ОГЛАВЛЕНИЕ

Вступление
КНИГА О ЯЗЫКЕ ПРОГРАММИРОВАНИЯ PHP

Я бы упал в обморок, но вы мне и так поверите.

Из телесериала «Альф»

В этой книге речь пойдет о языке программирования PHP. Это простой, красивый и элегантный язык, а еще он, в некотором смысле, особенный. Дело в том, что язык PHP предназначен для выполнения кодов на стороне сервера. Другими словами, мало знать сам язык, необходимо еще и понимать, как и для чего он используется. Это важное обстоятельство, поскольку понимание того, зачем нужен язык, часто является ключевым фактором для его эффективного применения.

НА ЗАМЕТКУ

Можно сказать, что PHP — язык, предназначенный не просто для программирования, а для веб-программирования.

Что такое PHP

Язык PHP используют для создания сайтов и веб-приложений. Он обладает довольно богатой историей, чрезвычайно популярен среди разработчиков и поддерживается большинством хост-серверов.

ПОДРОБНОСТИ

Хостинг — это услуга по предоставлению ресурсов и места на сервере для размещения данных и информации (например, в виде веб-страниц). Соответственно, хост-сервером называется сервер, на котором размещена соответствующая информация. Для простоты и большей конкретики будем под хост-сервером подразумевать компьютер (сервер), на котором размещена веб-страница

пользователя (то есть та страница, при работе с которой предполагается использовать PHP).

У языка PHP есть автор — датский программист Расмус Лердорф. Начинался проект с разработки скриптов для поддержки персональной веб-страницы и имел рабочее название Personal Home Page Tools, или, сокращенно, PHP Tools. Постепенно все это трансформировалось в полноценный и эффективный программный продукт. Сегодня название PHP обычно ассоциируется с фразой Hypertext Preprocessor (букв. *«препроцессор гипертекста»* — прим. ред.), что не далеко от истины.

НА ЗАМЕТКУ

На момент написания книги актуальной является версия PHP 8. Тем не менее опыт свидетельствует о том, что самая последняя версия языка далеко не сразу начинает использоваться на практике. Здесь прослеживается некоторая инертность, обусловленная как объективными, так и субъективными факторами. Поэтому мы будем в основном рассматривать универсальные подходы, актуальные для нескольких последних версий языка. Интересен также тот факт, что шестой версии PHP нет: после пятой следует седьмая. Причина в том, что попытка выпустить в свет шестую версию оказалась крайне неудачной.

Язык PHP является скриптовым (сценарным) и интерпретируемым языком. Последнее означает, что программный код выполняется под управлением специальной программы, которая называется интерпретатором. Интерпретатор считывает построчно код программы и выполняет соответствующие инструкции.

СТАНДАРТ PHP8

В версии PHP 8 анонсировано появление JIT-компилятора (сокращение от Just in Time), предназначенного для компилирования кодов PHP, что призвано повысить скорость выполнения программ. При компиляции программные инструкции переводятся в команды уровня процессора.

Сценарные языки обычно относятся к высокоуровневым. Сценарии, в отличие от обычных программ, как правило, содержат инструкции для управления уже готовыми программными компонентами. Другими словами, сценарный язык — это когда все просто и понятно. Хотя, конечно, бывают и исключения.

ПОДРОБНОСТИ

Синтаксис языка PHP ощутил на себе влияние языка C. Поэтому читатели, изучавшие такие языки, как C, C++, C# и Java обнаружат для себя много знакомых синтаксических конструкций.

Безусловно, PHP можно рассматривать как один из языков программирования, которых на сегодняшний день существует довольно много. Но особенность все же есть — хотя бы в том, как используются коды, написанные на PHP.

Если мы имеем дело с любым обычным языком программирования, то процесс написания и использования программ выглядит примерно так: сначала мы создаем программный код — проще говоря, пишем программу. Затем эту программу необходимо выполнить. Здесь все зависит от языка, но общая схема связана с тем, что программный код компилируется или интерпретируется. При компиляции специальная программа-компилятор переводит программный код в машинные инструкции (или нечто похожее на них), после чего эти инструкции выполняются. При интерпретации специальная программа-интерпретатор считывает программный код и выполняет команды из этого кода. Но что бы ни происходило, важно то, что все это мы делаем на одном и том же компьютере. На этом компьютере мы и программу запускаем, и результат получаем. С результатом выполнения программы мы можем делать все что угодно, главное, что в принципе одного компьютера достаточно.

