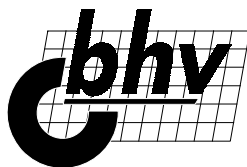


Александр Матросов
Александр Сергеев
Михаил Чаунин

HTML 4.0



Санкт-Петербург

Дюссельдорф ♦ Киев ♦ Москва ♦ Санкт-Петербург

УДК 681.3.06

Представлен весь спектр технологий создания Web-документов (начиная от простейших — статических — и до документов на основе динамического HTML), включая форматирование текста, создание списков, таблиц, форм, применение графики, каскадных таблиц стилей, встраивание различных объектов, использование средств интерактивного общения с пользователем, баз данных, мультимедиа-объектов и пр. Рассматриваются объектно-ориентированные технологии и программирование на языке Perl, а также создание CGI-программ и написание сценариев на языках JavaScript и VBScript.

Приводятся сведения о браузерах Netscape Communicator и Microsoft Internet Explorer и таблице HTML-тэгов.

Для Web-дизайнеров

Группа подготовки издания:

Главный редактор	<i>Екатерина Кондукова</i>
Зав. редакцией	<i>Наталья Таркова</i>
Ответственный редактор	<i>Борис Желваков</i>
Компьютерная верстка	<i>Ольга Сергиенко</i>
Корректурa	<i>Зинаида Дмитриева</i>
Дизайн обложки	<i>Наталья Смирнова</i>
Производство	<i>Николай Тверских</i>

Матросов А. В., Сергеев А. О., Чаунин М. П.

HTML 4.0. — СПб.: БХВ — Санкт-Петербург, 1999. — 672 с.: ил.

ISBN 5-8206-0072-X

© Матросов А. В., Сергеев А. О., Чаунин М. П., 1999

© Оформление, издательство "БХВ — Санкт-Петербург", 1999

Лицензия ЛР № 065953 от 15.06.98. Подписано в печать 12.11.99.

Формат 70×100^{1/16}. Печать офсетная. Усл. печ. л. 54,18.

Тираж 3000 экз. Заказ

"БХВ — Санкт-Петербург", 198005, Санкт-Петербург, Измайловский пр., 29.

Гигиеническое заключение на продукцию, товар, № 77.99.1.953.П.950.3.99 от 01.03.1999 г. выдано Департаментом ГСЭН Минздрава России.

Отпечатано с готовых диапозитивов
в Академической типографии "Наука" РАН.
199034, Санкт-Петербург, 9 линия, 12.

Содержание

ЧАСТЬ I. ОСНОВЫ HTML	11
Глава 1. Правила построения HTML-документов	13
Что такое HTML.....	13
Спецификации HTML.....	16
Структура документа	18
Раздел документа HEAD.....	19
Название документа.....	19
Связь с другими документами	20
Тэг <i><META></i>	22
Другие элементы заголовка.....	24
Раздел документа BODY.....	24
Форматирование текста	25
Тэги уровня блока и последовательные тэги	25
Логическое и физическое форматирование	26
Тэги логического форматирования текста	27
Тэги физического форматирования текста	31
Форматирование HTML-документа	39
Разделение на абзацы	39
Перевод строки.....	40
Тэги <i><NOBR></i> и <i><WBR></i>	41
Заголовки внутри HTML-документа.....	42
Горизонтальные линии.....	43
Использование предварительно отформатированного текста	44
Тэг <i><DIV></i>	45
Тэг <i><CENTER></i>	46
Включение комментариев в документ	46
Тэг <i><BLOCKQUOTE></i>	46
Тэг <i><ADDRESS></i>	48
Специальные символы	48
Ссылки на другие документы и файлы.....	49
Организация ссылок	49
Правила записи ссылок.....	50
Прочие тэги.....	54

