

РУКОВОДСТВО ПО РАЗРАБОТКЕ
В ПРИМЕРАХ ОТ СООБЩЕСТВА STACK OVERFLOW

PHP

ДЛЯ ПРОФЕССИОНАЛОВ

Издательство АСТ
г. Москва

УДК 004.438
ББК 32.973.2
ПЗ2

*Последняя версия книги «PHP Notes for Professionals» на английском языке доступна для скачивания по адресу: <https://goalkicker.com/PHPBook>.
Делитесь этим PDF-документом со всеми, кому он может быть полезен.
Книга составлена на основе документации Stack Overflow (<https://archive.org/details/documentation-dump.7z>), содержание которой написано замечательными участниками сообщества Stack Overflow.
Текстовые материалы распространяются на условиях лицензии Creative Commons BY-SA. Список авторов, участвовавших в создании различных разделов, приведен в конце книги. Авторские права на изображения принадлежат их владельцам, если не указано иное.*

PHP для профессионалов. Руководство по разработке в примерах ПЗ2 от сообщества Stack Overflow. — Москва : Издательство АСТ, 2026. — 512 с. : ил. — (Быстрый старт в программирование).

ISBN 978-5-17-187779-8

На протяжении многих лет язык PHP остается одним из ключевых инструментов веб-разработки. Он широко используется для создания динамических сайтов, веб-сервисов, систем управления контентом и высоконагруженных интернет-приложений. Благодаря своей гибкости и развитой экосистеме PHP применяется как в небольших проектах, так и в крупных корпоративных системах.

Перед вами не классический учебник, а практическое руководство, построенное в формате набора решений для реальных задач. Оно ориентировано на разработчиков, которые хотят не только освоить синтаксис языка, но и научиться эффективно применять его в повседневной работе. Издание можно использовать как справочник для поиска оптимального выхода из сложных и нестандартных ситуаций.

Особое внимание уделено вопросам, которые редко рассматриваются в базовых курсах: работе с современными возможностями языка, организации кода, обработке ошибок, взаимодействию с базами данных, безопасности приложений и оптимизации производительности. Рассматриваются практические кейсы, с которыми разработчики сталкиваются в реальных проектах.

Каждый раздел посвящен отдельной теме и сопровождается примерами, основанными на реальном опыте профессионалов. Такой подход позволяет не только понять, «как это работает», но и увидеть, как правильно применять возможности PHP на практике.

Книга будет полезна и начинающим разработчикам, стремящимся выйти на профессиональный уровень, и опытным специалистам, которым требуется удобный и надежный справочник для решения ежедневных задач.

УДК 004.438
ББК 32.973.2

ISBN 978-5-17-187779-8

Перевод на русский язык: ООО «Интеджер».
Издание на русском языке: ООО «Издательство АСТ».

Содержание

Раздел 1. Начало работы с PHP	13
Глава 1.1. HTML-вывод с веб-сервера	14
Глава 1.2. Hello, World!	15
Глава 1.3. Вывод с веб-сервера, отличный от HTML.....	15
Глава 1.4. Встроенный сервер PHP	17
Глава 1.5. PHP CLI.....	17
Глава 1.6. Разделение инструкций	19
Глава 1.7. Теги PHP	19
Раздел 2. Переменные	21
Глава 2.1. Динамический доступ к переменной по имени (переменные переменных).....	21
Глава 2.2. Типы данных.....	23
Глава 2.3. Рекомендации по работе с глобальными переменными	25
Глава 2.4. Значения по умолчанию для неинициализированных переменных	26
Глава 2.5. Значения переменных в логическом контексте и оператор тождественности	27
Раздел 3. Область видимости переменной	31
Глава 3.1. Суперглобальные переменные	31
Глава 3.2. Статические свойства и переменные.....	31
Глава 3.3. Глобальные переменные, определяемые пользователем.....	32
Раздел 4. Суперглобальные переменные PHP	34
Глава 4.1. Что такое суперглобальные переменные?.....	34
Глава 4.2. Суперглобальные переменные в PHP 5	41
Раздел 5. Вывод значения переменной	46
Глава 5.1. echo и print	46
Глава 5.2. Вывод структурированного представления массивов и объектов.....	47
Глава 5.3. Конкатенация строк с помощью echo.....	49
Глава 5.4. Функции printf и sprintf.....	50
Глава 5.5. Вывод больших целых чисел	50
Глава 5.6. Вывод многомерного массива с индексами и значениями в таблицу.....	51
Раздел 6. Константы	53
Глава 6.1. Определение констант	53
Глава 6.2. Константы классов	54
Глава 6.3. Проверка, определена ли константа	55
Глава 6.4. Использование констант	56
Глава 6.5. Массивы констант	56
Раздел 7. Магические константы	58
Глава 7.1. Разница между __FUNCTION__ и __METHOD__	58
Глава 7.2. Разница между __CLASS__, get_class() и get_called_class()	58
Глава 7.3. Константы файлов и каталогов	59
Раздел 8. Комментарии	60
Глава 8.1. Однострочные комментарии.....	60
Глава 8.2. Многострочные комментарии.....	60
Раздел 9. Типы	61
Глава 9.1. Сравнение типов	61
Глава 9.2. Логический тип (Boolean)	61
Глава 9.3. Числа с плавающей точкой (float)	62
Глава 9.4. Строки.....	63
Глава 9.5. Callable-выражения	65
Глава 9.6. Ресурсы.....	66
Глава 9.7. Приведение типов	66
Глава 9.8. Неявное приведение типов.....	67
Глава 9.9. Null	67
Глава 9.10. Целые числа	67
Раздел 10. Операторы	69
Глава 10.1. Оператор объединения с null (??).....	69

