

Владимир Дронов



# PHP, MySQL, HTML 5 и CSS 3

## Разработка современных динамических Web-сайтов

АУДИО И ВИДЕО HTML 5  
ПРЕОБРАЗОВАНИЯ  
И АНИМАЦИЯ CSS 3  
JavaScript, DOM и AJAX  
БИБЛИОТЕКА Yii  
ИНТЕРАКТИВНЫЕ  
ЭЛЕМЕНТЫ: СПОЙЛЕР,  
ЛАЙТБОКС И БЛОКНОТ  
УНИВЕРСАЛЬНОЕ  
ФАЙЛОВОЕ ХРАНИЛИЩЕ  
ПОДДЕРЖКА BBCode  
ПУБЛИКАЦИЯ САЙТА

**PRO**  
ПРОФЕССИОНАЛЬНОЕ  
ПРОГРАММИРОВАНИЕ



Материалы  
на [www.bhv.ru](http://www.bhv.ru)

УДК 004.43+004.738.5

ББК 32.973.26-018.1

Д75

**Дронов В. А.**

Д75 PHP, MySQL, HTML5 и CSS 3. Разработка современных динамических Web-сайтов. — СПб.: БХВ-Петербург, 2016. — 688 с.: ил. — (Профессиональное программирование)

ISBN 978-5-9775-3529-8

Книга посвящена разработке динамических Web-сайтов с применением HTML5, CSS 3, PHP и MySQL. Описаны возможности HTML5 по работе с текстом, графикой, аудио и видео, таблицами, средствами навигации и Web-формами, а также способы представления, преобразования и анимации Web-страниц с помощью CSS 3. Рассказано о языке JavaScript, объектной модели документа DOM, разработке Web-сценариев и технологии AJAX. Рассмотрены серверное программирование, язык PHP и сервер данных MySQL. Дано описание библиотеки Yii, предоставляющей Web-программисту готовый набор инструментов для написания серверных приложений. На практических примерах показана разработка дизайна страниц, интерактивных элементов — спойлера, лайтбокса и блокнота, создание универсального файлового хранилища и реализации поддержки тегов BBCode для форматирования текста. Рассмотрен процесс создания полнофункционального сайта и его публикации в Интернете. Все исходные коды доступны для загрузки с сайта издательства.

*Для Web-программистов*

УДК 004.43+004.738.5

ББК 32.973.26-018.1

**Группа подготовки издания:**

Главный редактор	<i>Екатерина Кондукова</i>
Зам. главного редактора	<i>Евгений Рыбаков</i>
Зав. редакцией	<i>Екатерина Капалыгина</i>
Редактор	<i>Анна Кузьмина</i>
Компьютерная верстка	<i>Ольги Сергиенко</i>
Корректор	<i>Зинаида Дмитриева</i>
Дизайн серии	<i>Инны Тачиной</i>
Оформление обложки	<i>Марины Дамбиевой</i>

Подписано в печать 31.08.15.

Формат 70×100<sup>1/16</sup>. Печать офсетная. Усл. печ. л. 55,47.

Тираж 1000 экз. Заказ №

"БХВ-Петербург", 191036, Санкт-Петербург, Гончарная ул., 20.

Первая Академическая типография "Наука"

199034, Санкт-Петербург, 9 линия, 12/28

ISBN 978-5-9775-3529-8

© Дронов В. А., 2016

© Оформление, издательство "БХВ-Петербург", 2016

# Оглавление

<b>Введение</b> .....	<b>19</b>
Некоторые замечания .....	19
Типографские соглашения .....	20
Благодарности .....	22
<b>ЧАСТЬ I. СОДЕРЖИМОЕ WEB-СТРАНИЦ. ЯЗЫК HTML5</b> .....	<b>23</b>
<b>Глава 1. Современный Web-дизайн. Web 2.0</b> .....	<b>25</b>
Современный Web-дизайн. Концепция Web 2.0 .....	25
Что требуется от современного Web-сайта .....	25
Концепция Web 2.0 .....	28
Интернет: как все это работает .....	30
Клиенты и серверы Интернета. Интернет-адреса .....	30
Web-сайты и Web-серверы .....	31
Что дальше? .....	33
<b>Глава 2. Введение в язык HTML5</b> .....	<b>34</b>
Основные принципы HTML .....	34
Первая Web-страница .....	34
Теги и атрибуты тегов .....	36
Вложенность тегов .....	37
Форматирование Web-страницы .....	38
Секции Web-страницы .....	38
Метаданные и метатеги .....	39
Что дальше? .....	40
<b>Глава 3. Структурирование текста</b> .....	<b>41</b>
Простейшие средства структурирования текста .....	41
Абзацы и заголовки .....	41
Блочные элементы HTML .....	43
Списки .....	43
Цитаты и адреса .....	45
Текст фиксированного формата .....	46
Блочные контейнеры .....	47

Семантическая разметка текста.....	47
Горизонтальные линии.....	49
Комментарии.....	49
Что дальше? .....	50
<b>Глава 4. Оформление текста.....</b>	<b>51</b>
Выделение фрагментов текста.....	51
Встроенные элементы HTML.....	52
Встроенные контейнеры .....	53
Разрыв строк .....	53
Вставка специальных символов. Литералы.....	53
Что дальше? .....	55
<b>Глава 5. Графика и мультимедиа .....</b>	<b>56</b>
Внедренные элементы Web-страниц .....	56
Графика.....	57
Форматы интернет-графики.....	57
Вставка графических изображений.....	58
Мультимедиа.....	59
Поддерживаемые форматы мультимедийных файлов.....	59
Вставка аудиоролика .....	60
Вставка видеоролика .....	61
Указание нескольких источников аудио и видео .....	63
Что дальше? .....	64
<b>Глава 6. Таблицы .....</b>	<b>65</b>
Создание таблиц .....	65
Заголовки и секции таблицы.....	68
Объединение ячеек таблиц .....	70
Что дальше? .....	73
<b>Глава 7. Средства навигации.....</b>	<b>74</b>
Текстовые гиперссылки .....	74
Создание гиперссылок.....	74
Интернет-адреса в WWW.....	76
Почтовые гиперссылки .....	77
Дополнительные возможности гиперссылок .....	77
Графические гиперссылки .....	79
Изображения-гиперссылки .....	79
Изображения-карты .....	79
Полоса навигации .....	81
Якоря.....	82
Что дальше? .....	83
<b>Глава 8. Web-формы и элементы управления .....</b>	<b>84</b>
Средства ввода данных .....	84
Создание Web-форм .....	85
Создание элементов управления .....	86
Общие вопросы создания элементов управления .....	86
Поле ввода .....	87

Поле ввода пароля .....	88
Поле ввода числового значения.....	89
Поле ввода интернет-адреса .....	89
Поле ввода адреса электронной почты .....	89
Флажок.....	90
Переключатель.....	90
Регулятор .....	91
Область редактирования .....	91
Список.....	92
Поле ввода файла.....	94
Скрытое поле.....	94
Кнопки .....	95
Элементы оформления и их создание.....	96
Надпись.....	96
Группа.....	96
Что дальше? .....	97

## **ЧАСТЬ II. ПРЕДСТАВЛЕНИЕ WEB-СТРАНИЦ. КАСКАДНЫЕ ТАБЛИЦЫ СТИЛЕЙ CSS 3 .....**

**99**

### **Глава 9. Введение в CSS 3.....**

**101**

Понятие о стилях CSS .....	101
Создание стилей CSS.....	102
Таблицы стилей. Встроенные стили .....	103
Правила каскадности и приоритет стилей .....	105
Наследование атрибутов стилей.....	108
Комментарии CSS.....	109
Что дальше? .....	109

