



# Оглавление

Предисловие .....	24
Для кого предназначена эта книга.....	24
Предположения, допущенные в книге.....	25
Структура издания.....	25
Дополнительная литература .....	26
Условные обозначения.....	26
Использование примеров кода.....	27
Благодарности.....	28
<b>Глава 1.</b> Введение в динамическое содержимое веб-страницы.....	29
HTTP и HTML: основы, заложенные Бернерсом-Ли .....	30
Процедура «запрос — ответ» .....	31
Преимущества использования PHP, MySQL, JavaScript и CSS .....	33
MariaDB: клон MySQL.....	34
PHP.....	35
MySQL.....	36
JavaScript.....	37
CSS.....	39
А теперь HTML5.....	40
Веб-сервер Apache .....	41
Обслуживание мобильных устройств.....	41
Несколько слов о программах с открытым исходным кодом .....	42
А теперь все это, вместе взятое .....	42
Вопросы .....	44
<b>Глава 2.</b> Установка сервера, предназначенного для разработки.....	46
Что такое WAMP, MAMP и LAMP .....	47
Установка AMPPS в системе Windows.....	47
Тестирование установки .....	52
Обращение к исходному источнику документов (Windows) .....	53
Другие системы WAMP .....	54

Установка AMPPS в системе macOS .....	55
Обращение к исходному источнику документов (macOS).....	56
Установка LAMP в Linux .....	57
Работа в удаленном режиме.....	58
Вход в систему .....	58
Использование FTP .....	58
Использование редактора программ .....	59
Использование IDE .....	60
Вопросы .....	62
<b>Глава 3. Введение в PHP.....</b>	<b>63</b>
Включение PHP в HTML.....	63
Примеры в этой книге.....	65
Структура PHP .....	66
Комментарии.....	66
Основной синтаксис .....	67
Переменные .....	68
Операторы .....	73
Присваивание значений переменным .....	76
Многострочные команды .....	79
Типы переменных .....	81
Константы.....	82
Предопределенные константы .....	83
Различие между командами echo и print .....	84
Функции .....	85
Область видимости переменной.....	86
Вопросы .....	92
<b>Глава 4. Выражения и управление процессом выполнения программы в PHP.....</b>	<b>94</b>
Выражения.....	94
TRUE или FALSE?.....	95
Литералы и переменные .....	96
Операторы.....	97
Приоритетность операторов .....	98
Взаимосвязанность операторов .....	100
Операторы отношения .....	102
Условия.....	106
Инструкция if .....	106
Инструкция else.....	108
Инструкция elseif .....	109
Инструкция switch .....	111
Оператор ? .....	113
Организация циклов .....	115
Циклы while .....	116
Циклы do..while .....	117

---

Циклы <code>for</code> .....	118
Прекращение работы цикла .....	120
Инструкция <code>continue</code> .....	121
Неявное и явное преобразование типов .....	122
Динамическое связывание в PHP .....	123
Динамическое связывание в действии .....	124
Вопросы .....	125
<b>Глава 5. Функции и объекты PHP</b> .....	126
Функции PHP .....	127
Определение функции .....	128
Возвращение значения .....	129
Возвращение массива .....	131
Передача аргументов по ссылке .....	131
Возвращение глобальных переменных .....	133
И еще раз об области видимости переменных .....	134
Включение и запрос файлов .....	134
Инструкция <code>include</code> .....	134
Инструкция <code>include_once</code> .....	135
Инструкции <code>require</code> и <code>require_once</code> .....	135
Совместимость версий PHP .....	136
Объекты PHP .....	137
Терминология .....	137
Объявление класса .....	139
Создание объекта .....	140
Доступ к объектам .....	140
Клонирование объектов .....	141
Конструкторы .....	143
Деструкторы .....	143
Написание методов .....	144
Объявление свойств .....	144
Объявление констант .....	145
Область видимости свойств и методов .....	146
Статические методы .....	147
Статические свойства .....	148
Наследование .....	149
Вопросы .....	152
<b>Глава 6. Массивы в PHP</b> .....	153
Основные подходы к массивам .....	153
Массивы с числовой индексацией .....	154
Ассоциативные массивы .....	155
Присваивание с использованием ключевого слова <code>array</code> .....	156
Цикл <code>foreach...as</code> .....	157
Многомерные массивы .....	159

Использование функций для работы с массивами.....	162
is_array.....	162
count.....	163
sort.....	163
shuffle.....	163
explode.....	164
extract.....	165
compact.....	165
reset.....	167
end.....	167
Вопросы.....	167
<b>Глава 7. Практикум по программированию на PHP.....</b>	<b>168</b>
Функция printf.....	168
Настройка представления данных.....	170
Дополнение строк.....	172
Функция sprintf.....	173
Функции даты и времени.....	173
Константы, связанные с датами.....	176
Функция checkdate.....	176
Работа с файлами.....	176
Проверка существования файла.....	177
Создание файла.....	177
Чтение из файлов.....	179
Копирование файлов.....	180
Перемещение файла.....	180
Удаление файла.....	181
Обновление файлов.....	181
Блокирование файлов при коллективном доступе.....	183
Чтение всего файла целиком.....	185
Загрузка файлов на веб-сервер.....	186
Системные вызовы.....	191
XHTML или HTML5?.....	193
Вопросы.....	193
<b>Глава 8. Введение в MySQL.....</b>	<b>195</b>
Основные характеристики MySQL.....	195
Сводка понятий, используемых в базах данных.....	196
Доступ к MySQL из командной строки.....	197
Начало работы с интерфейсом командной строки.....	197
Использование интерфейса командной строки.....	201
Команды MySQL.....	202
Типы данных.....	207

