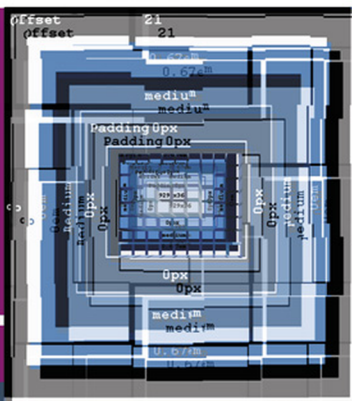


# HTML, JavaScript, PHP и MySQL

## Джентльменский набор Web-мастера

3-е издание

+ ВИДЕОКУРС



ОСНОВЫ CSS

ДИРЕКТИВЫ  
СЕРВЕРА Apache

ИСПОЛЬЗОВАНИЕ ПРОГРАММ  
phpMyAdmin, PHP Expert Editor,  
Aptana Studio, NetBeans,  
Notepad++, HeidiSQL и  
ШАБЛОНИЗАТОРА Smarty

РАБОТА  
С КОМАНДНОЙ СТРОКОЙ

ПУБЛИКАЦИЯ САЙТА  
В ИНТЕРНЕТЕ

ПРИМЕРЫ И СОВЕТЫ  
ИЗ ПРАКТИКИ

**Николай Прохоренок**

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

Санкт-Петербург

«БХВ-Петербург»

2010

**Прохоренок Н. А.**

П84 HTML, JavaScript, PHP и MySQL. Джентльменский набор Web-мастера. — 3-е изд., перераб. и доп. — СПб.: БХВ-Петербург, 2010. — 912 с.: ил. + Видеокурс (на CD-ROM) — (Профессиональное программирование)  
ISBN 978-5-9775-0540-6

Рассмотрены вопросы создания интерактивных Web-сайтов с помощью HTML, JavaScript, PHP и MySQL. Представлен материал о применении каскадных таблиц стилей (CSS) для форматирования Web-страниц. Даны основные конструкции языка PHP, на примерах показаны приемы написания сценариев, наиболее часто используемых при разработке Web-сайтов. Описаны приемы работы с базами данных MySQL при помощи PHP, а также администрирования баз данных с помощью программы phpMyAdmin. Особое внимание уделено созданию программной среды на компьютере разработчика и настройке Web-сервера Apache.

В 3-м издании книги описываются новые версии программ Apache 2.2.14, PHP 5.3.0 и MySQL 5.1.40. Добавлено описание программ Notepad++, Aptana Studio, NetBeans и HeidiSQL, рассмотрены возможности шаблонизатора Smarty, а также переработаны и дополнены все главы книги.

На прилагаемом компакт-диске содержатся листинги более чем двухсот примеров, описанных в книге, руководство по созданию динамического сайта, электронная версия самоучителя языка Perl и видеоролики.

*Для Web-разработчиков*

УДК 681.3.06  
ББК 32.973.26-018.2

**Группа подготовки издания:**

Главный редактор	<i>Екатерина Кондукова</i>
Зам. главного редактора	<i>Евгений Рыбаков</i>
Зав. редакцией	<i>Григорий Добин</i>
Редактор	<i>Екатерина Капалыгина</i>
Компьютерная верстка	<i>Натальи Смирновой</i>
Корректор	<i>Наталья Першакова</i>
Дизайн серии	<i>Инны Тачиной</i>
Оформление обложки	<i>Елены Беляевой</i>
Зав. производством	<i>Николай Тверских</i>

Лицензия ИД № 02429 от 24.07.00. Подписано в печать 30.01.10.

Формат 70×100<sup>1/16</sup>. Печать офсетная. Усл. печ. л. 73,53.

Тираж 2500 экз. Заказ №

"БХВ-Петербург", 190005, Санкт-Петербург, Измайловский пр., 29.

Санитарно-эпидемиологическое заключение на продукцию № 77.99.60.953.Д.005770.05.09 от 26.05.2009 г. выдано Федеральной службой по надзору в сфере защиты прав потребителей и благополучия человека.

Отпечатано с готовых диапозитивов  
в ГУП "Типография "Наука"  
199034, Санкт-Петербург, 9 линия, 12

# Оглавление

<b>ВВЕДЕНИЕ.....</b>	<b>1</b>
<b>ГЛАВА 1. ОСНОВЫ HTML. СОЗДАЕМ ДИЗАЙН САЙТА .....</b>	<b>5</b>
1.1. Основные понятия .....	5
1.2. Первый HTML-документ .....	9
1.3. Структура документа.....	12
1.3.1. Раздел <i>HEAD</i> . Техническая информация о документе.....	13
1.3.2. Раздел <i>BODY</i> . Основная часть документа .....	15
1.4. Форматирование отдельных символов.....	16
1.4.1. Выделение фрагментов текста.....	17
1.4.2. Создание нижних и верхних индексов .....	17
1.4.3. Вывод текста заданным шрифтом.....	18
1.5. Форматирование документа .....	19
1.5.1. Тег комментария .....	20
1.5.2. Перевод строки .....	20
1.5.3. Горизонтальная линия .....	21
1.5.4. Заголовки .....	22
1.5.5. Разделение на абзацы .....	22
1.6. Списки.....	23
1.6.1. Маркированные списки.....	23
1.6.2. Нумерованные списки .....	24
1.6.3. Списки определений.....	26
1.7. Графика.....	27
1.7.1. Изображение на Web-странице .....	27
1.7.2. Изображение в качестве фона .....	29
1.8. Гиперссылки.....	30
1.8.1. Внешние гиперссылки.....	30
1.8.2. Внутренние гиперссылки .....	32
1.8.3. Гиперссылки на адрес электронной почты .....	33
1.9. Таблицы .....	34
1.9.1. Вставка таблицы в документ .....	35
1.9.2. Заголовок таблицы.....	36
1.9.3. Строки таблицы .....	36
1.9.4. Ячейки таблицы .....	37