Глава 10.2. Оператор «космический корабль» (<=>).....	70
Глава 10.3. Оператор выполнения (` `).....	70
Глава 10.4. Операторы инкремента (++) и декремента (--)	71
Глава 10.5. Тернарный оператор (?:).....	71
Глава 10.6. Логические операторы (&&/AND и /OR).....	72
Глава 10.7. Операторы для работы со строками (. и .=).....	72
Глава 10.8. Операторы объектов и классов	72
Глава 10.9. Составные операторы присваивания (+= и т. д.)	74
Глава 10.10. Изменение приоритета операторов (с помощью скобок).....	75
Глава 10.11. Базовая операция присваивания (=).....	75
Глава 10.12. Ассоциативность операторов	75
Глава 10.13. Операторы сравнения.....	76
Глава 10.14. Побитовые операторы	77
Глава 10.15. Оператор instanceof (оператор типа).....	80
Раздел 11. Ссылки	82
Глава 11.1. Присваивание по ссылке	82
Глава 11.2. Возвращение значения по ссылке	82
Глава 11.3. Передача по ссылке	83
Раздел 12. Массивы	85
Глава 12.1. Инициализация массива	85
Глава 12.2. Проверка существования ключа.....	87
Глава 12.3. Проверка типа массива	88
Глава 12.4. Создание массива переменных	88
Глава 12.5. Проверка содержания значения в массиве	89
Глава 12.6. Интерфейсы ArrayAccess и Iterator.....	90
Раздел 13. Итерация массивов	94
Глава 13.1. Итерация нескольких массивов вместе.....	94
Глава 13.2. Использование инкремента индекса.....	95
Глава 13.3. Использование внутренних указателей массивов.....	95
Глава 13.4. Использование конструкции foreach.....	96
Глава 13.5. Использование итератора ArrayObject.....	98
Раздел 14. Работа с массивами.....	99
Глава 14.1. Применение функции к каждому элементу массива	99
Глава 14.2. Разбиение массива на части.....	100
Глава 14.3. Преобразование массива в строку	101
Глава 14.4. «Деструктуризация» массивов с помощью функции list()	101
Глава 14.5. Функция array_reduce	102
Глава 14.6. Добавление значения в массив	103
Раздел 15. Действия с массивами	104
Глава 15.1. Применение фильтров к массиву.....	104
Глава 15.2. Удаление элементов из массива	105
Глава 15.3. Сортировка массива.....	106
Глава 15.4. Белый список из некоторых разрешенных ключей массива	110
Глава 15.5. Добавление элемента в начало массива.....	111
Глава 15.6. Обмен местами ключей с их значениями	112
Глава 15.7. Слияние двух массивов в один	112
Раздел 16. Совместная обработка нескольких массивов.....	113
Глава 16.1. Пересечение массивов	113
Глава 16.2. Слияние и конкатенация массивов	113
Глава 16.3. Преобразование многомерного массива в ассоциативный	114
Глава 16.4. Объединение двух массивов (ключи из одного, значения из другого).....	115
Раздел 17. Класс DateTime.....	116
Глава 17.1. Создание неизменяемой версии объекта DateTime из изменяемой (в версии до PHP 5.6).....	116
Глава 17.2. Добавление и вычитание временных интервалов	116
Глава 17.3. Метод getTimestamp.....	116
Глава 17.4. Метод setDate	117
Глава 17.5. Создание объекта DateTime из пользовательского формата	117
Глава 17.6. Вывод на экран объектов DateTime	117
Раздел 18. Работа с датой и временем	119
Глава 18.1. Получение разницы между двумя датами / моментами времени	119