---

Индексы.....	218
Создание индекса.....	218
Создание запросов к базе данных MySQL.....	224
Объединение таблиц.....	234
Использование логических операторов.....	237
Функции MySQL.....	237
Работа с MySQL через phpMyAdmin.....	238
Вопросы.....	239
<b>Глава 9. Освоение MySQL.....</b>	<b>240</b>
Проектирование базы данных.....	240
Первичные ключи: ключи к реляционным базам данных.....	241
Нормализация.....	242
Первая нормальная форма.....	244
Вторая нормальная форма.....	246
Третья нормальная форма.....	249
Когда не следует проводить нормализацию.....	251
Отношения.....	252
«Один к одному».....	252
«Один ко многим».....	253
«Многие ко многим».....	254
Базы данных и анонимность.....	255
Транзакции.....	256
Ядра (механизмы хранения) транзакций.....	256
Команда BEGIN.....	257
Команда COMMIT.....	258
Команда ROLLBACK.....	258
Команда EXPLAIN.....	259
Резервное копирование и восстановление данных.....	260
Команда mysqldump.....	260
Создание файла резервной копии.....	262
Восстановление данных из файла резервной копии.....	264
Выгрузка данных в файлы формата CSV.....	264
Планирование резервного копирования.....	265
Вопросы.....	266
<b>Глава 10. Доступ к MySQL с использованием PHP.....</b>	<b>267</b>
Запросы к базе данных MySQL с помощью PHP.....	267
Процесс.....	267
Создание файла регистрации.....	268
Подключение к базе данных MySQL.....	269
Практический пример.....	275
Массив \$_POST.....	278
Удаление записи.....	278

Отображение формы.....	279
Запросы к базе данных.....	280
Запуск программы.....	281
Практическая работа с MySQL.....	282
Создание таблицы.....	282
Описание таблицы.....	283
Удаление таблицы.....	284
Добавление данных.....	284
Извлечение данных.....	285
Обновление данных.....	286
Удаление данных.....	286
Свойство AUTO_INCREMENT.....	287
Выполнение дополнительных запросов.....	288
Предотвращение попыток взлома.....	290
Возможные меры противодействия.....	291
Указатели мест заполнения.....	292
Предотвращение внедрения HTML-кода.....	295
Процедурный метод использования mysqli.....	296
Вопросы.....	298
<b>Глава 11. Обработка форм.....</b>	<b>299</b>
Создание форм.....	299
Извлечение отправленных данных.....	301
Значения по умолчанию.....	302
Типы элементов ввода данных.....	303
Обезвреживание введенных данных.....	312
Пример программы.....	314
Усовершенствования, появившиеся в HTML5.....	317
Атрибут autocomplete.....	317
Атрибут autofocus.....	317
Атрибут placeholder.....	318
Атрибут required.....	318
Атрибуты подмены.....	318
Атрибуты width и height.....	319
Атрибуты min и max.....	319
Атрибут step.....	319
Атрибут form.....	319
Атрибут list.....	320
Тип ввода color.....	320
Типы ввода number и range.....	320
Окно выбора даты и времени.....	320
Вопросы.....	320

---

<b>Глава 12. Cookie, сессии и аутентификация.....</b>	<b>322</b>
Использование cookie в PHP .....	322
Установка cookie .....	324
Доступ к cookie .....	325
Удаление cookie.....	325
HTTP-аутентификация.....	326
Сохранение имен пользователей и паролей .....	329
Пример программы .....	332
Использование сессий .....	335
Начало сессии .....	336
Завершение сессии .....	339
Установка времени ожидания .....	340
Безопасность сессии .....	340
Вопросы .....	345
<b>Глава 13. Изучение JavaScript.....</b>	<b>346</b>
JavaScript и текст HTML .....	347
Использование сценариев в заголовке документа .....	349
Устаревшие и нестандартные браузеры.....	349
Включение файлов JavaScript .....	350
Отладка кода JavaScript.....	350
Комментарии.....	351
Точка с запятой .....	352
Переменные .....	352
Строковые переменные.....	353
Числовые переменные.....	353
Массивы .....	354
Операторы.....	355
Арифметические операторы .....	355
Операторы присваивания .....	355
Операторы сравнения .....	356
Логические операторы .....	356
Инкремент, декремент переменной и краткая форма присваивания.....	357
Объединение строк.....	357
Управляющие символы .....	357
Типизация переменных.....	358
Функции .....	359
Глобальные переменные .....	360
Локальные переменные .....	360
Объектная модель документа .....	361
Еще одно использование знака \$ .....	363
Использование DOM .....	364