1.10. Фреймы .....	40
1.10.1. Разделение окна Web-браузера на несколько областей .....	40
1.10.2. Структура HTML-документа, содержащего фреймы .....	44
1.10.3. Описание фреймовой структуры .....	45
1.10.4. Описание отдельных областей .....	46
1.10.5. Тег <code>&lt;noframes&gt;</code> .....	47
1.10.6. Загрузка документа в определенный фрейм .....	47
1.10.7. Тег <code>&lt;iframe&gt;</code> . Добавление фрейма в обычный документ .....	48
1.11. Карты-изображения .....	50
1.11.1. Карта-изображение как панель навигации .....	51
1.11.2. Структура карт-изображений .....	52
1.11.3. Тег <code>&lt;map&gt;</code> .....	53
1.11.4. Описание активной области на карте-изображении .....	53
1.12. Формы .....	55
1.12.1. Создание формы для регистрации сайта .....	55
1.12.2. Структура документа с формами .....	57
1.12.3. Добавление формы в документ .....	57
1.12.4. Описание элементов управления .....	59
1.12.5. Тег <code>&lt;label&gt;</code> .....	64
1.12.6. Группировка элементов формы .....	67
1.13. Теги <code>&lt;div&gt;</code> и <code>&lt;span&gt;</code> . Группировка элементов страницы .....	67
1.14. Отличия XHTML 1.0 от HTML 4.01 .....	69
1.15. Проверка HTML-документов на соответствие стандартам .....	73
1.16. Специальный тег в Web-браузере Internet Explorer .....	74

## **ГЛАВА 2. ОСНОВЫ CSS. ФОРМАТИРУЕМ WEB-СТРАНИЦУ**

<b>С ПОМОЩЬЮ СТИЛЕЙ .....</b>	<b>77</b>
2.1. Основные понятия .....	77
2.2. Способы встраивания определения стиля .....	78
2.2.1. Встраивание определения стиля в тег .....	78
2.2.2. Встраивание определения стилей в заголовок HTML-документа .....	78
2.2.3. Вынесение таблицы стилей в отдельный файл .....	83
2.2.4. Приоритет применения стилей .....	85
2.3. Единицы измерения в CSS .....	87
2.4. Форматирование шрифта .....	88
2.4.1. Имя шрифта .....	88
2.4.2. Стиль шрифта .....	89
2.4.3. Размер шрифта .....	89
2.4.4. Цвет шрифта .....	89
2.4.5. Жирность шрифта .....	90

2.5. Форматирование текста.....	90
2.5.1. Расстояние между символами в словах .....	90
2.5.2. Расстояние между словами .....	90
2.5.3. Отступ первой строки.....	91
2.5.4. Вертикальное расстояние между строками.....	91
2.5.5. Горизонтальное выравнивание текста.....	91
2.5.6. Вертикальное выравнивание текста.....	92
2.5.7. Подчеркивание, надчеркивание и зачеркивание текста.....	92
2.5.8. Изменение регистра символов.....	93
2.5.9. Обработка пробелов между словами .....	93
2.6. Отступы .....	94
2.6.1. Внешние отступы.....	94
2.6.2. Внутренние отступы .....	95
2.7. Рамки.....	96
2.7.1. Стиль линии рамки .....	96
2.7.2. Толщина линии рамки .....	98
2.7.3. Цвет линии рамки .....	98
2.7.4. Одновременное задание атрибутов рамки .....	99
2.8. Фон элемента.....	99
2.8.1. Цвет фона.....	99
2.8.2. Фоновый рисунок.....	100
2.8.3. Режим повтора фонового рисунка .....	100
2.8.4. Прокрутка фонового рисунка .....	100
2.8.5. Положение фонового рисунка .....	101
2.8.6. Одновременное задание атрибутов фона .....	101
2.9. Списки.....	102
2.9.1. Вид маркера списка .....	102
2.9.2. Изображение в качестве маркера списка.....	102
2.9.3. Компактное отображение списка .....	103
2.10. Вид курсора .....	103
2.11. Псевдостили гиперссылок. Отображение ссылок разными цветами .....	104
2.12. Форматирование блоков .....	106
2.12.1. Указание типа блока .....	106
2.12.2. Установка размеров .....	108
2.12.3. Атрибут <i>overflow</i> .....	109
2.12.4. Управление обтеканием .....	111
2.12.5. Позиционирование блока.....	113
2.12.6. Последовательность отображения слоев.....	115
2.13. Управление отображением элемента.....	117
2.14. Проверка CSS-кода на соответствие стандартам .....	119

