

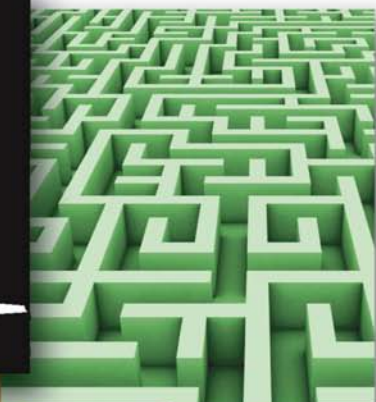
Н. А. Прохоренок, В. А. Дронов



HTML, JavaScript, PHP и MySQL

Джентльменский набор Web-мастера

4-е издание



HTML 5 и CSS 3
Web-СЕРВЕР Apache
АУДИО И ВИДЕО
АНИМАЦИЯ
ПРЕОБРАЗОВАНИЯ
AJAX
ГЕОЛОКАЦИЯ
ЛОКАЛЬНОЕ ХРАНИЛИЩЕ
ДАННЫХ
ТРАНЗАКЦИИ MySQL
ПРИМЕРЫ И СОВЕТЫ
ИЗ ПРАКТИКИ

PRO
ПРОФЕССИОНАЛЬНОЕ
ПРОГРАММИРОВАНИЕ



Материалы
на www.bhv.ru

Николай Прохоренок
Владимир Дронов

**HTML, JavaScript,
PHP и MySQL**
**Джентльменский набор
Web-мастера**
4-е издание

Санкт-Петербург
«БХВ-Петербург»

2015

УДК 004.43+004.738.5
ББК 32.973.26-018.1
П84

Прохоренок, Н. А.

П84 HTML, JavaScript, PHP и MySQL. Джентльменский набор Web-мастера. — 4-е изд., перераб. и доп. / Н. А. Прохоренок, В. А. Дронов. — СПб.: БХВ-Петербург, 2015. — 768 с.: ил. — (Профессиональное программирование)
ISBN 978-5-9775-3130-6

Рассмотрены вопросы создания интерактивных Web-сайтов с помощью HTML, JavaScript, PHP и MySQL, форматирования Web-страниц при помощи CSS. Даны основы PHP и примеры написания типичных сценариев. Описаны приемы работы и администрирования баз данных MySQL при помощи PHP и программы phpMyAdmin. Особое внимание уделено созданию программной среды на компьютере разработчика и настройке Web-сервера Apache. Приведено описание текстового редактора Notepad++, шаблонизатора Smarty и прочих программ (Aptana Studio, NetBeans и HeidiSQL), необходимых Web-разработчику.

В 4-м издании содержится описание возможностей, предлагаемых HTML 5 (средства семантической разметки и размещения аудио и видео) и CSS 3 (градиенты, создание тени, анимация и преобразования), технологии AJAX, формата JSON, новых инструментов JavaScript (включая средства геолокации и локальное хранилище данных) и всех нововведений, появившихся в актуальных на данный момент версиях Apache, PHP и MySQL.

Электронный архив содержит листинги примеров, руководство по созданию динамического сайта, самоучитель языка Perl, руководство по публикации сайта, инструкции по установке дополнительных программ и видеоуроки.

Для Web-разработчиков

УДК 004.43+004.738.54
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>Леонид Кочин</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 30.04.15.
Формат 70×100¹/₁₆. Печать офсетная. Усл. печ. л. 61,92.
Тираж 2000 экз. Заказ №
"БХВ-Петербург", 191036, Санкт-Петербург, Гончарная ул., 20.

Первая Академическая типография "Наука"
199034, Санкт-Петербург, 9 линия, 12/28

ISBN 978-5-9775-3130-6

© Прохоренок Н. А., Дронов В. А., 2015
© Оформление, издательство "БХВ-Петербург", 2015

Оглавление

Введение	1
Глава 1. Основы HTML. Создаем дизайн сайта.....	5
1.1. Основные понятия	5
1.2. Первый HTML-документ	8
1.3. Структура документа.....	10
1.3.1. Раздел <i>HEAD</i> . Техническая информация о документе	11
1.3.2. Раздел <i>BODY</i> . Основная часть документа.....	13
1.4. Форматирование отдельных символов	14
1.4.1. Выделение фрагментов текста	15
1.4.2. Создание нижних и верхних индексов	15
1.4.3. Вывод текста заданным шрифтом	16
1.5. Форматирование документа	17
1.5.1. Тег комментария	17
1.5.2. Перевод строки.....	18
1.5.3. Горизонтальная линия	18
1.5.4. Заголовки	19
1.5.5. Разделение на абзацы.....	19
1.6. Списки	20
1.6.1. Маркированные списки	20
1.6.2. Нумерованные списки	21
1.6.3. Списки определений	22
1.7. Графика.....	23
1.7.1. Изображение на Web-странице.....	23
1.7.2. Изображение в качестве фона.....	25
1.8. Гиперссылки.....	25
1.8.1. Внешние гиперссылки	25
Абсолютный URL-адрес	26
Относительный URL-адрес.....	26
1.8.2. Внутренние гиперссылки	27
1.8.3. Гиперссылки на адрес электронной почты.....	28

1.9. Таблицы.....	28
1.9.1. Вставка таблицы в документ.....	29
1.9.2. Заголовок таблицы.....	30
1.9.3. Строки таблицы.....	30
1.9.4. Ячейки таблицы.....	31
1.10. Фреймы.....	34
1.10.1. Разделение окна Web-браузера на несколько областей.....	34
1.10.2. Структура HTML-документа, содержащего фреймы.....	37
1.10.3. Описание фреймовой структуры.....	38
1.10.4. Описание отдельных областей.....	38
1.10.5. Тег <code><noframes></code>	39
1.10.6. Загрузка документа в определенный фрейм.....	39
1.10.7. Тег <code><iframe></code> . Добавление фрейма в обычный документ.....	40
1.11. Карты-изображения.....	42
1.11.1. Карта-изображение как панель навигации.....	42
1.11.2. Структура карт-изображений.....	43
1.11.3. Тег <code><map></code>	44
1.11.4. Описание активной области на карте-изображении.....	44
1.12. Формы.....	46
1.12.1. Создание формы для регистрации сайта.....	46
1.12.2. Структура документа с формами.....	47
1.12.3. Добавление формы в документ.....	48
1.12.4. Описание элементов управления.....	49
Текстовое поле и поле ввода пароля.....	50
Кнопки <i>Сброс</i> , <i>Отправить</i> и командная кнопка.....	51
Скрытое поле <i>hidden</i>	51
Поле для установки флажка.....	51
Элемент-переключатель.....	51
Текстовая область.....	52
Список с предопределенными значениями.....	52
1.12.5. Тег <code><label></code>	54
1.12.6. Группировка элементов формы.....	56
1.13. Теги <code><div></code> и <code></code> . Группировка элементов страницы.....	56
1.14. Отличия XHTML 1.0 от HTML 4.01.....	58
1.15. Проверка HTML-документов на соответствие стандартам.....	61
1.16. Специальный тег в Web-браузере Internet Explorer.....	61
1.17. HTML 5.....	63
1.17.1. Требования к страницам, написанным на HTML 5.....	63
1.17.2. Семантическая разметка.....	64
1.17.3. Мультимедиа.....	66
Вставка аудиоролика.....	66
Вставка видеоролика.....	67
Указание нескольких источников аудио или видео.....	67
Обеспечение совместимости со старыми Web-обозревателями.....	68

1.17.4. Новые возможности форм и элементов управления	69
Новые возможности форм	69
Новые возможности элементов управления	69
Задание значений для автодополнения.....	71
1.17.5. Прочие нововведения	72
1.17.6. Теги и параметры, объявленные устаревшими	72
Глава 2. Основы CSS. Форматируем Web-страницу с помощью стилей.....	73
2.1. Основные понятия	73
2.2. Способы встраивания определения стиля	74
2.2.1. Встраивание определения стиля в тег	74
2.2.2. Встраивание определения стилей в заголовок HTML-документа	74
2.2.3. Вынесение таблицы стилей в отдельный файл.....	78
2.2.4. Приоритет применения стилей	80
2.3. Единицы измерения в CSS.....	81
2.4. Форматирование шрифта	82
2.4.1. Имя шрифта	82
2.4.2. Стиль шрифта	83
2.4.3. Размер шрифта	83
2.4.4. Цвет шрифта	83
2.4.5. Жирность шрифта	83
2.5. Форматирование текста	84
2.5.1. Расстояние между символами в словах.....	84
2.5.2. Расстояние между словами	84
2.5.3. Отступ первой строки.....	84
2.5.4. Вертикальное расстояние между строками	84
2.5.5. Горизонтальное выравнивание текста.....	85
2.5.6. Вертикальное выравнивание текста	85
2.5.7. Подчеркивание, надчеркивание и зачеркивание текста.....	86
2.5.8. Изменение регистра символов	86
2.5.9. Обработка пробелов между словами.....	87
2.6. Отступы	87
2.6.1. Внешние отступы	88
2.6.2. Внутренние отступы	88
2.7. Рамки	89
2.7.1. Стиль линии рамки.....	89
2.7.2. Толщина линии рамки	90
2.7.3. Цвет линии рамки.....	91
2.7.4. Одновременное задание атрибутов рамки	91
2.8. Фон элемента	91
2.8.1. Цвет фона.....	92
2.8.2. Фоновый рисунок.....	92
2.8.3. Режим повтора фонового рисунка.....	92
2.8.4. Прокрутка фонового рисунка.....	92
2.8.5. Положение фонового рисунка	93
2.8.6. Одновременное задание атрибутов фона.....	93

2.9. Списки	93
2.9.1. Вид маркера списка	94
2.9.2. Изображение в качестве маркера списка	94
2.9.3. Компактное отображение списка	94
2.10. Вид курсора	95
2.11. Псевдостили гиперссылок. Отображение ссылок разными цветами	95
2.12. Форматирование блоков	97
2.12.1. Указание типа блока	97
2.12.2. Установка размеров	99
2.12.3. Атрибут <i>overflow</i>	99
2.12.4. Управление обтеканием	101
2.12.5. Позиционирование блока	102
2.12.6. Последовательность отображения слоев	105
2.13. Управление отображением элемента	106
2.14. CSS 3	107
2.14.1. Новые селекторы	108
2.14.2. Новые единицы измерения размеров и способы задания цвета	112
2.14.3. Параметры фона	112
Размеры фонового изображения	112
Режим позиционирования фонового изображения	113
Режим заполнения для фонового изображения	113
2.14.4. Рамки со скругленными углами	113
Задание радиуса скругления для разных углов по отдельности	114
Задание радиуса скругления сразу для всех углов	114
2.14.5. Параметры таблиц	115
Просвет между ячейками	115
Режим рисования рамок	115
2.14.6. Параметры тени	115
Параметры тени у текста	115
Параметры тени у контейнера	116
2.14.7. Загружаемые шрифты	116
2.14.8. Режимы установки размеров	117
Режим установки размеров для контейнеров	117
Режим установки размеров для таблиц	118
2.14.9. Градиентные фоны	118
Введение в градиенты	118
Создание линейных градиентов	119
Создание радиальных градиентов	120
Создание повторяющихся градиентов	122
2.14.10. Анимация с двумя состояниями	122
Введение в анимацию с двумя состояниями	122
Задание продолжительности анимации	123
Задание анимируемых атрибутов	123
Задержка перед началом анимации	124
Закон анимации	124

