

O'REILLY®

6-е издание
включает React,
PHP 8 & MySQL 8

Создаем динамические
веб-сайты с помощью

PHP, MySQL, JavaScript, CSS и HTML5



Робин Никсон

SIXTH EDITION

Learning PHP, MySQL & JavaScript

*A Step-by-Step Guide
to Creating Dynamic Websites*

Robin Nixon

Beijing • Boston • Farnham • Sebastopol • Tokyo

O'REILLY®

Создаем динамические
веб-сайты с помощью

PHP, MySQL, JavaScript, CSS и HTML5

Робин Никсон



Санкт-Петербург · Москва · Минск

2023

ББК 32.988.02-018
УДК 004.738.5
Н64

Никсон Робин

Н64 Создаем динамические веб-сайты с помощью PHP, MySQL, JavaScript, CSS и HTML5. 6-е изд. — СПб.: Питер, 2023. — 832 с.: ил. — (Серия «Бестселлеры O'Reilly»).

ISBN 978-5-4461-1970-7

Новое издание бестселлера описывает как клиентские, так и серверные аспекты веб-разработки. Книга, наполненная ценными практическими советами и подробным теоретическим материалом, поможет вам освоить динамическое веб-программирование с применением самых современных технологий. Для закрепления усвоенных знаний автор расскажет, как создать полнофункциональный сайт, работающий по принципу социальной сети.

16+ (В соответствии с Федеральным законом от 29 декабря 2010 г. № 436-ФЗ.)

ББК 32.988.02-018
УДК 004.738.5

Права на издание получены по соглашению с O'Reilly. Все права защищены. Никакая часть данной книги не может быть воспроизведена в какой бы то ни было форме без письменного разрешения владельцев авторских прав.

Информация, содержащаяся в данной книге, получена из источников, рассматриваемых издательством как надежные. Тем не менее, имея в виду возможные человеческие или технические ошибки, издательство не может гарантировать абсолютную точность и полноту приводимых сведений и не несет ответственности за возможные ошибки, связанные с использованием книги. Издательство не несет ответственности за доступность материалов, ссылки на которые вы можете найти в этой книге. На момент подготовки книги к изданию все ссылки на интернет-ресурсы были действующими.

ISBN 978-1492093824 англ.

Authorized Russian translation of the English edition of Learning PHP, MySQL & JavaScript, 6th Edition ISBN 9781492093824 © 2021 Robin Nixon
This translation is published and sold by permission of O'Reilly Media, Inc., which owns or controls all rights to publish and sell the same
© Перевод на русский язык ООО «Прогресс книга», 2022
© Издание на русском языке, оформление ООО «Прогресс книга», 2022
© Серия «Бестселлеры O'Reilly», 2022

ISBN 978-5-4461-1970-7

Краткое содержание

Предисловие	28
Глава 1. Введение в динамическое содержимое веб-страницы.....	32
Глава 2. Установка сервера, предназначенного для разработки.....	49
Глава 3. Введение в PHP.....	67
Глава 4. Выражения и управление процессом выполнения программы в PHP.....	98
Глава 5. Функции и объекты PHP.....	131
Глава 6. Массивы в PHP.....	159
Глава 7. Практикум по программированию на PHP	175
Глава 8. Введение в MySQL.....	202
Глава 9. Освоение MySQL.....	248
Глава 10. Новое в PHP 8 и MySQL 8.....	276
Глава 11. Доступ к MySQL с использованием PHP	289
Глава 12. Обработка форм	321
Глава 13. Cookie-файлы, сессии и аутентификация.....	344
Глава 14. Изучение JavaScript.....	369
Глава 15. Выражения и управление процессом выполнения сценариев в JavaScript	394

Глава 16. Функции, объекты и массивы JavaScript.....	415
Глава 17. Проверка данных и обработка ошибок в JavaScript и PHP	437
Глава 18. Использование технологии асинхронного обмена данными.....	463
Глава 19. Введение в CSS	477
Глава 20. Продвинутое стилизация страниц с помощью CSS3.....	520
Глава 21. Доступ к CSS из JavaScript.....	548
Глава 22. Введение в jQuery	568
Глава 23. Введение в jQuery Mobile	630
Глава 24. Введение в React.....	651
Глава 25. Введение в HTML5.....	678
Глава 26. Холсты в HTML5.....	685
Глава 27. Аудио и видео в HTML5.....	738
Глава 28. Другие свойства HTML5.....	747
Глава 29. Объединение технологий	766
Приложение. Ответы на вопросы.....	804
Об авторе	829
Об обложке	830

Оглавление

Предисловие	28
Для кого предназначена эта книга.....	28
Предположения, допущенные в книге.....	28
Структура издания.....	29
Условные обозначения.....	30
Использование примеров кода.....	31
Благодарности.....	31
От издательства.....	31
Глава 1. Введение в динамическое содержимое веб-страницы.....	32
HTTP и HTML: основы, заложенные Бернерсом-Ли	33
Процедура «запрос — ответ»	33
Преимущества использования PHP, MySQL, JavaScript и CSS.....	36
MariaDB: клон MySQL	38
PHP.....	38
MySQL.....	39
JavaScript	41
CSS	42
А теперь HTML5.....	43
Веб-сервер Apache	44
Обслуживание мобильных устройств	45
Несколько слов о программах с открытым исходным кодом	45
Сочетаем все технологии.....	46
Вопросы	48
Глава 2. Установка сервера, предназначенного для разработки.....	49
Что такое WAMP, MAMP и LAMP.....	50
Установка AMPPS в систему Windows.....	50
Тестирование установки	55
Обращение к исходному источнику документов (Windows).....	57
Другие системы WAMP	58

Установка AMPPS в системе macOS	58
Обращение к исходному источнику документов (macOS)	59
Установка LAMP в Linux	60
Работа в удаленном режиме	61
Вход в систему	61
Использование SFTP или FTPS	62
Использование редактора кода	62
Использование интегрированных сред разработки	64
Вопросы	66
Глава 3. Введение в PHP	67
Включение PHP в HTML	67
Примеры в этой книге	69
Структура PHP	69
Комментарии	69
Основной синтаксис	70
Переменные	71
Операторы	76
Присваивание значений переменным	80
Многострочные команды	83
Типы переменных	85
Константы	86
Предопределенные константы	87
Различие между командами echo и print	88
Функции	89
Область видимости переменной	90
Вопросы	96
Глава 4. Выражения и управление процессом выполнения программы в PHP	98
Выражения	98
Истина или ложь?	99
Литералы и переменные	100
Операторы	101
Приоритетность операторов	102
Взаимосвязанность операторов	104
Операторы отношения	106