## ГЛАВА 3. ОСНОВЫ JAVASCRIPT.

ДЕЛАЕМ СТРАНИЦЫ, РЕАГИРУЮЩИЕ НА ДЕЙСТВИЯ ПОЛЬЗОВАТЕЛЕЙ.....	121
3.1. Основные понятия .....	121
3.2. Первая программа на JavaScript .....	121
3.3. Комментарии в JavaScript .....	124
3.4. Вывод результатов работы программы и ввод данных .....	124
3.4.1. Окно с сообщением и кнопкой <i>OK</i> .....	125
3.4.2. Окно с сообщением и кнопками <i>OK</i> и <i>Cancel</i> .....	126
3.4.3. Окно с полем ввода и кнопками <i>OK</i> и <i>Cancel</i> .....	127
3.5. Переменные.....	127
3.6. Типы данных и инициализация переменных.	
Определение типа данных переменной.....	128
3.7. Операторы JavaScript.....	130
3.7.1. Математические операторы.....	130
3.7.2. Операторы присваивания.....	132
3.7.3. Двоичные операторы .....	132
3.7.4. Оператор обработки строк .....	133
3.7.5. Приоритет выполнения операторов .....	134
3.8. Преобразование типов данных.....	135
3.9. Специальные символы. Разбиение сообщения в диалоговом окне на несколько строк.....	138
3.10. Массивы.....	139
3.11. Функции. Разделение программы на фрагменты .....	142
3.11.1. Основные понятия .....	142
3.11.2. Расположение функций внутри HTML-документа.....	144
3.11.3. Рекурсия. Вычисление факториала .....	146
3.11.4. Глобальные и локальные переменные.....	147
3.12. Условные операторы. Выполнение блоков кода только при соответствии условию.....	149
3.12.1. Операторы сравнения.....	149
3.12.2. Оператор ветвления <i>if...else</i> . Проверка ввода пользователя.....	150
3.12.3. Оператор <i>?</i> Проверка числа на четность .....	153
3.12.4. Оператор выбора <i>switch</i> .....	154
3.13. Операторы циклов. Многократное выполнение блока кода .....	156
3.13.1. Цикл <i>for</i> .....	156
3.13.2. Цикл <i>while</i> .....	158
3.13.3. Цикл <i>do...while</i> .....	159
3.13.4. Оператор <i>continue</i> . Переход на следующую итерацию цикла.....	160
3.13.5. Оператор <i>break</i> . Прерывание цикла.....	160

3.14. Ошибки в программе.....	161
3.14.1. Синтаксические ошибки .....	161
3.14.2. Логические ошибки .....	162
3.14.3. Ошибки времени выполнения .....	162
3.14.4. Обработка ошибок .....	163
3.14.5. Модуль Firebug для Web-браузера Firefox.....	164
3.15. Встроенные классы JavaScript .....	167
3.15.1. Основные понятия .....	167
3.15.2. Класс <i>Global</i> .....	168
3.15.3. Класс <i>Number</i> . Работа с числами.....	170
3.15.4. Класс <i>String</i> . Обработка строк .....	171
3.15.5. Класс <i>Array</i> . Работа с массивами и их сортировка.....	173
3.15.6. Класс <i>Math</i> . Использование математических функций .....	180
3.15.7. Класс <i>Date</i> . Получение текущей даты и времени. Вывод даты и времени в окне Web-браузера .....	182
3.15.8. Класс <i>Function</i> (функции) .....	186
3.15.9. Класс <i>Arguments</i> . Функции с произвольным количеством аргументов .....	187
3.15.10. Класс <i>RegExp</i> . Проверка значений с помощью регулярных выражений .....	188
3.16. События .....	199
3.16.1. Основные понятия .....	199
3.16.2. События мыши .....	199
3.16.3. События клавиатуры.....	200
3.16.4. События документа.....	200
3.16.5. События формы.....	201
3.16.6. Последовательность событий .....	201
3.16.7. Всплывание событий .....	203
3.16.8. Действия по умолчанию и их отмена.....	205
3.16.9. Написание обработчиков событий .....	207
3.16.10. Объект <i>event</i> . Вывод координат курсора и кода нажатой клавиши. Вывод сообщений при нажатии комбинации клавиш .....	214
3.17. Объектная модель Microsoft Internet Explorer .....	220
3.17.1. Структура объектной модели .....	220
3.17.2. Объект <i>window</i> . Вывод сообщения в строку состояния Web-браузера.....	221
3.17.3. Работа с окнами. Создание нового окна без строки меню, адресной строки и панели инструментов .....	227
3.17.4. Модальные диалоговые окна. Использование модальных окон вместо встроенных диалоговых окон .....	231
3.17.5. Таймеры. Создание часов на Web-странице .....	234



3.17.6. Объект <i>navigator</i> . Получение информации о Web-браузере пользователя. Перенаправление клиента на разные страницы в зависимости от Web-браузера.....	236
3.17.7. Объект <i>screen</i> . Получение информации о мониторе пользователя.....	241
3.17.8. Объект <i>location</i> . Разбор составляющих URL-адреса документа. Создание многостраничных HTML-документов.....	242
3.17.9. Объект <i>history</i> . Получение информации о просмотренных страницах. Реализация перехода на предыдущую просмотренную страницу.....	247
3.17.10. Объект <i>document</i> . Получение полной информации о HTML-документе.....	248
3.17.11. Обращение к элементам документа. Выравнивание заголовков по центру.....	255
3.17.12. Работа с элементами документа. Изменение URL-адреса и текста ссылки. Преобразование ссылки в обычный текст.....	257
3.17.13. Объект <i>style</i> . Работа с таблицами стилей при помощи JavaScript.....	265
3.17.14. Объект <i>selection</i> . Проверка наличия выделенного фрагмента.....	268
3.17.15. Объект <i>TextRange</i> . Поиск фрагмента в текстовом поле или документе. Расширение или сжатие выделенного фрагмента текста.....	272
3.17.16. Работа с буфером обмена. Выделение фрагмента от позиции щелчка до конца документа и копирование его в буфер обмена.....	282
3.17.17. Реализация ссылок "Добавить сайт в Избранное" и "Сделать стартовой страницей".....	284
3.17.18. Сохранение данных на компьютере клиента. Определение возможности использования cookies. Сохранение русского текста в cookies.....	285
3.18. Работа с элементами формы.....	289
3.18.1. Элементы управления.....	290
3.18.2. Коллекция <i>Forms</i> . Доступ к элементу формы из скрипта.....	291
3.18.3. Свойства объекта формы.....	291
3.18.4. Методы объекта формы.....	292
3.18.5. События объекта формы.....	292
3.18.6. Текстовое поле и поле ввода пароля. Проверка правильности ввода E-mail и пароля. Получение данных из элемента формы.....	292
3.18.7. Поле для ввода многострочного текста. Добавление слов из текстового поля в поле <code>&lt;textarea&gt;</code> .....	295
3.18.8. Список с возможными значениями. Возможность добавления нового пункта. Применение списков вместо гиперссылок.....	298