О функции document.write.....	365
Использование console.log.....	365
Использование alert.....	365
Запись в элементы.....	365
Использование document.write.....	366
Вопросы.....	366
<b>Глава 14. Выражения и управление процессом выполнения сценариев</b> <b>в JavaScript.....</b>	<b>368</b>
Выражения.....	368
Литералы и переменные.....	369
Операторы.....	370
Приоритетность операторов.....	371
Взаимосвязанность.....	372
Операторы отношения.....	372
Инструкция with.....	375
Использование события onerror.....	376
Конструкция try...catch.....	377
Условия.....	378
Инструкция if.....	379
Инструкция else.....	379
Инструкция switch.....	380
Оператор ?.....	382
Циклы.....	382
Циклы while.....	382
Циклы do...while.....	383
Циклы for.....	384
Прекращение работы цикла.....	384
Инструкция continue.....	385
Явное преобразование типов.....	386
Вопросы.....	386
<b>Глава 15. Функции, объекты и массивы JavaScript.....</b>	<b>388</b>
Функции JavaScript.....	388
Определение функции.....	388
Возвращение значения.....	390
Возвращение массива.....	392
Объекты JavaScript.....	393
Объявление класса.....	393
Создание объекта.....	395
Доступ к объектам.....	395
Ключевое слово prototype.....	396
Массивы в JavaScript.....	399
Числовые массивы.....	399
Ассоциативные массивы.....	400

---

Многомерные массивы .....	401
Методы массивов .....	403
Вопросы .....	408
<b>Глава 16.</b> Проверка данных и обработка ошибок в JavaScript и PHP.....	409
Проверка данных, введенных пользователем, средствами JavaScript .....	409
Документ validate.html (часть первая) .....	410
Документ validate.html (часть вторая) .....	412
Регулярные выражения.....	416
Соответствие, закладываемое в метасимволы .....	416
Нестрогое символьное соответствие.....	417
Группировка с помощью скобок .....	419
Символьный класс.....	419
Указание диапазона .....	420
Инвертирование .....	420
Более сложные примеры .....	420
Сводная таблица метасимволов.....	423
Общие модификаторы .....	425
Использование регулярных выражений в JavaScript .....	426
Использование регулярных выражений в PHP.....	426
Повторное отображение формы после проверки данных PHP-программой.....	427
Вопросы .....	433
<b>Глава 17.</b> Использование технологии асинхронного обмена данными.....	435
Что такое асинхронный обмен данными.....	436
XMLHttpRequest.....	436
Ваша первая программа, использующая асинхронный обмен данными.....	439
Использование GET вместо POST.....	443
Отправка XML-запросов.....	446
Использование специальных сред для асинхронного обмена данными .....	451
Вопросы .....	451
<b>Глава 18.</b> Введение в CSS .....	452
Импортирование таблицы стилей.....	453
Импортирование CSS из HTML-кода.....	454
Встроенные настройки стиля .....	454
Идентификаторы (ID).....	454
Классы .....	455
Точки с запятой.....	455
Правила CSS .....	456
Множественные задания стиля .....	456
Комментарии .....	457
Типы стилей .....	458
Исходные стили.....	458
Пользовательские стили .....	458
Внешние таблицы стилей.....	459

Внутренние стили.....	460
Внедренные стили .....	460
Селекторы CSS.....	460
Селектор типа.....	460
Селектор потомков.....	461
Селектор дочерних элементов.....	462
Селектор элементов, имеющих идентификатор .....	463
Селектор класса.....	464
Селектор атрибутов.....	464
Универсальный селектор .....	465
Групповая селекция .....	466
Каскадность CSS.....	466
Создатели таблиц стилей .....	467
Методы создания таблиц стилей .....	467
Селекторы таблиц стилей .....	468
Разница между элементами Div и Span.....	470
Измерения.....	472
Шрифты и оформление.....	474
font-family .....	474
font-style .....	475
font-size .....	475
font-weight .....	476
Управление стилями текста.....	476
Оформление.....	476
Разрядка .....	477
Выравнивание .....	477
Преобразование.....	477
Отступы.....	477
Цвета CSS.....	478
Сокращенные цветовые строки .....	479
Градиенты.....	479
Позиционирование элементов.....	481
Абсолютное позиционирование .....	481
Относительное позиционирование .....	482
Фиксированное позиционирование.....	482
Псевдоклассы.....	484
Сокращенная запись правил .....	487
Модель блока и макет страницы.....	488
Установка полей.....	488
Применение границ.....	490
Настройка отступов .....	492
Содержимое объекта.....	493
Вопросы .....	493

---

<b>Глава 19. Расширение CSS с помощью CSS3</b> .....	495
Селекторы атрибутов.....	496
Свойство box-sizing.....	498
Создание фона в CSS3.....	498
Свойство background-clip.....	498
Свойство background-origin.....	500
Свойство background-size.....	501
Использование нескольких фонов.....	501
Границы CSS3.....	503
Свойство border-color.....	503
Свойство border-radius.....	504
Прямоугольные тени.....	507
Выход элемента за пределы размеров.....	508
Разметка с использованием нескольких колонок.....	508
Цвета и непрозрачность.....	510
Цвета HSL.....	510
Цвета HSLA.....	511
Цвета RGB.....	511
Цвета RGBA.....	512
Свойство opacity.....	512
Эффекты, применяемые к тексту.....	512
Свойство text-shadow.....	512
Свойство text-overflow.....	513
Свойство word-wrap.....	513
Веб-шрифты.....	514
Трансформации.....	516
Переходы.....	518
Свойства, применяемые к переходам.....	519
Продолжительность перехода.....	519
Задержка перехода.....	519
Задание скорости перехода.....	520
Сокращенный синтаксис.....	520
Вопросы.....	522
<b>Глава 20. Доступ к CSS из JavaScript</b> .....	523
Еще одно обращение к функции getElementById.....	523
Функция O.....	524
Функция S.....	524
Функция C.....	525
Включение функций.....	526
Обращение к свойствам CSS из JavaScript.....	526
Некоторые общие свойства.....	527
Другие свойства.....	528