### **Глава 10. Селекторы стилей. Единицы измерения CSS.....**

**110**

Селекторы стилей .....	110
Введение в селекторы стилей .....	110
Компоненты указателей .....	111
Основные указатели .....	111
Указатели на атрибуты тега .....	113
Псевдоклассы .....	114
Псевдоэлементы.....	118
Разделители .....	118
Единицы измерения и вычисления CSS.....	119
Важные атрибуты стилей.....	121
Что дальше? .....	121

### **Глава 11. Параметры текста и фона.....**

**122**

Параметры текста .....	122
Параметры шрифта.....	122
Загружаемые шрифты. Директивы CSS.....	126
Параметры, управляющие разрывом строк.....	127
Параметры вертикального выравнивания .....	129
Параметры тени у текста.....	130

Параметры фона .....	131
Что дальше? .....	135
<b>Глава 12. Параметры абзацев, списков и отображения.</b>	
<b>Генерируемое содержимое .....</b>	<b>136</b>
Параметры вывода текста .....	136
Параметры списков .....	137
Параметры отображения .....	139
Параметры курсора .....	141
Генерируемое содержимое .....	141
Статичное генерируемое содержимое .....	142
Создание нумерации .....	143
Что дальше? .....	145
<b>Глава 13. Параметры блоков. Блочный Web-дизайн .....</b>	<b>146</b>
Параметры блочных элементов .....	146
Параметры размеров .....	146
Параметры размещения. Плавающие элементы .....	147
Параметры переполнения. Элементы с прокруткой .....	149
Параметры тени у блочного элемента .....	150
Позиционируемые блоки .....	151
Понятие позиционируемого блока .....	151
Создание позиционируемых элементов .....	152
Основы блочного Web-дизайна .....	155
Что дальше? .....	159
<b>Глава 14. Параметры отступов, рамки и выделения .....</b>	<b>160</b>
Параметры отступов .....	160
Параметры рамки .....	163
Параметры выделения .....	166
Что дальше? .....	168
<b>Глава 15. Параметры таблиц .....</b>	<b>169</b>
Параметры выравнивания .....	169
Параметры отступов и рамок .....	170
Параметры размеров .....	171
Прочие параметры .....	172
Что дальше? .....	173
<b>Глава 16. Специальные эффекты CSS 3 .....</b>	<b>174</b>
Градиентные фоны .....	174
Введение в градиенты и градиентные фоны .....	174
Создание градиентных фонов .....	175
Линейный градиент .....	175
Радиальный градиент .....	177
Повторяющийся градиент .....	178
Преобразования .....	179
Как задаются преобразования и их параметры .....	179
Двухмерные преобразования .....	179
Смещение .....	179

Масштабирование .....	180
Наклон .....	181
Поворот.....	181
Трехмерные преобразования .....	182
Перспектива .....	182
Трехмерные преобразования .....	182
Точка зрения и ее местоположение.....	183
Скрытие обратной стороны элемента .....	184
Режим проецирования элементов-потомков .....	185
Позиционирование точки начала координат .....	187
Сложные преобразования .....	188
Анимация.....	188
Анимация с двумя состояниями .....	188
Простейшая анимация .....	189
Обратная анимация.....	191
Сложная анимация.....	192
Анимация с несколькими состояниями .....	193
Состояния анимации.....	193
Параметры анимации .....	194
Что дальше? .....	197
<b>Глава 17. Медиазапросы.....</b>	<b>198</b>
Классификация и использование медиазапросов .....	198
Медиазапросы HTML.....	199
Введение в медиазапросы HTML .....	199
Указатели медиазапросов .....	200
Разделители медиазапросов .....	202
Медиазапросы CSS .....	203
Некоторые соображения о целесообразности использования медиазапросов .....	204
Управление печатью страниц .....	204
Что дальше? .....	205
<b>ЧАСТЬ III. ПОВЕДЕНИЕ WEB-СТРАНИЦ. WEB-СЦЕНАРИИ .....</b>	<b>207</b>
<b>Глава 18. Язык JavaScript.....</b>	<b>209</b>
Основные понятия JavaScript.....	209
Типы данных JavaScript .....	211
Переменные.....	213
Именованые переменных.....	213
Объявление переменных .....	213
Операторы .....	214
Арифметические операторы .....	214
Оператор объединения строк.....	215
Операторы присваивания .....	215
Операторы сравнения .....	216
Логические операторы .....	217
Оператор получения типа <i>typeof</i> .....	218
Преобразование типов данных .....	218
Приоритет операторов.....	219

Сложные выражения .....	221
Блоки.....	221
Условные выражения .....	221
Условный оператор ? .....	222
Выражения выбора .....	222
Циклы.....	223
Цикл со счетчиком.....	223
Цикл с постусловием .....	224
Цикл с предусловием.....	225
Прерывание и перезапуск цикла.....	225
Функции.....	226
Объявление функций.....	226
Функции и переменные. Локальные переменные.....	227
Вызов функций.....	227
Присваивание функций. Функциональный тип данных .....	228
Массивы .....	229
Ссылки.....	231
Объекты.....	231
Понятия объекта и экземпляра объекта .....	232
Создание экземпляра объекта.....	232
Работа с экземпляром объекта.....	233
Встроенные объекты языка JavaScript .....	234
Объект <i>Object</i> и использование его экземпляров.....	236
Цикл по свойствам объекта .....	237
Комментарии JavaScript .....	238
Как Web-сценарии вставляются в Web-страницу .....	238
Что дальше? .....	240
<b>Глава 19. Доступ к элементам Web-страницы и управление ими .....</b>	<b>241</b>
Объектная модель документа .....	241
Доступ к странице и ее элементам .....	243
Доступ к странице.....	243
Доступ к элементам страницы .....	243
Доступ к ключевым элементам .....	243
Прямой доступ к элементу.....	243
Доступ по имени тега или стилевого класса. Коллекции.....	244
Доступ по селекторам CSS.....	245
Доступ к родителю и потомкам.....	246
Доступ через стандартные коллекции.....	248
Работа со страницей и ее элементами.....	248
Работа с параметрами страницы.....	248
Работа с параметрами элемента.....	249
Работа с основными параметрами.....	249
Работа с атрибутами тега и их значениями .....	252
Работа со стилями.....	252
Работа с содержимым элемента .....	254
Добавление нового содержимого .....	255
Добавление и удаление элементов страницы .....	255
Что дальше? .....	258



<b>Глава 20. Обработка событий</b> .....	<b>259</b>
Событие. Обработчик события.....	259
События, поддерживаемые страницей и ее элементами.....	260
Привязка обработчиков событий.....	261
Получение сведений о событии.....	263
Особые случаи обработки событий.....	265
Всплытие событий.....	265
Режим перехвата событий.....	266
Действие по умолчанию.....	267
Что дальше?.....	268
<b>Глава 21. Управление интерактивными и внедренными элементами</b> .....	<b>269</b>
Интерактивные элементы.....	269
Гиперссылки.....	269
Web-формы.....	270
Элементы управления.....	271
Внедренные элементы.....	277
Графические изображения.....	277
Аудио- и видеоролики.....	278
Что дальше?.....	283
<b>Глава 22. Работа с Web-обозревателем</b> .....	<b>284</b>
Окна Web-обозревателя.....	284
Интернет-адрес текущей страницы.....	288
Список истории Web-обозревателя.....	289
Параметры экрана.....	289
Сведения о Web-обозревателе.....	290
Стандартные диалоги и сообщения.....	291
Интервалы и тайм-ауты.....	292
Что дальше?.....	293
<b>Глава 23. AJAX. Регулярные выражения</b> .....	<b>294</b>
AJAX.....	294
Введение в AJAX.....	294
Программная реализация AJAX.....	295
Объект <i>XMLHttpRequest</i> .....	295
Отправка запроса.....	296
Получение результата.....	298
Формат JSON.....	300
AJAX-навигация.....	301
Недостаток AJAX.....	305
Регулярные выражения.....	305
Написание регулярных выражений.....	305
Работа с регулярными выражениями.....	308
Что дальше?.....	310