С языком PHP дела обстоят несколько иначе. Однако для того, чтобы понять, в чем именно заключается разница, стоит сначала рассмотреть общий механизм того, что происходит в сети в случае обращения к веб-страницам и какое место в этом процессе занимает PHP.

Клиент и сервер

Рассмотрим в самых общих чертах схему, в соответствии с которой выполняется просмотр сайта в сети. Главными действующими «персонажами» в данном случае выступают два компьютера: тот, на котором мы хотим посмотреть веб-страницу, и тот, на котором эта страница фактически находится. Первый компьютер (на котором мы пытаемся посмотреть веб-страницу) называется *клиентом*, второй (на котором веб-страница находится)– *сервером*. Между этими компьютерами существует связь через глобальную сеть,

поэтому они могут обмениваться информацией. Мы работаем на компьютере-клиенте и хотим просмотреть веб-страницу. Для этого мы запускаем специальную программу, называемую браузером. Мы открываем браузер (например, Chrome, Opera или Edge) и в адресной строке вводим адрес нужного нам сайта. Браузер инициирует запрос на сервер. Сервер получает запрос, обрабатывает его, и возвращает ответ на компьютер клиента. Фактически он возвращает код документа, который браузер должен отобразить на экране клиента. Общая схема этого «взаимодействия» проиллюстрирована на рис. В.1.

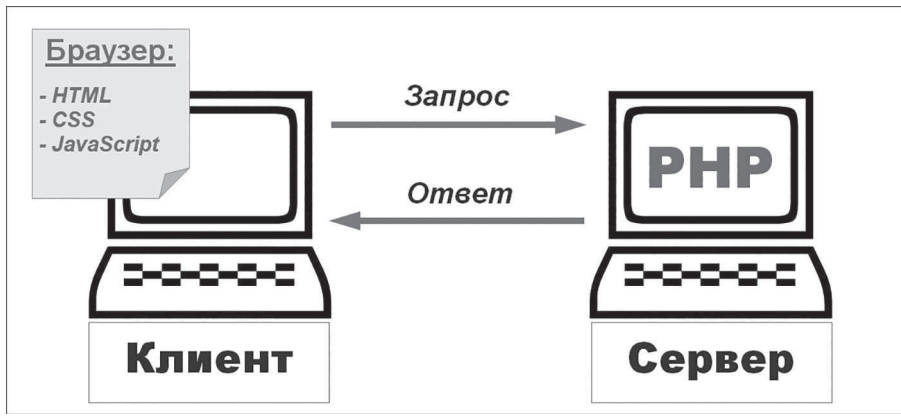

Рис. В.1. Схема взаимодействия между клиентом и сервером

Браузер обрабатывает документ, представленный в формате HTML (от Hypertext Markup Language). Это текст, содержащий специальную разметку. Браузер данную разметку «понимает» и отображает документ в соответствии с тем, как предписано встроенными в документ инструкциями. Помимо собственно разметки HTML, документ может содержать и нечто большее. Например, в нем может применяться форматирование, определяемое каскадными таблицами стилей (аббревиатура CSS от Cascading Style Sheets). Он также может содержать сценарии на языке JavaScript. В последнем случае браузер выполняет эти сценарии. То есть получается, что сервер высылает набор команд, а браузер их выполняет. Важно то, что выполняются команды на том же компьютере, с которого осуществлялся запрос.

НА ЗАМЕТКУ

Выражаясь простым языком, сервер сообщает, что нужно делать, а браузер на компьютере-клиенте выполняет необходимые операции. Удобно, но не всегда безопасно.

Где же в этой схеме место для PHP? Ответ такой: на этапе обработки запроса сервером. Когда сервер получает запрос от клиента, он обрабатывает его, и в процессе обработки могут выполняться сценарии — в данном случае написанные на языке PHP.

ПОДРОБНОСТИ

Часто результатом выполнения сценария является сгенерированный код HTML, который передается клиенту.