Условия.....	110
Инструкция if	111
Инструкция else	113
Инструкция elseif	114
Инструкция switch.....	116
Оператор ? (тернарный)	118
Циклы.....	120
Цикл while	120
Циклы do...while.....	122
Циклы for	123
Прекращение работы цикла	125
Инструкция continue.....	126
Неявное и явное преобразование типов.....	127
Динамическое связывание в PHP	128
Динамическое связывание в действии.....	129
Вопросы	130
Глава 5. Функции и объекты PHP	131
Функции PHP.....	132
Определение функции.....	133
Возвращение значения.....	134
Возвращение массива	136
Передача аргументов по ссылке	137
Возвращение глобальных переменных	138
И еще раз об области видимости переменных	139
Включение и запрос файлов	139
Инструкция include	140
Инструкция include_once.....	140
Инструкции require и require_once.....	141
Совместимость версий PHP.....	141
Объекты PHP	142
Терминология.....	143
Объявление класса	144
Создание объекта.....	145
Доступ к объектам.....	145
Клонирование объектов	147
Конструкторы.....	148

Деструкторы	149
Написание методов	149
Объявление свойств	150
Объявление констант	151
Область видимости свойств и методов.....	151
Статические методы	153
Статические свойства	153
Наследование.....	154
Вопросы	158
Глава 6. Массивы в PHP	159
Основные подходы к массивам	159
Массивы с числовой индексацией	159
Ассоциативные массивы	161
Присваивание с использованием ключевого слова array.....	162
Цикл foreach...as.....	163
Многомерные массивы.....	165
Использование функций для работы с массивами.....	169
Функция is_array.....	169
Функция count.....	169
Функция sort	170
Функция shuffle.....	170
Функция explode.....	170
Функция extract	171
Функция compact.....	172
Функция reset	173
Функция end	174
Вопросы	174
Глава 7. Практикум по программированию на PHP	175
Функция printf.....	175
Настройка представления данных.....	177
Дополнение строк.....	179
Функция sprintf	180
Функции даты и времени.....	180
Константы, связанные с датами	183
Функция checkdate.....	183

Работа с файлами.....	184
Проверка существования файла.....	184
Создание файла.....	184
Чтение файлов.....	186
Копирование файлов.....	187
Перемещение файла.....	188
Удаление файла.....	188
Обновление файлов.....	189
Блокирование файлов при коллективном доступе.....	190
Чтение всего файла целиком.....	192
Загрузка файлов на веб-сервер.....	192
Системные вызовы.....	199
XHTML или HTML5?.....	200
Вопросы.....	201
Глава 8. Введение в MySQL.....	202
Основные характеристики MySQL.....	202
Сводка понятий, используемых в базах данных.....	203
Доступ к MySQL из командной строки.....	203
Начало работы с интерфейсом командной строки.....	204
Использование интерфейса командной строки.....	208
Команды MySQL.....	209
Типы данных.....	215
Индексы.....	226
Создание индекса.....	226
Создание запросов к базе данных MySQL.....	231
Объединение таблиц.....	242
Использование логических операторов.....	245
Функции MySQL.....	245
Работа с MySQL через phpMyAdmin.....	245
Вопросы.....	247
Глава 9. Освоение MySQL.....	248
Проектирование базы данных.....	248
Первичные ключи: ключи к реляционным базам данных.....	249
Нормализация.....	250
Первая нормальная форма.....	252
Вторая нормальная форма.....	254

Третья нормальная форма	257
Когда не следует проводить нормализацию	259
Отношения	260
«Один к одному»	260
«Один ко многим»	261
«Многие ко многим»	262
Базы данных и анонимность	264
Транзакции	264
Ядра (механизмы хранения) транзакций	265
Команда BEGIN	266
Команда COMMIT	266
Команда ROLLBACK	267
Команда EXPLAIN	268
Резервное копирование и восстановление данных	269
Команда mysqldump	269
Создание файла резервной копии	270
Восстановление данных из файла резервной копии	273
Выгрузка данных в файлы формата CSV	273
Планирование резервного копирования	274
Вопросы	275
Глава 10. Новое в PHP 8 и MySQL 8	276
Об этой главе	276
PHP	277
Именованные параметры	277
Атрибуты	278
Свойства конструктора	278
Компиляция Just In Time	279
Типы соединений	279
Оператор нулевого значения Null-safe	279
Выражение match	280
Новые функциональные возможности	280
MySQL 8	284
Обновления в SQL	285
JSON (JavaScript Object Notation)	285
Географическое сопровождение	285
Надежность	286
Скорость и производительность	286

Администрирование.....	286
Безопасность.....	287
Вопросы.....	287
Глава 11. Доступ к MySQL с использованием PHP	289
Запросы к базе данных MySQL с помощью PHP.....	289
Процесс.....	289
Создание файла регистрации.....	290
Подключение к базе данных MySQL.....	291
Практический пример.....	296
Массив \$_POST.....	299
Удаление записи.....	300
Отображение формы.....	301
Запросы к базе данных.....	302
Запуск программы.....	302
Практическая работа с MySQL.....	304
Создание таблицы.....	304
Описание таблицы.....	305
Удаление таблицы.....	306
Добавление данных.....	306
Извлечение данных.....	307
Обновление данных.....	308
Удаление данных.....	309
Свойство AUTO_INCREMENT.....	309
Выполнение дополнительных запросов.....	311
Предотвращение попыток взлома.....	312
Возможные меры противодействия.....	314
Указатели мест заполнения.....	315
Предотвращение внедрения JavaScript в HTML.....	318
Вопросы.....	320
Глава 12. Обработка форм	321
Создание форм.....	321
Извлечение отправленных данных.....	323
Значения по умолчанию.....	324
Типы элементов ввода данных.....	325
Обезвреживание введенных данных.....	334
Пример программы.....	336

Усовершенствования, появившиеся в HTML5.....	339
Атрибут autocomplete	339
Атрибут autofocus.....	339
Атрибут placeholder.....	340
Атрибут required.....	340
Атрибуты подмены	340
Атрибуты width и height.....	341
Атрибуты min и max	341
Атрибут step	341
Атрибут form	341
Атрибут list.....	342
Тип ввода color	342
Типы ввода number и range	342
Окно выбора даты и времени.....	342
Вопросы	342
Глава 13. Cookie-файлы, сессии и аутентификация	344
Использование cookie в PHP	344
Установка cookie	346
Доступ к cookie	347
Удаление cookie.....	347
HTTP-аутентификация	348
Сохранение имен пользователей и паролей	352
Пример программы.....	354
Использование сессий.....	358
Начало сессии	359
Завершение сессии.....	362
Установка времени ожидания	363
Безопасность сессии.....	363
Вопросы	368
Глава 14. Изучение JavaScript	369
JavaScript и текст HTML	370
Использование сценариев в заголовке документа	372
Устаревшие и нестандартные браузеры.....	372
Включение файлов JavaScript	373
Отладка кода JavaScript	374
Использование комментариев	374

