинг 2.3. Использование тегов <?= ... ?>

```
<html>
  <head>
    <title>Пример</title>
  </head>
  <body>
    <?= "Hello world!" ?>
  </body>
</html>
```

Следует отметить, что HTML-страница может содержать более чем одну PHP-вставку. В листинге 2.4 приводится пример, который содержит две вставки: одна задает название страницы (в HTML-теге `<title>`), а вторая определяет содержимое страницы (в HTML-теге `<body>`).

> ***ЗАМЕЧАНИЕ***
>
> В листинге 2.4 используется функция `date()`, которая возвращает текущую дату и время. Более подробно эта функция обсуждается в *главе 11*, посвященной функциям даты и времени.

Листинг 2.4. Допускается несколько PHP-вставок в HTML-код

```
<html>
  <head>
    <title><?php echo "Вывод текущей даты" ?></title>
  </head>
  <body>
    <?php
      echo "Текущая дата:<br>";
      echo date(DATE_RSS);
    ?>
  </body>
</html>
```

Результат работы скрипта из листинга 2.4 представлен на рис. 2.3.