Встроенный JavaScript.....	530
Ключевое слово this .....	531
Привязка событий к объектам в сценарии.....	532
Прикрепление к другим событиям.....	532
Добавление новых элементов.....	533
Удаление элементов.....	535
Альтернативы добавлению и удалению элементов .....	535
Использование прерываний .....	536
Использование функции setTimeout .....	537
Отмена тайм-аута.....	538
Функция setInterval .....	538
Использование прерываний для анимации.....	540
Вопросы .....	542
<b>Глава 21. Введение в jQuery.....</b>	<b>543</b>
Почему же именно jQuery?.....	543
Включение jQuery .....	544
Выбор подходящей версии .....	544
Загрузка.....	546
Использование сети доставки контента.....	547
Заказная сборка jQuery.....	548
Синтаксис jQuery .....	548
Простой пример.....	548
Как избежать конфликтов библиотек .....	549
Селекторы .....	550
Метод css .....	550
Селектор элемента.....	551
Селектор идентификатора.....	552
Селектор класса.....	552
Сочетание селекторов .....	552
Обработка событий.....	553
Ожидание готовности документа .....	555
Функции и свойства событий.....	556
События blur и focus .....	556
Ключевое слово this .....	558
События click и dblclick.....	558
Событие keypress.....	559
Деликатное программирование .....	561
Событие mousemove.....	562
Другие события, связанные с мышью.....	565
Альтернативные методы работы с мышью .....	566
Событие submit .....	566
Специальные эффекты .....	568

---

Исчезновение и появление.....	569
Метод toggle.....	570
Проявление и растворение.....	571
Скольжение элементов вверх и вниз.....	572
Анимация.....	573
Остановка анимации.....	577
Работа с DOM.....	577
Разница между методами text и html.....	578
Методы val и attr.....	579
Добавление и удаление элементов.....	580
Динамическое применение классов.....	583
Работа с размерами.....	583
Методы width и height.....	584
Методы innerWidth и innerHeight.....	586
Методы outerWidth и outerHeight.....	586
Обход объектов DOM.....	587
Родительские элементы.....	587
Дочерние элементы.....	592
Одноуровневые элементы.....	592
Выбор следующих и предыдущих элементов.....	594
Обход элементов, выбранных с помощью методов jQuery.....	595
Метод is.....	597
Использование jQuery без селекторов.....	599
Метод \$.each.....	599
Метод \$.map.....	600
Использование асинхронного обмена данными.....	600
Метод POST.....	601
Метод GET.....	601
Дополнительные модули (плагины).....	602
Пользовательский интерфейс jQuery.....	602
Другие дополнительные модули.....	603
Вопросы.....	604
<b>Глава 22. Введение в jQuery Mobile.....</b>	<b>605</b>
Включение jQuery Mobile.....	606
Начало работы.....	607
Связывание страниц.....	609
Синхронная связанность.....	610
Связанность внутри многостраничного документа.....	611
Смена страниц.....	611
Стильные кнопки.....	615
Обработка списков.....	618
Фильтруемые списки.....	620
Разделители списков.....	621

---

А что же дальше? .....	624
Вопросы .....	625
<b>Глава 23. Введение в HTML5</b> .....	626
Холст .....	626
Геолокация .....	628
Аудио и видео .....	630
Формы .....	631
Локальное хранилище .....	631
Рабочие веб-процессы .....	631
Микроданные .....	632
Вопросы .....	632
<b>Глава 24. Холсты в HTML5</b> .....	633
Создание холста и доступ к нему .....	633
Функция toDataURL .....	635
Указание типа изображения .....	637
Метод fillRect .....	637
Метод clearRect .....	638
Метод strokeRect .....	638
Сочетание всех этих команд .....	638
Метод createLinearGradient .....	639
Метод addColorStop в подробностях .....	642
Метод createRadialGradient .....	643
Использование узоров для заливки .....	644
Запись текста на холсте .....	646
Метод strokeText .....	647
Свойство textBaseLine .....	647
Свойство font .....	647
Свойство textAlign .....	648
Метод fillText .....	648
Метод measureText .....	649
Рисование линий .....	649
Свойство lineWidth .....	649
Свойства lineCap и lineJoin .....	650
Свойство miterLimit .....	652
Использование путей .....	652
Методы moveTo и lineTo .....	653
Метод stroke .....	653
Метод rect .....	653
Заливка областей .....	654
Метод clip .....	655
Метод isPointInPath .....	658

---

Работа с кривыми линиями .....	659
Метод arc .....	659
Метод arcTo .....	661
Метод quadraticCurveTo .....	663
Метод bezierCurveTo .....	664
Обработка изображений .....	665
Метод drawImage .....	665
Изменение размеров изображения .....	666
Выбор области изображения .....	666
Копирование с холста .....	668
Добавление теней .....	668
Редактирование на уровне пикселей .....	670
Метод getImageData .....	670
Массив data .....	671
Метод putImageData .....	673
Метод createImageData .....	673
Более сложные графические эффекты .....	673
Свойство globalCompositeOperation .....	674
Свойство globalAlpha .....	676
Преобразования .....	677
Метод scale .....	677
Методы save и restore .....	678
Метод rotate .....	679
Метод translate .....	680
Метод transform .....	681
Метод setTransform .....	683
Вопросы .....	683
<b>Глава 25. Аудио и видео в HTML5</b> .....	685
О кодеках .....	686
Элемент audio .....	688
Поддержка браузеров, не работающих с HTML5 .....	690
Элемент video .....	691
Видеокодеки .....	692
Поддержка устаревших браузеров .....	695
Вопросы .....	696
<b>Глава 26. Другие свойства HTML5</b> .....	697
Геолокация и служба GPS .....	697
Другие методы определения местоположения .....	698
Геолокация и HTML5 .....	699
Локальное хранилище .....	702
Использование локального хранилища .....	703
Объект localStorage .....	704