<b>ЧАСТЬ IV. СЕРВЕРНЫЕ ПРИЛОЖЕНИЯ. PHP. MYSQL. БИБЛИОТЕКА Yii .....</b>	<b>311</b>
<b>Глава 24. Web-сайт как программа. Серверные приложения .....</b>	<b>313</b>
Статические и динамические сайты .....	313
Как работают серверные приложения .....	314
Что дальше? .....	315
<b>Глава 25. Язык PHP .....</b>	<b>316</b>
Основные принципы, типы данных, переменные и операторы .....	316
Регулярные выражения .....	318
Сложные выражения .....	318
Функции .....	319
Массивы .....	320
Классы и объекты .....	321
Доступ к свойствам и методам объекта .....	321
Объявление классов .....	321
Наследование классов .....	322
Конструкторы и деструкторы .....	324
Модификаторы доступа .....	325
Статические свойства и методы .....	326
Константы класса .....	327
Принципы написания программного кода PHP .....	327
Что дальше? .....	328
<b>Глава 26. Сервер данных MySQL .....</b>	<b>329</b>
Реляционные базы данных .....	329
Введение в реляционные базы данных .....	329
Поля .....	330
Индексы и ключи .....	331
Связи .....	333
Язык SQL .....	334
Сервер данных MySQL .....	336
Типы данных, поддерживаемые MySQL .....	336
Атрибуты полей и индексов .....	337
Пользователи и их права .....	337
Что дальше? .....	339
<b>Глава 27. Введение в библиотеку Yii .....</b>	<b>340</b>
Два подхода к разработке сайтов .....	340
Основные понятия Yii-программирования .....	341
Модели, контроллеры, действия и шаблоны .....	341
Маршрутизация .....	343
Установка библиотеки Yii .....	344
Создание сайта .....	344
Структура папок Yii-сайта .....	345
Настройка сайта .....	346
Настройка соединения с базой данных .....	347
Прочие настройки .....	348

Средства прототипирования .....	349
Что дальше? .....	351
<b>Глава 28. Модели.....</b>	<b>352</b>
Требование к таблицам и моделям.....	352
Прототипирование моделей.....	353
Объявление класса модели .....	355
Основные методы класса модели .....	355
Задание надписей для элементов управления .....	356
Задание правил для значений полей.....	357
Сценарии Yii .....	362
Задание связей.....	363
Простейшие случаи создания связей .....	363
Задание параметров связи.....	365
Получение статистической информации .....	368
Расширение функциональности модели .....	369
Объявление дополнительных свойств и методов.....	369
Задание дополнительных действий для модели.....	370
Что дальше? .....	371
<b>Глава 29. Контроллеры и действия .....</b>	<b>372</b>
Требования к контроллерам и действиям.....	372
Прототипирование контроллеров.....	373
Прототипирование CRUD-контроллера.....	373
Прототипирование "пустого" контроллера .....	374
Объявление класса контроллера.....	375
Обработка данных в контроллерах .....	376
Получение данных от посетителя.....	376
Получение доступа к модели .....	377
Задание сценария для модели .....	377
Поиск записей .....	377
Получение значений полей записи.....	380
Получение связанных записей.....	380
Создание сложных запросов.....	381
Подготовительные действия .....	381
Простая выборка записей.....	382
Связывание таблиц .....	383
Получение статистических сведений о записях модели.....	383
Ограничение количества выбираемых записей.....	384
Выборка записей.....	385
Повторное использование сложного запроса.....	386
Использование пагинатора .....	386
Сложные запросы к модели .....	388
Получение сведений об интернет-запросе.....	389
Получение сведений о контроллере и действии.....	390
Получение сведений о сайте .....	391
Вывод данных .....	391
Вывод посредством шаблона.....	391

Вывод в формате JSON .....	393
Перенаправление .....	393
Что дальше? .....	394
<b>Глава 30. Шаблоны .....</b>	<b>395</b>
Требования к шаблонам .....	395
Основные принципы написания шаблонов .....	395
Средства Yii по созданию шаблонов .....	397
Генерирование фрагментов HTML-кода .....	397
Вывод значений с форматированием .....	400
Шаблоны разметки .....	402
Задание шаблона разметки .....	402
Простейший шаблон разметки .....	403
Вложенные шаблоны разметки .....	404
Подшаблоны .....	406
Использование виджетов .....	406
Требования к виджетам .....	407
Создание виджетов .....	407
Вызов виджета .....	408
Что дальше? .....	409
<b>Глава 31. Ввод данных .....</b>	<b>410</b>
Формы, основанные на моделях .....	410
Создание самих форм .....	410
Создание элементов управления .....	411
Создание полей ввода, области редактирования и регулятора .....	411
Создание флажков и переключателей .....	412
Создание списков .....	413
Создание групп переключателей и флажков .....	414
Создание скрытых полей .....	415
Имена элементов управления .....	415
Создание кнопок .....	416
Создание надписей .....	416
Вывод сообщений об ошибках .....	417
Дополнительные параметры форм и элементов управления .....	418
Инструменты моделей для ввода и правки данных .....	419
Инструменты высокого уровня .....	419
Создание записи .....	419
Правка записи .....	421
Удаление записи .....	422
Инструменты низкого уровня .....	423
Массовые правка и удаление записей .....	424
Обычные формы. Модели форм .....	425
Выгрузка и сохранение файлов .....	427
Подготовительные действия .....	427
Обработка выгруженного файла .....	428
Использование CAPTCHA .....	432
Что дальше? .....	435

<b>Глава 32. Разграничение доступа.....</b>	<b>436</b>
Как реализуется разграничение доступа.....	436
Настройка разграничения доступа.....	437
Модель списка пользователей.....	438
Класс сведений о пользователе.....	439
Указание прав доступа.....	441
Реализация входа на сайт.....	444
Получение сведений о пользователе.....	446
Реализация выхода с сайта.....	447
Что дальше?.....	448
<b>Глава 33. Маршрутизация.....</b>	<b>449</b>
Настройки маршрутизации.....	449
Написание привязок.....	450
Написание простейших привязок.....	450
Написание параметризованных привязок.....	451
Несколько слов о генерировании интернет-адресов.....	452
Удаление из интернет-адресов имени файла index.php.....	453
Что дальше?.....	454
<b>Глава 34. Кэширование.....</b>	<b>455</b>
Настройки кэширования.....	455
Реализация высокоуровневого кэширования.....	456
Кэширование фрагментов страниц.....	456
Условия устаревания кэшируемого фрагмента.....	458
Кэширование запросов.....	460
Кэширование целых страниц.....	461
Кэширование на стороне сервера.....	461
Управление кэшированием на стороне клиента.....	462
Кэширование на низком уровне.....	463
Что дальше?.....	465
<b>ЧАСТЬ V. РАЗРАБОТКА САЙТА — СВОДИМ ВСЕ ВОЕДИНО.....</b>	<b>467</b>
<b>Глава 35. Планирование и предварительные действия.....</b>	<b>469</b>
Планирование сайта.....	469
Основные этапы планирования сайта.....	469
Дизайн сайта.....	470
Логическая и физическая структуры сайта.....	471
Сайт электронных публикаций "СЭП".....	472
Создание сайта.....	472
База данных сайта.....	473
Прочие настройки.....	473
Что дальше?.....	475
<b>Глава 36. Создание дизайна страниц.....</b>	<b>476</b>
Особенности создания оформления для страниц.....	476
Страницы для традиционных компьютеров.....	477
Разметка.....	477