Одновременное задание всех параметров анимации	125
Создание обратной анимации.....	125
Сложная анимация	126
2.14.11. Анимация с несколькими состояниями.....	127
Описание набора состояний для анимации.....	127
Указание набора состояний и продолжительности анимации.....	128
Указание задержки перед началом анимации и ее закона	128
Задание количества повторений анимации	129
Направление анимации	129
Текущее состояние анимации.....	129
Положение анимированного элемента по окончанию анимации.....	130
Одновременное задание всех параметров анимации	130
Сложная анимация	130
Создание кроссплатформенной анимации	131
2.14.12. Двумерные преобразования	131
Как задаются преобразования и их параметры.....	131
Смещение	132
Масштабирование	132
Наклон	133
Поворот	133
Позиционирование точки начала координат для двумерных преобразований.....	134
Сложные двумерные преобразования	134
Кроссплатформенные двумерные преобразования	135
2.14.13. Трехмерные преобразования.....	135
Перспектива	135
Выполнение трехмерных преобразований.....	136
Задание точки зрения	137
Скрытие обратной стороны элемента.....	137
Режим проецирования элементов на контейнер	138
Позиционирование точки начала координат для трехмерных преобразований	140
Сложные трехмерные преобразования.....	140
Кроссплатформенные трехмерные преобразования	140
2.15. Проверка CSS-кода на соответствие стандартам.....	140

Глава 3. Основы JavaScript. Создаем страницы, реагирующие на действия пользователей.....	143
3.1. Основные понятия	143
3.2. Первая программа на JavaScript	143
3.3. Комментарии в JavaScript	145
3.4. Вывод результатов работы программы и ввод данных	146
3.4.1. Окно с сообщением и кнопкой <i>OK</i>	146
3.4.2. Окно с сообщением и кнопками <i>OK</i> и <i>Cancel</i>	147
3.4.3. Окно с полем ввода и кнопками <i>OK</i> и <i>Cancel</i>	147
3.5. Переменные.....	148
3.6. Типы данных и инициализация переменных. Определение типа данных переменной.....	149

3.7. Операторы JavaScript.....	150
3.7.1. Математические операторы	150
3.7.2. Операторы присваивания	152
3.7.3. Двоичные операторы	152
3.7.4. Оператор обработки строк	153
3.7.5. Приоритет выполнения операторов	154
3.8. Преобразование типов данных	154
3.9. Специальные символы. Разбиение сообщения в диалоговом окне на несколько строк	157
3.10. Массивы	158
3.11. Функции. Разделение программы на фрагменты.....	160
3.11.1. Основные понятия.....	160
3.11.2. Расположение функций внутри HTML-документа	162
3.11.3. Рекурсия. Вычисление факториала	163
3.11.4. Глобальные и локальные переменные.....	164
3.12. Условные операторы. Выполнение блоков кода только при соответствии условию.....	166
3.12.1. Операторы сравнения	166
3.12.2. Оператор ветвления <i>if...else</i> . Проверка ввода пользователя.....	167
3.12.3. Оператор <i>?</i> Проверка числа на четность.....	169
3.12.4. Оператор выбора <i>switch</i>	169
3.13. Операторы циклов. Многократное выполнение блока кода	171
3.13.1. Цикл <i>for</i>	171
3.13.2. Цикл <i>while</i>	173
3.13.3. Цикл <i>do...while</i>	173
3.13.4. Оператор <i>continue</i> . Переход на следующую итерацию цикла.....	174
3.13.5. Оператор <i>break</i> . Прерывание цикла	174
3.14. Ошибки в программе.....	175
3.14.1. Синтаксические ошибки.....	175
3.14.2. Логические ошибки.....	176
3.14.3. Ошибки времени выполнения.....	176
3.14.4. Обработка ошибок	177
3.14.5. Модуль <i>Firebug</i> для Web-браузера <i>Firefox</i>	177
3.15. Встроенные классы JavaScript	181
3.15.1. Основные понятия.....	181
3.15.2. Класс <i>Global</i>	181
3.15.3. Класс <i>Number</i> . Работа с числами	183
3.15.4. Класс <i>String</i> . Обработка строк	183
3.15.5. Класс <i>Array</i> . Работа с массивами и их сортировка.....	186
Многомерные массивы	189
Ассоциативные массивы. Перебор ассоциативных массивов	190
3.15.6. Класс <i>Math</i> . Использование математических функций	191
3.15.7. Класс <i>Date</i> . Получение текущей даты и времени. Вывод даты и времени в окне Web-браузера	193
3.15.8. Класс <i>Function</i> (функции).....	196
3.15.9. Класс <i>Arguments</i> . Функции с произвольным количеством аргументов.....	197

3.15.10. Класс <i>RegExp</i> . Проверка значений с помощью регулярных выражений	198
Метасимволы, используемые в регулярных выражениях.	
Проверка правильности ввода дат и адресов электронной почты	201
Логическое ИЛИ.....	205
Глобальный класс <i>RegExp</i> . Составные части адресов электронной почты и URL-адресов	206
3.16. События	207
3.16.1. Основные понятия.....	207
3.16.2. События мыши	207
3.16.3. События клавиатуры.....	208
3.16.4. События документа.....	208
3.16.5. События формы.....	208
3.16.6. Последовательность событий	208
3.16.7. Всплывание событий	210
3.16.8. Действия по умолчанию и их отмена	212
3.16.9. Написание обработчиков событий	213
3.16.10. Объект <i>event</i> . Вывод координат курсора и кода нажатой клавиши. Вывод сообщений при нажатии комбинации клавиш.....	219
3.17. Объектная модель Microsoft Internet Explorer	223
3.17.1. Структура объектной модели	223
3.17.2. Объект <i>window</i> . Вывод сообщения в строку состояния Web-браузера	225
3.17.3. Работа с окнами. Создание нового окна без строки меню, адресной строки и панели инструментов	229
3.17.4. Модальные диалоговые окна. Использование модальных окон вместо встроенных диалоговых окон.....	233
3.17.5. Таймеры. Создание часов на Web-странице.....	235
3.17.6. Объект <i>navigator</i> . Получение информации о Web-браузере пользователя. Перенаправление клиента на разные страницы в зависимости от Web-браузера	237
3.17.7. Объект <i>screen</i> . Получение информации о мониторе пользователя	240
3.17.8. Объект <i>location</i> . Разбор составляющих URL-адреса документа. Создание многостраничных HTML-документов.....	241
3.17.9. Объект <i>history</i> . Получение информации о просмотренных страницах. Реализация перехода на предыдущую просмотренную страницу	246
3.17.10. Объект <i>document</i> . Получение полной информации о HTML-документе	246
Общие свойства и методы элементов Web-страницы	248
Получение информации о HTML-документе.....	250
3.17.11. Обращение к элементам документа. Выравнивание заголовков по центру.....	252
3.17.12. Работа с элементами документа. Изменение URL-адреса и текста ссылки. Преобразование ссылки в обычный текст	254
3.17.13. Объект <i>style</i> . Работа с таблицами стилей при помощи JavaScript.....	260
3.17.14. Объект <i>selection</i> . Проверка наличия выделенного фрагмента	262
3.17.15. Объект <i>TextRange</i> . Поиск фрагмента в текстовом поле или документе. Расширение или сжатие выделенного фрагмента текста.....	266
3.17.16. Работа с буфером обмена. Выделение фрагмента от позиции щелчка до конца документа и копирование его в буфер обмена	274

3.17.17. Реализация ссылок "Добавить сайт в Избранное" и "Сделать стартовой страницей"	275
3.17.18. Сохранение данных на компьютере клиента. Определение возможности использования cookies. Сохранение русского текста в cookies.....	276
3.18. Работа с элементами формы	280
3.18.1. Элементы управления	280
3.18.2. Коллекция <i>Forms</i> . Доступ к элементу формы из скрипта.....	281
3.18.3. Свойства объекта формы.....	281
3.18.4. Методы объекта формы.....	282
3.18.5. События объекта формы	282
3.18.6. Текстовое поле и поле ввода пароля. Проверка правильности ввода E-mail и пароля. Получение данных из элемента формы	282
3.18.7. Поле для ввода многострочного текста. Добавление слов из текстового поля в поле <code><textarea></code>	284
3.18.8. Список с возможными значениями. Возможность добавления нового пункта. Применение списков вместо гиперссылок.....	286
3.18.9. Флажок и переключатели. Получение значения выбранного переключателя при помощи цикла и проверка установки флажка	291
3.18.10. Кнопки. Обработка нажатия кнопки. Деактивация кнопки. Создание клавиши быстрого доступа и вывод текста на кнопке определенным цветом	293
3.18.11. Проверка корректности данных. Создание формы регистрации пользователя	295
3.19. Пользовательские объекты	299
3.19.1. Создание объектов	299
3.19.2. Прототипы	303
3.19.3. Пространства имен.....	305
3.20. AJAX.....	307
3.20.1. Подготовка к загрузке данных	307
Стандартный способ.....	307
Способ, применяемый в Internet Explorer 5 и 6.....	307
Универсальный способ	308
3.20.2. Отправка запроса	308
Синхронный или асинхронный запрос?	308
Задание параметров запроса.....	309
Задание MIME-типа отправляемых данных.....	309
Собственно отправка запроса.....	310
Отправка данных с запросом.....	310
3.20.3. Получение данных	311
Оформление обработчика данных	311
Определение успешного получения данных	312
Собственно получение данных.....	312
3.20.4. Формат JSON.....	313
Описание формата JSON	313
Декодирование данных JSON: стандартный способ	314
Декодирование данных JSON: способ, применяемый в устаревших Web-обозревателях.....	314
Декодирование данных JSON: универсальный способ	314

3.21. DOM 3.....	316
3.21.1. Обращение к элементам страницы.....	316
Обращение по имени класса.....	316
Обращение по селектору.....	316
3.21.2. Управление содержимым страницы.....	317
Создание, изменение и удаление элементов страницы.....	317
Работа с содержимым элементов.....	317
Работа со стилями элементов.....	318
3.21.3. Обработка событий.....	318
Указание обработчиков событий.....	318
Новый набор событий.....	318
Новый объект <i>event</i>	318
3.21.4. Работа с формами и элементами управления.....	319
Работа с формами.....	319
Работа с элементами управления.....	320
Обработка списков с возможностью выбора нескольких пунктов.....	321
Расширенная проверка значения, занесенного в поле ввода.....	322
3.21.5. Работа с графическими изображениями.....	323
3.21.6. Работа с мультимедиа.....	324
Свойства, методы и события аудио- и видеороликов.....	324
Создание элементов для управления воспроизведением ролика.....	328
3.21.7. Канва HTML 5. Программируемая графика.....	330
Канва.....	330
Контекст рисования.....	330
Прямоугольники.....	331
Задание цвета, уровня прозрачности и толщины линий.....	331
Рисование сложных фигур.....	333
Вывод текста.....	338
Использование сложных цветов.....	340
Вывод внешних изображений.....	344
Создание тени у рисуемой графики.....	345
Преобразование системы координат.....	346
Управление наложением графики.....	349
Использование масок.....	350
Работа с отдельными пикселями.....	351
3.21.8. Хранилище.....	354
Сессионное и локальное хранилища.....	354
Работа с хранилищем.....	354
Использование локального хранилища для временного хранения данных.....	356
3.21.9. Средства геолокации.....	357
Доступ к средствам геолокации.....	357
Получение данных геолокации.....	357
Обработка нештатных ситуаций.....	358
Задание дополнительных параметров.....	359
Отслеживание местоположения компьютера.....	360
3.22. JavaScript-библиотеки.....	360

Глава 4. Программное обеспечение Web-сервера.