Рабочие веб-процессы.....	706
Перетаскивание.....	709
Обмен сообщениями между документами .....	711
Другие теги HTML5 .....	715
Вопросы .....	715
<b>Глава 27. Объединение технологий .....</b>	<b>716</b>
Проектирование приложения социальной сети .....	717
Информация на сайте .....	717
Файл functions.php.....	718
Файл header.php.....	720
Файл setup.php .....	723
Файл index.php .....	724
Файл signup.php .....	726
Проверка возможности применения желаемого имени пользователя .....	726
Регистрация .....	726
Файл checkuser.php.....	729
Файл login.php .....	730
Файл profile.php .....	733
Добавление текста в поле About Me (Обо мне) .....	733
Добавление изображения профиля .....	734
Обработка изображения .....	734
Отображение текущего профиля .....	735
Файл members.php.....	737
Просмотр профилей пользователей .....	738
Добавление и удаление друзей.....	738
Вывод списка всех участников.....	738
Файл friends.php.....	741
Файл messages.php.....	744
Файл logout.php .....	748
Файл styles.css.....	749
Файл javascript.js .....	751
<b>Приложение А. Ответы на контрольные вопросы .....</b>	<b>752</b>
Глава 1 .....	752
Глава 2 .....	753
Глава 3 .....	753
Глава 4 .....	755
Глава 5 .....	756
Глава 6 .....	757
Глава 7 .....	757
Глава 8 .....	758
Глава 9 .....	759
Глава 10.....	760

---

Глава 11 .....	760
Глава 12 .....	761
Глава 13 .....	762
Глава 14 .....	763
Глава 15 .....	764
Глава 16 .....	764
Глава 17 .....	765
Глава 18 .....	766
Глава 19 .....	767
Глава 20 .....	768
Глава 21 .....	769
Глава 22 .....	770
Глава 23 .....	771
Глава 24 .....	771
Глава 25 .....	772
Глава 26 .....	773
<b>Приложение Б. Интернет-ресурсы</b> .....	774
Сайты, относящиеся к PHP .....	774
Сайты, относящиеся к MySQL .....	774
Сайты, относящиеся к JavaScript .....	775
Сайты, относящиеся к CSS .....	775
Сайты, относящиеся к HTML5 .....	775
Сайты, относящиеся к асинхронному обмену данными .....	775
Сайты с разнообразными ресурсами .....	776
<b>Приложение В. Стоповые слова MySQL для FULLTEXT</b> .....	777
<b>Приложение Г. Функции MySQL</b> .....	780
Строковые функции .....	780
Функции для работы с датами .....	783
Функции для работы с временем .....	788
<b>Приложение Д. Селекторы, объекты и методы jQuery</b> .....	790
Селекторы jQuery .....	790
Объекты jQuery .....	795
Методы jQuery .....	797



# 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 первые браузер и веб-сервер.

Теперь эти средства воспринимаются нами как должное, но в то время концепция их применения носила революционный характер. До этого основной объем соединений приходился на пользователей домашних модемов, дозванивавшихся и подключававшихся к электронным доскам объявлений, которые базировались на отдельном компьютере и позволяли общаться и обмениваться данными только с другими пользователями этой 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 текущий потенциал 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

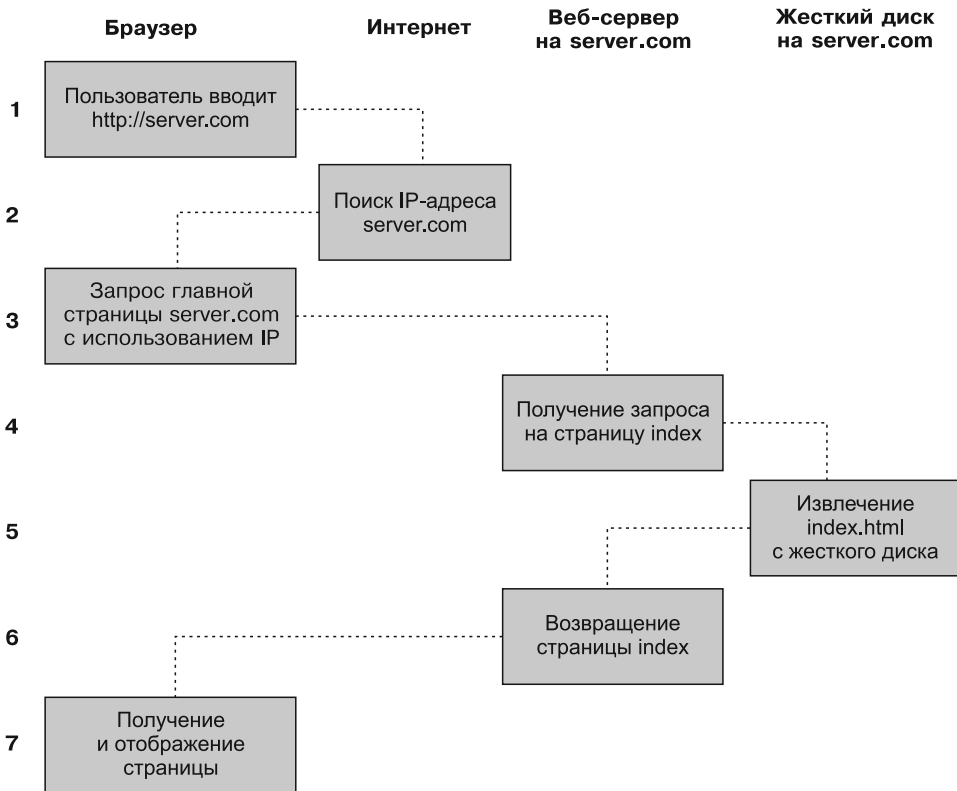
HTTP представляет собой стандарт взаимодействия, регулирующий порядок направления запросов и получения ответов — процесса, происходящего между браузером, запущенным на компьютере конечного пользователя, и веб-сервером. Задача сервера состоит в том, чтобы принять запрос от клиента и попытаться дать на 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 к веб-серверу отправить ему веб-страницу и выполнения браузером данной просьбы. После этого браузер занимается отображением 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. Сервер извлекает веб-страницу и отправляет ее по обратному маршруту в адрес браузера.
6. 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.