Но это еще не все. Многие программы пытаются обмениваться информацией через сеть. Процессы на компьютере-клиенте посылают сигналы процессам на сервере и обратно, и нужно знать, какой сигнал какому процессу предназначен. Для этого используются порты — специальные целочисленные идентификаторы, с помощью которых процессы «находят» предназначенные для них сигналы. Соответственно, запросы с браузера клиента и ответы сервера должны быть синхронизированы по порту. Таким образом, для эффективной работы с PHP нужно решить довольно много технических проблем. Все это мы будем рассматривать постепенно, по мере необходимости.

Как выполняется код PHP

Перейдем к практической стороне изучения PHP. А именно, зададимся вопросом: как выполнить программный код, написанный на языке PHP? Если рассматривать «естественный» режим использования PHP-кода, то для этой цели нам понадобятся сервер и клиент. То есть два компьютера. В данном случае сценарий (программа) размещается на сервере, а посмотреть результат его выполнения можно обратившись к веб-странице на сервере через браузер клиента. Но даже если все эти ресурсы есть под рукой, такой режим, мягко говоря, не очень удобен, поскольку редактировать программный код придется на одном компьютере (сервере), а проверять результат — на другом (клиенте). Конечно, нужно более комфортное решение.

Альтернативный способ заключается в том, чтобы «обмануть» браузер — создать иллюзию, что сервером является сам клиент. Имеется в виду использование локального сервера. В такой режим перейти несложно. Преимущество подхода связано с тем, что и программа, и ее результат локализованы в пределах одного компьютера. При этом программный код на PHP можно выполнять с помощью интерпретатора — то есть примерно так же, как и в случае с другими интерпретируемыми языками. Наверное, это самый

простой способ увидеть результат выполнения программы. Правда, не следует забывать, что пишутся программы на PHP совсем не для того, чтобы выполняться интерпретатором на компьютере клиента. Так что здесь тоже есть свои тонкости.

Программное обеспечение

Книга содержит много примеров, и в процессе их изучения желательно не только разбирать программный код, но еще и исследовать результат его выполнения. Для этого понадобится специальное программное обеспечение.

Прежде всего, нужно установить поддержку для PHP. С этой целью переходим на страницу `www.php.net`, как показано на рис. В.2.

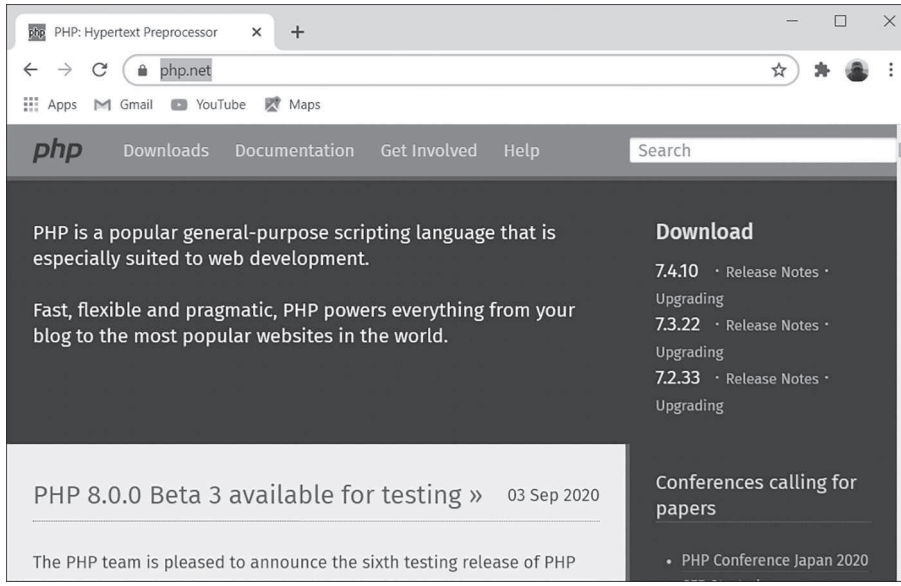

Рис. В.2. Окно поддержки PHP по адресу `www.php.net`

На данной странице необходимо найти раздел загрузки программного обеспечения, и скачать нужные файлы.