Устанавливаем и настраиваем программы под Windows	363
4.1. Необходимые программы	363
4.2. Установка, настройка и запуск сервера Apache	364
4.2.1. Настройка сервера Apache.....	364
4.2.2. Запуск и останов Apache.....	367
4.2.3. Установка Apache как службы Windows	367
4.3. Структура каталогов сервера Apache	369
4.4. Файл конфигурации httpd.conf.....	370
4.4.1. Основные понятия.....	370
4.4.2. Разделы файла конфигурации	371
4.4.3. Общие директивы. Создание домашнего каталога пользователя, доступного при запросе http://localhost/~nik/	372
4.4.4. Переменные сервера и их использование	374
4.4.5. Директивы управления производительностью	375
4.4.6. Директивы обеспечения постоянного соединения.....	376
4.4.7. Директивы работы с языками	376
4.4.8. Директивы перенаправления.....	377
4.4.9. Обработка ошибок	377
4.4.10. Настройки MIME-типов	378
4.4.11. Управление листингом каталога.....	380
4.4.12. Директивы протоколирования	382
4.4.13. Файл конфигурации .htaccess. Управляем сервером Apache из обычной папки.....	384
4.4.14. Защита содержимого папки паролем	385
4.4.15. Управление доступом	388
4.4.16. Регулярные выражения, используемые в директивах.....	390
4.4.17. Создание виртуальных серверов.....	391
4.5. Установка PHP	393
4.6. Установка MySQL	400
4.7. Установка phpMyAdmin	408

Глава 5. Основы PHP. Создаем динамические Web-страницы

413	
5.1. Основные понятия	413
5.2. Первая программа на PHP	413
5.2.1. Особенности создания скриптов в кодовой таблице UTF-8.....	416
5.3. Методы встраивания PHP-кода	417
5.4. Комментарии в PHP-сценариях	417
5.5. Вывод результатов работы скрипта	418
5.6. Переменные.....	420
5.7. Типы данных и инициализация переменных.....	420
5.8. Проверка существования переменной	422
5.9. Удаление переменной	423
5.10. Константы. Создание и использование констант.....	423
5.11. Операторы PHP.....	425
5.11.1. Математические операторы	425

5.11.2. Операторы присваивания	426
5.11.3. Двоичные операторы	426
5.11.4. Оператор конкатенации строк. Подстановка значений переменных. Запуск внешних программ	427
5.11.5. Приоритет выполнения операторов	429
5.12. Преобразование типов данных	430
5.13. Специальные символы	432
5.14. Массивы	432
5.14.1. Инициализация массива	432
5.14.2. Получение и изменение элемента массива. Определение числа элементов массива.....	433
5.14.3. Многомерные массивы	433
5.14.4. Ассоциативные массивы	434
5.14.5. Слияние массивов	435
5.14.6. Перебор элементов массива	435
Перебор элементов массива без использования циклов	437
5.14.7. Добавление и удаление элементов массива	438
5.14.8. Переворачивание и перемешивание массива	439
5.14.9. Сортировка массива. Создание пользовательской сортировки	440
5.14.10. Получение части массива	442
5.14.11. Преобразование переменных в массив	442
5.14.12. Преобразование массива в переменные	443
5.14.13. Заполнение массива числами	443
5.14.14. Преобразование массива в строку	444
5.14.15. Проверка наличия значения в массиве.....	445
5.15. Строки.....	446
5.15.1. Функции для работы со строками.....	446
5.15.2. Настройка локали.....	449
5.15.3. Функции для работы с символами.....	450
5.15.4. Поиск и замена в строке	450
5.15.5. Функции для сравнения строк.....	451
5.15.6. Кодирование строк.....	451
5.15.7. Преобразование кодировок	452
5.15.8. Регулярные выражения. Разбираем адрес электронной почты на составные части. Проверяем правильность введенной даты.....	454
Метасимволы, используемые в регулярных выражениях.....	456
Логическое ИЛИ.....	459
5.15.9. Perl-совместимые регулярные выражения	459
5.15.10. Функции для работы со строками в кодировке UTF-8	467
5.15.11. Перегрузка строковых функций.....	474
5.16. Функции для работы с числами.....	475
5.17. Функции для работы с датой и временем. Получение текущей даты, даты создания файла и проверка корректности введенной даты	477
5.18. Функции. Разделение программы на фрагменты.....	480
5.18.1. Основные понятия.....	480

5.18.2. Расположение описаний функций	482
5.18.3. Операторы <i>require</i> и <i>include</i> . Выносим функции в отдельный файл. Создаем шаблоны для множества страниц	482
5.18.4. Операторы <i>require_once</i> и <i>include_once</i>	485
5.18.5. Рекурсия. Вычисляем факториал	485
5.18.6. Глобальные и локальные переменные. Использование глобальных переменных внутри функций	486
5.18.7. Статические переменные	489
5.18.8. Переменное число параметров в функции. Сумма произвольного количества чисел	489
5.19. Условные операторы. Выполнение блоков кода только при соответствии условию	490
5.19.1. Операторы сравнения	490
5.19.2. Оператор ветвления <i>if...else</i> . Проверка выбранного элемента из списка	491
5.19.3. Оператор <i>?</i> Проверка числа на четность	494
5.19.4. Оператор выбора <i>switch</i> . Использование оператора <i>switch</i> вместо <i>if...else</i>	495
5.20. Операторы циклов. Многократное выполнение блока кода	496
5.20.1. Цикл <i>for</i>	497
5.20.2. Цикл <i>while</i>	498
5.20.3. Цикл <i>do...while</i>	499
5.20.4. Цикл <i>foreach</i>	499
5.20.5. Оператор <i>continue</i> . Переход на следующую итерацию цикла	500
5.20.6. Оператор <i>break</i> . Прерывание цикла	500
5.21. Завершение выполнения сценария. Навигация при выборе значения из списка	501
5.22. Ошибки в программе	502
5.22.1. Синтаксические ошибки	502
5.22.2. Логические ошибки	503
5.22.3. Ошибки времени выполнения	503
5.22.4. Обработка ошибок	503
5.22.5. Инstrukция <i>or die()</i>	504
5.23. Переменные окружения	505
5.23.1. Суперглобальные массивы	505
5.23.2. Часто используемые переменные окружения	505
5.24. Заголовки HTTP	506
5.24.1. Основные заголовки	508
5.24.2. Функции для работы с заголовками. Перенаправление клиента на другой URL-адрес. Запрет кэширования страниц. Реализация ссылки <i>Скачать</i> . Просмотр заголовков, отправляемых сервером	510
5.24.3. Работа с cookies. Создаем индивидуальный счетчик посещений	514
5.25. Работа с файлами и каталогами	515
5.25.1. Основные понятия	515
5.25.2. Функции для работы с файлами. Создание файла, запись в файл, вывод содержимого файла в список	515
5.25.3. Перемещение внутри файла	519
5.25.4. Создание списка рассылки с возможностью добавления, изменения и удаления E-mail-адресов	519

5.25.5. Чтение CSV-файлов. Преобразование CSV-файла в HTML-таблицу.....	523
5.25.6. Права доступа в операционной системе UNIX.....	525
5.25.7. Функции для манипулирования файлами	527
5.25.8. Загрузка файлов на сервер.....	528
5.25.9. Функции для работы с каталогами. Создаем программу для просмотра всех доступных каталогов и файлов на диске.....	530
5.25.10. Получение информации из сети Интернет.....	533
Использование библиотеки CURL.....	538
5.26. Отправка писем с сайта. Рассылка писем по E-mail-адресам из файла	542
5.27. Аутентификация с помощью PHP. Создание Личного кабинета	545
5.28. Работа с графикой.....	549
5.28.1. Информация об установленной библиотеке GD	549
5.28.2. Получение информации об изображении	550
5.28.3. Работа с готовыми изображениями	553
5.28.4. Создание нового изображения	555
5.28.5. Работа с цветом	555
5.28.6. Рисование линий и фигур	557
5.28.7. Вывод текста в изображение. Создаем счетчик посещений.....	560
5.28.8. Изменение размеров и копирование изображений	564
5.29. Обработка данных формы.....	567
5.29.1. Текстовое поле, поле ввода пароля и скрытое поле.....	567
5.29.2. Поле для ввода многострочного текста	568
5.29.3. Список с возможными значениями	569
5.29.4. Флажок.....	570
5.29.5. Элемент-переключатель	571
5.29.6. Кнопка <i>Submit</i>	571
5.29.7. Проверка корректности данных. Создание формы регистрации пользователя.....	572
5.30. Другие полезные функции	576
5.30.1. Выделение фрагментов исходного кода	576
5.30.2. Получение информации об интерпретаторе	576
5.30.3. Вывод всех доступных сценарию функций	577
5.30.4. Засыпание сценария	578
5.30.5. Изменение значения директив во время выполнения сценария.....	578
5.30.6. Выполнение команд, содержащихся в строке	580
5.31. Объектно-ориентированное программирование.....	580
5.31.1. Создание класса.....	581
5.31.2. Конструктор и деструктор.....	581
5.31.3. Наследование.....	582
5.31.4. Статические свойства и методы.....	585
5.31.5. Объявление констант внутри класса	585
5.31.6. Определение области видимости.....	586
5.31.7. Абстрактные классы и методы.....	588
5.31.8. Интерфейсы	589
5.31.9. Оператор проверки типа <i>instanceof</i>	590
5.31.10. Создание шаблона сайта при помощи класса.....	591

5.32. Поддержка AJAX со стороны сервера. Кодирование данных в формате JSON	592
5.33. Шаблонизатор Smarty.....	595
5.33.1. Установка и настройка	596
5.33.2. Управляющие конструкции.....	599
5.33.3. Модификаторы переменных	607
5.33.4. Кэширование страниц.....	611
Глава 6. Основы MySQL. Работаем с базами данных	615
6.1. Основные понятия	615
6.2. Нормализация базы данных.....	615
6.3. Типы данных полей.....	618
6.3.1. Числовые типы	619
6.3.2. Строковые типы	619
6.3.3. Дата и время	620
6.4. Основы языка SQL.....	620
6.4.1. Создание базы данных.....	621
6.4.2. Создание пользователя базы данных.....	622
6.4.3. Создание таблицы	624
6.4.4. Добавление данных в таблицу	626
6.4.5. Обновление записей.....	629
6.4.6. Удаление записей из таблицы	630
6.4.7. Изменение структуры таблицы	630
6.4.8. Выбор записей.....	631
6.4.9. Выбор записей из нескольких таблиц	633
6.4.10. Индексы. Ускорение выполнения запросов.....	637
6.4.11. Удаление таблицы и базы данных	642
6.5. Доступ к базе данных из PHP с помощью библиотеки php_mysql.dll.....	643
6.5.1. Установка соединения	643
6.5.2. Выбор базы данных	644
6.5.3. Выполнение запроса к базе данных.....	644
6.5.4. Обработка результата запроса	645
6.6. Доступ к базе данных из PHP с помощью библиотеки php_mysql.dll.....	650
6.6.1. Установка соединения	651
6.6.2. Выбор базы данных	652
6.6.3. Выполнение запроса к базе данных.....	653
6.6.4. Обработка результата запроса	654
6.7. Операторы MySQL	661
6.7.1. Математические операторы	661
6.7.2. Двоичные операторы	663
6.7.3. Операторы сравнения	663
6.7.4. Приоритет выполнения операторов	665
6.7.5. Преобразование типов данных	666
6.8. Поиск по шаблону	666
6.9. Поиск с помощью регулярных выражений	669
6.9.1. Метасимволы, используемые в регулярных выражениях	670

6.10. Режим полнотекстового поиска	673
6.10.1. Создание индекса <i>FULLTEXT</i>	673
6.10.2. Реализация полнотекстового поиска	675
6.10.3. Режим логического поиска.....	675
6.10.4. Поиск с расширением запроса	676
6.11. Функции MySQL.....	677
6.11.1. Функции для работы с числами	677
6.11.2. Функции даты и времени.....	681
6.11.3. Функции для обработки строк	693
6.11.4. Функции для шифрования строк.....	698
6.11.5. Информационные функции	700
6.11.6. Прочие функции.....	701
6.12. Переменные SQL	705
6.13. Временные таблицы	706
6.14. Вложенные запросы	707
6.14.1. Заполнение таблицы с помощью вложенного запроса	708
6.14.2. Применение вложенных запросов в инструкции <i>WHERE</i>	710
6.14.3. Применение вложенных запросов в инструкции <i>FROM</i>	712
6.15. Внешние ключи	712
6.16. Транзакции	715
6.16.1. Запуск, подтверждение и отмена транзакций	715
6.16.2. Изоляция транзакций	717
Введение в изоляцию транзакций	717
Уровни изоляции транзакций	718
6.16.3. Автозавершение транзакций и его отключение	720
6.16.4. Поддержка транзакций библиотекой <i>php_mysqli.dll</i>	721
Приложение. Описание электронного архива.....	723
Предметный указатель	725

Введение

Если вы хотите научиться своими руками создавать сайты, свободно владеть HTML, CSS, JavaScript, PHP и MySQL, то эта книга для вас. Большинство подобных книг предлагают изучение или только клиентских технологий (HTML, CSS, JavaScript), или только серверных (PHP, MySQL). Но разделять эти инструменты нельзя, т. к. они могут существовать только совместно, а значит, и изучать их нужно как единое целое.