3.18.9. Флажок и переключатели. Получение значения выбранного переключателя при помощи цикла и проверка установки флажка.....	303
3.18.10. Кнопки. Обработка нажатия кнопки. Деактивация кнопки. Создание клавиши быстрого доступа и вывод текста на кнопке определенным цветом .....	306
3.18.11. Проверка корректности данных. Создание формы регистрации пользователя.....	309
3.19. Пользовательские объекты .....	314
3.19.1. Создание объектов.....	314
3.19.2. Прототипы .....	318
3.19.3. Пространства имен .....	321
3.20. JavaScript-библиотеки.....	323

## **ГЛАВА 4. ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ WEB-СЕРВЕРА.**

### **УСТАНОВЛИВАЕМ И НАСТРАИВАЕМ ПРОГРАММЫ ПОД WINDOWS .....325**

4.1. Необходимые программы .....	325
4.2. Установка сервера Apache .....	326
4.3. Структура каталогов сервера Apache.....	333
4.4. Файл конфигурации httpd.conf .....	334
4.4.1. Основные понятия .....	334
4.4.2. Разделы файла конфигурации.....	335
4.4.3. Общие директивы. Создание домашней директории пользователя, доступной при запросе http://localhost/~nik/ .....	337
4.4.4. Директивы управления производительностью .....	339
4.4.5. Директивы обеспечения постоянного соединения .....	340
4.4.6. Директивы работы с языками .....	341
4.4.7. Директивы перенаправления .....	341
4.4.8. Обработка ошибок .....	342
4.4.9. Настройки MIME-типов .....	343
4.4.10. Управление листингом каталога .....	345
4.4.11. Директивы протоколирования.....	348
4.4.12. Файл конфигурации .htaccess. Управляем сервером Apache из обычной папки.....	350
4.4.13. Защита содержимого папки паролем .....	351
4.4.14. Управление доступом.....	355
4.4.15. Регулярные выражения, используемые в директивах .....	356
4.4.16. Создание виртуальных серверов .....	357
4.5. Настройка сервера Apache .....	360
4.6. Установка PHP .....	362
4.7. Установка MySQL.....	370

4.8. Установка phpMyAdmin .....	379
4.9. Знакомьтесь — Денвер .....	385
4.9.1. Установка Денвера .....	385
4.9.2. Запуск и остановка Денвера .....	393
4.9.3. Создание виртуальных хостов .....	393
4.9.4. Конфигурационные файлы Денвера .....	395
4.10. Установка и настройка PHP Expert Editor .....	395
4.11. Установка и настройка Aptana Studio .....	401
4.12. Установка и настройка NetBeans .....	412
4.13. Программа HeidiSQL .....	418

## **Глава 5. ОСНОВЫ PHP. СОЗДАЕМ ДИНАМИЧЕСКИЕ WEB-СТРАНИЦЫ.....421**

5.1. Основные понятия .....	421
5.2. Первая программа на PHP .....	421
5.3. Методы встраивания PHP-кода .....	425
5.4. Комментарии в PHP-сценариях .....	426
5.5. Вывод результатов работы скрипта .....	427
5.6. Переменные .....	429
5.7. Типы данных и инициализация переменных .....	429
5.8. Проверка существования переменной .....	431
5.9. Удаление переменной .....	432
5.10. Константы. Создание и использование констант .....	433
5.11. Операторы PHP .....	434
5.11.1. Математические операторы .....	435
5.11.2. Операторы присваивания .....	436
5.11.3. Двоичные операторы .....	436
5.11.4. Оператор конкатенации строк. Подстановка значений переменных. Запуск внешних программ .....	437
5.11.5. Приоритет выполнения операторов .....	440
5.12. Преобразование типов данных .....	441
5.13. Специальные символы .....	443
5.14. Массивы .....	444
5.14.1. Инициализация массива .....	444
5.14.2. Получение и изменение элемента массива. Определение количества элементов массива .....	444
5.14.3. Многомерные массивы .....	445
5.14.4. Ассоциативные массивы .....	445
5.14.5. Слияние массивов .....	447
5.14.6. Перебор элементов массива .....	447
5.14.7. Добавление и удаление элементов массива .....	451