Глава 18.2. Преобразование даты в другой формат.....	120
Глава 18.3. Разбор (парсинг) текстового представления даты на английском языке в формат Date	121
Глава 18.4. Использование предопределенных констант для формата даты.....	122
Раздел 19. Управляющие конструкции	123
Глава 19.1. if else	123
Глава 19.2. Альтернативный синтаксис для управляющих конструкций	123
Глава 19.3. while.....	124
Глава 19.4. do-while.....	124
Глава 19.5. goto.....	124
Глава 19.6. declare	125
Глава 19.7. require и include	125
Глава 19.8. return.....	126
Глава 19.9. for	127
Глава 19.10. foreach.....	127
Глава 19.11. if elseif else.....	127
Глава 19.12. if.....	128
Глава 19.13. switch	128
Раздел 20. Циклы	130
Глава 20.1. continue.....	130
Глава 20.2. break	131
Глава 20.3. foreach.....	132
Глава 20.4. do . . . while	133
Глава 20.5. for	133
Глава 20.6. while	134
Раздел 21. Функции	135
Глава 21.1. Списки аргументов переменной длины	135
Глава 21.2. Необязательные параметры	136
Глава 21.3. Передача аргументов по ссылке	137
Глава 21.4. Использование функций	138
Глава 21.5. Область видимости функции.....	138
Раздел 22. Программирование с использованием функций.....	139
Глава 22.1. Замыкание.....	139
Глава 22.2. Присваивание анонимных функций переменным	140
Глава 22.3. Объекты как функции.....	140
Глава 22.4. Использование внешних переменных	141
Глава 22.5. Анонимная функция.....	141
Глава 22.6. Чистые функции	142
Глава 22.7. Распространенные функциональные методы в PHP	142
Глава 22.8. Использование встроенных функций в качестве callback-функций.....	143
Глава 22.9. Область видимости	143
Глава 22.10. Передача callback-функции в качестве параметра	144
Раздел 23. Альтернативный синтаксис для управляющих конструкций.....	145
Глава 23.1. Альтернативный оператор if/else	145
Глава 23.2. Альтернативный оператор for	145
Глава 23.3. Альтернативный оператор while	145
Глава 23.4. Альтернативный оператор foreach	146
Глава 23.5. Альтернативный оператор case.....	146
Раздел 24. Форматирование строк	147
Глава 24.1. Интерполяция строк.....	147
Глава 24.2. Извлечение/замена подстрок.....	148
Раздел 25. Разбор строк	150
Глава 25.1. Разбиение строки по разделителям	150
Глава 25.2. Подстрока	150
Глава 25.3. Поиск подстроки с помощью функции strops.....	152
Глава 25.4. Разбор строки с помощью регулярных выражений.....	153
Раздел 26. Классы и объекты.....	155
Глава 26.1. Константы классов	155
Глава 26.2. Абстрактные классы	158
Глава 26.3. Позднее статическое связывание	160
Глава 26.4. Пространство имен и автозагрузка	161

Глава 26.5. Видимость методов и свойств.....	162
Глава 26.6. Интерфейсы	165
Глава 26.7. Ключевое слово <code>final</code>	168
Глава 26.8. Автоматическая загрузка классов.....	169
Глава 26.9. Вызов конструктора родительского класса при создании экземпляра дочернего класса	171
Глава 26.10. Динамическое связывание	171
Глава 26.11. Ключевые слова <code>\$this</code> , <code>self</code> и <code>static</code> плюс синглтон.....	172
Глава 26.12. Определение базового класса	175
Глава 26.13. Анонимные классы.....	176
Раздел 27. Пространства имен	178
Глава 27.1. Объявление пространств имен	178
Глава 27.2. Ссылка на класс или функцию в пространстве имен.....	178
Глава 27.3. Объявление подпространств имен.....	180
Глава 27.4. Что такое пространства имен?	180
Раздел 28. Сессии	181
Глава 28.1. Параметры <code>session_start()</code>	181
Глава 28.2. Блокировка сессий.....	181
Глава 28.3. Манипулирование данными сессии	182
Глава 28.4. Уничтожение всей сессии	183
Глава 28.5. Безопасный запуск сессии без ошибок.....	183
Глава 28.6. Название сессии	184
Раздел 29. Файлы <code>cookie</code>	185
Глава 29.1. Изменение файла куки (<code>cookie</code>).....	185
Глава 29.2. Установка куки (<code>cookie</code>)	186
Глава 29.3. Проверка установки <code>cookie</code> -файла.....	186
Глава 29.4. Удаление файлов <code>cookie</code>	186
Глава 29.5. Получение файлов <code>cookie</code>	187
Раздел 30. Буферизация вывода	188
Глава 30.1. Основное использование: получение содержимого между буферами и очистка	188
Глава 30.2. Обработка буфера с помощью функции обратного вызова (<code>callback-функции</code>)	189
Глава 30.3. Вложенные буферы вывода	189
Глава 30.4. Запуск буферизации перед выводом любого содержимого	190
Глава 30.5. Передача потока вывода клиенту	191
Глава 30.6. Использование буфера вывода для сохранения содержимого в файл (полезно для отчетов, счетов и т.д.)	191
Глава 30.7. Типичное использование функции <code>ob_start</code> и когда ее лучше всего применять.....	192
Глава 30.8. Перехват буфера вывода для последующего использования.....	192
Раздел 31. JSON	194
Глава 31.1. Декодирование строки JSON.....	194
Глава 31.2. Кодирование строки в формат JSON	197
Глава 31.3. Отладка ошибок JSON.....	200
Глава 31.4. Использование интерфейса <code>JsonSerializable</code> в объекте.....	202
Глава 31.5. Заголовок JSON и возвращаемый ответ	204
Раздел 32. SOAP-клиент	205
Глава 32.1. Режим WSDL.....	205
Глава 32.2. Режим без WSDL	205
Глава 32.3. Карты классов (<code>Classmaps</code>).....	205
Глава 32.4. Трассировка SOAP-запросов и ответов	206
Раздел 33. Использование <code>cURL</code> в PHP	208
Глава 33.1. Основное использование (<code>GET</code> -запросы)	208
Глава 33.2. <code>POST</code> -запросы.....	209
Глава 33.3. Использование файлов <code>cookie</code>	209
Глава 33.4. Использование <code>multi_curl</code> для выполнения нескольких <code>POST</code> -запросов.....	210
Глава 33.5. Отправка в одном запросе многомерных данных и нескольких файлов с помощью функции <code>CurlFile</code>	212
Глава 33.6. Создание и отправка запроса с помощью пользовательского метода.....	214
Глава 33.7. Получение и установка в PHP пользовательских <code>HTTP</code> -заголовков	215