Все главы книги расположены в порядке возрастания уровня сложности материала. Если вы начинающий Web-мастер, то книгу следует изучать последовательно, главу за главой. Если же материал какой-либо из глав был изучен ранее, то можно сразу переходить к следующей главе.

Что же можно создать с использованием описываемых технологий? Давайте кратко рассмотрим возможности этих технологий, а также содержание глав книги.

Язык разметки HTML, рассматриваемый в *главе 1*, позволяет задать местоположение элементов Web-страницы в окне Web-браузера. С помощью HTML можно отформатировать отдельные символы или целые фрагменты текста, вставить изображение, таблицу или форму, создать панель навигации с помощью карт-изображений, разделить окно Web-браузера на несколько областей, вставить гиперссылку и многое другое. А новая версия языка HTML — HTML 5 — даже позволяет поместить на страницу аудио- или видеоролик, который будет воспроизводиться самим Web-браузером, без необходимости устанавливать какие бы то ни было плагини.

При помощи каскадных таблиц стилей (CSS), о которых идет речь в *главе 2*, можно задавать точные характеристики практически всех элементов Web-страницы. Это позволяет контролировать внешний вид Web-страницы в окне Web-браузера и приближает возможности Web-дизайна к настольным издательским системам. Разработчик может указать параметры шрифта, цвет текста и фона, выравнивание, создать рамку и расположить элементы на странице произвольным образом. Новая версия CSS — CSS 3 — также предоставляет инструменты для задания градиентного фона, теней у текста и самого элемента страницы и даже для создания анимации.

У Web-страниц, созданных с использованием HTML и CSS, есть существенный недостаток — они являются статическими, т. е. не могут меняться, реагируя на действия пользователя. Внедрение программ, написанных на языке JavaScript, в HTML-код позволит "оживить" Web-страницу, сделать ее интерактивной или, другими словами, заставить взаимодействовать с пользователем. С помощью JavaScript можно обрабатывать данные формы до отправки на сервер, получать информацию о Web-браузере пользователя и его мониторе и соответствующим образом менять форматирование страницы, создавать новые окна, изменять любые элементы HTML-документа в ответ на какое-либо событие, создавать часы на Web-странице, показывающие текущее время с точностью до секунды, скрывать или отображать элементы Web-страницы и выполнять многие другие действия. Как все это сделать, рассказано в *главе 3*.

В *главе 3* также описывается технология AJAX, позволяющая программно подгружать с сервера произвольные данные без перезагрузки самой страницы. Это могут быть как фрагменты HTML-кода, выводимые на страницу непосредственно, так и данные, закодированные в формате JSON и предназначенные для использования в JavaScript-программах. Применение технологии AJAX позволит значительно расширить функциональность создаваемых сайтов.

Глава 4 повествует, как установить и настроить специальное программное обеспечение для тестирования сайтов: Web-сервер Apache, среду для выполнения серверных скриптов, написанных на языке PHP, и сервер баз данных MySQL. Таким образом, можно проверить работоспособность создаваемого сайта непосредственно на своем компьютере, еще до его публикации в Интернете.

Огромные возможности открывают серверные технологии, среди которых для целей данной книги выбран язык программирования PHP. Это наиболее распространенный в настоящее время язык для написания серверных скриптов. Используя его (или другие программные платформы, применяемые для создания динамических Web-страниц), можно изменять HTML-код, получаемый Web-браузером, в зависимости от вводимых пользователем данных, типа и версии установленного Web-браузера и других факторов. Большое количество расширений и готовых программных продуктов, а также легкость освоения сделали PHP очень популярным языком программирования для Интернета. С помощью PHP можно работать с файлами и каталогами, обрабатывать данные формы на сервере, рассылать письма, загружать файлы на сервер, создавать для каждого пользователя Личный кабинет, программировать гостевые книги, форумы, блоги, интернет-магазины и многое другое. Писать программы на PHP мы научимся в *главе 5*.

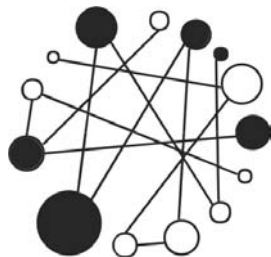
На сегодняшний день ни один крупный портал не обходится без использования баз данных. В Web-разработках чаще всего применяется быстрая, бесплатная и обладающая большими возможностями система управления базами данных (СУБД) MySQL. С помощью MySQL можно эффективно добавлять, изменять и удалять данные, получать нужную информацию по запросу. Работа с MySQL, в том числе с базами данных этого формата из программ, написанных на PHP, обсуждается в *главе 6*.

В электронном приложении, доступном на сайте издательства "БХВ-Петербург", собраны следующие материалы:

- ❑ описание процесса установки и настройки специализированных редакторов, которые позволят значительно упростить создание сайта и сделают процесс изучения материала книги более эффективным;
- ❑ описание процесса публикации сайта в Интернете (подбор подходящей площадки, работа с FTP-клиентом, настройка Web-сервера Apache, выполнение автоматического запуска программ в определенное время, подготовка сайта к индексации и т. д.);
- ❑ дополнительные руководства (описание фильтров и преобразований, которые доступны в Web-браузере Internet Explorer, и электронный самоучитель языка Perl);
- ❑ описание процесса разработки полнофункционального Web-сайта с использованием всех изученных технологий. Это каталог сайтов, включающий личный кабинет для пользователей с защитой средствами PHP, а также личный кабинет для администратора, защищенный средствами сервера Apache;
- ❑ все листинги, встречающиеся в тексте книги.

Авторы желают приятного прочтения и надеются, что эта книга станет верным спутником в вашей программистской практике.

ГЛАВА 1



Основы HTML. Создаем дизайн сайта

1.1. Основные понятия

HTML (HyperText Markup Language) — это язык разметки документа, описывающий форму отображения информации на экране компьютера.

При создании документа часто приходится выделять какую-либо часть текста полужирным шрифтом, изменять размер или цвет шрифта, выравнивать текст по центру страницы и т. д. В текстовом редакторе для этого достаточно выделить нужный фрагмент и применить к нему форматирование. Например, чтобы пометить текст курсивом, нужно выделить его и нажать кнопку **Курсив**. На языке HTML тот же эффект достигается следующей строкой кода:

```
<i>Текст</i>
```

Символ `<i>` указывает, что текст надо выделить, начиная с этого места, а `</i>` отмечает конец выделенного фрагмента.

Символы `<i>` и `</i>` принято называть *тегами*. С помощью тегов описывается вся структура документа. Теги выделяются угловыми скобками "`<`" и "`>`", между которыми указывается имя тега. Большинство тегов являются парными, т. к. есть открывающий тег (`<i>`) и соответствующий ему закрывающий (`</i>`). Закрывающий тег отличается наличием косой черты ("`/`") перед его именем. Есть также теги, вообще не имеющие закрывающего тега, например тег переноса строки `
`.

Некоторые теги могут иметь *параметры* (иногда их называют *атрибутами*). Параметры указываются после имени тега через пробел в формате `параметр="значение"`. Если параметров несколько, то они перечисляются через пробел. Например:

```
<meta http-equiv="Content-Type" content="text/html; charset=windows-1251">
```

В этом примере параметру `http-equiv` тега `<meta>` присвоено значение `Content-Type`, а параметру `content` — значение `text/html; charset=windows-1251`.

Теги могут вкладываться друг в друга. Например:

```
<p><i>Текст</i></p>
```


При вложении тегов необходимо соблюдать последовательность их закрытия. Например, такой код использовать нельзя:

```
<p><i>Текст</p></i>
```

ПРИМЕЧАНИЕ

В HTML названия тегов и параметров можно записывать в любом регистре, а в языке XHTML только в нижнем регистре.

Просматривать HTML-документы можно с помощью специальных программ, которые называют Web-браузерами. Web-браузеры отображают документы с форматированием, выполненным на основе исходного кода, описывающего структуру документа.

Результат интерпретации HTML-документа, отображаемый в окне Web-браузера, называется Web-страницей. В отличие от HTML-документа Web-страница может содержать не только текст, но и графику, видео, звуковое сопровождение, может реагировать на действия пользователя и т. д. Кроме того, Web-страница может быть результатом интерпретации сразу нескольких HTML-документов.

Документы в формате HTML имеют расширение html или htm.

Прежде чем изучать язык HTML, советуем установить на компьютер один из редакторов — FCKeditor или tinyMCE. Эти редакторы написаны на языке программирования JavaScript и работают в Web-браузере.

Скачать FCKeditor можно со страницы <http://ckeditor.com/download>. После распаковки архива запустите файл sample07.html (расположен в папке fckeditor_samples\html\). Если вы используете Web-браузер Firefox, то для работы редактора необходимо выполнить следующие действия:

1. В адресной строке вводим `about:config` и нажимаем клавишу `<Enter>`.
2. Находим директиву `security.fileuri.strict_origin_policy` и двойным щелчком на строке устанавливаем значение `false`.

На рис. 1.1 можно увидеть, как выглядит редактор FCKeditor, запущенный в Web-браузере Firefox. Если вы раньше работали с текстовым редактором Microsoft Word, то большинство кнопок на панели инструментов будет вам знакомо. Принцип работы в FCKeditor точно такой же, как и в Word. После ввода текста и его форматирования редактор автоматически генерирует HTML-код. Посмотреть исходный HTML-код можно нажав кнопку **Источник** на панели инструментов (рис. 1.2). Следует заметить, что при изменении исходного HTML-кода автоматически изменяется и внешний вид документа.