Глава 2. Списки.....	56
Маркированный список	56
Тэги <code></code> и <code></code>	57
Графические маркеры списка	59
Нумерованный список.....	61
Тэги <code></code> и <code></code>	61
Список определений	65
Списки типа <code><DIR></code> и <code><MENU></code>	67
Вложенные списки.....	68
Глава 3. Графика.....	71
Общие соображения.....	72
Способы хранения изображений	73
Фоновые изображения	75
Встраивание изображений в HTML-документы	78
Выравнивание изображений.....	78
Задание размеров выводимого изображения	83
Отделение изображения от текста	84
Рамки вокруг изображений	85
Альтернативный текст	86
Использование изображения в качестве ссылки.....	87
Параметр <i>LOWSRC</i>	88
Использование миниатюрных версий изображений	88
Формат GIF.....	90
Формат JPG.....	96
Какой формат предпочесть — GIF или JPG	97
Некоторые проблемы использования цвета	99
Создание анимации на основе GIF-файлов.....	100
Программа GIF Construction Set.....	102
Блок <i>HEADER</i>	103
Блок <i>LOOP</i>	104
Блок <i>CONTROL</i>	104
Блок <i>IMAGE</i>	107
Использование мастера анимации.....	108
Дополнительные возможности пакета GIF Construction Set	110
Глава 4. Таблицы в HTML	114
Создание простейших HTML-таблиц	116
Представление таблиц на странице	118
Заголовок таблицы <code><CAPTION></code>	118
Параметры тэга <code><TABLE></code>	119
Параметр <i>BORDER</i>	120
Параметр <i>CELLSPACING</i>	121
Параметр <i>CELLPADDING</i>	122
Параметры <i>WIDTH</i> и <i>HEIGHT</i>	123
Параметр <i>ALIGN</i>	124
Форматирование данных внутри таблицы	128

Вложенные таблицы	132
Особенности построения таблиц	136
Отображение пустых ячеек в таблицах	136
Выравнивание данных в столбцах таблицы	137
Задание цвета рамок таблицы	139
Задание фонового рисунка для таблицы	140
Тэги структурирования таблицы <i><THEAD></i> , <i><TBODY></i> и <i><TFOOT></i>	140
Задание числа столбцов таблицы	143
Вертикальное выравнивание таблиц	143
Альтернатива табличному представлению	144
Подготовка таблиц.....	144
Глава 5. Фреймы	148
Сферы применения фреймов	148
Правила описания фреймов.....	155
Тэг <i><FRAMESET></i>	157
Тэг <i><FRAME></i>	159
Тэг <i><NOFRAMES></i>	162
Особенности описания фреймовых структур	162
Примеры фреймов.....	164
Особенности навигации при использовании фреймов	167
Взаимодействие между фреймами.....	168
Примеры более сложного взаимодействия между фреймами	174
Различие между фреймами и окнами браузера.....	179
Дополнительные возможности браузеров.....	181
Возможности браузера Netscape.....	181
Возможности браузера Microsoft Internet Explorer	183
Плавающие фреймы	185
Средства создания документов, содержащих фреймы	186
Редактор фреймов FrameGang.....	187
Редактор фреймов Frame-It	190
Информация об использовании фреймов на WWW.....	193
Глава 6. Карты-изображения	194
Основы использования карт-изображений.....	196
Терминология	197
Графическое представление карты-изображения.....	197
Описание конфигурации карты-изображения	197
Варианты реализации карт-изображений	198
Преимущества и недостатки карт-изображений	198
Серверный вариант реализации карт-изображений	200
Формат CERN.....	202
Формат NCSA	203
Клиентский вариант карты-изображения	204
Тэг <i><MAP></i>	205
Тэг <i><AREA></i>	205
Параметр <i>SHAPE</i>	205