Начальное оформление .....	478
"Шапка" .....	479
Панель навигации .....	481
Блок основного содержимого .....	484
Параметры самого блока основного содержимого .....	484
Параметры текста .....	485
Параметры нумерации заголовков .....	486
Параметры внедренных элементов .....	487
Параметры форм и элементов управления .....	489
"Поддон" .....	491
Страницы для мобильных устройств .....	492
Разметка .....	493
"Шапка" .....	493
Блок основного содержимого .....	494
"Поддон" .....	495
Печатная версия страницы .....	496
Что дальше? .....	497
<b>Глава 37. Интерактивные элементы .....</b>	<b>498</b>
Спойлер .....	498
Формирование спойлера .....	498
Оформление спойлера .....	499
Программирование спойлера .....	501
Лайтбокс .....	503
Формирование лайтбокса .....	503
Оформление лайтбокса .....	504
Программирование лайтбокса .....	507
Блокнот .....	510
Формирование блокнота .....	510
Оформление блокнота .....	511
Программирование блокнота .....	513
Задание размеров видео .....	515
Что дальше? .....	516
<b>Глава 38. Статичные страницы .....</b>	<b>517</b>
Маршрутизация .....	517
Базовый класс контроллера. Определение обращения с мобильного устройства .....	518
Контроллер .....	520
Шаблоны .....	520
Шаблоны разметки .....	520
Шаблоны страниц .....	522
Виджет панели навигации .....	523
Тестирование мобильной версии сайта .....	525
Параметры кэширования .....	526
Что дальше? .....	528
<b>Глава 39. Список пользователей и разграничение доступа .....</b>	<b>529</b>
Настройки .....	529
Таблица базы данных .....	530

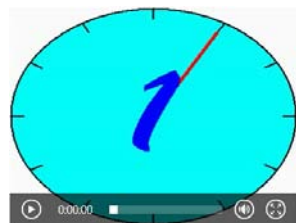
Модель.....	531
Вход и выход.....	532
Вход на сайт .....	532
Выход с сайта.....	533
Виджет панели навигации.....	533
Работа со списком пользователей.....	534
Список пользователей .....	534
Создание пользователя.....	537
Правка пользователя.....	538
Удаление пользователя.....	539
Разграничение доступа.....	540
Что дальше? .....	541
<b>Глава 40. Категории и подкатегории .....</b>	<b>542</b>
Маршрутизация.....	542
Таблицы базы данных .....	543
Модели.....	544
Базовый класс модели .....	544
Модель категорий.....	545
Модель подкатегорий.....	546
Виджет панели навигации.....	547
Вывод списков категорий и подкатегорий .....	549
Работа с категориями и подкатегориями.....	551
Автоматическое генерирование слогов.....	551
Страницы для работы с подкатегориями.....	552
Список подкатегорий .....	552
Создание и правка подкатегорий.....	553
Разграничение доступа.....	554
Кэширование .....	555
Что дальше? .....	557
<b>Глава 41. Статьи.....</b>	<b>558</b>
Маршрутизация.....	558
Таблица базы данных .....	559
Модель.....	559
Вывод статей.....	561
Вывод списка статей, относящихся к выбранной подкатегории .....	561
Вывод списка последних пяти статей, относящихся к выбранной категории.....	565
Поиск статей.....	567
Вывод статьи.....	570
Форматирование текста статей. BBCode .....	572
Набор тегов BBCode, поддерживаемых нашим сайтом .....	572
Собственно форматирование текстов статей .....	573
Страницы для работы со статьями.....	576
Разграничение доступа.....	577
Кэширование .....	578
Что дальше? .....	580

<b>Глава 42. Комментарии.....</b>	<b>581</b>
Маршрутизация.....	581
Таблица базы данных .....	581
Модель.....	582
Список комментариев, относящихся к выбранной статье .....	584
Страницы для работы с комментариями .....	587
Кэширование .....	589
Что дальше? .....	590
<b>Глава 43. Хранилище файлов .....</b>	<b>591</b>
Маршрутизация.....	591
Таблица базы данных .....	591
Модель.....	592
Контроллер.....	598
Шаблон .....	600
Оформление .....	603
Web-сценарий .....	605
Что дальше? .....	612
 <b>ЧАСТЬ VI. НАНЕСЕНИЕ ПОСЛЕДНИХ ШТРИХОВ И ПУБЛИКАЦИЯ САЙТА.....</b>	
<b>Глава 44. Программируемая графика HTML5 .....</b>	<b>615</b>
Канва.....	615
Контекст рисования.....	615
Рисование простейших фигур.....	616
Задание цвета, уровня прозрачности и толщины линий .....	616
Рисование сложных фигур.....	618
Как рисуются сложные контуры.....	618
Перо. Перемещение пера .....	618
Прямые линии .....	619
Дуги.....	619
Кривые Безье.....	620
Прямоугольники .....	621
Задание стиля линий.....	621
Вывод текста .....	623
Использование сложных цветов .....	624
Линейный градиент .....	624
Радиальный градиент.....	626
Графический цвет .....	627
Вывод внешних изображений.....	628
Создание тени у рисуемой графики .....	629
Преобразования системы координат .....	629
Сохранение и загрузка состояния.....	630
Перемещение начала координат канвы .....	630
Поворот системы координат.....	631
Изменение масштаба системы координат .....	632
Управление наложением графики.....	633



Создание маски .....	634
Что дальше? .....	635
<b>Глава 45. Хранение данных на стороне клиента .....</b>	<b>636</b>
Хранилище HTML5 .....	636
Временное хранение текста статей на стороне клиента .....	637
Что дальше? .....	639
<b>Глава 46. Публикация сайта .....</b>	<b>640</b>
Подготовка сайта к публикации .....	640
Указание окончательных настроек .....	640
Удаление ненужных файлов .....	641
Создание страницы сообщений об ошибках .....	642
Публикация сайта .....	643
<b>Заключение .....</b>	<b>645</b>
<b>ПРИЛОЖЕНИЯ .....</b>	<b>647</b>
<b>Приложение 1. Установка и настройка пакета Open Server .....</b>	<b>649</b>
Установка .....	649
Запуск и управление серверами .....	651
Настройка .....	651
<b>Приложение 2. Использование программы phpMyAdmin для работы с базами данных MySQL .....</b>	<b>653</b>
Запуск и вход .....	653
Работа с пользователями .....	654
Создание пользователя и базы данных .....	654
Правка и удаление пользователей .....	656
Работа с таблицами .....	656
Создание таблиц .....	656
Правка таблиц .....	658
Удаление таблиц .....	659
Работа с индексами .....	660
Создание индексов .....	660
Правка и удаление индексов .....	660
Работа со связями .....	661
Работа с содержимым таблиц .....	663
Перенос содержимого из одной базы данных в другую .....	664
Экспорт данных .....	664
Импорт данных .....	665
Выход .....	666
<b>Приложение 3. Использование утилиты SUPER для перекодирования аудио- и видеофайлов .....</b>	<b>667</b>
<b>Приложение 4. Описание электронного архива .....</b>	<b>671</b>
<b>Предметный указатель .....</b>	<b>673</b>

# ГЛАВА 1



## Современный Web-дизайн. Web 2.0

Всемирная паутина, WWW (World Wide Web), Web-дизайн, Web-сайт, Web-страница — все знают, что это такое. Но что такое современная Всемирная паутина, современный Web-дизайн и современная Web-страница? Именно с ответов на все эти вопросы начнется данная книга.

### Современный Web-дизайн. Концепция Web 2.0

Раньше доступ в Интернет можно было получить только с компьютеров. Потом в Интернет стали выходить с мобильных телефонов. Сейчас к Сети подключились мультимедийные плееры, устройства чтения электронных книг, появившиеся недавно планшетные компьютеры и "умные" телевизоры. А завтра — кто знает; может быть, мы будем выходить на Web-сайты с холодильника или пылесоса...

"Я буду везде, — заявляет Интернет. — Я стану вездесущим. Все готовьтесь к моему приходу!"