Точка с запятой	375
Переменные.....	375
Строковые переменные	376
Числовые переменные	376
Массивы.....	376
Операторы.....	377
Арифметические операторы.....	378
Операторы присваивания.....	378
Операторы сравнения	379
Логические операторы	379
Инкремент, декремент переменной и краткая форма присваивания.....	380
Объединение строк.....	380
Управляющие символы.....	380
Типизация переменных	381
Функции	383
Глобальные переменные	383
Локальные переменные.....	383
Использование let и const	384
Объектная модель документа	387
Еще одно использование знака \$	389
Использование DOM.....	390
О функции document.write	391
Использование console.log	391
Использование alert.....	391
Запись в элементы	391
Использование document.write	392
Вопросы	392

Глава 15. Выражения и управление процессом выполнения

сценариев в JavaScript	394
Выражения	394
Литералы и переменные.....	395
Операторы.....	396
Приоритетность операторов	397
Взаимосвязанность	398
Операторы отношения	399
Инструкция with.....	402
Использование события onerror	403

Конструкция try...catch	404
Условия.....	405
Инструкция if	405
Инструкция else	406
Инструкция switch.....	407
Оператор ?	408
Циклы.....	409
Циклы while	409
Циклы do...while.....	410
Циклы for	410
Прекращение работы цикла	411
Инструкция continue.....	412
Явное преобразование типов	413
Вопросы	413
Глава 16. Функции, объекты и массивы JavaScript	415
Функции JavaScript.....	415
Определение функции.....	415
Возвращение значения.....	417
Возвращение массива	420
Объекты JavaScript.....	420
Объявление класса.....	421
Создание объекта.....	422
Доступ к объектам.....	423
Ключевое слово prototype	423
Массивы в JavaScript	427
Числовые массивы.....	427
Ассоциативные массивы	428
Многомерные массивы.....	429
Методы массивов	430
Вопросы	436
Глава 17. Проверка данных и обработка ошибок в JavaScript и PHP	437
Проверка данных, введенных пользователем, средствами JavaScript	437
Документ validate.html (часть первая)	438
Документ validate.html (часть вторая).....	440
Регулярные выражения.....	444
Соответствие, закладываемое в метасимволы	444
Нестрогое символьное соответствие	445

Группировка с помощью скобок	447
Символьный класс	447
Указание диапазона	448
Инвертирование	448
Более сложные примеры	449
Сводная таблица метасимволов	451
Общие модификаторы	454
Использование регулярных выражений в JavaScript	454
Использование регулярных выражений в PHP	455
Повторное отображение формы после проверки данных PHP-программой	456
Вопросы	462
Глава 18. Использование технологии асинхронного обмена данными.....	463
Что такое асинхронный обмен данными	463
Использование XMLHttpRequest	464
Ваша первая программа, использующая асинхронный обмен данными	465
Использование GET вместо POST	469
Отправка XML-запросов	471
Использование специальных платформ для асинхронного обмена данными	475
Вопросы	475
Глава 19. Введение в CSS	477
Импортирование таблицы стилей	478
Импортирование CSS из HTML-кода	479
Встроенные настройки стиля	479
Идентификаторы (ID)	479
Классы	480
Точки с запятой	480
Правила CSS	481
Множественные задания стиля	481
Использование комментариев	482
Типы стилей	483
Исходные стили	483
Пользовательские стили	483
Внешние таблицы стилей	484
Внутренние стили	484
Внедренные стили	485

Селекторы CSS	485
Селектор типа.....	485
Селектор потомков.....	485
Селектор дочерних элементов.....	486
Селектор элементов, имеющих идентификатор	488
Селектор класса	488
Селектор атрибутов.....	489
Универсальный селектор	490
Групповая селекция.....	490
Каскадность CSS.....	491
Создатели таблиц стилей.....	491
Методы создания таблиц стилей.....	492
Селекторы таблиц стилей.....	492
Разница между элементами div и span	495
Измерения	497
Шрифты и оформление	498
Свойство font-family	498
Свойство font-style	500
Свойство font-size.....	500
Свойство font-weight	501
Управление стилями текста.....	501
Оформление	501
Разрядка.....	502
Выравнивание.....	502
Преобразование	503
Отступы	503
Цвета в CSS	504
Сокращенные цветовые строки	504
Градиенты.....	505
Позиционирование элементов.....	506
Абсолютное позиционирование.....	506
Относительное позиционирование	507
Фиксированное позиционирование.....	507
Псевдоклассы.....	510
Сокращенная запись правил	512
Модель блока и макет страницы.....	513
Установка полей.....	513
Применение границ	515

Настройка отступов.....	517
Содержимое объекта.....	518
Вопросы	518
Глава 20. Продвинутая стилизация страниц с помощью CSS3.....	520
Селекторы атрибутов	521
Соответствие частям строк	522
Свойство box-sizing	523
Создание фона в CSS3	523
Свойство background-clip.....	523
Свойство background-origin	524
Свойство background-size.....	526
Использование значения auto	526
Использование нескольких фонов	526
Границы в CSS3	528
Свойство border-color.....	529
Свойство border-radius.....	529
Прямоугольные тени	532
Выход элемента за пределы размеров.....	532
Разметка с использованием нескольких колонок	533
Цвета и непрозрачность	534
Цвета HSL.....	534
Цвета HSLA.....	535
Цвета RGB	536
Цвета RGBA	536
Свойство opacity	536
Эффекты, применяемые к тексту	537
Свойство text-shadow	537
Свойство text-overflow.....	537
Свойство word-wrap	538
Веб-шрифты	539
Веб-шрифты Google.....	540
Трансформации.....	540
Трехмерная трансформация	542
Переходы	543
Свойства, применяемые к переходам	543
Продолжительность перехода	544
Задержка перехода.....	544

Задание скорости перехода	544
Сокращенный синтаксис	545
Вопросы	546
Глава 21. Доступ к CSS из JavaScript.....	548
Еще одно обращение к функции getElementById	548
Функция O	548
Функция S	549
Функция C	550
Включение функций	551
Обращение к свойствам CSS из JavaScript	551
Некоторые общие свойства	552
Другие свойства	553
Встроенный JavaScript	555
Ключевое слово this	556
Привязка событий к объектам в сценарии	556
Прикрепление к другим событиям	557
Добавление новых элементов	558
Удаление элементов	560
Альтернативы добавлению и удалению элементов	560
Использование прерываний	561
Использование функции setTimeout	561
Отмена тайм-аута	563
Функция setInterval	563
Использование прерываний для анимации	565
Вопросы	567
Глава 22. Введение в jQuery	568
Почему же именно jQuery?	569
Включение jQuery	569
Выбор подходящей версии	570
Загрузка	571
Использование сети доставки контента	572
Заказная сборка jQuery	573
Синтаксис jQuery	573
Простой пример	573
Как избежать конфликтов библиотек	574