Раздел 34. Отражение (Reflection)	216
Глава 34.1. Определение характеристик классов или объектов.....	216
Глава 34.2. Тестирование методов с модификаторами доступа private/protected.....	217
Глава 34.3. Доступ к членам класса с модификаторами доступа private/protected.....	218
Раздел 35. Внедрение зависимостей	221
Глава 35.1. Внедрение через конструктор.....	221
Глава 35.2. Внедрение через сеттеры.....	222
Глава 35.3. Внедрение контейнера	223
Раздел 36. XML	225
Глава 36.1. Создание XML с помощью класса DOMDocument	225
Глава 36.2. Чтение XML-документа с помощью класса DOMDocument	226
Глава 36.3. Использование XML с помощью библиотеки PHP SimpleXML	227
Глава 36.4. Создание XML-файла с помощью класса XMLWriter	230
Глава 36.5. Чтение XML-документа с помощью SimpleXML.....	230
Раздел 37. SimpleXML	232
Глава 37.1. Загрузка XML-данных в SimpleXML.....	232
Раздел 38. Парсинг HTML	233
Глава 38.1. Парсинг HTML из строки.....	233
Глава 38.2. Использование XPath.....	233
Глава 38.3. SimpleXML	233
Раздел 39. Регулярные выражения (regex/PCRE)	235
Глава 39.1. Глобальный поиск на основе регулярного выражения (RegExp).....	235
Глава 39.2. Сопоставление строк с помощью регулярных выражений	237
Глава 39.3. Разбиение строки на массив с помощью регулярного выражения	237
Глава 39.4. Замена строк с помощью регулярного выражения.....	238
Глава 39.5. Замена строк с использованием функций обратного вызова (callback-функций)	238
Раздел 40. Трейты (Traits)	239
Глава 40.1. Что такое трейт?.....	239
Глава 40.2. Трейты для облегчения горизонтального повторного использования кода.....	240
Глава 40.3. Разрешение конфликтов	241
Глава 40.4. Реализация паттерна «Одиночка» (Singleton) с помощью трейтов	242
Глава 40.5. Трейты для поддержания чистоты классов	243
Глава 40.6. Использование нескольких трейтов.....	244
Глава 40.7. Изменение области видимости методов.....	245
Раздел 41. Менеджер зависимостей Composer	246
Глава 41.1. Что такое Composer?	246
Глава 41.2. Автозагрузка с помощью Composer	247
Глава 41.3. Разница между 'composer install' и 'composer update'	248
Глава 41.4. Доступные команды Composer.....	248
Глава 41.5. Преимущества использования Composer	250
Глава 41.6. Установка Composer	250
Раздел 42. Магические методы	252
Глава 42.1. __call() и __callStatic().....	252
Глава 42.2. __get(), __set(), __isset() и __unset().....	253
Глава 42.3. __construct() и __destruct()	255
Глава 42.4. __toString()	255
Глава 42.5. __clone().....	256
Глава 42.6. __invoke()	257
Глава 42.7. __sleep() и __wakeup().....	257
Глава 42.8. __debugInfo().....	258
Раздел 43. Работа с файлами	259
Глава 43.1. Удобные функции	259
Глава 43.2. Удаление файлов и каталогов	261
Глава 43.3. Получение информации о файле	262
Глава 43.4. Поточковый ввод-вывод файлов	263
Глава 43.5. Перемещение и копирование файлов и каталогов.....	265
Глава 43.6. Минимизация использования памяти при работе с большими файлами	266