### Что требуется от современного Web-сайта

Будем готовиться... Но что нам, как будущим Web-дизайнерам, для этого следует сделать? Соблюсти три несложных правила.

1. Строго соблюдать все интернет-стандарты.
2. Тщательно продумать наполнение Web-страниц.
3. Позаботиться о доступности Web-страниц.

Рассмотрим их подробнее.

Интернет грозит прийти на самые разные устройства, которые могут быть основаны на различных аппаратных и программных платформах, зачастую сильно отличающихся друг от друга. Так, персональные компьютеры построены на аппаратной платформе Intel и программной платформе Microsoft Windows (по крайней

мере, большинство). Мобильный телефон и планшет автора работают под управлением операционной системы Google Android. А на подходе — уже так называемые "интернет-вещи", "умные вещи", обладающие возможностью подключаться к Сети, и на каких платформах они будут работать, пока неизвестно...

Одно объединяет все это аппаратно-программное многообразие — строгое соблюдение интернет-стандартов. Устройства, не соблюдающие эти стандарты, в лучшем случае будут отображать Web-страницы неправильно, в худшем — вообще не будут работать.

Из этого следует *первое правило* из перечисленных ранее — Web-дизайнеры при создании Web-страниц обязаны строго придерживаться современных интернет-стандартов, чтобы их творения одинаково (ну, или почти одинаково) отображались на всех устройствах.

Первое правило также требует отказа от устаревших и закрытых, фирменных интернет-технологий. С устаревшими технологиями все понятно: старье — не помощник новому. Закрытые же технологии неудобны тем, что зачастую контролируются единственной фирмой, которая единолично "заказывает музыку" и далеко не всегда прислушивается к мнению интернет-сообщества. К таким технологиям относится, в частности, до сих пор популярная, но постепенно сдающая позиции Adobe Flash.

Открытыми интернет-стандартами, в том числе и Web-стандартами, занимается организация World Wide Web Consortium (Консорциум Всемирной паутины), или сокращенно W3C. Она разрабатывает стандарты, согласует их с требованиями участников рынка и публикует на своем Web-сайте <http://www.w3.org>. Все опубликованные там стандарты обязательны к применению всеми разработчиками устройств, интернет-программ и Web-страниц.

Интернет когда-то начинался как сеть ученых, которым было нужно обмениваться результатами исследований. А что представляли собой эти результаты? В основном, текст, возможно, с иллюстрациями. Ученые — публика в этом смысле невзыскательная, им вполне хватало скромных возможностей тогдашнего WWW.

Теперь же абсолютное большинство пользователей Интернета — обычные обыватели. Им мало простого текста с парой картинок, им подавай хорошо оформленный текст, музыку и видео. Они требовательнее первых обитателей Сети.

Отсюда вытекает *второе правило* — Web-дизайнеры должны заботиться о полноте и удобстве наполнения страниц.

- ◆ Структура Web-страниц должна быть хорошо продумана, чтобы посетитель сразу смог найти на них все, что ему нужно.
- ◆ Web-страницы должны легко читаться и не "резать" глаза.
- ◆ К важным материалам желательно привлечь внимание посетителя, а маловажные скрыть. В этом могут помочь динамические элементы: раскрывающиеся при щелчке мышью абзацы (спойлеры), гиперссылки, выделяющиеся при наведении курсора мыши, раскрывающиеся меню и пр.

- ◆ Если сайт посвящен музыке или видео, все это должно быть доступно для воспроизведения прямо на его страницах, без загрузки как собственно аудио- или видеофайла, так и какой-либо дополнительной программы.
- ◆ Одним словом — все для удобства посетителя! (Пожалуй, это правило следовало бы поставить в начале списка...)

Интернет грозит прийти на самые разные устройства с различными характеристиками: быстродействием процессора, объемом памяти, разрешением экрана, скоростью доступа к Сети. Но все они должны обеспечивать единообразный вывод Web-страниц. Как этого достигнуть?

Вот и *третье правило* — Web-дизайнеры должны заботиться о доступности страниц.

- ◆ Web-страницы следует делать как можно более компактными. Чем компактнее файл, тем быстрее он загружается по сети — это аксиома.
- ◆ Web-страницы не должны быть чересчур сложными. Чем сложнее страница, тем больше времени и системных ресурсов требует ее обработка и вывод.
- ◆ Web-страницы не должны требовать для отображения никакого дополнительного программного обеспечения. В идеале для их вывода должно быть достаточно только Web-обозревателя.
- ◆ Наконец, Web-страницы должны автоматически адаптироваться под устройства с различными параметрами экрана.

Но как эти правила реализуются на практике? Давайте откроем какой-нибудь современный Web-сайт, например, принадлежащий организации W3C (рис. 1.1). Как мы помним, его можно найти по интернет-адресу <http://www.w3.org>.

Что же мы здесь видим?

- ◆ Web-сайт создан с учетом всех современных интернет-стандартов. Он отображается во всех Web-обозревателях практически одинаково.
- ◆ Web-сайт не использует ни устаревших, ни закрытых интернет-технологий.
- ◆ Структура страниц исключительно ясна — мы можем без проблем найти все, что нужно. Слева находится набор гиперссылок, ведущих на другие страницы сайта (панель навигации), посередине — список новостей и гиперссылки на избранные статьи, справа — гиперссылки на дополнительные материалы.
- ◆ Web-страница прекрасно читается. Тонкий шрифт без засечек, спокойная серо-голубая цветовая гамма, тонкие рамочки, минимум графики — ничто не бросается в глаза.
- ◆ Web-страница быстро загружается и мгновенно выводится на экран.
- ◆ Web-страница ничего не требует для своего вывода, кроме Web-обозревателя.

Налицо и соблюдение стандартов, и наполнение, и доступность. Три из трех!

Именно такие страницы мы и будем учиться создавать в данной книге.

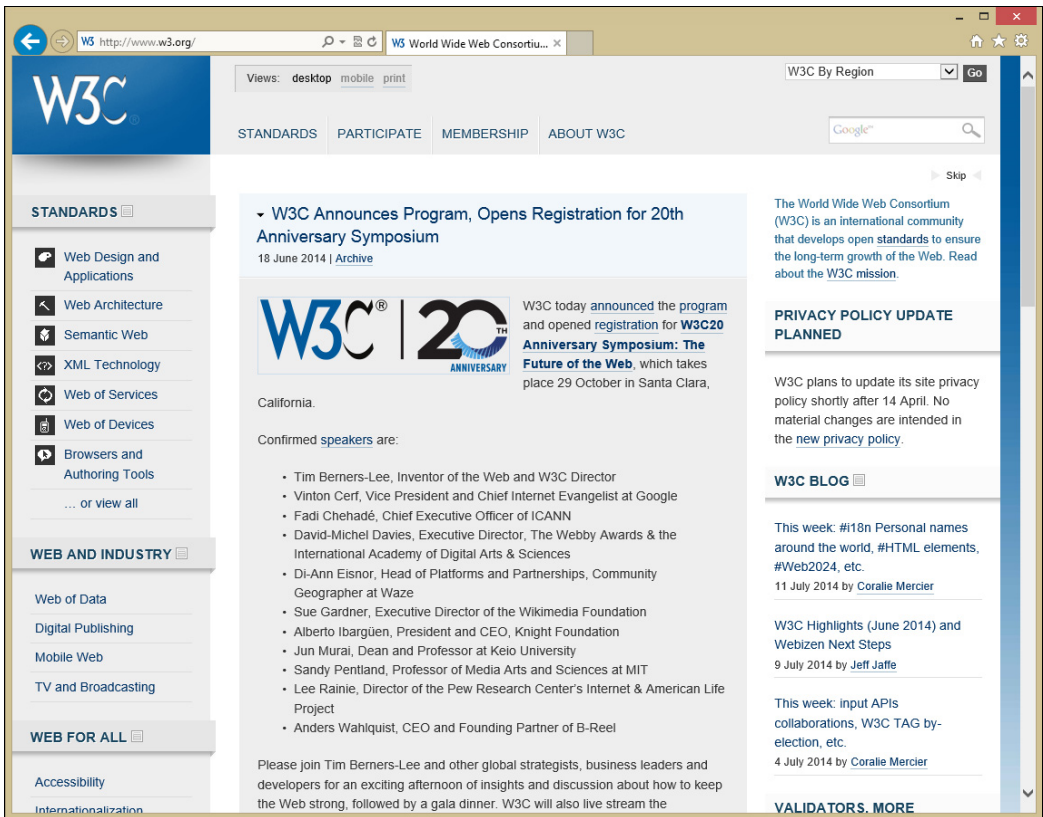


Рис. 1.1. Главная страница Web-сайта организации W3C

## Концепция Web 2.0

Давайте еще раз обратимся к рассмотренным ранее правилам и немного расширим их.