Селекторы.....	575
Метод css	575
Селектор элемента.....	576
Селектор идентификатора	576
Селектор класса	577
Сочетание селекторов	577
Обработка событий	578
Ожидание готовности документа	580
Функции и свойства событий	581
События blur и focus	581
Ключевое слово this	582
События click и dblclick.....	583
Событие keypress	584
Деликатное программирование	586
Событие mousemove.....	586
Другие события, связанные с мышью	589
Альтернативные методы работы с мышью.....	590
Событие submit.....	591
Специальные эффекты.....	593
Исчезновение и появление	594
Метод toggle	595
Проявление и растворение	596
Скольжение элементов вверх и вниз.....	597
Анимация	598
Остановка анимации.....	601
Работа с DOM.....	602
Разница между методами text и html	603
Методы val и attr.....	603
Добавление и удаление элементов.....	604
Динамическое применение классов.....	607
Работа с размерами.....	608
Методы width и height.....	608
Методы innerWidth и innerHeight.....	611
Методы outerWidth и outerHeight.....	611
Обход объектов DOM	611
Родительские элементы	612

Дочерние элементы	616
Одноуровневые элементы	617
Выбор следующих и предыдущих элементов	619
Обход элементов, выбранных с помощью методов jQuery.....	620
Метод is	622
Использование jQuery без селекторов	624
Метод \$.each	624
Метод \$.map.....	625
Использование асинхронного обмена данными	625
Использование метода POST.....	626
Использование метода GET	626
Дополнительные модули (плагины)	627
Пользовательский интерфейс jQuery	627
Другие дополнительные модули.....	628
Вопросы	628
Глава 23. Введение в jQuery Mobile	630
Включение jQuery Mobile.....	631
Начало работы.....	632
Связывание страниц.....	634
Синхронная связанность	635
Связанность внутри многостраничного документа.....	635
Смена страниц	636
Стильные кнопки.....	640
Обработка списков.....	643
Фильтруемые списки	645
Разделители списков	646
А что же дальше?	649
Вопросы	650
Глава 24. Введение в React	651
В чем же суть React?	652
Получение доступа к файлам React	653
Включение babel.js.....	654
Наш первый проект в React.....	655
Использование функции вместо класса	656
Чистый и нечистый код: золотое правило	657
Совместное использование класса и функции.....	658

Свойства и компоненты.....	659
Различия между использованием класса и функции	660
Состояние и жизненный цикл React.....	660
Использование хуков (если вы пользуетесь Node.js)	663
События в React.....	664
Встроенные условные утверждения JSX.....	666
Использование списков и ключей	667
Уникальные ключи.....	667
Обработка форм	669
Использование текстовых полей	670
Использование текстовых областей.....	672
Использование атрибута select.....	673
React Native.....	675
Создание приложений React Native.....	675
Дополнительная литература.....	676
Вывод React на новый уровень.....	676
Вопросы	676
Глава 25. Введение в HTML5.....	678
Холст.....	678
Геолокация	680
Аудио и видео	682
Формы	683
Локальное хранилище	683
Рабочие веб-процессы.....	683
Вопросы	684
Глава 26. Холсты в HTML5	685
Создание холста и доступ к нему.....	685
Функция toDataURL.....	687
Указание типа изображения.....	689
Метод fillRect.....	689
Метод clearRect	690
Метод strokeRect	690
Сочетание всех этих команд.....	690
Метод createLinearGradient	691
Метод addColorStop в подробностях	694
Метод createRadialGradient	695
Использование узоров для заливки.....	696

Запись текста на холсте	698
Метод <code>strokeText</code>	699
Свойство <code>textBaseline</code>	699
Свойство <code>font</code>	699
Свойство <code>textAlign</code>	700
Метод <code>fillText</code>	700
Метод <code>measureText</code>	701
Рисование линий	701
Свойство <code>lineWidth</code>	701
Свойства <code>lineCap</code> и <code>lineJoin</code>	702
Свойство <code>miterLimit</code>	704
Использование путей	704
Методы <code>moveTo</code> и <code>lineTo</code>	705
Метод <code>stroke</code>	705
Метод <code>rect</code>	705
Заливка областей	706
Метод <code>clip</code>	707
Метод <code>isPointInPath</code>	711
Работа с кривыми линиями	711
Метод <code>arc</code>	711
Метод <code>arcTo</code>	714
Метод <code>quadraticCurveTo</code>	715
Метод <code>bezierCurveTo</code>	717
Обработка изображений	718
Метод <code>drawImage</code>	718
Изменение размеров изображения	718
Выбор области изображения	719
Копирование с холста	720
Добавление теней	720
Редактирование на уровне пикселей	722
Метод <code>getImageData</code>	722
Метод <code>putImageData</code>	726
Метод <code>createImageData</code>	726
Более сложные графические эффекты	726
Свойство <code>globalCompositeOperation</code>	727
Свойство <code>globalAlpha</code>	729

Преобразования	730
Метод scale.....	730
Методы save и restore.....	731
Метод rotate	732
Метод translate	733
Метод transform.....	734
Метод setTransform	736
Вопросы	736
Глава 27. Аудио и видео в HTML5	738
О кодеках.....	739
Элемент <audio>.....	740
Элемент <video>	743
Видеокодеки	743
Вопросы	746
Глава 28. Другие свойства HTML5	747
Геолокация и служба GPS	747
Другие методы определения местоположения.....	748
Геолокация и HTML5.....	749
Локальное хранилище	752
Использование локального хранилища	753
Объект localStorage	754
Рабочие веб-процессы.....	756
Перетаскивание	759
Обмен сообщениями между документами	761
Другие теги HTML5.....	765
Вопросы	765
Глава 29. Объединение технологий.....	766
Проектирование приложения социальной сети	767
Информация на сайте.....	767
Файл functions.php.....	767
Функции	768
Файл header.php.....	770
Файл setup.php	773
Файл index.php	775

Файл signup.php.....	776
Проверка возможности указания желаемого имени пользователя	777
Регистрация	777
Файл checkuser.php.....	780
Файл login.php	781
Файл profile.php.....	783
Добавление текста в поле About Me (Обо мне)	784
Добавление изображения профиля	784
Обработка изображения.....	785
Отображение текущего профиля.....	785
Файл members.php	788
Просмотр профилей пользователей.....	789
Добавление и удаление друзей	789
Вывод списка всех участников	789
Файл friends.php.....	792
Файл messages.php	795
Файл logout.php	799
Файл styles.css	800
Файл javascript.js	802
Вопросы	803
Приложение. Ответы на вопросы	804
Об авторе	829
Об обложке	830

*Джулии, Наоми, Гарри, Мэтью,
Лоре, Ханне, Рэйчел и Дэвиду*

Предисловие

Сочетание PHP и MySQL — самый удобный подход к динамическому веб-конструированию, основанному на использовании баз данных. Этот подход удерживает свои позиции, несмотря на вызовы, брошенные другими интегрированными средами разработки, освоение которых дается значительно труднее. Благодаря открытости исходных кодов, его можно использовать совершенно бесплатно, поэтому он очень популярен у веб-разработчиков.