Раздел 44. Потоки	268
Глава 44.1. Регистрация обертки потока (stream wrapper).....	268
Раздел 45. Указание типа (type hint)	270
Глава 45.1. Указание типов данных для классов и интерфейсов	270
Глава 45.2. Указания типов для скалярных типов, массивов и вызываемых объектов (callable).....	271
Глава 45.3. Указания для Nullable-типов	272
Глава 45.4. Указания типов для обобщенных объектов.....	273
Глава 45.5. Указание типа для данных типа Void.....	274
Раздел 46. Фильтры и функции фильтрации	275
Глава 46.1. Проверка логических значений.....	275
Глава 46.2. Проверка, что число является числом с плавающей точкой.....	276
Глава 46.3. Проверка MAC-адреса	277
Глава 46.4. Очистка (нормализация) адресов электронной почты (Sanitze).....	277
Глава 46.5. Очистка целых чисел	277
Глава 46.6. Очистка URL-адресов	278
Глава 46.7. Проверка (валидация) адреса электронной почты	278
Глава 46.8. Проверка, что значение является целым числом	279
Глава 46.9. Проверка, что целое число находится в заданном диапазоне.....	280
Глава 46.10. Валидация URL-адреса	280
Глава 46.11. Проверка на соответствие числам с плавающей точкой	282
Глава 46.12. Проверка IP-адресов.....	284
Глава 46.13. Фильтры нормализации	286
Раздел 47. Генераторы (Generators)	287
Глава 47.1. Ключевое слово Yield.....	287
Глава 47.2. Чтение большого файла с помощью генератора.....	288
Глава 47.3. Зачем использовать генератор?.....	288
Глава 47.4. Использование функции send() для передачи значений генератору.....	289
Раздел 48. UTF-8	291
Глава 48.1. Ввод данных.....	291
Глава 48.2. Вывод данных.....	291
Глава 48.3. Хранение данных и доступ к ним	291
Раздел 49. Поддержка Unicode в PHP	293
Глава 49.1. Преобразование символов Unicode в формат «\uxxxx» с помощью PHP.....	293
Глава 49.2. Преобразование символов Unicode в их числовое значение и/или HTML-сущности с помощью PHP	293
Глава 49.3. Расширение Intl для поддержки Unicode.....	295
Раздел 50. URL-адреса	296
Глава 50.1. Разбор (парсинг) URL-адреса.....	296
Глава 50.2. Построение строки запроса в кодировке URL на основе массива.....	296
Глава 50.3. Перенаправление на другой URL-адрес	297
Раздел 51. Как разбить URL-адрес на составляющие	299
Глава 51.1. Использование функции parse_url()	299
Глава 51.2. Использование функции explode()	300
Глава 51.3. Использование basename()	301
Раздел 52. Сериализация объектов	302
Глава 52.1. Сериализация / десериализация	302
Глава 52.2. Интерфейс Serializable.....	302
Раздел 53. Сериализация	304
Глава 53.1. Сериализация различных типов	304
Глава 53.2. Проблемы безопасности при использовании функции unserialize.....	306
Раздел 54. Замыкания (Closure)	309
Глава 54.1. Базовое использование замыкания.....	309
Глава 54.2. Использование внешних переменных	310
Глава 54.3. Базовая привязка замыкания (bindTo)	310
Глава 54.4. Привязка замыкания и область видимости	311
Глава 54.5. Привязка замыкания для одного вызова.....	312
Глава 54.6. Использование замыканий для реализации паттерна «наблюдатель».....	313

Раздел 55. Чтение данных запроса	315
Глава 55.1. Чтение необработанных данных запроса POST	315
Глава 55.2. Чтение данных из запроса POST	315
Глава 55.3. Чтение данных из запроса GET	316
Глава 55.4. Обработка ошибок при загрузке файлов	316
Глава 55.5. Передача массивов через POST	317
Глава 55.6. Загрузка файлов с помощью HTTP PUT	318
Раздел 56. Приведение типов и проблемы нестрогого сравнения	319
Глава 56.1. Что такое приведение типов?	319
Глава 56.2. Чтение из файла	319
Глава 56.3. Ножиданности при использовании оператора switch.....	320
Глава 56.4. Строгая типизация	321
Раздел 57. Сокеты	322
Глава 57.1. Клиентский сокет TCP	322
Глава 57.2. Серверный сокет TCP	323
Глава 57.3. Серверный сокет UDP	324
Глава 57.4. Обработка ошибок сокетов	324
Раздел 58. PDO	325
Глава 58.1. Предотвращение SQL-инъекций с помощью параметризованных запросов.....	325
Глава 58.2. Подключение и получение данных PDO	326
Глава 58.3. Транзакции баз данных с помощью PDO	327
Глава 58.4. PDO — подключение к серверу MySQL/MariaDB	330
Глава 58.5. PDO — получение количества строк, затронутых запросом.....	330
Глава 58.6. PDO::lastInsertId()	331
Раздел 59. PHP MySQLi	332
Глава 59.1. Закрытие соединения	332
Глава 59.2. Подключение MySQLi	332
Глава 59.3. Перебор результатов MySQLi	333
Глава 59.4. Подготовленные запросы в MySQLi	333
Глава 59.5. Экранирование строк	335
Глава 59.6. Отладка SQL-запросов в MySQLi	335
Глава 59.7. Запрос MySQLi (query).....	335
Глава 59.8. Как получить данные из подготовленного запроса.....	336
Глава 59.9. Идентификатор вставки MySQLi Insert ID.....	338
Раздел 60. SQLite3	340
Глава 60.1. Самоучитель по SQLite3.....	340
Глава 60.2. Выполнение запросов к базе данных.....	342
Глава 60.3. Получение только одного результата	342
Раздел 61. Использование MongoDB	343
Глава 61.1. Подключение к MongoDB	343
Глава 61.2. Получение нескольких документов — find()	343
Глава 61.3. Получение одного документа — findOne()	343
Глава 61.4. Вставка документа.....	343
Глава 61.5. Обновление документа.....	344
Глава 61.6. Удаление документа	344
Раздел 62. mongo-php	345
Глава 62.1. Взаимодействие между MongoDB и Php	345
Глава 63. Использование Redis в PHP	348
Глава 63.1. Подключение к экземпляру Redis.....	348
Глава 63.2. Установка PHP Redis на Ubuntu.....	348
Глава 63.3. Выполнение команд Redis в PHP.....	348
Раздел 64. Отправка электронной почты	349
Глава 64.1. Отправка электронной почты — основы, дополнительные сведения и полный пример.....	349
Глава 64.2. Отправка электронной почты в формате HTML с помощью mail()	352
Глава 64.3. Отправка письма с вложением с помощью функции mail()	352
Глава 64.4. Отправка текстовых сообщений электронной почты с помощью PHPMailer	354
Глава 64.5. Отправка электронной почты в формате HTML с помощью PHPMailer	355