Параметр <i>COORDS</i>	206
Параметры <i>HREF</i> и <i>NOHREF</i>	206
Параметр <i>TARGET</i>	207
Параметр <i>ALT</i>	207
Комбинация клиентского и серверного вариантов	208
Особенности использования карт-изображений.....	209
Альтернативные средства навигации.....	209
Средства создания карт-изображений.....	211
Программа MapEdit.....	212
Программа Map THIS!.....	217
Программа CrossEye.....	220
Заключительный пример.....	221
Глава 7. Звук.....	223
Средства воспроизведения звука	223
Как компьютер работает со звуком.....	224
Модуль LiveAudio	226
Управление модулем LiveAudio.....	226
Встраивание звуковых файлов в Web-страницу.....	227
Другие звуковые модули.....	232
Технология RealAudio	235
Программа-плеер RealPlayer Plus G2.....	238
Встраивание в страницу звуковых файлов формата RealAudio	239
Ресурсы RealAudio в Интернете.....	246
Звуковые файлы формата MP3.....	249
Воспроизведение файлов формата MP3.....	251
Установка программы Winamp.....	252
Управление программой Winamp.....	256
Подключаемые модули программы Winamp.....	257
Изменение внешнего вида программы Winamp.....	259
Скрытые возможности программы Winamp.....	261
Декодирование файлов формата MP3.....	262
Где найти файлы MP3?.....	263
Создание звуковых файлов MP3.....	265
Программы вычленения файлов с аудио CD.....	266
Программы кодирования.....	266
Выбор параметров кодирования.....	270
Переносные плееры звуковых файлов MP3.....	271
Потоковое воспроизведение.....	274
Звуковые файлы формата VQF.....	274
Звуковые файлы формата AAC.....	277
HomeboyAAC.....	278
AT&T a2bAAC.....	278
Liquifier Pro AAC.....	279
Astrid/Quartex AAC.....	279
Звуковые файлы формата PAC.....	280
Новый формат хранения звука MP4.....	281
Формат WMA.....	282

Глава 8. Разработка HTML-страниц при помощи текстового процессора Microsoft Word	283
Создание Web-страниц	285
Создание маркированных и нумерованных списков на Web-страницах	285
Вставка горизонтальной линии в Web-страницу	288
Выбор фона создаваемого документа.....	289
Изменение цвета и форматирование текста Web-страниц	291
Работа со стилями на Web-страницах.....	294
Непосредственное редактирование HTML-кода.....	295
Предварительный просмотр Web-страницы в процессе редактирования	296
Таблицы на Web-страницах.....	296
Работа с рисунками на Web-страницах.....	299
Создание ссылок в документе.....	301
Создание форм на Web-страницах	302
Сохранение существующего документа Word в формате HTML	303
Проблемы преобразования полей.....	308
Использование примечаний в документе.....	314
Версии и реализации Microsoft Word.....	315
ЧАСТЬ II. ИНТЕРАКТИВНЫЕ WEB-ДОКУМЕНТЫ.....	319
Глава 9. Выполняемые сценарии	321
Основы объектно-ориентированных технологий.....	321
Что такое программный объект	321
Событийные приложения.....	323
Объектные модели языков сценариев.....	324
Язык создания сценариев JavaScript.....	325
Общий обзор языка.....	326
Синтаксис языка	327
Размещение операторов языка на странице.....	327
Использование тэга <i><SCRIPT></i>	327
Задание файла с кодом JavaScript.....	328
Элементы JavaScript в параметрах тэгов HTML	329
Обработчики событий.....	329
Язык ядра JavaScript	333
Переменные и литералы.....	333
Выражения и операторы.....	335
Стандартные объекты и функции	339
Операторы управления	345
Объекты клиента и обработка событий.....	350
Иерархия объектов.....	351
Свойства и методы ключевых объектов.....	352
Обработчики событий.....	359
Практические примеры.....	364
Часы JavaScript	364
Простое меню.....	365
Документ с фреймами	368

Язык VBScript.....	369
Основные понятия.....	370
Типы данных	370
Переменные, массивы и константы.....	373
Операторы.....	375
Операторы условия и цикла.....	377
Процедуры	383
Объектная модель и взаимодействие с элементами документа.....	385
Функции и объекты ядра VBScript.....	385
Объекты MS Internet Explorer	391
Процедуры обработки событий.....	393
Практические примеры.....	396
Плавающий фрейм.....	396
Баннер	398
CGI-сценарии и язык Perl.....	401
Основные понятия.....	401
HTML-формы	402
Тэг <i><FORM></i>	403
Тэг <i><INPUT></i>	404
Тэг <i><SELECT></i>	407
Тэг <i><TEXTAREA></i>	408
Пример формы	409
Передача информации CGI-программе	411
Кодирование и пересылка данных формы.....	411
CGI-сценарии	414
Общие сведения	414
Переменные в языке Perl.....	418
Переменные среды CGI.....	426
Поиск и замена текста. Регулярные выражения.....	429
Обработка данных формы	432
Подпрограммы, библиотеки, модули.....	438
Глава 10. Динамический HTML.....	449
Каскадные таблицы стилей	450
Общие положения	451
Встраивание таблиц стилей в документ	452
Группирование и наследование	454
Селекторы.....	455
Селектор <i>CLASS</i>	456
Селектор <i>ID</i>	457
Контекстные селекторы.....	457
Псевдоклассы	458
Псевдоклассы связей	459
Применение таблиц стилей.....	459
Модель форматирования	462
Блочные элементы	463
Встроенные элементы	465