Скачать tinyMCE можно со страницы <http://tinymce.moxiecode.com/download.php>. После загрузки распаковываем архив в текущую папку. Для русификации редактора со страницы http://tinymce.moxiecode.com/download_i18n.php необходимо скачать архив с файлами для русского языка. Архив следует разместить в папке `tinymce\jscripts\tiny_mce\`, а затем распаковать в текущую папку. Все файлы будут автоматически распределены по каталогам. Чтобы подключить поддержку русского

языка, необходимо в файле full.html (расположен в папке tinymce\examples\)\ добавить строку

```
language: "ru",
```

сразу после строки

```
tinymce.init({
```

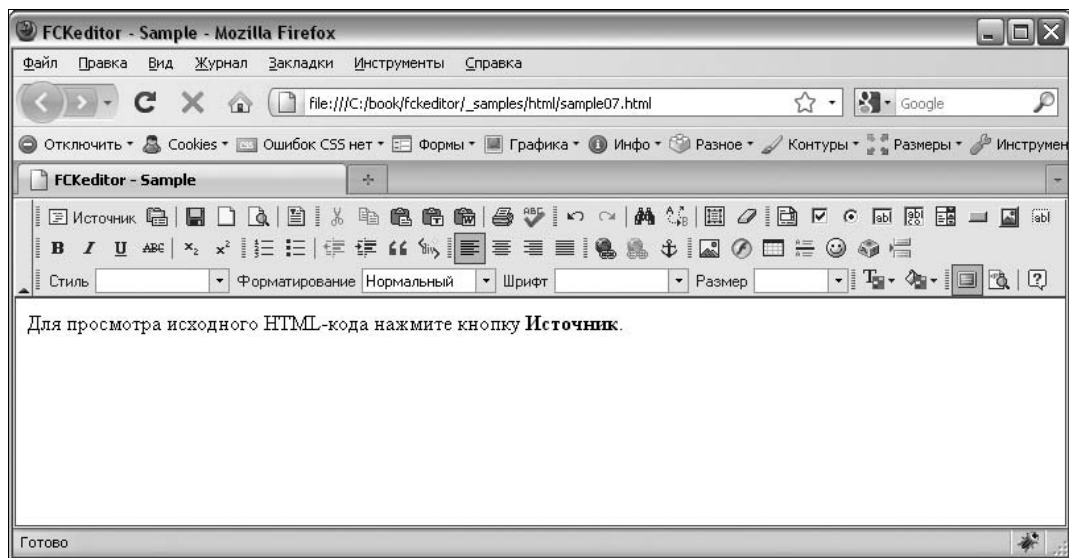


Рис. 1.1. Редактор FCKeditor, запущенный в Web-браузере Firefox

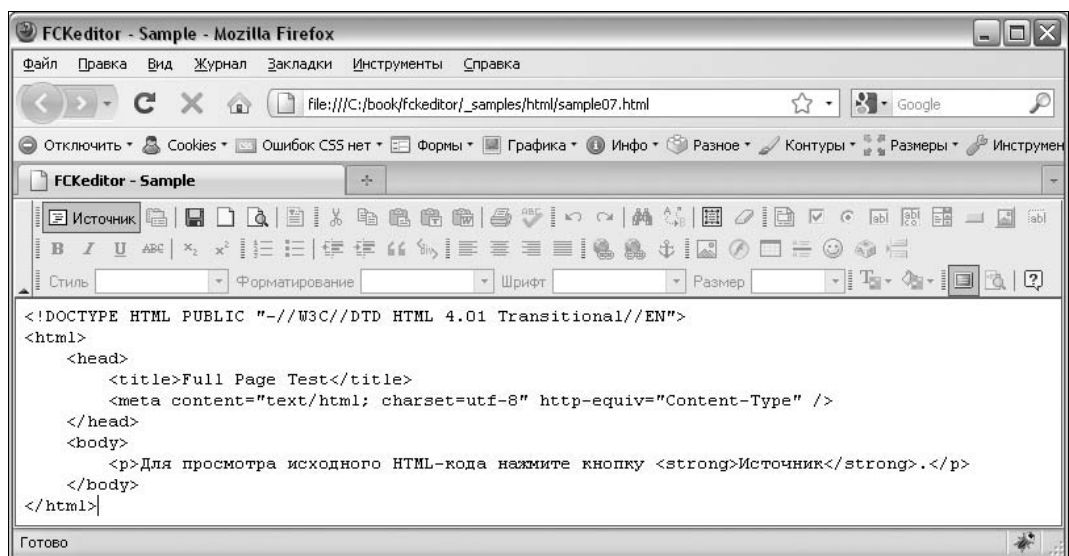


Рис. 1.2. Результат нажатия кнопки **Источник** в редакторе FCKeditor

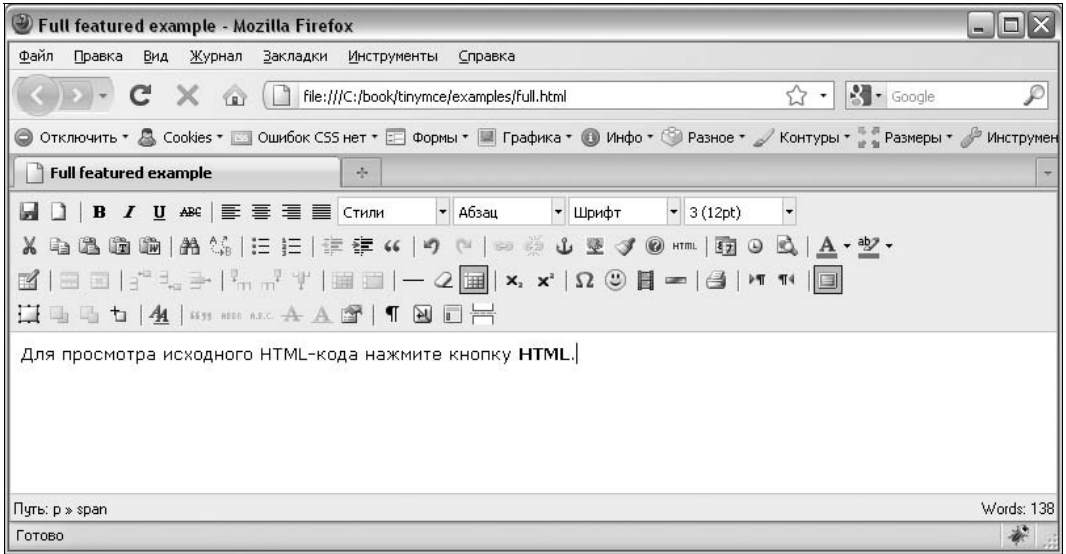


Рис. 1.3. Редактор tinyMCE, запущенный в Web-браузере Firefox

Теперь файл full.html открываем с помощью Web-браузера. На рис. 1.3 можно увидеть, как выглядит редактор tinyMCE, запущенный в Web-браузере Firefox.

1.2. Первый HTML-документ

Попробуем создать наш первый HTML-документ. Для его создания можно воспользоваться любым текстовым редактором. Самым распространенным редактором является обычный Блокнот. Открываем Блокнот и набираем содержимое листинга 1.1.

Листинг 1.1. Первый HTML-документ

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
    "http://www.w3.org/TR/html4/strict.dtd">
<html>
<head>
  <title>Заголовок страницы</title>
  <meta http-equiv="Content-Type" content="text/html; charset=windows-1251">
</head>
<body>
  <p>
    <strong>Этот текст выделен полужирным шрифтом</strong>
  </p>
</body>
</html>
```

Сохраняем введенный текст в формате HTML, например, под именем test.html. Для этого в меню **Файл** выбираем пункт **Сохранить как**. В открывшемся окне в строке **Имя файла** вводим "test.html", а в списке **Тип файла** указываем **Все файлы**. Выбираем папку, например, Рабочий стол, и нажимаем **Сохранить**. Закрываем Блокнот.

ПРИМЕЧАНИЕ

Если в списке **Тип файла** оставить **Текстовые документы (*.txt)**, то в строке **Имя файла** название файла необходимо заключить в кавычки, иначе к имени файла будет добавлено расширение txt.

Запускаем Web-браузер, например, Internet Explorer. С помощью пункта **Открыть** меню **Файл** открываем сохраненный файл test.html. Если все сделано правильно, то в окне Web-браузера будет показана выделенная надпись "Этот текст выделен полужирным шрифтом", а в строке заголовка будет надпись "Заголовок страницы — Microsoft Internet Explorer". Теги в окне Web-браузера не отображаются!

Теперь попробуем изменить заголовок в окне Web-браузера. Для этого необходимо открыть исходный текст в формате HTML. Это можно сделать тремя способами:

- в меню **Вид** выбрать пункт **Просмотр HTML-кода**;
- правой кнопкой мыши щелкнуть в любом месте окна Web-браузера. В появившемся контекстном меню выбрать пункт **Просмотр HTML-кода**;

ПРИМЕЧАНИЕ

В некоторых случаях результат этих двух действий может быть разным. Если Web-страница состоит из нескольких HTML-документов, то первый способ отобразит только код структуры Web-страницы, а не исходный код каждого из HTML-документов. Второй способ позволяет отобразить исходный код лишь одного HTML-документа, а от места щелчка зависит, код какого HTML-документа будет отображен. В нашем случае результат будет одним и тем же.

- открыть файл, содержащий исходный код, с помощью Блокнота или другого текстового редактора. Этот способ является самым универсальным. Настоятельно рекомендуем использовать именно его.

В итоге исходный текст будет доступен для редактирования. Изменим строчку

```
<title>Заголовок страницы</title>
```

на

```
<title>Моя первая Web-страница</title>
```

и сохраним файл (меню **Файл**, пункт **Сохранить**). Теперь вернемся в Web-браузер и обновим Web-страницу. Обновить можно следующими способами:

- в меню **Вид** выбрать пункт **Обновить**;
- выбрать этот же пункт в контекстном меню;
- нажать кнопку **Обновить** на Панели инструментов;
- на клавиатуре нажать клавишу <F5>.

В результате строка заголовка изменится на "Моя первая Web-страница — Microsoft Internet Explorer".

Таким образом, изменяя что-либо в исходном коде, можно визуально оценивать результаты произведенных действий. Алгоритм такой: открываем исходный код, вносим корректировку, сохраняем, а затем обновляем Web-страницу.

ПРИМЕЧАНИЕ

Необходимо заметить, что все описанные действия возможны только для локально сохраненных HTML-документов. Если HTML-документ опубликован в Интернете, то можно лишь созерцать исходный код, а вот изменить его таким способом нельзя.

Очень хорошей альтернативой Блокноту является программа Notepad++. Она позволяет корректно работать как с кодировкой windows-1251, так и с кодировкой UTF-8, а также имеет подсветку синтаксиса HTML, JavaScript, PHP и др. Именно этой программой мы будем пользоваться на протяжении всей книги.

Скачать программу Notepad++ можно абсолютно бесплатно со страницы <http://notepad-plus.sourceforge.net/ru/site.htm>. Из двух вариантов (ZIP-архив и инсталлятор) советую выбрать именно инсталлятор, т. к. при установке можно будет указать язык интерфейса программы. Установка Notepad++ предельно проста и в комментариях не нуждается.

Запускаем программу Notepad++. В меню **Кодировки** устанавливаем флажок **Кодировать в ANSI**. Набираем код, представленный в листинге 1.1, а затем в меню **Файл** выбираем пункт **Сохранить как**. В открывшемся окне в строке **Имя файла** вводим "test.html". Выбираем папку, например Рабочий стол, и нажимаем **Сохранить**. Для просмотра открываем файл с помощью Web-браузера.

Чтобы открыть какой-либо файл на редактирование, в меню **Файл** выбираем пункт **Открыть** или щелкаем правой кнопкой мыши на ярлыке файла в Проводнике Windows и из контекстного меню выбираем пункт **Edit with Notepad++**.

ПРИМЕЧАНИЕ

Вместо Notepad++ можно воспользоваться редакторами PHP Expert Editor, Aptana Studio или NetBeans. Эти редакторы помимо подсветки синтаксиса предоставляют множество дополнительных функций. Тем не менее для быстрого редактирования файла удобнее пользоваться Notepad++. Описание редакторов вы найдете в *главе 4*.

1.3. Структура документа

Итак, мы изучили технологию создания HTML-документов, научились сохранять, отображать и изменять исходный код. Пришла пора вернуться к языку HTML. В листинге 1.2 представлена структура, характерная для любого HTML-документа.

Листинг 1.2. Структура HTML-документа

```
<!DOCTYPE> <!-- Объявление формата документа -->
<html>
```

```
<head>
  <!-- Техническая информация о документе -->
</head>
<body>
  <!-- Основная часть документа -->
</body>
</html>
```

Тег `<!DOCTYPE>` позволяет определить Web-браузеру формат файла и правильно отобразить все его инструкции. Допустимые форматы для HTML 4.01:

- **Strict** — строгий формат. Не содержит тегов и параметров, помеченных как устаревшие или не одобряемые. Объявление формата:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
  "http://www.w3.org/TR/html4/strict.dtd">
```

- **Transitional** — переходный формат. Содержит устаревшие теги в целях совместимости и упрощения перехода со старых версий HTML. Объявление формата:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
  "http://www.w3.org/TR/html4/loose.dtd">
```

- **Frameset** — аналогичен переходному формату, но содержит также теги для создания фреймов. Объявление формата:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Frameset//EN"
  "http://www.w3.org/TR/html4/frameset.dtd">
```

Если тег `<!DOCTYPE>` не указан, то Web-браузер Internet Explorer переходит в режим совместимости (Quirks Mode). В этом режиме отличается тип блочной модели. Поэтому при отсутствии тега `<!DOCTYPE>` разные Web-браузеры могут по-разному отображать Web-страницу.