> **НА ЗАМЕТКУ**
>
> В самом простом варианте установка сводится к распаковке архива, загруженного с сайта `www.php.net`. Скорее всего, этого будет достаточно для работы с интерпретатором PHP (файл `php.exe`)

в режиме командной строки. Для более «комфортной» работы, с привлечением вспомогательных программных средств, возможно, придется выполнить дополнительные настройки. В таком случае следует воспользоваться справочной информацией на странице `www.php.net` и справкой по соответствующему программному продукту (например, редактору кодов).

ПОДРОБНОСТИ

Узнать версию PHP можно с помощью инструкции `php -v` в командной строке. Для получения справки по PHP используют команду `php -h`. Более полную информацию о PHP позволяет получить команда `php -i`.

Если используется операционная система Windows, то для перехода в режим терминала в адресном поле Проводника (Windows Explorer) можно ввести инструкцию `cmd`. Затем в окне терминала следует перейти в каталог с PHP. Например, если PHP находится в папке `C:\PHP`, соответствующая команда будет выглядеть так: `cd C:\PHP`. Альтернатива — сначала перейти в каталог с PHP, а уже затем в адресной строке Проводника ввести инструкцию `cmd`.

Во многих операционных системах семейства Linux PHP предустановлен. Но если это не так, то для установки можно воспользоваться командой `sudo apt install php`.

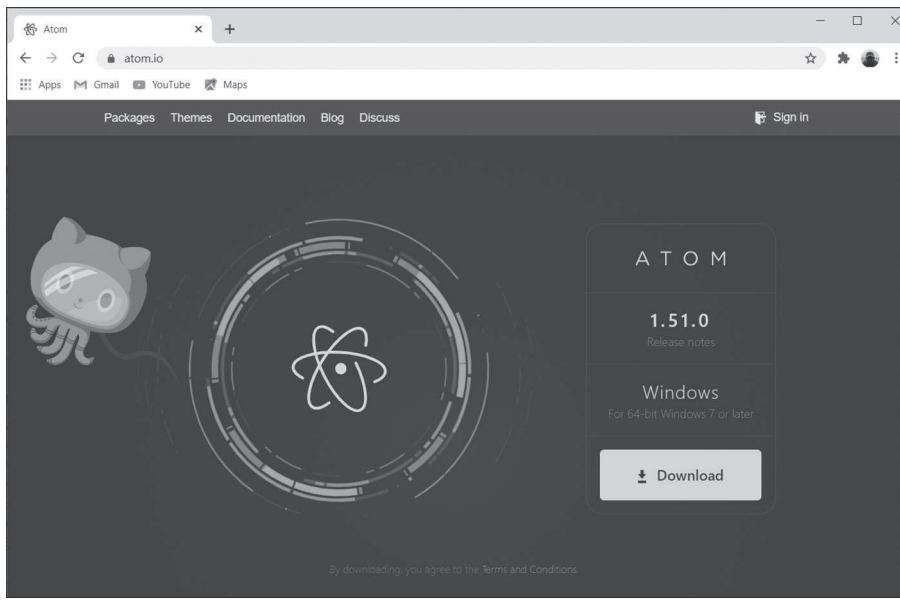

Рис. В.3. Страница загрузки редактора Atom `www.atom.io`

В целом на этом можно остановиться. Среди загруженных утилит должен присутствовать интерпретатор PHP, позволяющий выполнять программные коды. Однако сами коды нужно где-то набирать. Для этого подойдет и обычный текстовый редактор. Но лучше установить что-то более продвинутое, с поддержкой синтаксиса PHP (то есть редактор, «понимающий» специальные инструкции PHP). Например, можно использовать редактор Atom (адрес `www.atom.io`). Окно браузера, открытое на странице поддержки проекта, представлено на рис. В.3.

Конечно, есть и другие варианты. Достаточно удобна среда разработки Visual Studio Code (адрес `https://code.visualstudio.com`). Окно браузера, открытое на странице проекта, показано на рис. В.4.