Свойства форматирования элементов	466
Шрифты	468
Свойство <i>font-family</i>	468
Свойство <i>font-style</i>	469
Свойство <i>font-variant</i>	469
Свойство <i>font-weight</i>	470
Свойство <i>font-size</i>	470
Свойство <i>font</i>	471
Свойство <i>@font-face</i>	471
Цвет и фон	471
Форматирование текста	474
Блоки	476
Визуальное форматирование	478
Абсолютное позиционирование	479
Относительное позиционирование	481
Статическое позиционирование	482
Визуальные эффекты	482
Отображение списков	486
Объектная модель документа	488
Структура документа	489
Объектная модель DHTML в MS Internet Explorer 4.0	490
Иерархия объектов	491
Свойства и методы объектов	494
Событийная модель	497
Цикл жизни события	497
Объект <i>event</i>	500
Объектная модель документа в MS Internet Explorer 5.0	503
Динамический HTML в Internet Explorer	509
Динамическое изменение документа	510
Раскрывающийся список	510
Движущийся элемент	514
Поиск в документе	516
Фильтры и переходы	519
Общие свойства некоторых фильтров	523
Описание фильтров	525
Описание переходов	530
Связывание данных с документом	533
Архитектура привязки данных	533
Объекты-источники данных	536
Динамический HTML в Netscape Navigator	543
Применение каскадных таблиц стилей	543
Позиционирование и объектная модель сценария	547
Динамическое позиционирование	551
Загружаемые шрифты	553
Глава 11. Встраиваемые компоненты	555
Элементы управления ActiveX	555
Встраивание в HTML-страницу	557

Элементы управления ActiveX и сценарии.....	561
Редактор FrontPage 98.....	565
Безопасность и элементы управления ActiveX.....	573
Цифровая подпись.....	574
Безопасное использование элементов управления ActiveX.....	576
Лицензирование элементов управления ActiveX.....	579
Элементы управления на HTML-страницах.....	581
Элемент управления <i>TabStrip</i>	582
Элемент управления <i>TreeView</i>	588
Java-апплеты.....	597
Приложения и апплеты.....	599
Структура приложения.....	600
Интерфейсы.....	602
Апплеты.....	603
Встраивание апплета в HTML-документ.....	607
Жизненный цикл апплета.....	608
Создание графического интерфейса пользователя.....	612
Обработка событий.....	615
Рисунки в апплетах.....	620
Архивы.....	621
Java и JavaScript.....	622
Заключение.....	630
Приложение 1. Поддержка тэгов и параметров HTML-браузерами.....	633
Приложение 2. Названия и коды цветов для HTML.....	650
Приложение 3. Совместимость приводов CD-ROM и программ для вычленения звуковых файлов.....	656
Предметный указатель.....	664

ЧАСТЬ I

1. Правила построения HTML-документов
2. Списки
3. Графика
4. Таблицы в HTML
5. Фреймы
6. Карты-изображения
7. Звук
8. Разработка HTML-страниц при помощи текстового процессора Microsoft Word

ОСНОВЫ HTML

Данная книга посвящена вопросам создания документов, предназначенных для использования на Web-страницах. В ней представлен весь спектр технологий создания различных Web-документов, начиная от простейших статических документов, использующих "чистый" HTML-код, до сложных документов, использующих динамическую генерацию содержимого, средства интерактивного общения с пользователем, базы данных, мультимедиа-объекты и др.