5.14.8. Переворачивание и перемешивание массива.....	452
5.14.9. Сортировка массива. Создание пользовательской сортировки .....	453
5.14.10. Получение части массива.....	455
5.14.11. Преобразование переменных в массив .....	456
5.14.12. Преобразование массива в переменные .....	456
5.14.13. Заполнение массива числами.....	457
5.14.14. Преобразование массива в строку.....	458
5.14.15. Проверка наличия значения в массиве .....	459
5.15. Строки.....	460
5.15.1. Функции для работы со строками .....	460
5.15.2. Настройка локали.....	465
5.15.3. Функции для работы с символами .....	466
5.15.4. Поиск и замена в строке.....	466
5.15.5. Функции для сравнения строк .....	467
5.15.6. Кодирование строк .....	468
5.15.7. Преобразование кодировок.....	469
5.15.8. Регулярные выражения. Разбираем адрес электронной почты на составные части. Проверяем правильность введенной даты.....	470
5.15.9. Perl-совместимые регулярные выражения .....	478
5.15.10. Функции для работы со строками в кодировке UTF-8.....	487
5.15.11. Перегрузка строковых функций .....	496
5.16. Функции для работы с числами.....	497
5.17. Функции для работы с датой и временем. Получение текущей даты, даты создания файла и проверка корректности введенной даты .....	499
5.18. Функции. Разделение программы на фрагменты .....	503
5.18.1. Основные понятия .....	503
5.18.2. Расположение описаний функций.....	505
5.18.3. Операторы <i>require</i> и <i>include</i> . Выносим функции в отдельный файл. Создаем шаблоны для множества страниц.....	506
5.18.4. Операторы <i>require_once</i> и <i>include_once</i> .....	509
5.18.5. Рекурсия. Вычисляем факториал .....	510
5.18.6. Глобальные и локальные переменные. Передача параметров по ссылке. Использование глобальных переменных внутри функций.....	510
5.18.7. Статические переменные .....	514
5.18.8. Переменное число параметров в функции. Сумма произвольного количества чисел.....	515
5.19. Условные операторы. Выполнение блоков кода только при соответствии условию.....	516
5.19.1. Операторы сравнения.....	516
5.19.2. Оператор ветвления <i>if...else</i> . Проверка выбранного элемента из списка .....	518

5.19.3. Оператор ? Проверка числа на четность .....	520
5.19.4. Оператор выбора <i>switch</i> . Использование оператора <i>switch</i> вместо <i>if...else</i> .....	522
5.20. Операторы циклов. Многократное выполнение блока кода .....	524
5.20.1. Цикл <i>for</i> .....	524
5.20.2. Цикл <i>while</i> .....	526
5.20.3. Цикл <i>do...while</i> .....	526
5.20.4. Цикл <i>foreach</i> .....	527
5.20.5. Оператор <i>continue</i> . Переход на следующую итерацию цикла .....	528
5.20.6. Оператор <i>break</i> . Прерывание цикла .....	529
5.21. Завершение выполнения сценария. Навигация при выборе значения из списка .....	529
5.22. Ошибки в программе .....	531
5.22.1. Синтаксические ошибки .....	531
5.22.2. Логические ошибки .....	532
5.22.3. Ошибки времени выполнения .....	532
5.22.4. Обработка ошибок .....	532
5.22.5. Инструкция <i>or die()</i> .....	534
5.23. Переменные окружения .....	534
5.23.1. Массив <i>\$GLOBALS</i> .....	535
5.23.2. Часто используемые переменные окружения .....	538
5.24. Заголовки HTTP .....	539
5.24.1. Основные заголовки .....	542
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 .....	544
5.24.3. Работа с cookies. Создаем индивидуальный счетчик посещений .....	548
5.25. Работа с файлами и каталогами .....	549
5.25.1. Основные понятия .....	549
5.25.2. Функции для работы с файлами. Создание файла, запись в файл, вывод содержимого файла в список .....	550
5.25.3. Перемещение внутри файла .....	554
5.25.4. Создание списка рассылки с возможностью добавления, изменения и удаления E-mail-адресов .....	555
5.25.5. Чтение CSV-файлов. Преобразование CSV-файла в HTML-таблицу .....	560
5.25.6. Права доступа в операционной системе UNIX .....	562
5.25.7. Функции для манипулирования файлами .....	564
5.25.8. Загрузка файлов на сервер .....	566
5.25.9. Функции для работы с каталогами. Создаем программу для просмотра всех доступных каталогов и файлов на диске .....	568
5.25.10. Получение информации из сети Интернет .....	572

5.26. Отправка писем с сайта. Рассылка писем по E-mail-адресам из файла .....	583
5.27. Аутентификация с помощью PHP. Создание Личного кабинета.....	587
5.28. Работа с графикой.....	592
5.28.1. Информация об установленной библиотеке GD.....	592
5.28.2. Получение информации об изображении.....	593
5.28.3. Работа с готовыми изображениями.....	597
5.28.4. Создание нового изображения .....	599
5.28.5. Работа с цветом .....	600
5.28.6. Рисование линий и фигур.....	602
5.28.7. Вывод текста в изображение. Создаем счетчик посещений.....	604
5.28.8. Изменение размеров и копирование изображений.....	609
5.29. Обработка данных формы.....	612
5.29.1. Текстовое поле, поле ввода пароля и скрытое поле .....	612
5.29.2. Поле для ввода многострочного текста .....	613
5.29.3. Список с возможными значениями.....	613
5.29.4. Флажок.....	615
5.29.5. Элемент-переключатель.....	616
5.29.6. Кнопка <i>Submit</i> .....	616
5.29.7. Проверка корректности данных. Создание формы регистрации пользователя.....	617
5.30. Другие полезные функции .....	622
5.30.1. Выделение фрагментов исходного кода .....	623
5.30.2. Получение информации об интерпретаторе.....	623
5.30.3. Вывод всех доступных сценарию функций.....	623
5.30.4. Засыпание сценария.....	624
5.30.5. Изменение значения директив во время выполнения сценария .....	625
5.30.6. Выполнение команд, содержащихся в строке .....	627
5.31. Объектно-ориентированное программирование .....	627
5.31.1. Создание класса .....	628
5.31.2. Конструктор и деструктор.....	629
5.31.3. Наследование .....	631
5.31.4. Статические свойства и методы .....	633
5.31.5. Объявление констант внутри класса.....	634
5.31.6. Определение области видимости .....	634
5.31.7. Создание шаблона сайта при помощи класса .....	637
5.32. Шаблонизатор Smarty.....	639
5.32.1. Установка и настройка .....	639
5.32.2. Управляющие конструкции .....	643
5.32.3. Модификаторы переменных .....	649
5.32.4. Кэширование страниц .....	654