Глава 64.6. Отправка электронной почты с вложением с помощью PHPMailer	355
Глава 64.7. Отправка обычной текстовой почты с помощью Sendgrid	356
Глава 64.8. Отправка электронной почты с вложением с помощью Sendgrid	357
Раздел 65. Использование SQLSRV	358
Глава 65.1. Получение сообщений об ошибках	358
Глава 65.2. Извлечение результатов запроса	358
Глава 65.3. Создание соединения	359
Глава 65.4. Выполнение простого запроса	359
Глава 65.5. Вызов хранимой процедуры	360
Глава 65.6. Создание параметризованного запроса	360
Раздел 66. Интерфейс командной строки (Command Line Interface, CLI)	361
Глава 66.1. Обработка параметров программы	361
Глава 66.2. Обработка аргументов	362
Глава 66.3. Обработка ввода и вывода	363
Глава 66.4. Коды состояния	364
Глава 66.5. Ограничение выполнения скрипта командной строкой	364
Глава 66.6. Различия в поведении PHP при запуске программ из командной строки	365
Глава 66.7. Запуск скрипта	365
Глава 66.8. Граничные случаи работы getopt ()	366
Глава 66.9. Запуск встроенного веб-сервера	367
Раздел 67. Локализация	368
Глава 67.1. Локализация строк с помощью функции gettext ()	368
Раздел 68. Управление HTTP-заголовками	369
Глава 68.1. Основные настройки заголовка	369
Раздел 69. Соглашения по написанию кода	371
Глава 69.1. Теги PHP	371
Раздел 70. Асинхронное программирование	372
Глава 70.1. Преимущества генераторов	372
Глава 70.2. Использование цикла событий Icicle	372
Глава 70.3. Порождение неблокирующих процессов с помощью proc_open ()	373
Глава 70.4. Чтение последовательного порта с использованием Event и DIO	375
Глава 70.5. HTTP-клиент на основе модуля Event	377
Глава 70.6. HTTP-клиент на основе модуля Ev	380
Глава 70.7. Использование цикла событий Amp	385
Раздел 71. Как определить IP-адрес клиента	386
Глава 71.1. Корректное использование HTTP_X_FORWARDED_FOR	386
Раздел 72. Создание PDF-файлов в PHP	388
Глава 72.1. Начало работы с PDFlib	388
Раздел 73. YAML в PHP	389
Глава 73.1. Установка модуля YAML	389
Глава 73.2. Использование YAML для хранения конфигурации приложения	389
Раздел 74. Обработка изображений с помощью модуля GD	391
Глава 74.1. Вывод изображений	391
Глава 74.2. Создание изображения	393
Глава 74.3. Обрезка и масштабирование изображений	393
Раздел 75. Imagick	396
Глава 75.1. Первые шаги	396
Глава 75.2. Преобразование изображения в строку с кодировкой base64	396
Раздел 76. Сервер SOAP	398
Глава 76.1. Базовый сервер SOAP	398
Раздел 77. Машинное обучение	399
Глава 77.1. Классификация с использованием PHP-ML	399
Глава 77.2. Регрессия	401
Глава 77.3. Кластеризация	402
Раздел 78. Кеширование	404
Глава 78.1. Кеширование с помощью memcache	404
Глава 78.2. Кеширование с использованием APC	405