Книга состоит из двух частей. Первая часть посвящена типовым вопросам создания документов: форматирование текста, применение графики, создание списков, таблиц и др. В этой части приводятся основные тэги языка HTML и способы их применения. Для чтения первой части книги не потребуются знания из области программирования, достаточно лишь владеть основами работы с компьютером.

Вторая часть книги, хотя и не требует изначальных знаний в области программирования, однако предназначена читателям, обладающим некоторым "программистским" мышлением.

Книга построена таким образом, чтобы первая часть была доступна практически любому читателю. Поэтому в ней рассматриваются вопросы непосредственного форматирования документов, оставляя за рамками методы современного построения документов, основанных на стилевом оформлении. Опыт обучения слушателей методам составления разнообразных документов показывает, что на начальном этапе не удастся сразу пояснить необходимость применения современных методов. Действительно, человеку, который составляет свой первый документ в текстовом редакторе, невозможно объяснить, зачем, например, нужно применять стили или шаблоны вместо конкретного указания размеров шрифта и его названия.

Аналогично учащимся, постигающим азы программирования и написавшим свою первую программу, невозможно объяснить преимущества объектно-ориентированного подхода. Маленькая программа, написанная традиционным способом, будучи переписанная с использованием объектов, их методов и свойств, покажется начинающему программисту громоздкой и непонятной.

Понимание необходимости использования современных методов разработки документов приходит со временем. Поэтому в первой части книги используются только традиционные формы разработки документов, на которых можно научиться основам HTML-разметки документов. Приобретая необходимый опыт, можно приступить к изучению второй части.

В данной книге главы 1—8 и приложения написаны Сергеевым А. О., а главы 9—11 — Матросовым А. В. и Чауниным М. П.

Замечание и пожелания по книге просим направлять авторам по электронной почте **sergeev@mail.ifmo.ru**.

Правила построения HTML-документов

Что такое HTML

Всемирная паутина World Wide Web (WWW) соткана из Web-страниц, которые создаются с помощью так называемого языка разметки гипертекста HTML (HyperText Markup Language). Хотя многие говорят о программировании на этом языке, HTML вовсе не является языком программирования в традиционном понимании. HTML — *язык разметки* документа. При разработке HTML-документа выполняется разметка текстового документа точно так же, как это делает редактор при помощи красного карандаша. Эти пометки служат для указания формы представления информации, содержащейся в документе.

Специальные программы просмотра HTML-документов, которые часто называют *браузерами*, служат для интерпретации файлов, размеченных по правилам языка HTML, форматирования их в виде Web-страниц и отображении их содержимого на экране компьютера пользователя. Существует большое количество программ-браузеров, разработанных различными компаниями, однако, на сегодняшний день из всего разнообразия программ явно выделяются две программы-лидера — Netscape Communicator и Microsoft Internet Explorer.

Программа Netscape Navigator разработана компанией Netscape Communications Corporation. Как и у многих программных продуктов, существует ряд версий этой программы. Последней версией программы Netscape Communicator на момент написания книги являлась версия 4.7. Программа Internet Explorer разработана компанией Microsoft. Последняя версия этой программы — 5.0.

Другие браузеры значительно отстают по популярности. Несколько лет назад браузер компании Netscape занимал ведущее место среди браузеров, более двух третей пользователей применяли именно эту программу просмотра. Выпустив свой браузер, компания Microsoft приложила огромные усилия для завоевания этой части рынка. В средствах массовой информации часто

встречались сообщения о войне между браузерами за пользователей. Сейчас эти два браузера сравнимы по популярности. Росту популярности браузера Microsoft способствует включение браузера в состав операционной системы Windows 98, однако, в конечном счете, выбор браузера остается за пользователем.

Современные браузеры обладают широкими возможностями, но основным для них является интерпретация документов, размеченных по правилам HTML. Описанию этих правил, в основном, и посвящена данная книга. В первой части мы рассмотрим лишь основополагающие принципы построения HTML-документов.