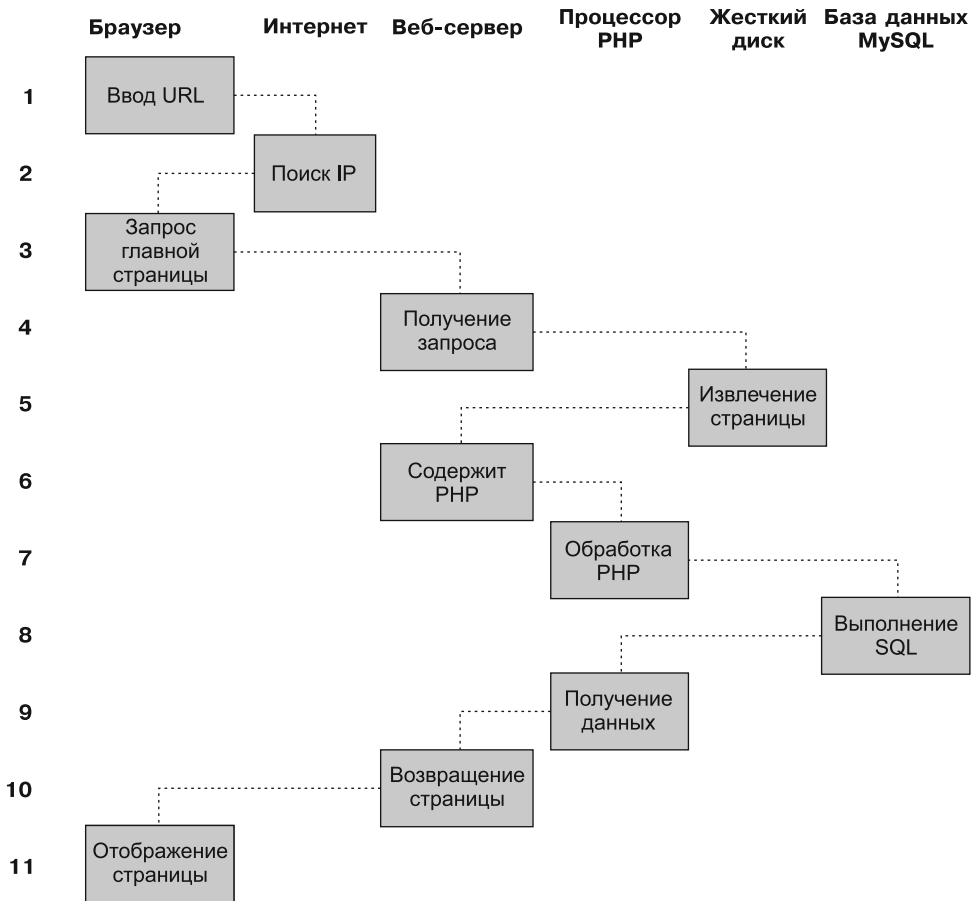
Обратите внимание на то, что на шаге 2 браузер ищет IP-адрес, принадлежащий доменному имени `server.com`. У каждой машины, подключенной к Интернету, включая и ваш компьютер, есть свой IP-адрес. Но, как правило, доступ к веб-серверам осуществляется по именам, таким как `google.com`. Вам должно быть известно,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

При передаче динамических веб-страниц процедура состоит из большего 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 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`. Запрос проходит по Сети и поступает на веб-сервер `server.com`.
4. Веб-сервер, получивший запрос, ищет веб-страницу на своем жестком диске.
5. 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.
6. Интерпретатор PHP выполняет PHP-код.
7. Кое-какие фрагменты кода PHP содержат MySQL-инструкции, которые интерпретатор PHP, в свою очередь, передает процессору базы данных MySQL.
8. База данных MySQL возвращает результаты выполнения инструкции интерпретатору PHP.
9. Интерпретатор PHP возвращает веб-серверу результаты выполнения кода PHP, а также результаты, полученные от базы данных MySQL.
10. 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 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. Достаточно упомянуть широко известную библиотеку JQuery, применение которой в настоящее время, наверное, является наиболее востребованным способом доступа программистов к функциям асинхронного обмена данными.