Раздел 79. Автозагрузка	407
Глава 79.1. Автозагрузка как неотъемлемая часть фреймворка.....	407
Глава 79.2. Встроенное определение класса, загрузка не требуется.....	408
Глава 79.3. Загрузка классов вручную с помощью require.....	408
Глава 79.4. Автозагрузка как альтернатива ручной загрузке классов.....	409
Глава 79.5. Автозагрузка с использованием Composer.....	409
Раздел 80. Структуры данных SPL	411
Глава 80.1. SplFixedArray.....	411
Раздел 81. IMAP	415
Глава 81.1. Подключение к почтовому ящику.....	415
Глава 81.2. Установка модуля IMAP.....	416
Глава 81.3. Получение списка всех папок почтового ящика.....	417
Глава 81.4. Поиск сообщений в почтовом ящике.....	418
Раздел 82. Аутентификация HTTP	420
Глава 82.1. Простая аутентификация.....	420
Раздел 83. WebSockets	421
Глава 83.1. Простой TCP/IP-сервер.....	421
Раздел 84. Модуль математических операций с числами произвольной точности BCMath	423
Глава 84.1. Использование модуля bcmath для чтения/записи двоичного числа типа long в 32-битных системах.....	425
Глава 84.2. Сравнение арифметических операций Bcmath и float.....	426
Раздел 85. Развертывание приложения с использованием Docker	427
Глава 85.1. Получение Docker-образа для PHP.....	427
Глава 85.2. Написание Dockerfile.....	427
Глава 85.3. Сборка образа.....	427
Глава 85.4. Запуск контейнера приложения.....	428
Раздел 86. APCu	429
Глава 86.1. Итерация по записям.....	429
Глава 86.2. Простое сохранение и извлечение данных.....	429
Глава 86.3. Информация о хранилище.....	429
Раздел 87. Встроенный сервер PHP	430
Глава 87.1. Запуск встроенного сервера.....	430
Глава 87.2. Встроенный сервер с определенным каталогом и скриптом маршрутизатора.....	430
Раздел 88. PSR	431
Глава 88.1. PSR-4: автозагрузка.....	431
Глава 88.2. PSR-1: базовый стандарт кодирования.....	432
Раздел 89. PHPDoc	433
Глава 89.1. Описание переменной.....	433
Глава 89.2. Добавление метаданных к функциям.....	433
Глава 89.3. Описание параметров.....	434
Глава 89.4. Коллекции.....	435
Глава 89.5. Добавление метаданных к файлам.....	436
Глава 89.6. Наследование метаданных от родительских структур.....	437
Раздел 90. Паттерны проектирования	438
Глава 90.1. Цепочка вызовов методов (Method Chaining) в PHP.....	438
Раздел 91. Компиляция модуля PHP	440
Глава 91.1. Компиляция в ОС Linux.....	440
Раздел 92. Распространенные ошибки	442
Глава 92.1. Вызов функции fetch_assoc для логических (boolean) значений.....	442
Глава 92.2. Unexpected \$end.....	442
Раздел 93. Сборник ошибок и предупреждений	444
Глава 93.1. Parse error: syntax error, unexpected T_PAAMAYIM_NEKUDOTAYIM.....	444
Глава 93.2. Уведомление: неопределенный индекс (Notice: Undefined index).....	444
Глава 93.3. Warning: Cannot modify header information – headers already sent.....	444

Раздел 94. Обработка исключений и отчет об ошибках	446
Глава 94.1. Настройка и отображение сообщений об ошибках	446
Глава 94.2. Регистрация фатальных ошибок	446
Раздел 95. Отладка	448
Глава 95.1. Вывод содержимого переменных.....	448
Глава 95.2. Отображение ошибок.....	448
Глава 95.3. <code>phpinfo()</code>	449
Глава 95.4. <code>Xdebug</code>	450
Глава 95.5. Настройка отображения информации об ошибках	450
Глава 95.6. <code>phpversion()</code>	450
Раздел 96. Модульное тестирование (Unit Testing)	452
Глава 96.1. Тестирование поведения класса	452
Глава 96.2. Поставщики данных в PHPUnit	455
Глава 96.3. Тестирование исключений	459
Раздел 97. Производительность	461
Глава 97.1. Профилирование с помощью <code>Xdebug</code>	461
Глава 97.2. Использование памяти.....	462
Глава 97.3. Профилирование с помощью <code>XHProf</code>	463
Раздел 98. Мультипроцессность	465
Глава 98.1. Мультипроцессность с использованием встроенных функций <code>fork</code>	465
Глава 98.2. Создание дочернего процесса с помощью <code>fork</code>	465
Глава 98.3. Межпроцессное взаимодействие	466
Раздел 99. Модуль для организации работы многопоточности	467
Глава 99.1. Начало работы.....	467
Глава 99.2. Использование пулов и работников (Pools and Workers).....	467
Раздел 100. Безопасная реализация опции «Remember Me»	470
Глава 100.1. «Остаться в системе» — лучший подход.....	470
Раздел 101. Безопасность	471
Глава 101.1. Утечка версии PHP.....	471
Глава 101.2. Тип атаки межсайтовый скриптинг (также существует функция, XSS)	471
Глава 101.3. Подделка межсайтовых запросов (Cross-Site Request Forgery, CSRF).....	473
Глава 101.4. Внедрение в командную строку (Command Line Injection)	474
Глава 101.5. Удаление HTML-тегов	475
Глава 101.6. Подключение файлов.....	476
Глава 101.7. Отчеты об ошибках	477
Глава 101.8. Загрузка файлов	478
Раздел 102. Криптография	481
Глава 102.1. Симметричное шифрование и дешифрование больших файлов с использованием <code>OpenSSL</code>	481
Глава 102.2. Симметричный шифр.....	483
Раздел 103. Функции хеширования паролей	485
Глава 103.1. Создание хеша пароля	485
Глава 103.2. Определение возможности обновления существующего хеша пароля до более стойкого алгоритма.....	487
Глава 103.3. Проверка пароля по хешу.....	487
Раздел 104. Внесение вклада в руководство по PHP	489
Глава 104.1. Улучшение официальной документации	489
Глава 104.2. Советы по внесению вклада в руководство.....	489
Раздел 105. Участие в разработке ядра PHP	491
Глава 105.1. Настройка базовой среды разработки.....	491
Приложение А. Установка окружения PHP в Windows	492
Глава А.1. Загрузка, установка и использование WAMP	492
Глава А.2. Установка PHP и использование его с IIS.....	492
Глава А.3. Загрузка и установка XAMPP	493
Приложение В. Установка в средах Linux/Unix	495
Глава В.1. Установка из командной строки с использованием APT для PHP 7.....	495
Глава В.2. Установка в дистрибутивах Enterprise Linux (CentOS, Scientific Linux и т. д.).....	495
Благодарности.....	497