Чтобы понять, что собой представляет язык разметки, вспомним старые добрые времена, когда многие работали с текстовыми редакторами типа WordStar. В них для выделения какой-либо фразы, например, полужирным шрифтом, в ее начале и в конце ставились специальные отметки (/в и /b):

/ВЭтот текст будет выведен полужирным шрифтом/б

При выводе такого текста на печатающее устройство (о дисплеях еще речь не идет, в те далекие времена их еще или не было вообще или существовали алфавитно-цифровые дисплеи, не позволяющие изменять шрифты) символы /в заставляли использовать полужирный шрифт до тех пор, пока не встретятся символы /b.

HTML работает точно так же. Если есть необходимость выделить текст на экране полужирным шрифтом, то это можно сделать аналогично:

<В>Этот текст будет выведен полужирным шрифтом</В>

Символы <в> включают полужирное начертание, а символы </в> выключают его. Такие символы, которые управляют отображением текста и при этом сами не отображаются на экране, в языке HTML принято называть *тэгами* (от английского слова tag — ярлык, признак).

Все тэги языка HTML выделяются символами-ограничителями (< и >), между которыми записывается идентификатор (имя) тэга (в нашем примере это в), и, возможно, его *параметры*. Единственным исключением из этого правила являются тэги комментария с более сложными ограничителями (<!-- и -->). Названия тэгов, а также их параметров можно записывать на любом регистре. Для единообразия в данной книге большинство тэгов записывается прописными буквами.

Большинство тэгов HTML используется попарно, т. е. для определенного тэга, назовем его *открывающим*, в документе имеется соответствующий *закрывающий* тэг. По правилам HTML закрывающий тэг записывается так же, как и открывающий, но с символом / (прямой слэш) перед именем тэга. Единственным принципиальным различием парных тэгов является то, что закрывающие тэги не используют параметры.

Тэги, которые нуждаются в соответствующих завершающих тэгах, будем называть *тэгами-контейнерами*. Все, что записано между соответствующим открывающим и закрывающим тэгом, будем называть *содержимым* тэга-контейнера. Иногда завершающий тэг можно опускать. Например, для тэга, описывающего данные для ячейки таблицы <TD>, соответствующий закрывающий тэг </TD> можно всегда опускать. Окончание данных для ячейки таблицы будет распознано по появлению очередного тэга <TD> или тэга окончания строки таблицы </TR>.

Есть ряд тэгов, для которых завершающие тэги опускаются большинством авторов документов. Примером может служить тэг элемента списка или тэг абзаца <P>. Современные браузеры во многих случаях правильно формируют документы, если опущены некоторые завершающие тэги, однако такая практика не может быть рекомендована.

Ряд тэгов в принципе не нуждается в завершающих тэгах. Примерами могут служить тэг вставки изображений , принудительного перевода строки
, указания базового шрифта <BASEFONT> и др. Часто из самого предназначения тэга можно догадаться, нуждается ли он в завершающем.

Существуют общие правила интерпретации тэгов браузерами. В отличие от языков программирования, в которых ошибочные операторы приводят к выдаче соответствующих сообщений на этапе компиляции программы и требуют правки, в HTML не принято реагировать на неверную запись тэгов. Неверно записанный тэг или его параметр должен просто игнорироваться браузером. Это общее правило для всех браузеров, под действие которого подпадают не только ошибочно записанные тэги, но и тэги, не распознаваемые данной версией браузера. Примером могут служить тэги, предложенные и реализованные для отдельного браузера и неизвестные для другого. Например, тэг-контейнер <NOFRAMES>, который служит для предоставления альтернативной информации браузерам, не обеспечивающим поддержку фреймовых структур, такими браузерами не будет распознан. Браузер же, поддерживающий фреймы, встретив тэг <NOFRAMES>, пропустит всю заключенную в нем информацию. А браузер, не знакомый с фреймами, естественно, не поймет и тэг <NOFRAMES>. Однако, согласно приведенному правилу, этот тэг будет просто пропущен, зато вся последующая информация будет отображена.