Любому нацеленному на результативность разработчику, использующему платформу UNIX/Linux или даже Windows, необходимо серьезно освоить эти технологии. В сочетании с партнерскими технологиями JavaScript, React, CSS и HTML5 можно создавать сайты калибра таких промышленных стандартов, как Facebook, Twitter и Gmail.

Для кого предназначена эта книга

Книга предназначена для тех, кто хочет изучить способы создания эффективных и динамичных сайтов. Сюда можно отнести веб-мастеров или специалистов по графическому дизайну, которые уже освоили создание статических веб-сайтов или CMS, таких как WordPress, и у которых есть желание вывести свое мастерство на следующий уровень, а также студентов вузов и колледжей, недавних выпускников этих учебных заведений и просто самоучек.

Фактически любой человек, стремящийся изучить основные принципы, заложенные в основу адаптивного веб-дизайна, сможет получить весьма обстоятельные сведения об основных технологиях: PHP, MySQL, JavaScript, CSS и HTML5, а также освоить основы библиотеки React.

Предположения, допущенные в книге

При написании данной книги автор предполагал, что читатель уже имеет элементарные понятия об HTML и способен как минимум скомпоновать простой статический сайт. Но при этом не обязательно наличие у читателя каких-либо знаний в области PHP, MySQL, JavaScript, CSS и HTML5, хотя, если такие знания имеются, изучение материала будет происходить значительно быстрее.

Структура издания

Главы книги расположены в определенном порядке. Сначала делается краткий обзор всех основных технологий, которые рассматриваются в книге, а затем описывается процесс их установки на сервер, предназначенный для разработки веб-приложений, чтобы подготовить читателя к практической работе с примерами.

В первой части книги преподносятся основы языка программирования PHP, включая основы синтаксиса, массивов, функций и объектно-ориентированного программирования.

Затем, после освоения основ PHP, можно переходить к введению в систему управления базами данных MySQL, рассмотрение которой начинается с изучения структуры базы данных MySQL и заканчивается составлением сложных запросов.

После этого рассказывается о том, как воспользоваться сочетанием PHP и MySQL, чтобы приступить к созданию собственных динамических веб-страниц путем интегрирования в это сочетание форм и других функциональных возможностей HTML. Затем будут рассмотрены подробности практических аспектов разработки на PHP и MySQL, включая описание различных полезных функций и способов работы с cookies и сессиями, а также способов поддержания высокого уровня безопасности.

В следующих нескольких главах даются основы JavaScript, начиная с простых функций и обработки событий и заканчивая доступом к объектной модели документа (DOM), проверкой введенных данных и обработкой ошибок в браузере. Это обстоятельный учебник для тех, кто приступает к использованию популярной библиотеки React для JavaScript.

После рассмотрения основных технологий описываются способы создания фоновых Ajax-вызовов и превращения сайтов в высокодинамичную среду.

После этого вам предстоит освоить еще две главы, в которых рассматривается, как использовать CSS для стилового оформления и подбора формата ваших веб-страниц, прежде чем будут раскрыты приемы существенного снижения трудозатрат по разработке приложений с помощью библиотеки React и дано описание интерактивных свойств, встроенных в HTML5, включающих геолокацию, аудио, видео и холст.

Получив все эти сведения, вы сможете создать полноценный набор программ, в совокупности представляющий собой работоспособный сайт социальной сети.

По мере изложения материала дается большое количество указаний и советов по выработке хорошего стиля программирования, а также подсказок, которые помогут читателям обнаружить и устранить скрытые ошибки программирования. Кроме того, делается много ссылок на сайты с дополнительными материалами, относящимися к рассматриваемым темам.

Условные обозначения

В книге применяются следующие условные обозначения.

Курсив

Курсивом выделены новые термины.

Моноширинный шрифт

Используется для листингов программ, а также внутри абзацев для обозначения таких элементов, как функции, базы данных, типы данных, переменные среды, операторы и ключевые слова, имена файлов и их расширений, команд и параметров командной строки, переменных и других кодовых элементов, тегов HTML и содержимого файлов, названий путей, утилит Unix, названий таблиц и столбцов.

Моноширинный жирный шрифт

Показывает команды или другой текст, который пользователь должен ввести самостоятельно.

Моноширинный курсив

Показывает текст, который должен быть заменен значениями, введенными пользователем, или значениями, определяемыми контекстом.

Шрифт без засечек

Используется для обозначения URL, адресов электронной почты, названий меню, параметров, кнопок, каталогов.



Так обозначаются замечания общего характера.



Так обозначаются предупреждения или предостережения.



Так обозначаются подсказки или советы.

Использование примеров кода

Дополнительные материалы (примеры кода, упражнения и т. д.) доступны для загрузки на сайте GitHub по адресу <https://github.com/RobinNixon/lpmj6>.

Эта книга предназначена для оказания помощи в выполнении стоящих перед вами задач. Вы можете использовать код, приведенный в ней, в своих программах и документации. Вам не нужно обращаться к нам за разрешением, до тех пор, пока вы не станете копировать значительную часть кода. Например, использование при написании программы нескольких фрагментов кода, взятых из данной книги, не требует специального разрешения. Но продажа и распространение набора примеров из книг издательства «Питер» — требует. Ответы на вопросы, в которых упоминаются материалы этой книги, и цитирование приведенных в ней примеров не требуют разрешения. Но включение существенного объема примеров кода, приводимых в данной книге, в документацию по вашему собственному продукту *требует* получения разрешения.

Ссылки на источник приветствуются, но не обязательны. В такие ссылки обычно включается название книги, имя ее автора, название издательства и номер ISBN. Например: «Создаем динамические веб-сайты с помощью PHP, MySQL, JavaScript, CSS и HTML5», шестое издание, автор Робин Никсон (Robin Nixon). 978-5-4461-1970-7.

Благодарности

Хочу выразить благодарность старшему редактору по сбору контента Аманде Куинн, редактору по разработке контента Мелиссе Поттер и всем, кто приложил немало усилий для выхода этой книги, в том числе Майклу Шпачеку и Дэвиду Макки за всеобъемлющую техническую рецензию, Кейтлин Геган — за общее руководство над выпуском книги, Ким Кофер — за редактуру, Ким Сандовал — за корректуру, Джудит Макконвилл — за создание приложения, Карен Монтгомери — за великолепную сахарную сумчатую летягу на обложке книги, Рэнди Камеру — за последний вариант обложки книги, моему главному редактору Энди Орам — за проверку первых пяти изданий, а также множеству других людей — их так много, что невозможно перечислить имена, — отправивших сообщения о замеченных ошибках и высказавших свои предложения относительно этого, нового издания.

От издательства

Ваши замечания, предложения, вопросы отправляйте по адресу comp@piter.com (издательство «Питер», компьютерная редакция).

Мы будем рады узнать ваше мнение!

На веб-сайте издательства www.piter.com вы найдете подробную информацию о наших книгах.