Раздел 1. Начало работы с PHP

PHP 8.x

Версия	Дата выпуска	Поддержка до
8.4	21.11.2024	31.12.2028
8.3	23.11.2023	31.12.2027
8.2	08.12.2022	31.12.2026
8.1	25.11.2021	31.12.2025
8.0	26.11.2020	26.11.2023

PHP 7.x

Версия	Дата выпуска	Поддержка до
7.4	28.11.2019	28.11.2022
7.3	06.12.2018	06.12.2021
7.2	30.11.2017	30.11.2020
7.1	01.12.2016	01.12.2019
7.0	03.12.2015	03.12.2018

PHP 5.x

Версия	Дата выпуска	Поддержка до
5.6	28.08.2014	31.12.2018
5.5	20.06.2013	21.07.2016
5.4	01.03.2012	03.09.2015
5.3	30.06.2009	14.08.2014
5.2	02.11.2006	06.01.2011
5.1	24.11.2005	24.08.2006
5.0	13.07.2004	05.09.2005

PHP 4.x

Версия	Дата выпуска	Поддержка до
4.4	11.07.2005	07.08.2008
4.3	27.12.2002	31.03.2005
4.2	22.04.2002	06.09.2002
4.1	10.12.2001	12.03.2002
4.0	22.05.2000	23.06.2001

Унаследованные версии

Версия	Дата выпуска	Поддержка до
3.0	06.06.1998	20.10.2000
2.0	01.11.1997	
1.0	08.06.1995	

Глава 1.1. HTML-вывод с веб-сервера

PHP можно использовать для добавления содержимого в HTML-файлы. В то время как HTML обрабатывается непосредственно веб-браузером, PHP-скрипты выполняются веб-сервером, а полученный HTML-код отправляется в браузер.

Следующая HTML-разметка содержит оператор PHP, который добавит в вывод строку Hello World!:

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

Если сохранить этот текстовый файл как PHP-скрипт и выполнить его на веб-сервере, браузеру пользователя будет отправлен следующий HTML-код:

```
<!DOCTYPE html>
<html>
  <head>
    <title>PHP!</title>
  </head>
  <body>
    <p>Hello world!</p>
  </body>
</html>
```

PHP 5.x Версия ≥ 5.4

У команды `echo` существует также сокращенный синтаксис, который позволяет сразу вывести значение. До версии PHP 5.4.0 этот сокращенный синтаксис работал только при включенном параметре `short_open_tag` в файле конфигурации.

Например, рассмотрим код:

```
<p> <?= "Hello world!" ?></p>
```

Его вывод будет идентичен выполнению следующего фрагмента программы:

```
<p> <?php echo "Hello world!"; ?></p>
```

В реальных приложениях все данные, выводимые PHP на HTML-страницу, должны быть корректно *экранированы*, чтобы предотвратить XSS-атаки (Cross-Site Scripting — «межсайтовый скриптинг») или повреждение текста.

См. также: главы «Строки» и «PSR-1», где описаны лучшие практики, включая корректное использование коротких тегов (`<?= . . . ?>`).

Глава 1.2. Hello, World!

Наиболее распространенная языковая конструкция для вывода данных в PHP — это языковая конструкция `echo`:

```
echo "Hello, World!\n";
```

В качестве альтернативы можно использовать команду `print`:

```
print "Hello, World!\n";
```

Обе конструкции выполняют одну и ту же функцию, однако имеют некоторые незначительные различия:

- `echo` возвращает значения типа `void`, а команда `print` возвращает целое число типа `int` со значением 1;
- `echo` может принимать несколько аргументов (только без скобок), в то время как `print` принимает только один аргумент;
- `echo` работает немного быстрее, чем `print`.

И `echo`, и `print` — не функции, а языковые конструкции. Это означает, что при их использовании нет необходимости заключать аргументы в круглые скобки. Тем не менее ради единообразия с функциями допускается писать их и в форме вызова с круглыми скобками.

Подробные примеры применения команд `echo` и `print` вы найдете в соответствующих разделах этой книги.

Кроме того, PHP поддерживает форматированный вывод с помощью функции `printf`, а также связанных с ней функций, работающих в стиле языка C. Например:

```
printf("%s\n", "Hello, World!");
```

Полное описание вывода переменных в PHP смотрите в главе «Вывод значения переменной».

Глава 1.3. Вывод с веб-сервера, отличный от HTML

В некоторых случаях при работе с веб-сервером может потребоваться переопределить тип содержимого (Content-Type), используемый веб-сервером по умолчанию. Например,