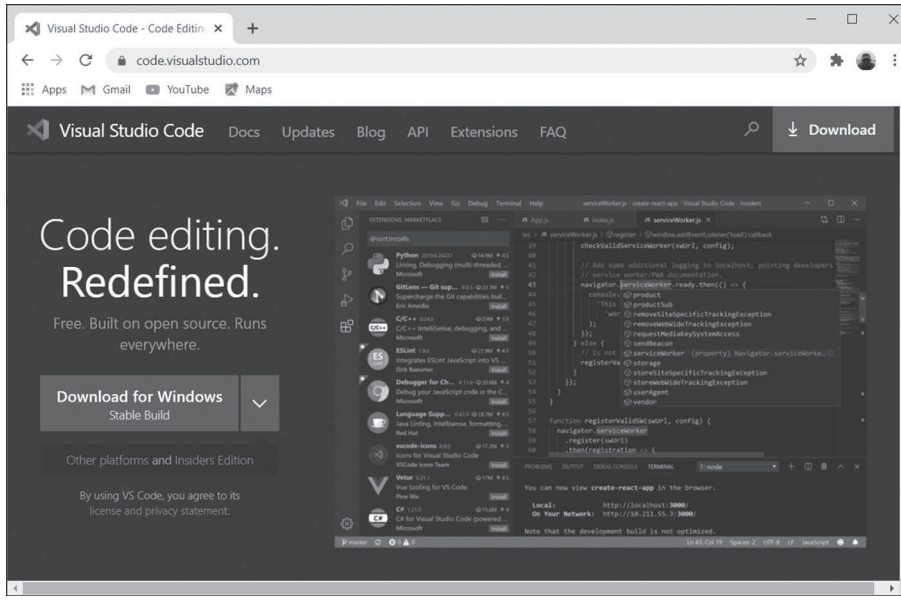

Рис. В.4. Страница поддержки проекта Visual Studio Code

ПОДРОБНОСТИ

При первом запуске приложения Visual Studio Code в разделе Customize следует подтвердить установку поддержки для PHP.

Еще один неплохой вариант — среда разработки NetBeans. Установочные файлы можно загрузить по адресу `http://netbeans.apache.org`. На рис. В.5 показано окно браузера, открытое на странице поддержки проекта NetBeans.

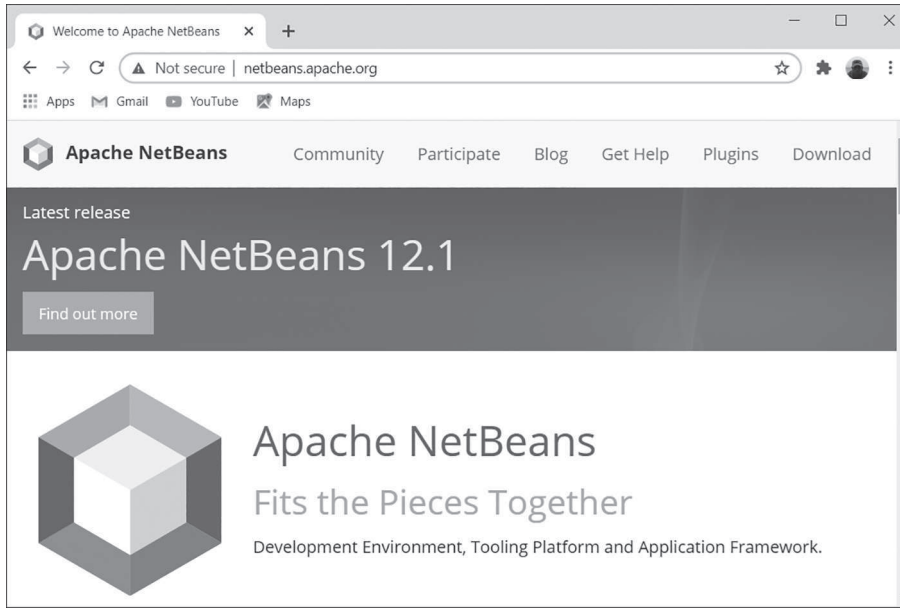

Рис. В.5. Страница поддержки проекта NetBeans

Существуют и другие полезные программные продукты, в том числе и коммерческие. Однако подчеркнем еще раз, что в целом достаточно установить PHP и подобрать приличный редактор кодов.

> **НА ЗАМЕТКУ**
>
> Методы работы с программным обеспечением (в небольшом объеме) мы будем изучать по мере рассмотрения примеров.