ГЛАВА 1

Введение в динамическое содержимое веб-страницы

Всемирная паутина — это непрерывно развивающаяся сеть, ушедшая далеко вперед от своей концепции ранних 1990-х, когда ее создание было обусловлено решением конкретных задач. Высокотехнологичные эксперименты в ЦЕРНе (Европейском центре физики высоких энергий, известном в наши дни в качестве обладателя Большого адронного коллайдера) выдавали невероятно большой объем данных, который был слишком велик для распространения среди участвующих в экспериментах ученых, разбросанных по всему миру.

К тому времени интернет уже существовал и к нему было подключено несколько сотен тысяч компьютеров, поэтому Тим Бернерс-Ли (специалист ЦЕРНа) придумал способ навигации между ними с использованием среды гиперссылок — так называемого протокола передачи гиперссылок (Hyper Text Transfer Protocol — HTTP). Он также создал специальный язык разметки, названный языком гипертекстовой разметки (Hyper Text Markup Language — HTML). Для того чтобы собрать все это воедино, он создал первый браузер и веб-сервер.

Теперь эти средства воспринимаются нами как данность, но в то время концепция их применения носила революционный характер. До этого основной объем соединений приходился на пользователей домашних модемов, дозванивавшихся и подключающихся к электронным доскам объявлений, которые базировались на отдельном компьютере и позволяли общаться и обмениваться данными только с другими пользователями данной службы. Следовательно, для эффективного электронного общения с коллегами и друзьями нужно было становиться участником многих электронных досок объявлений.

Но Бернерс-Ли изменил все это одним махом, и к середине 1990-х годов уже существовали три основных конкурирующих друг с другом графических браузера, пользовавшихся вниманием 5 млн посетителей. Однако вскоре стало очевидно, что кое-что было упущено. Конечно, текстовые и графические страницы, имеющие гиперссылки для перехода на другие страницы, были блестящей концепцией, но результаты не отражали текущего потенциала компьютеров и интернета

по удовлетворению насущных потребностей пользователей в динамическом изменении контекста. Всемирная паутина оставляла весьма невыразительное впечатление даже при наличии прокрутки текста и анимированных GIF-картинок. Корзины покупателей, поисковые машины и социальные сети внесли существенные коррективы в порядок использования Всемирной паутины. В этой главе будет дан краткий обзор различных компонентов, формирующих ее облик, и программного обеспечения, способствующего обогащению и оживлению наших впечатлений от ее использования.



Пришло время воспользоваться аббревиатурами. Прежде чем делать это, я старался дать им четкое объяснение. Но если сразу не удастся разобраться, какое именно понятие они замещают или что означают, переживать не стоит, поскольку все подробности прояснятся по мере чтения книги.

HTTP и HTML: основы, заложенные Бернерсом-Ли

HTTP представляет собой стандарт взаимодействия, регулирующий порядок направления запросов и получения ответов — процесса, происходящего между браузером, запущенным на компьютере конечного пользователя, и веб-сервером. Задача сервера состоит в том, чтобы принять запрос от клиента и попытаться дать на него содержательный ответ, обычно передавая ему запрошенную веб-страницу. Именно поэтому и используется термин «*сервер*» («обслуживающий»). Партнером, взаимодействующим с сервером, является *клиент*, поэтому данное понятие применяется как к браузеру, так и к компьютеру, на котором он работает.

Между клиентом и сервером может располагаться ряд других устройств, например маршрутизаторы, модули доступа, шлюзы и т. д. Они выполняют различные задачи по обеспечению безошибочного перемещения запросов и ответов между клиентом и сервером. Как правило, для отправки этой информации используется интернет. Некоторые из этих промежуточных устройств могут также ускорить интернет путем локального сохранения страниц или информации в так называемом кэше, обслуживая затем данное содержимое для клиентов непосредственно из кэша, без постоянного извлечения его из сервера-источника.

Обычно веб-сервер может обрабатывать сразу несколько подключений, а при отсутствии связи с клиентом он находится в режиме ожидания входящего подключения. При поступлении запроса на подключение сервер подтверждает его получение отправкой ответа.

Процедура «запрос — ответ»

В наиболее общем виде процесс «запрос — ответ» состоит из просьбы браузера или другой платформы к веб-серверу отправить ему веб-страницу и выполнения

браузером данной просьбы. После этого браузер занимается отображением или рендерингом страницы (рис. 1.1).

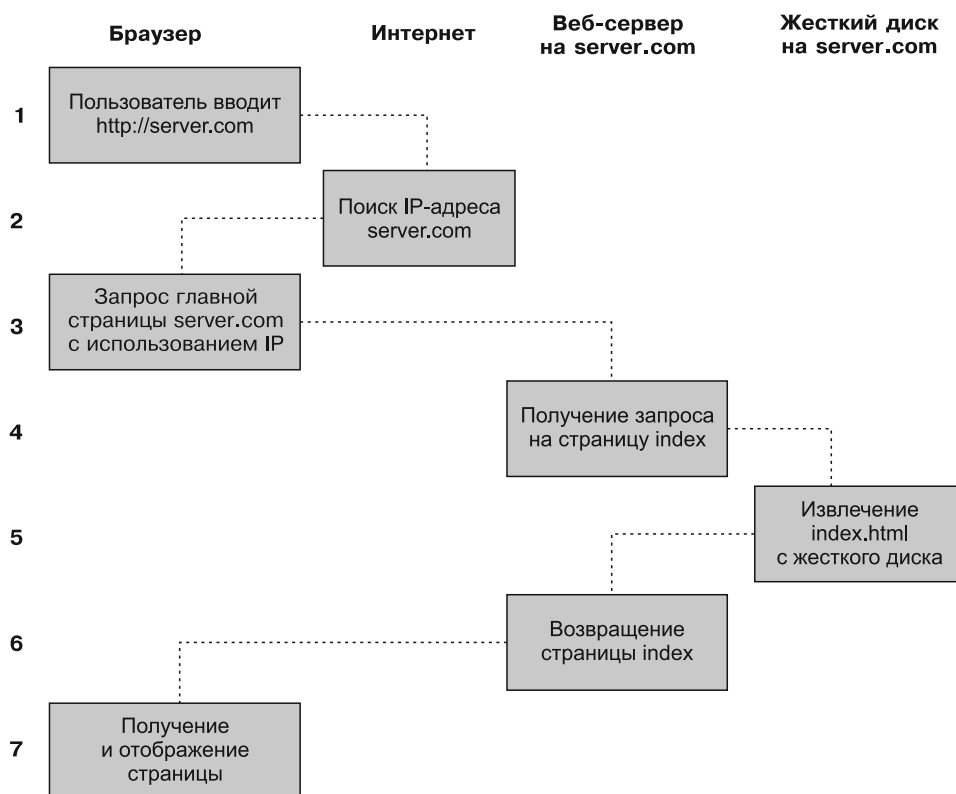


Рис. 1.1. Основная последовательность процесса «запрос — ответ» между клиентом и сервером

При этом соблюдается такая последовательность действий.