**ГЛАВА 6. ОСНОВЫ MYSQL. РАБОТАЕМ С БАЗАМИ ДАННЫХ.....661**

6.1. Основные понятия .....	661
6.2. Нормализация базы данных.....	661
6.3. Типы данных полей .....	665
6.3.1. Числовые типы .....	666
6.3.2. Строковые типы .....	666
6.3.3. Дата и время .....	667
6.4. Основы языка SQL.....	668
6.4.1. Создание базы данных .....	668
6.4.2. Создание пользователя базы данных .....	669
6.4.3. Создание таблицы.....	671
6.4.4. Вставка данных в таблицу .....	674
6.4.5. Обновление записей .....	677
6.4.6. Удаление записей из таблицы .....	678
6.4.7. Изменение свойств таблицы .....	679
6.4.8. Выбор записей.....	680
6.4.9. Выбор записей из нескольких таблиц.....	682
6.4.10. Индексы. Ускорение выполнения запросов .....	687
6.4.11. Удаление таблицы и базы данных.....	692
6.5. Доступ к базе данных из PHP с помощью библиотеки <code>php_mysql.dll</code> .....	693
6.5.1. Установка соединения.....	693
6.5.2. Выбор базы данных .....	694
6.5.3. Выполнение запроса к базе данных .....	694
6.5.4. Обработка результата запроса .....	695
6.6. Доступ к базе данных из PHP с помощью библиотеки <code>php_mysqli.dll</code> .....	702
6.6.1. Установка соединения.....	702
6.6.2. Выбор базы данных .....	704
6.6.3. Выполнение запроса к базе данных .....	704
6.6.4. Обработка результата запроса .....	706
6.7. Операторы MySQL .....	714
6.7.1. Математические операторы .....	715
6.7.2. Двоичные операторы .....	717
6.7.3. Операторы сравнения .....	717
6.7.4. Приоритет выполнения операторов .....	719
6.7.5. Преобразование типов данных .....	720
6.8. Поиск по шаблону .....	721
6.9. Поиск с помощью регулярных выражений.....	725
6.10. Режим полнотекстового поиска .....	729
6.10.1. Создание индекса <i>FULLTEXT</i> .....	729
6.10.2. Реализация полнотекстового поиска.....	731

6.10.3. Режим логического поиска .....	732
6.10.4. Поиск с расширением запроса .....	733
6.11. Функции MySQL .....	733
6.11.1. Функции для работы с числами .....	733
6.11.2. Функции даты и времени .....	738
6.11.3. Функции для обработки строк .....	750
6.11.4. Функции для шифрования строк .....	757
6.11.5. Информационные функции .....	758
6.11.6. Прочие функции .....	760
6.12. Переменные SQL .....	764
6.13. Временные таблицы .....	765
6.14. Вложенные запросы .....	767
6.14.1. Заполнение таблицы с помощью вложенного запроса .....	768
6.14.2. Применение вложенных запросов в инструкции <i>WHERE</i> .....	770
6.15. Внешние ключи .....	772

## **ГЛАВА 7. ПУБЛИКАЦИЯ САЙТА. ДЕЛАЕМ САЙТ ДОСТУПНЫМ ДЛЯ ВСЕХ .....775**

7.1. Определение цели .....	775
7.2. Выбор доменного имени .....	776
7.3. Виды хостинга .....	779
7.4. Бесплатный хостинг Narod.ru .....	780
7.4.1. Регистрация и обзор возможностей .....	781
7.4.2. Создание страницы "Обратная связь" .....	782
7.4.3. Загрузка контента на сервер .....	784
7.4.4. Управление гостевой книгой, форумом и чатом .....	786
7.5. Платный виртуальный хостинг .....	791
7.5.1. Выбор тарифного плана .....	791
7.5.2. Регистрация и обзор возможностей .....	792
7.5.3. Структура Контрольной панели .....	793
7.5.4. Структура каталогов сервера и загрузка контента на сервер .....	796
7.5.5. Настройка Web-сервера Apache с помощью файла <i>.htaccess</i> .....	804
7.5.6. Файл <i>favicon.ico</i> .....	806
7.5.7. Защита содержимого папки с помощью Web-сервера Apache .....	807
7.5.8. Загрузка файлов на сервер с помощью формы .....	809
7.5.9. Создание базы данных MySQL .....	812
7.5.10. Управление базой данных с помощью <i>phpMyAdmin</i> .....	813
7.5.11. Отправка почты с сайта .....	817
7.5.12. Анализ статистики и работа с логами сервера .....	818
7.5.13. Автоматический запуск программ в заданное время .....	820