## MariaDB: клон MySQL

После того как компания Oracle приобрела Sun Microsystems (владельца MySQL), возникли опасения, что полностью открытым код MySQL может не остаться, и в итоге 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 должно вызывать никаких опасений, поскольку все показанное в данной книге одинаково успешно работает как под MySQL, так и под MariaDB, имеющей ту же кодовую базу, что и MySQL Server 5.5. В любом случае 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 георепликацию 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 " Today is " . date('T') . ". ";
?>
Here's the latest news.
```

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 `Here's the latest news` 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:

**Today is Wednesday. Here's the latest news.**

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

```
Today is <?php echo date("l"); ?>. Here's the latest news.
```

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 подобный 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 системой, оказывается как нельзя кстати. Она представляет собой надежную и исключительно быстродействующую систему управления базами данных, использующую команды, похожие на простые английские слова.

Высшим уровнем структуры MySQL является база данных, внутри которой можно иметь одну или несколько таблиц, содержащих ваши данные. Предположим, вы работаете над таблицей под названием `users` (пользователи), внутри которой были созданы столбцы для фамилий — `surname`, имен — `firstname` и адресов электронной

почты — `email`, и теперь нужно добавить еще одного пользователя. Одна из команд, которую можно применить для этого, выглядит следующим образом:

```
INSERT INTO users VALUES('Smith', 'John', 'jsmith@mysite.com');
```

Для создания базы данных и таблицы, а также настройки всех нужных полей понадобится сначала выдать и другие команды, но используемая здесь SQL-команда `INSERT` демонстрирует простоту добавления в базу данных новой информации. `Structured Query Language (SQL)` является языком, разработанным в начале 1970-х годов и напоминающим один из старейших языков программирования — `COBOL`. Тем не менее он хорошо подходит для запросов к базе данных, что и предопределило его использование в течение столь длительного времени.

Так же просто выполняется и поиск данных. Предположим, что имеется адрес электронной почты пользователя и нужно найти имя его владельца. Для этого можно ввести следующий запрос MySQL:

```
SELECT surname,firstname FROM users WHERE email='jsmith@mysite.com';
```

После этого MySQL вернет `Smith, John` и любые другие пары имен, которые могут быть связаны в базе данных с адресом электронной почты.

Нетрудно предположить, что возможности MySQL простираются значительно дальше выполнения простых команд вставки и выбора — `INSERT` и `SELECT`. Например, можно скомбинировать родственные наборы данных, чтобы собрать вместе взаимосвязанные части информации, запросить результаты, выбрав порядок их выдачи из множества вариантов, найти частичные совпадения, если известна только часть искомой строки, вернуть конкретно заданное количество результатов и сделать многое другое.

При использовании PHP все эти вызовы можно направлять непосредственно к MySQL без необходимости самостоятельного применения ее интерфейса командной строки. Это значит, что для того, чтобы докопаться до нужного вам элемента данных, вы можете сохранять результаты в массивах для их обработки и осуществления множества поисковых операций, каждая из которых зависит от результатов, возвращенных предыдущими операциями.

Далее будет показано, что для придания еще большей мощности прямо в MySQL встроено несколько дополнительных функций, которые можно вызвать с целью эффективного выполнения наиболее часто встречающихся в MySQL операций, избавляясь от необходимости их составления из нескольких PHP-вызовов в адрес MySQL.

## JavaScript

Самая старая из трех основных технологий, рассматриваемых в данной книге, — JavaScript — была создана для получения доступа из сценариев ко всем элементам HTML-документа. Иными словами, она предоставляет средства для динамического

взаимодействия с пользователем, например для проверки приемлемости адресов электронной почты в формах ввода данных, отображения подсказок наподобие «Вы действительно подразумевали именно это?» и т. д. (хотя с точки зрения безопасности, которая всегда должна реализовываться на веб-сервере, на эту технологию положиться нельзя).

В сочетании с CSS JavaScript закладывает основу мощности динамических веб-страниц, которые изменяются буквально на глазах, в отличие от новой страницы, возвращаемой сервером.

Тем не менее с использованием JavaScript могут возникнуть осложнения, обусловленные некоторыми существенными различиями в способах реализации этого языка, выбранных разными разработчиками браузеров. В основном эти различия возникают, когда некоторые производители пытаются придать своим браузерам дополнительные функциональные возможности, не обращая внимания на совместимость с продуктами своих конкурентов.

К счастью, разработчики в большинстве своем уже взяли себя за ум, и теперь оптимизация вашего кода для различных браузеров утратила прежнюю актуальность. Но остаются миллионы экземпляров устаревших браузеров, которыми будут пользоваться на протяжении еще многих лет. Тем не менее и для них существуют решения проблем несовместимости, и позже в этой книге будут рассмотрены библиотеки и технологии, позволяющие без каких-либо опасений игнорировать существующие различия.

А сейчас взглянем на то, как можно воспользоваться обычным JavaScript-кодом, воспринимаемым всеми браузерами:

```
<script type="text/javascript">
  document.write("Today is " + Date() );
</script>
```

Этот фрагмент кода предписывает браузеру интерпретировать все, что находится внутри тегов `<script>`, в качестве кода JavaScript, что затем браузер и сделает, записав в текущий документ текст "Today is ", а также дату, полученную за счет использования принадлежащей JavaScript функции `Date`. В результате получится нечто подобное следующему:

**Today is Sun Jan 01 2017 01:23:45**



Если не требуется указывать конкретную версию JavaScript, то, как правило, можно опустить `type="text/javascript"` и использовать для начала интерпретации JavaScript тег `<script>`.

---

Ранее было упомянуто, что изначально JavaScript разрабатывался для того, чтобы получить возможность динамического управления различными элементами, нахо-

дящимися внутри HTML-документа, и это его предназначение по-прежнему является основным. Но все чаще JavaScript применяется для осуществления процесса доступа к веб-серверу в фоновом режиме с целью *асинхронного обмена данными*.

Асинхронный обмен данными позволил веб-страницам стать похожими на автономные программы, поскольку для отображения нового содержимого их уже не нужно загружать целиком. Вместо этого асинхронный вызов может использоваться для извлечения и обновления отдельно взятого элемента веб-страницы, например может быть изменена ваша фотография на сайте социальной сети или заменена кнопка, на которой нужно щелкнуть, отвечая на вопрос. Полностью эта тема будет рассмотрена в главе 17.

Затем в главе 21 мы присмотримся к среде jQuery, которую можно использовать, чтобы не изобретать колесо в случае возникновения потребностей в быстродействующем, кросс-браузерном коде для управления веб-страницами. Конечно, доступны и другие подобные среды, но jQuery является самой популярной библиотекой и, исходя из показателей, накопленных за весьма длительный срок ее использования, обладает исключительной надежностью и является основным средством в арсенале многих опытных разработчиков.

## CSS

CSS является ключевым дополнением к HTML, обеспечивая соответствующую разметку HTML-текста и встроенных изображений, соотносясь с параметрами экрана пользователя. После появления третьего стандарта (CSS3) CSS предлагает уровень динамической интерактивности, которая прежде поддерживалась только с помощью JavaScript. Например, вы можете не только придать стиль любому элементу HTML, чтобы изменить его размеры, цвета, границы, интервалы, но и, используя всего лишь несколько строк CSS, добавить своим веб-страницам анимированные переходы и преобразования.

Применение CSS может просто заключаться во вставке правил между тегами `<style>` и `</style>`, расположенными в заголовке веб-страницы:

```
<style>
  p{
    text-align:justify;
    font-family:Helvetica;
  }
</style>
```

Эти правила будут изменять исходное выравнивание текста тега `<p>`, чтобы содержащиеся в нем абзацы были полностью выровнены и для них использовался шрифт Helvetica.

В главе 18 вы увидите, что существует множество различных способов задания правил CSS и их также можно включать непосредственно в теги или сохранять во

внешнем файле, предназначенном для отдельной загрузки. Такая гибкость позволяет проводить точную настройку стиля HTML. Вы также увидите, как с помощью CSS можно, например, создать встроенную функцию `hover` для анимирования объектов при проходе над ними указателя мыши. Кроме того, вы научитесь получать доступ ко всем свойствам CSS-элемента из JavaScript и из HTML.

## А теперь HTML5

Какими бы ни были полезными все эти дополнения к веб-стандартам, самым амбициозным разработчикам и их было мало. К примеру, так и не был придуман простой способ работы с графикой в браузере, не требующий обращения к таким дополнительным модулям, как Flash. То же самое происходило и в отношении аудио- и видеовставок в веб-страницы. Кроме того, можно отметить множество досадных несоответствий, вкравшихся в HTML в процессе его развития.

Итак, чтобы подчистить все эти шероховатости, перенести Интернет за пределы технологии Web 2.0 в его следующую фазу развития, был создан новый стандарт HTML, устраняющий перечисленные недостатки: *HTML5*. Он начал разрабатываться в далеком 2004 году, когда Mozilla Foundation и Opera Software (разработчики двух популярных браузеров) составили его первый проект. Но его окончательный проект был представлен World Wide Web Consortium (W3C), международной организацией, руководящей веб-стандартами, лишь в начале 2013 года.

На разработку HTML5 ушло несколько лет, но теперь мы располагаем весьма надежной и стабильной версией 5.1 (с 2016 года). Но цикл разработки никогда не заканчивается, и со временем в этот язык несомненно будут встроены дополнительные функциональные возможности. К некоторым наиболее ярким свойствам HTML5, позволяющим управлять медиаресурсами, можно отнести элементы `<audio>`, `<video>` и `<canvas>`, с помощью которых происходит добавление звука, видео и усовершенствованной графики. Все, что нужно знать об этих и всех других аспектах HTML5, весьма подробно изложено в данной книге, начиная с главы 23.



Кроме всего прочего, мне в спецификации HTML5 нравится, что для самозакрывающихся элементов больше не нужен синтаксис XHTML. Раньше перевод на новую строку можно было изобразить с помощью элемента `<br>`. Затем для обеспечения совместимости в будущем с XHTML (так и не состоявшейся заменой HTML) элемент был изменен на `<br />` с добавленным символом `/` (поскольку ожидалось, что характерной особенностью закрывающего тега всех элементов станет именно этот символ). Но теперь все вернулось на круги своя и можно использовать любую из версий таких элементов. Итак, для большей лаконичности и меньшего объема набираемого текста в данной книге я вернулся к прежнему стилю: `<br>`, `<hr>` и т. д.

---