ПРИМЕЧАНИЕ

Более подробную информацию о типах блочной модели можно получить в Интернете на странице консорциума W3C <http://www.w3.org/TR/CSS2/box.html> и на странице <http://www.quirksmode.org/css/quirksmode.html>.

Весь текст HTML-документа расположен между тегами `<html>` и `</html>`. HTML-документ состоит из двух разделов — заголовка (между тегами `<head>` и `</head>`) и содержательной части (между тегами `<body>` и `</body>`).

1.3.1. Раздел *HEAD*.

Техническая информация о документе

Раздел `HEAD` содержит техническую информацию о странице — заголовок, ее описание и ключевые слова для поисковых машин, данные об авторе и времени создания страницы, базовом адресе страницы, кодировке и т. д.

Единственным обязательным тегом в разделе `HEAD` является тег `<title>`. Текст, расположенный между тегами `<title>` и `</title>`, отображается в строке заголовка Web-браузера. Длина заголовка должна быть не более 60 символов, иначе он полностью не поместится в заголовке Web-браузера:

```
<title>Заголовок страницы</title>
```

СОВЕТ

Очень часто текст между тегами `<title>` и `</title>` используется в результатах, выдаваемых поисковым порталом, в качестве текста ссылки на эту страницу. По этой причине заголовок должен максимально полно описывать содержание страницы. Не следует писать что-то вроде "Главная страница", "Первая страница" и т. п.

С помощью одинарного тега `<meta>` можно задать описание содержимого страницы и ключевые слова для поисковых машин. Если текст между тегами `<title>` и `</title>` используется в качестве текста ссылки на эту страницу, то описание из тега `<meta>` будет отображено под ссылкой:

```
<meta name="description" content="Описание содержимого страницы">
<meta name="keywords" content="Ключевые слова через запятую">
```

Можно также указать несколько описаний на разных языках. Для этого в параметре `lang` следует указать используемый язык:

```
<meta name="description" lang="ru" content="Описание содержимого страницы">
<meta name="description" lang="en" content="Description">
<meta name="keywords" lang="ru" content="Ключевые слова через запятую">
<meta name="keywords" lang="en" content="Keywords">
```

Кроме того, тег `<meta>` позволяет запретить или разрешить индексацию Web-страницы поисковыми машинами:

```
<meta name="robots" content="<Индексация>, <Переход по ссылкам>">
```

В параметре `content` указывается комбинация следующих значений:

- `index` — индексация разрешена;
- `noindex` — индексация запрещена;
- `follow` — разрешено переходить по ссылкам, которые находятся на этой Web-странице;
- `nofollow` — запрещено переходить по ссылкам;
- `all` — комбинация `index` плюс `follow`;
- `none` — комбинация `noindex` плюс `nofollow`.

Приведем ряд примеров. Индексация и переход по ссылкам разрешены:

```
<meta name="robots" content="index, follow">
```

Индексация разрешена, а переход по ссылкам запрещен:

```
<meta name="robots" content="index, nofollow">
```

Индексация и переход по ссылкам запрещены:

```
<meta name="robots" content="noindex, nofollow">
```

Также с помощью тега `<meta>` можно указать кодировку текста:

```
<meta http-equiv="Content-Type" content="text/html; charset=windows-1251">
```

Для автоматической перезагрузки страницы через заданный промежуток времени следует воспользоваться свойством `refresh` тега `<meta>`:

```
<meta http-equiv="refresh" content="30">
```

В этом примере страница будет перезагружена через 30 секунд. Если необходимо сразу перебросить посетителя на другую страницу, то можно указать URL-адрес в параметре `url`:

```
<meta http-equiv="refresh" content="0; url=http://mail.ru/">
```

ПРИМЕЧАНИЕ

В разделе `HEAD` могут быть расположены также теги `<base>`, `<link>`, `<script>`, `<style>` и некоторые другие. Эти теги мы рассмотрим по мере изучения материала.

1.3.2. Раздел *BODY*. Основная часть документа

В этом разделе располагается все содержимое документа. Большинство тегов, рассмотренных в этой главе книги, должны находиться именно между тегами `<body>` и `</body>`.

Следует отметить, что в формате `Strict` содержимое тега `<body>` должно быть расположено внутри блочных элементов, например, `<p>`, `<div>` или др.:

```
<body>
  <p>Текст документа</p>
  <div>Текст документа</div>
</body>
```

Тег `<body>` имеет следующие параметры:

- ❑ `bghcolor` задает цвет фона Web-страницы. Даже если цветом фона является белый, все равно следует указать цвет.

Цвет определяется цифрами в шестнадцатеричном коде. Для каждой составляющей цвета (красного, зеленого и синего) задается значение в пределах от 00 до FF. Эти значения объединяются в одно число, перед которым добавляется символ "#", например, значение `#FF0000` соответствует красному цвету, `#00FF00` — ярко-зеленому, а `#FF00FF` — фиолетовому (смеси красного и синего);

- ❑ `background` позволяет задать фоновый рисунок для документа путем указания URL-адреса изображения;
- ❑ `alink` определяет цвет активной ссылки;

- link устанавливает цвет еще не просмотренных ссылок;
- vlink определяет цвет уже просмотренных ссылок;
- text устанавливает цвет текста.

Например, тег <body> может выглядеть так:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
  <title>Заголовок страницы</title>
  <meta http-equiv="Content-Type" content="text/html; charset=windows-1251">
</head>
<body alink="#FF0000" link="#000000" vlink="#000080" text="#000000">
Текст документа
</body>
</html>
```

ОБРАТИТЕ ВНИМАНИЕ

Все рассмотренные в этом разделе параметры являются устаревшими и поддерживаются только в формате Transitional. Использование их в формате Strict недопустимо.

Существуют и другие параметры, которые мы будем рассматривать по мере изучения языка.

1.4. Форматирование отдельных символов

Как уже говорилось, HTML — это язык разметки. Следовательно, важно уметь форматировать отдельные символы, а также целые фрагменты текста. Но прежде чем изучать теги, рассмотрим возможность отображения специальных символов. Такими символами, например, являются знаки меньше (<) и больше (>), т. к. с помощью этих символов описываются HTML-теги. Для отображения специальных символов используются так называемые *HTML-эквиваленты*. Например, для вывода такого текста

Текст между тегами <title> и </title> используется в результатах, выдаваемых поисковым порталом.

необходимо написать так

Текст между тегами <title> и </title> используется в результатах, выдаваемых поисковым порталом.

В этом примере мы заменили знак меньше (<) на <, а знак больше (>) — на >. Перечислим наиболее часто используемые HTML-эквиваленты:

- < — знак меньше (<);
- > — знак больше (>);

- `&` — амперсанд (&);
- ` ` — неразрывный пробел;
- `"` — кавычка ("");
- `©` — знак авторских прав (©);
- `®` — знак зарегистрированной торговой марки (®);
- `™` — торговая марка (™).

1.4.1. Выделение фрагментов текста

Тег `` отображает текст полужирным шрифтом:

```
<b>Полужирный шрифт</b>
```

Вместо тега `` лучше использовать тег логического форматирования ``:

```
<strong>Полужирный шрифт</strong>
```

Тег `<i>` отображает текст курсивом:

```
<i>Текст, выделенный курсивом</i>
```

Вместо тега `<i>` лучше использовать тег логического форматирования ``:

```
<em>Текст, выделенный курсивом</em>
```

Тег `<u>` отображает текст подчеркнутым:

```
<u>Подчеркнутый текст</u>
```

Теги `<strike>` и `<s>` отображают текст перечеркнутым:

```
<strike>Перечеркнутый текст</strike>
```

```
<s>Перечеркнутый текст</s>
```

ОБРАТИТЕ ВНИМАНИЕ

Теги `<u>`, `<strike>` и `<s>` являются устаревшими и поддерживаются только в формате Transitional. Использование их в формате Strict недопустимо.

1.4.2. Создание нижних и верхних индексов

Тег `<sub>` сдвигает текст ниже уровня строки и уменьшает размер шрифта. Он используется для создания нижних индексов, например, H₂O:

```
Формула воды H<sub>2</sub>O
```

Тег `<sup>` сдвигает текст выше уровня строки и уменьшает размер шрифта. Этот тег используется чаще всего для создания степеней, например, м²:

```
Единица измерения площади — м<sup>2</sup>
```

1.4.3. Вывод текста заданным шрифтом

Тег `` определяет размер, тип и цвет шрифта. Он имеет следующие параметры:

- `face` служит для указания типа шрифта:

```
<font face="Verdana">Текст</font>
```

Можно указать как один, так и несколько типов, разделяя их запятыми. При этом список шрифтов просматривается слева направо. Указанное название должно точно соответствовать названию типа шрифта. Если шрифт не найден на компьютере пользователя, то используется шрифт по умолчанию;

- `size` задает размер шрифта в условных единицах от 1 до 7. Размер, используемый Web-браузером по умолчанию, принято приравнять к 3. Размер шрифта можно указывать как цифрой от 1 до 7, так и в относительных единицах, указывая, на сколько единиц нужно увеличить (знак "+") или уменьшить (знак "-") размер шрифта относительно базового:

```
<font size="4">Текст</font>
```

```
<font size="+1">Текст</font>
```

```
<font size="-1">Текст</font>
```

- `color` позволяет указывать цвет шрифта. Цвета задаются так же, как для параметра `bgcolor` (см. разд. 1.3.2):

```
<font color="#FF0000">Текст</font>
```

Вместо цифр можно использовать названия цветов:

```
<font color="red">Текст</font>
```

Перечислим названия наиболее часто используемых цветов:

- `black` — #000000 — **черный**;
- `white` — #FFFFFF — **белый**;
- `yellow` — #FFFF00 — **желтый**;
- `silver` — #C0C0C0 — **серый**;
- `red` — #FF0000 — **красный**;
- `green` — #008000 — **зеленый**;
- `gray` — #808080 — **темно-серый**;
- `blue` — #0000FF — **синий**;
- `navy` — #000080 — **темно-синий**;
- `purple` — #800080 — **фиолетовый**.

ОБРАТИТЕ ВНИМАНИЕ

Тег `` является устаревшим и поддерживается только в формате Transitional. Использование его в формате Strict недопустимо.

Также для форматирования текста применяются и другие теги. Для вывода текста шрифтом большего размера используется парный тег `<big>`:

Текст `<big>большого</big>` размера

А для вывода текста шрифтом меньшего размера применяется парный тег `<small>`:

Текст `<small>меньшего</small>` размера

Для вывода текста моноширинным шрифтом используется тег `<tt>`:

`<tt>Моноширинный шрифт</tt>`

1.5. Форматирование документа

Практически все теги, рассмотренные в предыдущем разделе, являются тегами физического форматирования. Исключение составляют теги `` и ``. Эти теги являются тегами логического форматирования текста и используются для выделения очень важных и просто важных фрагментов соответственно. Теги логического форматирования используются для структурной разметки документа и могут отображаться разными Web-браузерами по-разному. Перечислим основные теги логического форматирования:

- `<cite>...</cite>` — применяется для отметки цитат, а также названий произведений;
- `<code>...</code>` — служит для отметки фрагментов программного кода;
- `<acronym>...</acronym>` — используется для отметки аббревиатур;
- `<kbd>...</kbd>` — отмечает фрагмент как вводимый пользователем с клавиатуры;
- `<q>...</q>` — используется для отметки коротких цитат;
- `<samp>...</samp>` — применяется для отметки результата, выдаваемого программой;
- `<var>...</var>` — отмечает имена переменных.

1.5.1. Тег комментария

Текст, заключенный между тегами `<!--` и `-->`, не отображается Web-браузером. Заметим, что это нестандартная пара тегов, т. к. открывающий тег не имеет закрывающей угловой скобки, а в закрывающем теге отсутствует открывающая угловая скобка:

```
<!-- Текст -->
```

СОВЕТ

Использование комментариев в исходном коде позволит быстро найти нужный фрагмент. Это особенно важно для начинающих Web-дизайнеров.

1.5.2. Перевод строки

Для разделения строк используется одинарный тег `
`.

Если в HTML-документе набрать текст

```
Строка1
Строка2
Строка3
```

то Web-браузер отобразит его в одну строку: "Строка1 Строка2 Строка3". Для того чтобы строки располагались друг под другом, необходимо добавить тег `
` в конец каждой строки:

```
Строка1<br>
Строка2<br>
Строка3<br>
```

Для вывода текста в том же виде, что и в исходном коде, можно воспользоваться парным тегом `<pre>`:

```
<pre>
Строка1
Строка2
Строка3
</pre>
```

В этом примере строки также будут располагаться друг под другом.

1.5.3. Горизонтальная линия

Одинарный тег `<hr>` позволяет провести горизонтальную линию.

Тег `<hr>` имеет следующие параметры:

- `size` — толщина линии:

```
<hr size="5">
```

- `width` — длина линии. Можно указывать значение как в пикселах, так и в процентах относительно ширины окна Web-браузера:

```
<hr size="5" width="100">
<hr size="5" width="100%">
```

- `align` — выравнивание линии. Параметр может принимать следующие значения:

- `center` — выравнивание по центру (значение по умолчанию):

```
<hr size="2" width="200" color="red" align="center">
```

- `left` — выравнивание по левому краю:

```
<hr size="2" width="200" color="red" align="left">
```

- `right` — выравнивание по правому краю:

```
<hr size="2" width="200" color="red" align="right">
```

- `noshade` — присутствие этого параметра отменяет рельефность линии:

```
<hr size="2" width="200" align="center" noshade>
```

ОБРАТИТЕ ВНИМАНИЕ

Все рассмотренные параметры тега `<hr>` являются устаревшими и поддерживаются только в формате `Transitional`. Использование их в формате `Strict` недопустимо.

1.5.4. Заголовки

Заголовки могут иметь шесть различных размеров:

```
<hx>Заголовок</hx>
```

где `x` — число от 1 до 6.

Заголовок с номером 1 является самым крупным:

```
<h1>Самый крупный заголовок</h1>
```

Заголовок с номером 6 является самым мелким:

```
<h6>Самый мелкий заголовок</h6>
```

Основным параметром является `align`, он задает выравнивание заголовка относительно окна Web-браузера. Он может принимать следующие значения:

- `center` — выравнивание по центру:

```
<h1 align="center">Заголовок первого уровня с выравниванием по центру</h1>
```

- `left` — выравнивание по левому краю (по умолчанию):

```
<h2 align="left">Заголовок второго уровня с выравниванием по левому краю</h2>
```

- `right` — выравнивание по правому краю:

```
<h6 align="right">Самый мелкий заголовок с выравниванием по правому краю</h6>
```

ОБРАТИТЕ ВНИМАНИЕ

Параметр `align` является устаревшим и поддерживается только в формате `Transitional`. Использование его в формате `Strict` недопустимо.

1.5.5. Разделение на абзацы

Тег `<p>` позволяет разбить текст на отдельные абзацы. Web-браузеры отделяют абзацы друг от друга пустой строкой. Закрывающий тег `</p>` не обязателен.

Основным параметром является `align`, он задает горизонтальное выравнивание. Параметр может принимать следующие значения:

❑ `center` — выравнивание по центру:

```
<p align="center">Абзац с выравниванием по центру</p>
```

❑ `left` — выравнивание по левому краю (по умолчанию):

```
<p align="left">Абзац с выравниванием по левому краю</p>
```

❑ `right` — выравнивание по правому краю:

```
<p align="right">Абзац с выравниванием по правому краю</p>
```

❑ `justify` — выравнивание по ширине (по двум сторонам):

```
<p align="justify">Абзац с выравниванием по ширине</p>
```

ОБРАТИТЕ ВНИМАНИЕ

Параметр `align` является устаревшим и поддерживается только в формате `Transitional`. Использование его в формате `Strict` недопустимо.

1.6. Списки

Список — это набор упорядоченных абзацев текста, помеченных специальными значками (маркированные списки) или цифрами (нумерованные списки). Рассмотрим каждый из вариантов в отдельности.

1.6.1. Маркированные списки

Маркированный список помещают внутри пары тегов `` и ``. Перед каждым пунктом списка необходимо поместить тег ``. Закрывающий тег `` не обязателен. В листинге 1.3 представлена структура маркированного списка.

Листинг 1.3. Маркированный список

```
<ul>
  <li>Первый пункт</li>
  <li>Второй пункт</li>
</ul>
```

Тег `` имеет параметр `type`, позволяющий задать значок, которым помечаются строки списка. Параметр может принимать следующие значения:

❑ `disc` — значки в форме кружков с заливкой:

```
<ul type="disc">
  <li>Первый пункт</li>
  <li>Второй пункт</li>
</ul>
```

- `circle` — значки в форме кружков без заливки:

```
<ul type="circle">
  <li>Первый пункт</li>
  <li>Второй пункт</li>
</ul>
```

- `square` — значки в форме квадрата с заливкой:

```
<ul type="square">
  <li>Первый пункт</li>
  <li>Второй пункт</li>
</ul>
```

ОБРАТИТЕ ВНИМАНИЕ

Параметр `type` является устаревшим и поддерживается только в формате `Transitional`. Использование его в формате `Strict` недопустимо.

1.6.2. Нумерованные списки

Нумерованный список помещают внутри пары тегов `` и ``. Перед каждым пунктом списка необходимо поместить тег ``. Закрывающий тег `` не обязателен.

В листинге 1.4 показана структура нумерованного списка.

Листинг 1.4. Нумерованный список

```
<ol>
  <li>Первый пункт</li>
  <li>Второй пункт</li>
</ol>
```

Тег `` имеет два параметра. Первый из них — `type` — позволяет задать формат, которым нумеруются строки списка.

Параметр может принимать следующие значения:

- `A` — пункты нумеруются прописными латинскими буквами:

```
<ol type="A">
  <li>Первый пункт</li>
  <li>Второй пункт</li>
</ol>
```

- `a` — пункты нумеруются строчными латинскими буквами:

```
<ol type="a">
  <li>Первый пункт</li>
  <li>Второй пункт</li>
</ol>
```


□ I — пункты нумеруются прописными римскими цифрами:

```
<ol type="I">
  <li>Первый пункт</li>
  <li>Второй пункт</li>
</ol>
```

□ i — пункты нумеруются строчными римскими цифрами:

```
<ol type="i">
  <li>Первый пункт</li>
  <li>Второй пункт</li>
</ol>
```

□ 1 — пункты нумеруются арабскими цифрами (по умолчанию):

```
<ol type="1">
  <li>Первый пункт</li>
  <li>Второй пункт</li>
</ol>
```

Второй параметр тега `` — `start` — задает номер, с которого будет начинаться нумерация строк:

```
<ol type="1" start="5">
  <li>Первый пункт</li>
  <li>Второй пункт</li>
</ol>
```

Тег `` также имеет параметр `value`, который позволяет изменить номер данного элемента списка:

```
<ol type="1">
  <li>Первый пункт</li>
  <li value="5">Второй пункт</li>
  <li>Третий пункт</li>
</ol>
```

В этом примере "Первый пункт" будет иметь номер 1, "Второй пункт" — номер 5, а "Третий пункт" — номер 6.

ОБРАТИТЕ ВНИМАНИЕ

Параметры `type`, `start` и `value` являются устаревшими и поддерживаются только в формате Transitional. Использование их в формате Strict недопустимо.

1.6.3. Списки определений

Списки определений состоят из пар "термин/определение". Описываются с помощью тега `<dl>`. Для вставки термина применяется тег `<dt>`, а для вставки определения — тег `<dd>`. Закрывающие теги `</dt>` и `</dd>` не обязательны. Пример использования списков определений приведен в листинге 1.5.

Листинг 1.5. Списки определений

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
    "http://www.w3.org/TR/html4/strict.dtd">
<html>
<head>
  <title>Списки определений</title>
  <meta http-equiv="Content-Type" content="text/html; charset=windows-1251">
</head>
<body>
  <dl>
    <dt>HTML (HyperText Markup Language)</dt>
    <dd>
      Язык разметки документа, описывающий форму отображения
      информации на экране компьютера
    </dd>
    <dt>CSS (Cascading Style Sheets)</dt>
    <dd>Каскадные таблицы стилей</dd>
  </dl>
</body>
</html>
```

1.7. Графика

Применение графики делает Web-страницу визуально привлекательнее. Изображения помогают лучше передать суть и содержание документа. В Интернете применяются графические форматы:

- GIF — использует только 256 цветов и поддерживает прозрачность. Кроме того, GIF-файл может содержать анимацию;
- JPEG — метод сжатия фотографий с потерей качества. Прозрачность и анимация не поддерживаются;
- PNG — формат хранения графики, использующий сжатие без потерь. Поддерживает прозрачность. Разрабатывался в качестве замены формата GIF.

ПРИМЕЧАНИЕ

Загромождение документа графикой приводит к увеличению времени загрузки Web-страницы. По этой причине применяйте графику только там, где это действительно оправданно.

1.7.1. Изображение на Web-странице

Изображения вставляются в Web-страницы с помощью одинарного тега ``. Сам тег `` должен быть расположен внутри блочного тега, например, `<p>`, `<div>` или др.

Тег имеет следующие параметры:

- `src` — URL-адрес файла графического изображения:

```
  

```

- `alt` — строка текста, которая будет выводиться на месте появления изображения до его загрузки или при отключенной графике, а также если изображение загрузить не удалось. Кроме того, при наведении курсора мыши на изображение текст, указанный в параметре `alt`, можно увидеть в качестве текста всплывающей подсказки:

```

```

- `width` — ширина изображения в пикселах:

```

```

- `height` — высота изображения в пикселах:

```

```

ПРИМЕЧАНИЕ

Значения параметров `width` и `height` могут не соответствовать реальным размерам изображения. В этом случае Web-браузер выполнит перемасштабирование. Если значение одного из параметров указать неправильно, то изображение будет искажено. Если указать только один параметр, то значение второго будет рассчитано пропорционально значению первого исходя из реальных размеров изображения.

СОВЕТ

Всегда указывайте значения параметров `width` и `height`, т. к. это позволит Web-браузеру отформатировать Web-страницу до загрузки изображений. В противном случае загрузка каждого изображения приведет к необходимости произвести форматирование еще раз, что в свою очередь приведет к перемещению других элементов Web-страницы. В результате картинка в окне Web-браузера будет дергаться.

Следующие параметры доступны только при использовании формата Transitional:

- `border` — толщина границы изображения:

```

```

- `align` — расположение изображения относительно текста или других элементов Web-страницы. Параметр может принимать следующие значения:

- `left` — изображение выравнивается по левому краю, а текст обтекает его с правой стороны:

```
<p>Текст</p>
```

- `right` — изображение выравнивается по правому краю, а текст обтекает его с левой стороны:

```
<p>Текст</p>
```

- top — изображение выравнивается по верху текущей строки:
`<p>Текст</p>`
- bottom — изображение выравнивается по низу текущей строки:
`<p>Текст</p>`
- middle — центр изображения выравнивается по базовой линии текущей строки:
`<p>Текст</p>`

□ hspace — отступ от изображения до текста по горизонтали:

```
<p>  
  Текст  
</p>
```

□ vspace — отступ от изображения до текста по вертикали:

```
<p>  
  Текст  
</p>
```

1.7.2. Изображение в качестве фона

Параметр background тега <body> позволяет задать фоновый рисунок для документа:

```
<body background="foto.gif" bgcolor="gray">Тело документа</body>
```

В параметре bgcolor следует указывать цвет, близкий к цвету фонового изображения, т. к. резкий переход, например, от светлого тона к темному вызовет неприятное мелькание, ведь фоновое изображение может загрузиться с некоторой задержкой.