7.6. Раскрутка сайта .....	823
7.6.1. Подготовка страниц сайта к индексации .....	823
7.6.2. Файл robots.txt .....	824
7.6.3. Регистрация в каталогах и рейтингах .....	825
7.6.4. Участие в баннерообменных сетях .....	827
7.6.5. Обмен ссылками с Web-мастерами .....	827
7.7. Заработок в сети .....	828
7.7.1. Партнерские программы .....	828
7.7.2. Рекламные сети .....	830
7.7.3. Электронные деньги .....	831
7.8. Перечень полезных сайтов .....	832

**ПРИЛОЖЕНИЕ. ОПИСАНИЕ КОМПАКТ-ДИСКА .....835**

П.1. Видеоролики.....	835
П.1.1. HTML .....	836
П.1.2. Работа с изображениями .....	844
П.1.3. Flash .....	851
П.1.4. PHP и MySQL.....	859
П.1.5. Редакторы .....	862

**ПРЕДМЕТНЫЙ УКАЗАТЕЛЬ .....869**

# 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 технологии нельзя, так как они могут существовать только совместно, а значит, и изучать их нужно только как единое целое.

Все главы книги расположены в порядке возрастания уровня сложности материала. Если вы начинающий Web-мастер, то книгу следует изучать именно в порядке расположения глав. Исключение составляет лишь *глава 7*, в которой говорится о размещении сайта в сети Интернет. К ней можно обратиться когда угодно, так как для создания простого статического сайта достаточно знать лишь язык разметки HTML. Если материал какой-либо из глав был изучен ранее, то можно сразу переходить к изучению следующей главы.

Что же можно создать с использованием изучаемых технологий? Давайте рассмотрим возможности этих технологий, а также предназначение 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.

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.

У Web-страниц, созданных с использованием HTML и CSS, есть существенный недостаток — они являются статическими, то есть не могут меняться, реагируя на действия пользователя. Внедрение в HTML программ на языке JavaScript 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 изменять 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 и отображать элементы Web-страницы и многое другое. Как все это сделать, рассказано в *главе 3*.

Еще большие возможности дает использование серверных технологий, 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 технологии, применяемые для создания динамических Web-страниц), можно изменять HTML-код, получаемый Web-браузером, в зависимости от вводимых пользователем данных, типа и версии используемого Web-браузера и других факторов. Огромное количество расширений и готовых программных продуктов, а также легкость освоения языка 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, размещать на сайте гостевую книгу, форум, чат, интернет-магазин и многое другое. Писать программы на PHP мы научимся в *главе 5*.

А в *главе 4* рассказывается, как установить и настроить специальное программное обеспечение для тестирования скриптов на PHP. Это позволит изучить основные настройки программ и удалить все ошибки из скриптов до их загрузки на сервер. Ведь сайт может стать очень популярным, а посетителям не очень понравится увидеть вместо необходимой информации сообщение об ошибке. Кроме того, в *главе 4* мы рассмотрим установку и настройку специализированных редакторов, которые позволяют значительно упростить создание сайта и сделают процесс изучения материала книги более эффективным.

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. Использование MySQL обсуждается в *главе 6*. PHP обеспечивает эффективную поддержку баз данных и позволяет работать с MySQL, Microsoft SQL Server, Oracle, Sybase и др. Зная и умея работать с MySQL, легко перейти и к другим базам данных, если возникнет такая необходимость.

В главе 7 мы рассмотрим все проблемы, связанные с размещением сайта в Интернете. Мы узнаем, как подбирать подходящую площадку для сайта, научимся работать с FTP и командной строкой, произведем настройку сервера Apache, изучим возможность автоматического запуска программ в определенное время, подготовим сайт к индексации и т. д.

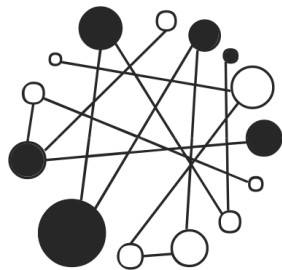
На главе 7 наше знакомство с Web-технологиями не заканчивается. На прилагаемом компакт-диске вы найдете описание фильтров и преобразований, которые можно использовать в Web-браузере Internet Explorer, а также электронную версию самоучителя языка Perl. Кроме того, на компакт-диске расположено описание процесса создания динамического сайта с использованием всех изученных технологий. Мы создадим полноценный каталог сайтов, включающий личный кабинет для пользователей с защитой средствами PHP, а также личный кабинет для администратора, защищенный средствами сервера Apache. Создаваемые программы научат правильно обрабатывать данные формы и работать с базами данных.

Все листинги из книги можно найти на прилагаемом компакт-диске. Настоятельно рекомендую обязательно рассматривать все примеры из книги и вначале самостоятельно набирать код. При наборе вы создадите множество ошибок. Именно умение находить эти ошибки сделает из вас настоящего Web-мастера.

Ваши замечания и пожелания вы можете оставить в гостевой книге на сайте <http://wwwadmin.ru/>. Все замеченные опечатки прошу присылать на E-mail [mail@bhv.ru](mailto:mail@bhv.ru).

Желаю приятного прочтения и надеюсь, что эта книга станет верным спутником в вашей повседневной жизни.

# 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, так как 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 `<br>`.

Некоторые теги могут иметь *параметры* (иногда их называют *атрибутами*). Параметры указываются после имени тега через пробел в формате пара-

метр="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, советую 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>`.