1. Вы вводите в адресную строку браузера `http://server.com`.
2. Ваш браузер ищет IP-адрес, соответствующий доменному имени `server.com`.
3. Браузер посылает запрос на главную страницу `server.com`.
4. Запрос проходит по интернету и поступает на веб-сервер `server.com`.
5. Веб-сервер, получивший запрос, ищет веб-страницу на своем жестком диске.
6. Сервер извлекает веб-страницу и отправляет ее по обратному маршруту в адрес браузера.
7. Браузер отображает веб-страницу.

При передаче типовой веб-страницы этот процесс также осуществляется для каждого имеющегося на ней объекта: элемента графики, встроенного видео- или Flash-ролика и даже шаблона CSS.

Обратите внимание на то, что на шаге 2 браузер ищет IP-адрес, принадлежащий доменному имени `server.com`. У каждой машины, подключенной к интернету, включая и ваш компьютер, есть свой IP-адрес. Но, как правило, доступ к веб-серверам осуществляется по именам, таким как `google.com`. Браузер обращается к вспомогательной интернет-службе, так называемой системе доменных имен (Domain Name System — DNS), чтобы найти связанный с сервером IP-адрес, а затем воспользоваться им для связи с компьютером.

При передаче динамических веб-страниц процедура состоит из большого количества действий, поскольку к ней могут привлекаться как PHP, так и MySQL. Например, можно щелкнуть кнопкой мыши на картинке с изображением плаща. После этого PHP составит запрос, используя стандартный язык базы данных, SQL — множество используемых для этого команд будет рассмотрено в данной книге, — и отправит запрос в адрес MySQL-сервера. Этот сервер возвратит информацию о выбранном вами плаще, и PHP-код заключит ее в некий код HTML, который будет отправлен сервером в адрес вашего браузера (рис. 1.2).

Выполняется такая последовательность действий.

1. Вы вводите в адресную строку браузера `http://server.com`.
2. Ваш браузер ищет IP-адрес, соответствующий доменному имени `server.com`.
3. Браузер посылает запрос на главную страницу `server.com`.
4. Запрос проходит по Сети и поступает на веб-сервер `server.com`.
5. Веб-сервер, получивший запрос, ищет веб-страницу на своем жестком диске.
6. Теперь, когда главная страница размещена в его памяти, веб-сервер замечает, что она представлена файлом, включающим в себя PHP-сценарии, и передает страницу интерпретатору PHP.
7. Интерпретатор PHP выполняет PHP-код.
8. Кое-какие фрагменты кода PHP содержат MySQL-инструкции, которые интерпретатор PHP, в свою очередь, передает процессору базы данных MySQL.
9. База данных MySQL возвращает результаты выполнения инструкции интерпретатору PHP.
10. Интерпретатор PHP возвращает веб-серверу результаты выполнения кода PHP, а также результаты, полученные от базы данных MySQL.
11. Веб-сервер возвращает страницу выдавшему запрос клиенту, который отображает эту страницу на экране.

Конечно, ознакомиться с этим процессом и узнать о совместной работе трех элементов не помешает, но на практике эти подробности не понадобятся, поскольку все происходит в автоматическом режиме.

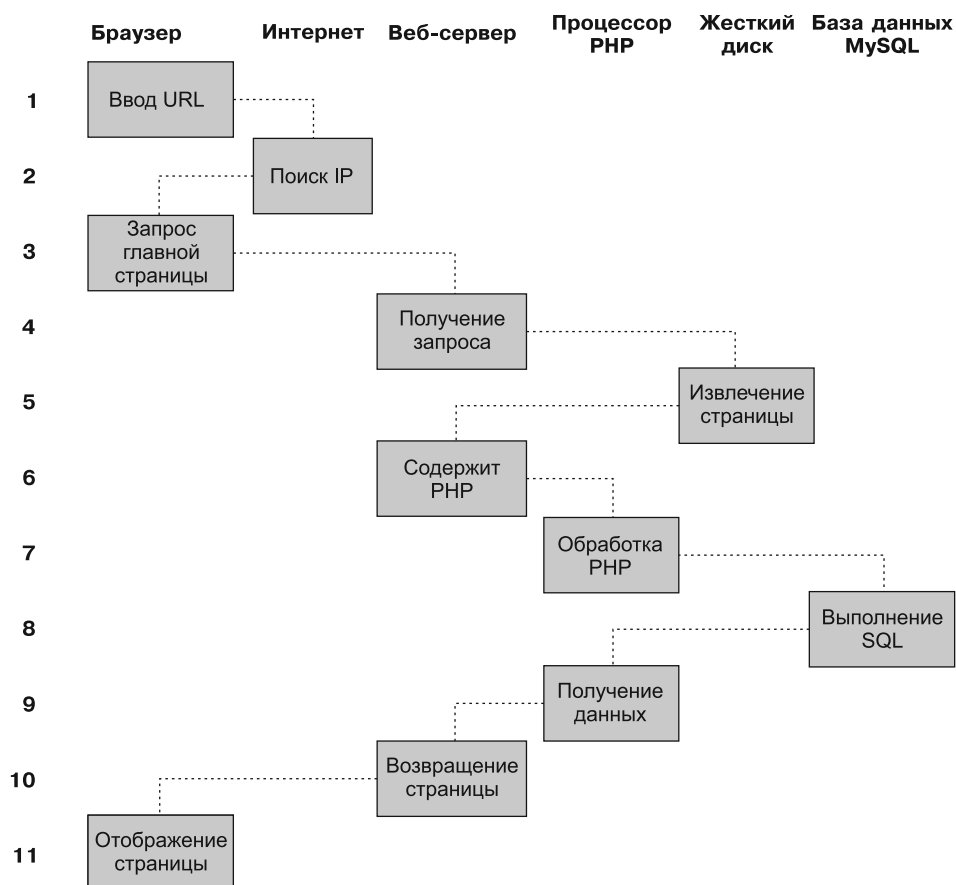


Рис. 1.2. Динамическая последовательность процесса «запрос — ответ», выполняемого клиентом и сервером

В каждом из примеров возвращенные браузеру HTML-страницы могут содержать также код JavaScript, интерпретируемый локально на машине клиента. Этот код может инициировать еще один запрос, точно так же запрос может быть инициирован встроенными объектами, например изображениями.

Преимущества использования PHP, MySQL, JavaScript и CSS

В начале этой главы был представлен мир технологии Web 1.0, но рывок к созданию технологии Web 1.1, вместе с которой были разработаны такие браузерные

расширения, как Java, JavaScript, JScript (несколько иной вариант JavaScript от корпорации Microsoft) и ActiveX, не заставил себя долго ждать. На серверной стороне прогресс был обеспечен за счет общего шлюзового интерфейса (Common Gateway Interface, CGI), использования таких языков сценариев, как Perl (альтернатива языку PHP), и выполнения *сценариев на стороне сервера* — динамической вставки содержимого одного файла (или выходных данных выполняемой локальной программы) в другой файл.