ОБРАТИТЕ ВНИМАНИЕ

Эти параметры являются устаревшими и поддерживаются только в формате Transitional. Использование их в формате Strict недопустимо.

1.8. Гиперссылки

Гиперссылки позволяют нажатием кнопки мыши быстро перемещаться от одного документа к другому. Именно гиперссылки связывают все Web-страницы в единую сеть.

1.8.1. Внешние гиперссылки

Внешние гиперссылки вставляются в HTML-документ с помощью тега <a>. Сам тег <a> должен быть расположен внутри блочного тега, например, <p>, <div> или др.

Основным параметром тега `<a>` является `href`. Именно этот параметр задает URL-адрес Web-страницы, которая будет загружена при щелчке мыши на указателе. В качестве указателя может быть текст

```
<a href="http://www.mysite.ru/file.html">Текст ссылки</a>
```

или изображение

```
<a href="http://www.mysite.ru/file.html">  
</a>
```

Если URL-адрес содержит символ "&", то его необходимо заменить на HTML-эквивалент `&`:

```
<a href="index.php?id=5&name=Nik">Текст ссылки</a>
```

ПРИМЕЧАНИЕ

Кроме HTML-документов можно ссылаться и на файлы других типов, например, изображения, архивы и т. д. При переходе по такой ссылке Web-браузер в зависимости от типа файла либо отобразит его, либо предложит сохранить.

URL-адреса бывают абсолютными и относительными.

Абсолютный URL-адрес

Абсолютный URL-адрес содержит обозначение протокола, доменный или IP-адрес компьютера, путь к файлу, а также имя файла. Например:

```
http://www.mysite.ru/folder/file.html
```

Если файл находится в корневой папке, то путь может отсутствовать:

```
http://www.mysite.ru/file.html
```

Имя файла также может отсутствовать. В этом случае загружается Web-страница, заданная по умолчанию в настройках Web-сервера:

```
http://www.mysite.ru/
```

```
http://www.mysite.ru/folder/
```

Относительный URL-адрес

При относительном задании URL-адреса путь определяется с учетом местоположения Web-страницы, на которой находится ссылка. Возможны следующие варианты:

- если нужная Web-страница находится в той же папке, что и Web-страница, содержащая ссылку, то URL-адрес может содержать только имя файла. Если с Web-страницы, находящейся по адресу <http://www.mysite.ru/folder1/folder2/file1.html>, нужно перейти на <http://www.mysite.ru/folder1/folder2/file2.html>, то ссылка будет такой:

```
<a href="file2.html">Текст ссылки</a>
```

- если с Web-страницы, находящейся по адресу `http://www.mysite.ru/folder1/folder2/file1.html`, нужно перейти на `http://www.mysite.ru/folder1/folder2/folder3/file2.html`, то ссылке можно указать так:

```
<a href="folder3/file2.html">Текст ссылки</a>
```

- если с Web-страницы, находящейся по адресу `http://www.mysite.ru/folder1/folder2/file1.html`, нужно перейти на `http://www.mysite.ru/folder1/file2.html`, то ссылка будет такой:

```
<a href="../file2.html">Текст ссылки</a>
```

А при переходе с `http://www.mysite.ru/folder1/folder2/folder3/file1.html` на `http://www.mysite.ru/folder1/file2.html` — такой:

```
<a href="../../file2.html">Текст ссылки</a>
```

Очень часто необходимо загрузить документ в новое окно Web-браузера. Для этого в параметре `target` тега `<a>` следует указать значение `_blank`:

```
<a href="http://www.mysite.ru/file.html" target="_blank">Ссылка</a>
```

Другие значения параметра `target` мы рассмотрим при изучении фреймов (см. разд. 1.10.6).

ОБРАТИТЕ ВНИМАНИЕ

Использование параметра `target` в формате `Strict` недопустимо.

1.8.2. Внутренние гиперссылки

С помощью внутренних гиперссылок можно создать ссылки на разные разделы текущей Web-страницы. Если документ очень большой, то наличие внутренних гиперссылок позволяет быстро перемещаться между разделами.

Внутренняя гиперссылка также вставляется при помощи тега `<a>` с одним отличием — параметр `href` содержит имя указателя, а не URL-адрес. Перед именем указателя ставится знак `#`:

```
<a href="#chapter1">Глава 1</a>
```

Указатель создается с помощью тега `<a>`, но вместо параметра `href` используется параметр `name`, который задает имя указателя:

```
<a name="chapter1"></a>
```

Иногда указатель называют "якорем". Также можно сослаться на "якорь" другого документа. Это делается так:

```
<a href="http://www.mysite.ru/file.html#chapter6">Текст</a>
```

Структура документа с внутренними ссылками приведена в листинге 1.6.

Листинг 1.6. Структура документа с внутренними ссылками

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"  
"http://www.w3.org/TR/html4/strict.dtd">
```

```
<html>
<head>
  <title>Создание внутренних ссылок</title>
  <meta http-equiv="Content-Type" content="text/html; charset=windows-1251">
</head>
<body>
  <h1>Название документа</h1>
  <h2>Оглавление</h2>
  <ul>
    <li><a href="#chapter1">Глава 1</a></li>
    <li><a href="#chapter2">Глава 2</a></li>
    <li><a href="#chapter3">Глава 3</a></li>
    <li><a href="#chapter4">Глава 4</a></li>
  </ul>
  <h2><a name="chapter1"></a>Глава 1</h2>
  <p>Содержание главы 1</p>
  <h2><a name="chapter2"></a>Глава 2</h2>
  <p>Содержание главы 2</p>
  <h2><a name="chapter3"></a>Глава 3</h2>
  <p>Содержание главы 3</p>
  <h2><a name="chapter4"></a>Глава 4</h2>
  <p>Содержание главы 4</p>
</body>
</html>
```

1.8.3. Гиперссылки на адрес электронной почты

Ссылка на адрес электронной почты выглядит так:

```
<a href="mailto:mail@mysite.ru">Текст</a>
```

Вместо URL-адреса указывается адрес электронной почты, перед которым добавляется слово "mailto:".

СОВЕТ

Не следует публиковать ссылку с адресом электронной почты на сайте. Такие ссылки автоматически собираются роботами, и в дальнейшем этот E-mail будет завален спамом.

1.9. Таблицы

В HTML-документе таблицы используются в следующих случаях:

- как средство представления данных;
- как элемент оформления страницы, с помощью которого можно точно разместить на странице текст и графику.

Начнем со структуры, описывающей таблицу (листинг 1.7).

Листинг 1.7. Структура HTML-таблиц

```
<table border="1" width="200">
  <caption>Заголовок таблицы</caption>
  <tbody>
    <tr>
      <td align="center">1</td>
      <td align="center">2</td>
    </tr>
    <tr>
      <td align="center">3</td>
      <td align="center">4</td>
    </tr>
  </tbody>
</table>
```

Эта структура описывает таблицу 2×2 с заголовком. Значения в ячейках выровнены по центру. Все ячейки таблицы пронумерованы от 1 до 4.

Таблица вставляется в HTML-документ с помощью парного тега `<table>`. Отдельная ячейка таблицы описывается тегом `<td>`, а ряд ячеек — с помощью тега `<tr>`. Тег `<caption>` позволяет задать заголовок таблицы.

Для логического форматирования таблицы предназначены теги `<thead>` и `<tbody>`. Тег `<thead>` описывает заголовок таблицы, а тег `<tbody>` — основное содержимое таблицы. Закрывающие теги `</thead>` и `</tbody>` не обязательны.

1.9.1. Вставка таблицы в документ

Тег `<table>` имеет следующие параметры:

- ❑ `border` управляет отображением линий сетки таблицы, а также задает толщину рамки вокруг таблицы. По умолчанию сетка не отображается:

```
<table><!-- Здесь сетка не отображается -->
<table border="0"><!-- Здесь сетка не отображается -->
<table border="5"><!-- В этом случае сетка отображается, а
толщина рамки вокруг таблицы равна 5 пикселям -->
```

- ❑ `cellspacing` задает толщину линий сетки внутри таблицы, точнее сказать, расстояние между рамками соседних ячеек. По умолчанию параметр имеет значение 2. Если параметру присвоить значение 0, то рамки смежных ячеек сольются в одну линию:

```
<table cellspacing="0">
```

- ❑ `cellpadding` указывает размер отступа между рамкой ячейки и данными внутри ячейки:

```
<table cellpadding="2">
```

По умолчанию параметр имеет значение 1;

- `width` определяет ширину таблицы в пикселах или в процентах от размера окна:

```
<table width="200">
<table width="100%">
```

Следующие параметры доступны только при использовании формата `Transitional`:

- `align` задает выравнивание таблицы, а также обтекание таблицы текстом. Он может принимать следующие значения:

- `left` — таблица выравнивается по левому краю, а текст обтекает ее справа:

```
<table align="left">
```

- `right` — таблица выравнивается по правому краю, а текст обтекает ее слева:

```
<table align="right">
```

- `center` — таблица выравнивается по центру:

```
<table align="center">
```

- `bgcolor` указывает цвет фона таблицы:

```
<table bgcolor="silver">
<table bgcolor="#C0C0C0">
```

1.9.2. Заголовок таблицы

Тег `<caption>` позволяет задать заголовок таблицы. Он имеет единственный параметр `align`. Этот параметр может принимать одно из двух значений:

- `top` — заголовок помещается над таблицей:

```
<caption align="top">Заголовок таблицы</caption>
```

- `bottom` — заголовок располагается под таблицей:

```
<caption align="bottom">Заголовок таблицы</caption>
```

ОБРАТИТЕ ВНИМАНИЕ

Параметр `align` является устаревшим и поддерживается только в формате `Transitional`. Использование его в формате `Strict` недопустимо.

1.9.3. Строки таблицы

С помощью парного тега `<tr>` описываются строки таблицы. Он имеет следующие параметры:

- `align` указывает горизонтальное выравнивание текста в ячейках таблицы. Параметр может принимать следующие значения:

- `left` — по левому краю (по умолчанию):

```
<tr align="left">
```

- `right` — по правому краю:
`<tr align="right">`
 - `center` — по центру:
`<tr align="center">`
 - `justify` — по ширине:
`<tr align="justify">`
- `valign` определяет вертикальное выравнивание текста в ячейках таблицы. Он может принимать следующие значения:
- `top` — по верхнему краю:
`<tr valign="top">`
 - `middle` — по центру:
`<tr valign="middle">`
 - `bottom` — по нижнему краю:
`<tr valign="bottom">`
 - `baseline` — по базовой линии:
`<tr valign="baseline">`
- `bgcolor` указывает цвет фона ячеек таблицы. Параметр является устаревшим и поддерживается только в формате `Transitional`. Использование его в формате `Strict` недопустимо.

1.9.4. Ячейки таблицы

С помощью тега `<td>` описываются ячейки таблицы. Тег `<td>` имеет следующие параметры:

- `align` и `valign` выполняют те же функции, что и в теге `<tr>`;
- `width` и `height` определяют ширину и высоту ячейки в пикселах или в процентах;
- `bgcolor` указывает цвет фона ячейки;

ОБРАТИТЕ ВНИМАНИЕ

Параметры `width`, `height` и `bgcolor` являются устаревшими и поддерживаются только в формате `Transitional`. Использование их в формате `Strict` недопустимо.

- `colspan` задает количество объединяемых ячеек по горизонтали;
- `rowspan` указывает количество объединяемых ячеек по вертикали.

В качестве примера объединения ячеек возьмем наш первоначальный фрагмент кода (листинг 1.7) и объединим горизонтально расположенные ячейки 1 и 2 в одну (листинг 1.8).