- 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](http://tinymce.moxiecode.com/download_i18n.php) необходимо скачать архив с файлами для русского языка. Архив следует разместить в папке `tinymce\jscripsts\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({
```

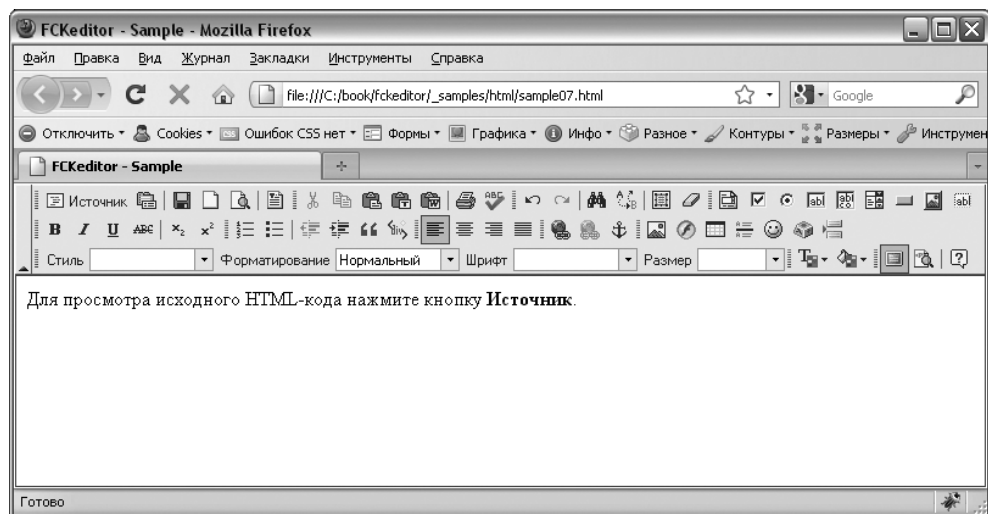


Рис. 1.1. Редактор FCKeditor, запущенный в Web-браузере Firefox

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

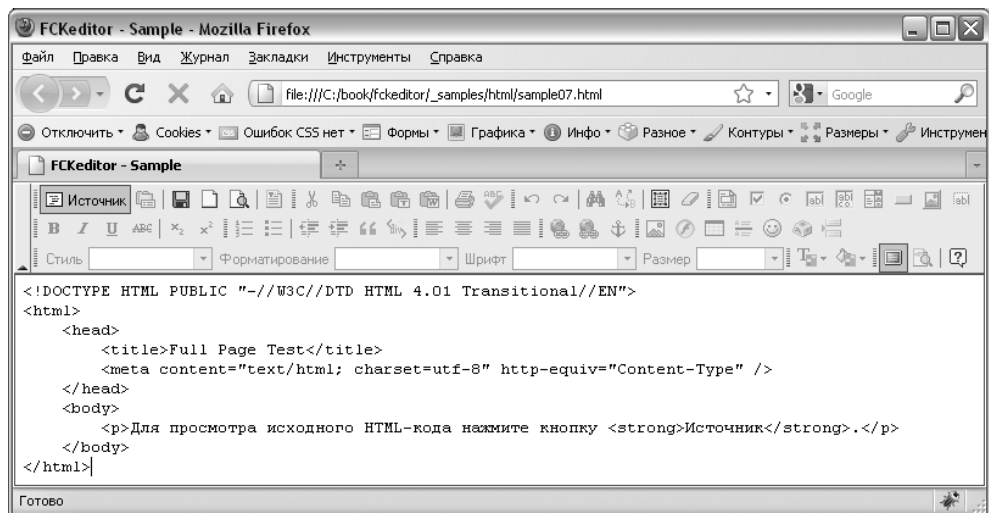


Рис. 1.2. Результат нажатия кнопки **Источник** в редакторе FCKeditor

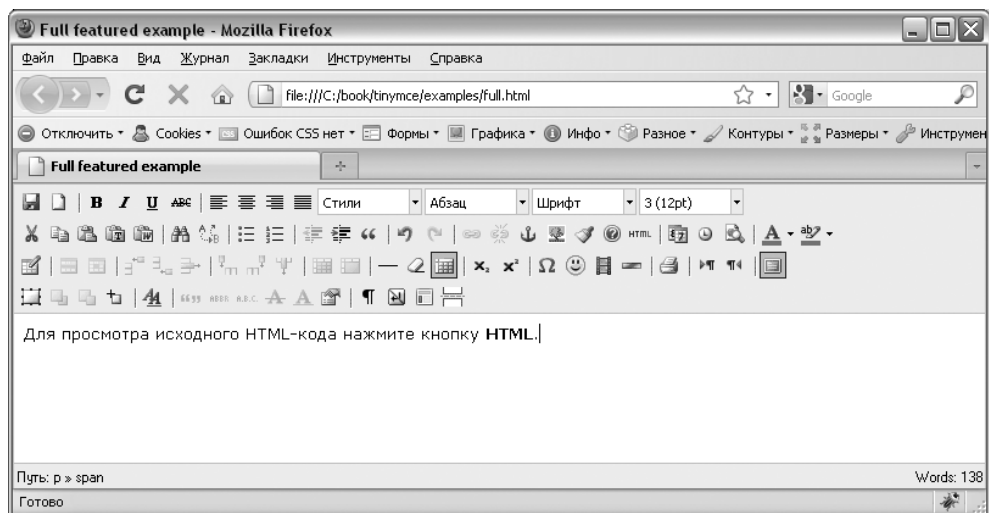


Рис. 1.3. Редактор tinYMCE, запущенный в Web-браузере Firefox



## 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 визуальнo оценить 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, так как 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 `<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/">
```