- ◆ При создании Web-страниц следует придерживаться современных интернет-стандартов. При этом нужно полностью отказаться от устаревших и закрытых интернет-технологий, как не укладывающихся в современную парадигму Web-дизайна и зачастую не поддерживаемых всеми Web-обозревателями.
- ◆ Особое внимание нужно обратить на структуру и наполнение Web-страниц. Структура страниц должна быть максимально простой, а наполнение — достаточно богатым, чтобы посетитель быстро нашел нужную ему информацию. Кроме того, необходимо создавать страницы так, чтобы дизайн не мешал восприятию информации.
- ◆ Web-страницы обязательно следует делать максимально доступными на любых устройствах. Страницы должны быстро загружаться и выводиться на экран. Страницы должны нормально отображаться на экранах с различными характеристиками. Наконец, страницы не должны требовать для отображения никакого дополнительного программного обеспечения.

Фактически здесь мы привели постулаты так называемой концепции *Web 2.0*. Это список правил, которым должен удовлетворять любой Web-сайт, претендующий на звание современного. Образно выражаясь, это флаг, который совместно несут труженики Web-индустрии, шагая в ногу со временем.

Также концепция Web 2.0 предусматривает четыре принципа, являющиеся "передним краем" Web-дизайна. Давайте их рассмотрим.

Принцип первый — разделение содержимого, представления и поведения Web-страницы. Здесь *содержимое* — это информация, которая выводится на Web-странице, *представление* описывает формат вывода этой информации, а *поведение* — реакцию Web-страницы или отдельных ее элементов на действия посетителя. Благодаря их разделению мы сможем править, скажем, содержимое, не затрагивая представление и поведение, или поручать создание содержимого, представления и поведения разным людям.

Принцип второй — *подгружаемое содержимое*. Вместо того чтобы обновлять всю Web-страницу в ответ на щелчок на гиперссылке, мы можем подгружать только ее часть, содержащую необходимую информацию. Это заметно: время загрузки страницы и объем передаваемой по сети информации (сетевой трафик) позволят выполнять какие-либо действия с данными после их подгрузки.

Принцип третий — *адаптируемое содержимое*. Страницы должны сами подстраиваться под экраны с различными параметрами. Так, на обычном компьютерном мониторе может выводиться полная версия страницы, а на экране телефона — ее упрощенная версия, на которой некоторые маловажные элементы скрыты.

Принцип четвертый — *семантическая разметка* данных. С ее помощью мы можем дать определенным фрагментам страницы какое-либо особое значение. Например, мы можем пометить часть страницы — рисунок и подпись к нему — как иллюстрацию, и тогда Web-обозреватель сможет выделить ее определенным образом, основываясь на заданном нами представлении.

В качестве примеров можно привести два сайта. Во-первых, сайт популярной социальной сети "ВКонтакте" (<http://vk.com/>). Вы, вероятно, замечали, что, стоит прокрутить страницу вниз, как на ней появляются более ранние записи — это работает механизм подгрузки данных, полученных от Web-приложения (о которых мы поговорим в *главе 24*). Во-вторых, это сайт интернет-энциклопедии Википедия (<http://ru.wikipedia.org/>) — его страницы успешно подстраиваются под самые разные экраны.

Очень многие сайты в настоящее время следуют концепции Web 2.0, выполняя если не все ее принципы, то большую их часть. И неудивительно, ведь это позволяет радикально сократить время разработки, повысить скорость загрузки, дать страницам дополнительную функциональность и без проблем адаптировать их для отображения на мобильных устройствах.

Да-да, Web 2.0 — это не самоцель, а всего лишь средство достижения задач, поставленных перед Web-дизайнерами и перечисленных в самом начале этой главы. Хотите разрабатывать современные, "легкие" и доступные для всех устройств сайты? Следуйте данной концепции. И читайте эту книгу!

## Интернет: как все это работает

Давайте еще раз посмотрим на Web-сайт организации W3C. И зададимся вопросом, вынесенным в заголовок данного раздела.

Как все это работает? Откуда Web-обозреватель получает нужную Web-страницу? Кто отвечает за работу сложного механизма под названием Всемирная паутина?

### Клиенты и серверы Интернета. Интернет-адреса

Возьмем для примера главную страницу сайта, который мы открыли. Она должна где-то храниться. Но где? На диске другого компьютера, подключенного к сети (в данном случае — к сети Интернет). Этот компьютер может принадлежать как автору Web-сайта, так и сторонней организации, предоставляющей доступ в Интернет (*интернет-провайдеру*) или площадку для публикации сайтов клиентов (*хостинг-провайдеру*). И хранится она в виде файла или набора файлов, таких же, какие в изобилии "водятся" на нашем собственном компьютере.

Но как мы смогли получить и просмотреть содержимое этого файла? Во-первых, посредством самой сети — она связала компьютер, хранящий файл, с нашим. Во-вторых, с помощью особых программ, которые, собственно, и выполнили передачу файла. Эти программы делятся на две группы.

Программы первой группы взаимодействуют непосредственно с пользователем: принимают от него запросы на информацию, которая хранится где-то в сети, получают ее, выводят на экран и, возможно, позволяют ее править и отправлять обратно. Такие программы называют *клиентами*.

Для просмотра страниц мы пользуемся Web-обозревателем. Это программа-клиент; она принимает от нас интернет-адреса страниц, получает файлы, хранящие их содержимое, и выводит это содержимое на экран. Программа почтового клиента позволяет как извлекать из почтового ящика полученные письма, так и создавать новые. Существуют также клиенты чата, систем мгновенных сообщений и пр.

Но клиенты не имеют прямого доступа к хранящейся на других компьютерах информации. Они не могут просто "залезть" на жесткий диск удаленного компьютера и прочесть оттуда файл. Так сделано из соображений безопасности. Вместо этого они отправляют запросы программам второй группы — серверам.

*Серверы* работают на компьютерах, хранящих информацию, которая должна быть доступна в сети. Они принимают запросы от клиентов, извлекают требуемую информацию из файлов и отправляют им. Также они могут получать введенную пользователями информацию от клиентов и сохранять их в файлах, при этом, возможно, как-то обработав. Можно сказать, что серверы выступают посредниками между клиентами и запрашиваемой ими информацией.

Для управления Web-сайтами используются *Web-серверы*, которые принимают запросы от клиентов и отправляют им содержимое требуемых файлов. Для управления почтовыми службами применяются серверы электронной почты; они сохраняют пришедшие письма в файлах, выдают их почтовым клиентам по запросу, при-

нимают от клиентов новые сообщения и отправляют их по указанному адресу — в общем, работают как своего рода почтовое отделение. Службы чатов и мгновенных сообщений также имеют свои серверы.

Клиенты — лицо Интернета. Серверы — его сердце.