О книге

Книга всецело посвящена языку PHP. Мы сфокусируемся на том, что это за язык, зачем он нужен и в чем заключаются его особенности. Мы рассмотрим основные синтаксические конструкции языка, а также подходы и механизмы, применяемые в программировании на PHP. Значительное внимание в книге уделено примерам. Диапазон обсуждаемых вопросов достаточен для того, чтобы после изучения материала писать эффективные программные коды на PHP. Основная часть книги состоит из двенадцати глав.

- В первой главе приводятся примеры простых программ, обсуждаются способы использования программного обеспечения и анализируются общие принципы создания программ на PHP.

- Во второй главе рассматриваются переменные и типы данных. Мы узнаем, как создаются переменные, какие операции с ними выполняются и в чем особенности использования данных разных типов.

- Третья глава посвящена управляющим инструкциям. Мы познакомимся с условным оператором, операторами выбора, операторами цикла, а также с инструкцией безусловного перехода. Все эти синтаксические конструкции критически важны для создания функциональных программ на PHP.

- Четвертая глава содержит информацию о массивах. В языке PHP массивы имеют ряд уникальных особенностей, и именно эти особенности являются предметом обсуждения. В этой главе рассматриваются одномерные и многомерные массивы, описывается оператор цикла по коллекции, а также дается обзор основных операций, выполняемых с массивами.

- В пятой главе описываются функции. Мы узнаем, как они создаются, обсудим механизмы передачи аргументов, способ возвращения результата, научимся задавать для аргументов значения по умолчанию, создавать функции с произвольным количеством аргументов, рассмотрим способ передачи аргументов по имени. Также в этой главе уделяется внимание рекурсии, анонимным функциям и ряду других тем, относящихся к функциям.

- В шестой главе рассматриваются ссылки, константы, глобальные и статистические переменные, работа с файлами и методы работы с многострочным текстом.

- В седьмой главе обсуждаются принципы объектно-ориентированного программирования (ООП) и механизмы реализации этих принципов в PHP. Кроме прочего, мы узнаем, как создаются классы и объекты, в чем особенность методов, познакомимся с конструкторами и деструкторами, статическими членами класса. В данной главе рассматривается задача о копировании объектов, а также иллюстрируются особенности использования закрытых полей и методов. Также в ней описываются некоторые специальные методы.

- Восьмая глава посвящена такому важному механизму ООП, как наследование. Мы узнаем, как создается дочерний класс и как переопределяются методы. Мы рассмотрим особенности использования конструкторов, закрытых и защищенных членов класса при наследовании. Кроме того, в данной главе обсуждается виртуальность

методов, описывается многоуровневое наследование, рассматриваются некоторые другие темы.

- В девятой главе описываются абстрактные классы и интерфейсы (в том числе, их реализация и наследование). Мы узнаем, что такое трейты, как с помощью интерфейса можно контролировать тип объекта, что такое пространство имен и как оно используется.
- Принципы обработки ошибок и исключений рассматриваются в десятой главе. Мы познакомимся с основными классами исключений, узнаем, как они генерируются, рассмотрим методы создания классов для пользовательских исключений. В данной главе освещаются также и некоторые другие вопросы, имеющие отношение к обработке ошибок и исключений.
- Одиннадцатая глава посвящена генераторам и итераторам. Мы научимся использовать функции-генераторы, применять их в разных ситуациях. Еще мы узнаем, что такое итераторы и как они создаются.
- В двенадцатой главе рассматриваются примеры использования PHP-программ на практике. Глава содержит детальный разбор нескольких задач, которые дают представление о том, какова роль кодов PHP в создании веб-документов.

В конце каждой главы для удобства приводится краткое *Резюме*, в котором перечисляются все основные моменты, рассмотренные в ней. Это способствует усвоению материала книги.

НА ЗАМЕТКУ

Книга рассчитана в первую очередь на тех, кто имеет минимальную подготовку в программировании. Соответственно, материал по возможности подается максимально просто.

Об авторе

Васильев Алексей Николаевич, доктор физико-математических наук, профессор кафедры теоретической физики Киевского национального университета имени Тараса Шевченко. Автор книг по программированию и математическому моделированию. Сфера научных интересов: физика жидкостей и жидких кристаллов, синергетика, биофизика, экономика и математическая лингвистика.