Когда ситуация окончательно прояснилась, на передовых позициях остались три основные технологии. Несмотря на то что язык сценариев Perl силами своих стойких приверженцев сохранил популярность, простота PHP и допустимость использования в нем встроенных ссылок на программу базы данных MySQL обеспечили этому языку более чем двойное превосходство по количеству пользователей. А JavaScript, ставший важнейшей составной частью уравнения, используемого для динамического манипулирования каскадными таблицами стилей (Cascading Style Sheets — CSS) и HTML, в настоящее время берет на себя наиболее трудоемкие задачи осуществления асинхронного обмена данными (осуществляемого между клиентом и сервером после загрузки веб-страницы). Используя асинхронный обмен данными, веб-страницы обрабатывают данные и отправляют запросы веб-серверу в фоновом режиме, не оповещая пользователя о происходящем.

Несомненно, своеобразный симбиоз PHP и MySQL способствует их продвижению, но что привлекает к ним разработчиков в первую очередь? На это следует дать простой ответ: та легкость, с которой эти технологии можно использовать для быстрого создания на сайтах динамических элементов. MySQL является быстродействующей и мощной, но при этом простой в использовании системой базы данных, предлагающей сайту практически все необходимое для поиска и обработки данных, которые предназначены для браузеров. Когда PHP для хранения и извлечения этих данных выступает в союзе с MySQL, вы получаете основные составляющие, необходимые для разработки сайтов социальных сетей и для перехода к технологии Web 2.0.

И когда вы также соедините вместе JavaScript и CSS, у вас появится рецепт для создания высокодинамичных и интерактивных сайтов, особенно в современных условиях доступности множества сложных функциональных сред JavaScript, вызов которых действительно позволяет ускорить веб-разработку. К ним относится широко известный jQuery, который до недавнего времени был для программистов одним из наиболее распространенных средств доступа к функциям асинхронного обмена данными, а также более свежая библиотека React JavaScript, популярность которой быстро растет. В настоящее время это один из наиболее широко загружаемых и внедренных фреймворков. Он распространен настолько, что с 2020 года на сайте с предложениями работы Indeed числится более чем в два раза больше вакансий для разработчиков React, чем для разработчиков jQuery.

MariaDB: клон MySQL

После того как компания Oracle приобрела Sun Microsystems (владельца MySQL), возникли опасения, что полностью открытым код MySQL может не остаться, и поэтому от этой СУБД отпочковалась MariaDB, дабы код оставался открытым в соответствии с положениями лицензии GNU GPL. Разработка MariaDB шла под руководством ряда первоначальных создателей MySQL, и эта СУБД сохранила максимальную совместимость с MySQL. Поэтому вероятность встречи на некоторых серверах MariaDB вместо MySQL весьма высока, что не вызывает никаких опасений, поскольку все показанное в данной книге одинаково успешно работает с MySQL. Для любых целей одна СУБД может заменяться другой, и вы при этом не заметите никакой разницы.

Впрочем, вышло так, что многие возникшие поначалу опасения были напрасны, поскольку код MySQL остался открытым, а компания Oracle просто сделала платной приобретение и поддержку тех выпусков, которые предоставляют дополнительные функциональные возможности, включающие гео-репликацию и автоматическое масштабирование. Тем не менее, в отличие от MariaDB, MySQL больше не поддерживается сообществом, но осознание того, что MariaDB никогда не лишится этой поддержки, позволит многим разработчикам спать спокойно и, вероятно, даст гарантии того, что код самой MySQL останется открытым.

PHP

Использование PHP существенно упрощает встраивание средств, придающих веб-страницам динамические свойства. Когда страницам присваивается расширение PHP, у них появляется прямой доступ к языку сценариев. Разработчику нужно лишь написать код, похожий на этот:

```
<?php
  echo " Сегодня " . date('T') . ". ";
?>
Последние новости.
```

Открывающий тег `<?php` дает веб-серверу разрешение на интерпретацию всего последующего кода вплоть до тега `?>`. Все, что находится за пределами этой конструкции, отправляется клиенту в виде простого HTML. Поэтому текст `Последние новости` просто выводится в браузер. А внутри PHP-тегов встроенная функция `date()` отображает текущий день недели, соответствующий системному времени сервера.

В итоге на выходе из этих двух частей получается примерно следующее:

Сегодня Wednesday. Последние новости.

PHP — довольно гибкий язык, и некоторые разработчики предпочитают помещать PHP-конструкцию непосредственно рядом с кодом PHP, как в этом примере:

```
Сегодня <?php echo date("l"); ?>. Последние новости.
```

Существуют также другие способы форматирования и вывода информации, которые будут рассмотрены в главах, посвященных PHP. Важно усвоить то, что, используя PHP, веб-разработчики получают язык сценариев, который хотя и не обладает быстротой кода, скомпилированного на C или ему подобных языках, но все же работает невероятно быстро и к тому же очень хорошо вписывается в разметку HTML.



Если вы собираетесь набирать встречающиеся в этой книге примеры на PHP в программе-редакторе, чтобы работать параллельно с моим повествованием, не забывайте предварять их тегом `<?php`, а в конце ставить тег `?>`, для того чтобы обеспечить их обработку интерпретатором PHP. Для упрощения этой задачи можно заранее подготовить файл `example.php`, содержащий эти теги.

Используя PHP, вы получаете средство управления своим веб-сервером с неограниченными возможностями. Если понадобится на лету внести изменения в HTML, обработать данные кредитной карты, добавить сведения о пользователе в базу данных или извлечь информацию из стороннего сайта, все это можно будет сделать из тех же самых PHP-файлов, в которых находится и сам код HTML.

MySQL

Разумеется, без средств отслеживания информации, предоставляемой пользователем, в ходе работы с вашим сайтом нельзя в полной мере говорить о возможностях динамического изменения выходного кода HTML. На заре создания Всемирной паутины многие сайты использовали неструктурированные текстовые файлы для хранения таких данных, как имена пользователей и пароли. Но такой подход мог вызвать ряд проблем, если файл не был надежно заблокирован от повреждений, возникающих при одновременном доступе к нему множества пользователей. К тому же неструктурированный файл мог разрастаться до таких размеров, что с ним непросто было работать, не говоря уже о трудностях, связанных с попытками объединения файлов и осуществления в них сложных поисковых операций за какое-нибудь мало-мальски приемлемое время.

Именно в таких случаях большое значение приобретает использование реляционных баз данных со структурированной системой запросов. И MySQL, будучи совершенно бесплатной и установленной на огромном количестве веб-серверов



- Lituz.com

Elektron kitoblar

**To'liq qismini Shu tugmani
bosish orqali sotib oling!**