Но как указать, какая информация и с какого сервера нам требуется? С помощью определенным образом составленного интернет-адреса.

Каждая единица информации — файл, ящик электронной почты, канал чата, — доступная в сети, однозначно идентифицируется интернет-адресом, который представляет собой строку из букв, цифр и некоторых других символов. Этот интернет-адрес включает в себя две части:

- ◆ интернет-адрес программы-сервера, работающей на компьютере;
- ◆ указатель на нужную единицу информации, например, путь к файлу, имя ящика электронной почты, имя канала чата и др. (может отсутствовать).

Рассмотрим несколько примеров интернет-адресов.

В интернет-адресе **http://www.somesite.ru/folder1/file1.htm** присутствуют обе части. Здесь **http://www.somesite.ru** — интернет-адрес программы-сервера (или, как еще говорят, *имя хоста*), в данном случае — Web-сервера, а **/folder1/file1.htm** — путь к запрашиваемому файлу.

В интернет-адресе **http://www.thersite.ru** присутствует только интернет-адрес Web-сервера. Какая информация в этом случае будет отправлена клиенту (Web-обозревателю), мы узнаем потом.

А в интернет-адресе **user@mail.someserver.ru** мы видим интернет-адрес сервера электронной почты (**mail.someserver.ru**) и имя почтового ящика (**user**).

Разговор об интернет-адресах еще не закончен. Мы вернемся к нему в *главе 7*, когда будем рассматривать средства навигации по Web-сайту, в частности, гиперссылки. А пока что давайте подробнее поговорим о Web-серверах и их нелегкой "работе".

## Web-сайты и Web-серверы

Как мы только что выяснили, все интернет-программы делятся на клиенты и серверы. Клиенты работают на стороне пользователя, получают от них интернет-адреса и выводят им полученную с этих адресов информацию. Серверы принимают запросы от клиентов, находят запрашиваемую ими информацию на дисках серверных компьютеров и отправляют ее клиентам.

Во Всемирной паутине WWW в качестве клиентов используются Web-обозреватели, а в качестве серверов — Web-серверы. Это мы тоже знаем.

Любая информация на дисках компьютера хранится в файлах. Ну, это знает любой более-менее подкованный пользователь...

Web-страницы также хранятся в файлах с расширением htm или html (или, с учетом описанных во введении типографских соглашений, htm[1]). Одна Web-страница занимает один или более файлов.



А теперь — внимание! Мы рассмотрим некоторые "интимные" подробности работы Web-серверов, которые знает не каждый интернетчик.

Прежде всего, для хранения всех файлов, составляющих Web-сайт, на диске серверного компьютера выделяется особая папка, называемая *корневой папкой Web-сайта*. Путь к этой папке указывается в настройках Web-сервера, чтобы он смог ее "найти".

Все, повторим — все файлы, составляющие Web-сайт, должны храниться в корневой папке или в папках, вложенных в нее. Файлы, расположенные вне корневой папки, с точки зрения Web-сервера не существуют. Это сделано для безопасности, чтобы злоумышленник не смог получить доступ к дискам серверного компьютера.

Когда в интернет-адресе указывается путь к запрашиваемому файлу, Web-сервер отсчитывает его относительно корневой папки. Это проще всего показать на примерах.

- ◆ **<http://www.somesite.ru/page1.htm>** — в ответ будет отправлен файл `page1.htm`, хранящийся в корневой папке сайта.
- ◆ **<http://www.somesite.ru/chapter2/page6.htm>** — в ответ будет отправлен файл `page6.htm`, хранящийся в папке `chapter2`, которая вложена в корневую папку сайта.
- ◆ **<http://www.somesite.ru/downloads/others/archive.zip>** — в ответ будет отправлен файл `archive.zip`, хранящийся в папке `others`, вложенной в папку `downloads`, которая, в свою очередь, вложена в корневую папку сайта.

Но ведь мы нечасто набираем интернет-адрес, включающий путь к запрашиваемому файлу. Гораздо чаще интернет-адреса включают только адрес программы-сервера, например, **<http://www.somesite.ru>**. Что в таком случае делает Web-сервер? Какой файл он отправляет в ответ?

Специально для этого предусмотрены так называемые *Web-страницы по умолчанию*. Такая Web-страница выдается клиенту, если он указал в интернет-адресе только путь к файлу, но не его имя. Обычно файл Web-страницы по умолчанию имеет имя `default.htm[1]` или `index.htm[1]`, хотя его можно изменить в настройках Web-сервера.

Так, если мы наберем интернет-адрес **<http://www.somesite.ru>**, Web-сервер вернет нам файл страницы по умолчанию, хранящийся в корневой папке сайта. Практически всегда это будет *главная Web-страница* — та, с которой начинается "путешествие" по сайту.

Мы можем набрать и интернет-адрес вида **<http://www.somesite.ru/chapter2/>**. Тогда Web-сервер отправит нам файл страницы по умолчанию, хранящийся в папке `chapter2`, вложенной в корневую папку сайта.

### **ВНИМАНИЕ!**

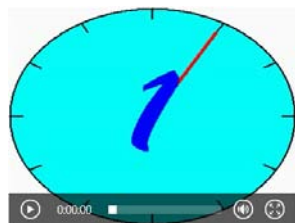
Если Web-сервер не обнаружит указанный файл страницы, он вернет Web-обозревателю сообщение об ошибке с кодом 404 — "запрошенная страница отсутствует". Такое сообщение говорит о проблемах в работе сайта, и эта проблема должна быть устранена.

## Что дальше?

В этой главе мы познакомились с современными веяниями в Web-дизайне, узнали о концепции Web 2.0 и поняли, как работает Интернет в целом и WWW в частности. В общем, узнали самые основы.

Теперь на какое-то время мы оставим в покое принципы и концепции и сосредоточимся на практике, а именно на языке HTML. В следующей главе мы рассмотрим основные его понятия: теги, их атрибуты и вложенность и структурирование страниц.

## ГЛАВА 2



# Введение в язык HTML5

В предыдущей главе мы ознакомились с современными тенденциями Web-дизайна, концепцией Web 2.0 и принципами работы Интернета. Все эти тенденции, концепции и принципы не отняли у нас много времени...

...Чего не скажешь о языке разметки страниц HTML. Ему будет посвящена вся первая часть книги.

## Основные принципы HTML

Язык *HTML* (HyperText Markup Language, язык гипертекстовой разметки) предназначен для описания содержимого Web-страниц, т. е. той информации, которая, собственно, будет выводиться на экран: абзацев, заголовков, списков, таблиц, графических изображений, гиперссылок, аудио и видео.

Первая версия HTML была представлена еще в 1992 году. В настоящее время в разработке находится пятая версия этого языка — HTML5, — и, хоть окончательная ее спецификация до сих пор не выпущена, все современные Web-обозреватели уже поддерживают, по крайней мере, большинство предлагаемых ею возможностей. Эту версию мы и будем использовать для написания наших страничек.

### ***ВНИМАНИЕ!***

Поддержка HTML5 появилась в Microsoft Internet Explorer, начиная с версии 9, в Mozilla Firefox — начиная с версии 3.5, в Opera — с версии 10, в Apple Safari — с версии 4, в Google Chrome — с самых первых версий.

Язык HTML5 довольно прост и основан на нескольких совсем не сложных принципах. Сейчас мы в этом убедимся.

## Первая Web-страница

Изучать HTML лучше всего на примере. Так что давайте сразу же создадим нашу первую Web-страничку. Сделать это можно в любом простейшем текстовом редакторе, например, входящем в комплект поставки Windows Блокноте.

Откроем Блокнот и наберем в нем текст (или, как говорят бывалые программисты, код), приведенный в листинге 2.1.

**Листинг 2.1**

```
<!doctype html>
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
    <title>Пример Web-страницы</title>
  </head>
  <body>
    <h1>Изучаем язык HTML</h1>
    <p>Язык <strong>HTML</strong> предназначен для создания содержимого
    Web-страниц.</p>
  </body>
</html>
```

Проверим набранный код на ошибки и сохраним в файл с именем 2.1.html. Только сделаем при этом две важные вещи.

1. Сохраним HTML-код в кодировке UTF-8. Для этого в диалоговом окне сохранения файла Блокнота найдем раскрывающийся список **Кодировка** и выберем в нем пункт **UTF-8**.
2. Заклучим имя файла в кавычки. Иначе Блокнот добавит к нему расширение txt, и наш файл получит имя 2.1.html.txt.

Все, наша первая Web-страница готова! Теперь осталось открыть ее в любом Web-обозревателе, поддерживающем язык HTML5 (автор применил Internet Explorer 10), и посмотреть на результат (рис. 2.1).

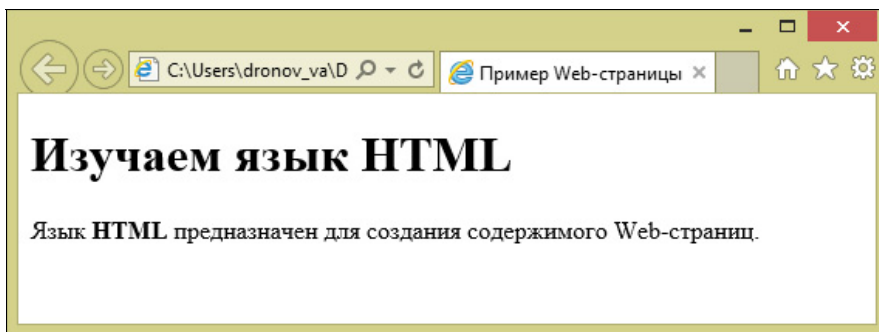


Рис. 2.1. Наша первая Web-страница

Видите? Мы создали Web-страницу, содержащую большой "кричащий" заголовок, абзац текста, который включает в свой состав фрагмент текста, выделенный полужирным шрифтом (аббревиатура "HTML"). И все это — в "голом" тексте, набранном в Блокноте!

## Теги и атрибуты тегов

Теперь посмотрим, что же мы такое написали в файле 1.1.htm. Пока что ограничимся следующим небольшим фрагментом HTML-кода:

```
<h1>Изучаем язык HTML</h1>
<p>Язык <strong>HTML</strong> предназначен для создания содержимого
Web-страниц.</p>
```

Здесь мы видим текст заголовка и абзаца. И еще странные слова, взятые в угловые скобки — символы < и >. Что это такое?

Это *теги* HTML, особые команды, задающие назначение того или иного фрагмента содержимого — будет он абзацем, заголовком или важным текстом, который следует выделить полужирным шрифтом.

Начнем с тегов <h1> и </h1>, поскольку они идут первыми. Эти теги превращают фрагмент текста, находящийся между ними, в заголовок. Тег <h1> помечает начало фрагмента, на который распространяется действие тега, и называется *открывающим*. А тег </h1> устанавливает конец "охватываемого" фрагмента и называется *закрывающим*. Что касается самого фрагмента, заключенного между открывающим и закрывающим тегами, то он называется *содержимым тега*. Именно к содержимому применяется действие тега.

Все теги HTML представляют собой символы < и >, внутри которых находится *имя тега*, определяющее назначение тега. (Имена тегов можно набирать как строчными, так и прописными буквами, но в HTML5 обычно используются строчные буквы.) Закрывающий тег должен иметь то же имя, что и открывающий; единственное отличие закрывающего тега — символ /, который ставится между символом < и именем тега.

Рассмотренные нами теги <h1> и </h1> в HTML фактически считаются одним тегом <h1>. Такой тег называется *парным*.

Другой парный тег — <p> — создает на Web-странице абзац из текста, являющегося содержимым этого тега. Такой абзац будет отображаться с отступами сверху и снизу. Если он полностью помещается по ширине в окне Web-обозревателя, то отобразится в одну строку; в противном случае сам Web-обозреватель разобьет его на несколько более коротких строк. (Это же справедливо и для заголовка.)

Третий парный тег — <strong> — помечает свое содержимое как важный текст, на который следует обратить особое внимание и который выделяется полужирным шрифтом. Как мы видим, этот тег вложен внутрь содержимого тега <p>. Это значит, что содержимое тега <strong> будет отображаться как часть абзаца (тега <p>).

А теперь рассмотрим вот такой тег:

```
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
```

Это *одинарный тег*, не имеющий закрывающей пары. Такие теги действуют в той точке HTML-кода, где они находятся, и либо задают какие-либо сведения о самой странице (как только что рассмотренный нами тег <meta>), либо помещают в соответствующее место страницы какой-либо элемент, не относящийся к тексту.

А еще мы видим, что в данном теге между символами < и >, помимо имени тега, присутствуют еще какие-то данные. Это *атрибуты тега*, задающие его параметры. В частности, атрибуты `http-equiv` и `content` тега `<meta>` указывают тип документа и его кодировку.

Каждый атрибут тега имеет *имя*, за которым ставится знак равенства и *значение* данного атрибута, взятое в двойные кавычки. Так, атрибут с именем `http-equiv` имеет значение `"Content-Type"`, указывающее, что данный тег задает тип документа. А атрибут с именем `content` имеет значение `"text/html; charset=utf-8"`, обозначает тип документа `"Web-страница"` и указывает, что он набран в кодировке UTF-8.

Атрибуты тегов бывают обязательными и необязательными. *Обязательные* атрибуты должны присутствовать в теге в обязательном порядке. *Необязательные* же атрибуты могут быть опущены; в таком случае тег ведет себя так, будто соответствующему атрибуту присвоено значение по умолчанию. Атрибуты `http-equiv` и `content` тега `<meta>` являются обязательными.

## Вложенность тегов

Если мы снова посмотрим на приведенный в листинге 2.1 HTML-код, то заметим, что одни теги вложены в другие. Так, тег `<strong>` вложен в тег `<p>`, являясь частью его содержимого. Тег `<p>`, в свою очередь, вложен в тег `<body>`, а тот — в тег `<html>`. (Теги `<body>` и `<html>` мы рассмотрим чуть позже.) Такая *вложенность тегов* в HTML — обычное явление.

Когда Web-обозреватель встречает тег, вложенный в другой тег, он как бы накладывает действие "внутреннего" тега на эффект "внешнего". Так, действие тега `<strong>` будет наложено на действие тега `<p>`, и фрагмент абзаца окажется выделенным полужирным шрифтом, при этом оставаясь частью этого абзаца.

Теперь — внимание! Порядок следования закрывающих тегов должен быть обратным тому, в котором следуют теги открывающие. Говоря иначе, теги со всем их содержимым должны полностью вкладываться в другие теги, не оставляя "хвостов" снаружи.

Осталось выучить несколько новых терминов. Тег, в который непосредственно вложен данный тег, называется *родительским*, или *родителем*. В свою очередь, тег, вложенный в данный тег, называется *дочерним*, или *потомком*. Так, для тега `<p>` в приведенном ранее листинге тег `<body>` — родительский, а тег `<strong>` — дочерний. Любой тег может иметь сколько угодно дочерних тегов, но только один родительский (что, впрочем, понятно — не может же он быть непосредственно вложен одновременно в два тега).

Аналогично, элемент Web-страницы, в который вложен элемент, создаваемый данным тегом, называется *родительским*, или *родителем*. А элемент Web-страницы, который вложен в данный элемент, — *дочерним*, или *потомком*.

*Уровень вложенности* того или иного тега показывает количество тегов, в которые он последовательно вложен. Так, если принять за точку отсчета тег `<html>`, то тег