Тэги могут записываться с *параметрами* или *атрибутами* (от англ. attribute). В этой книге будем чаще всего использовать термин *параметр*. Наборы допустимых параметров индивидуальны для каждого тэга. Общие правила записи параметров заключаются в следующем. После имени тэга могут следовать параметры, которые отделяются друг от друга пробелами. Порядок следования параметров тэга произволен. Многие параметры требуют указания их *значений*, однако некоторые параметры не имеют значений или могут записываться без них, принимая *значения по умолчанию*. Если параметр требует значения, то оно указывается после названия параметра через знак

равенства. Значение параметра может записываться в кавычках, так и без них. Единственным случаем, в котором без кавычек не обойтись, является случай, когда в значении параметра имеются пробелы. В значениях параметров (в отличие от названий тэгов и самих параметров) иногда важен регистр записи. Приведем пример записи тэга с параметрами:

```
<TABLE BORDER ALIGN="left">
```

Здесь для тэга `<TABLE>` задано два параметра. Первый параметр `BORDER` указан без значения. Второй параметр `ALIGN` имеет значение `left`.

В последующих главах первой части книги будет описано назначение тэгов языка HTML и их параметров. В общем, тэги могут иметь различные параметры, однако существует ряд параметров, единых практически для всех тэгов. Упомянем здесь общие параметры тэгов, чтобы более не говорить о них при описании каждого тэга.

Все тэги, которые допустимо использовать в разделе `<BODY>` документа HTML, могут иметь параметры `CLASS`, `ID`, `LANG`, `LANGUAGE`, `STYLE` и `TITLE`. Использование этих параметров полезно, прежде всего, при стилевом оформлении документов, речь о котором пойдет во второй части книги.

Параметры `CLASS`, `ID`, `STYLE` поддерживаются Internet Explorer, начиная с версии 3.0, и Netscape, начиная с версии 4.0. Эти параметры нужны при использовании стилей.

Параметры `LANG`, `LANGUAGE`, `TITLE` — поддерживаются только Internet Explorer, начиная с версии 4.0. Эти параметры указывают, соответственно, используемый язык (например, для России: `LANG=ru`), язык записи скриптов (например, `LANGUAGE=JavaScript`), а также текст подсказки, выдаваемой при наведении указателя мыши на данный элемент (`TITLE`).

В современном HTML, помимо тэгов языка и их содержимого, в исходном HTML-коде также записываются *коды сценариев* (JavaScript или VBScript). В первой части книги об этом практически нигде не упоминается, зато часть вторая целиком посвящена вопросам использования сценариев.

Завершая общий обзор HTML, отметим, что простейшие HTML-документы представляют собой обычные текстовые файлы, для просмотра и редактирования которых можно воспользоваться любым текстовым редактором. Эти файлы обычно имеют расширение `HTM` или `HTML`.

Спецификации HTML

Язык HTML приобрел популярность в середине 90-х годов, благодаря экспоненциальному росту сети Интернет. К этому времени назрела необходимость стандартизации языка, поскольку различные компании, разрабатывавшие программное обеспечение для доступа в Интернет, предлагали свои

варианты инструкций HTML, число которых все возрастало и возрастало. Настала пора прийти к какому-то единому соглашению в части применения тэгов языка HTML.

Работу по созданию спецификации HTML взяла на себя организация, называемая World Wide Web Consortium (сокращенно — W3C). В ее задачу входило составление спецификации, отражающей современный уровень развития возможностей языка с учетом разнообразных предложений компаний-разработчиков браузеров. Так, в ноябре 1995 г. появилась спецификация HTML 2.0, призванная формализовать сложившуюся к концу 1994 г. практику использования HTML.

Схема утверждения спецификаций состоит в следующем. Консорциум W3C выпускает проект спецификации, после обсуждения которого выпускается так называемый черновой, рабочий (draft) вариант спецификации и предлагает его к обсуждению на определенный период. После периода обсуждения рабочий вариант спецификации может стать рекомендацией, т. е. официально признанным вариантом спецификации HTML.

Вскоре после спецификации 2.0 была выпущена рабочая версия спецификации 3.0, срок окончания периода обсуждения которой истек в сентябре 1995 г. Эта спецификация так и не была принята в качестве официальной рекомендации. В нее планировалось включить большое разнообразие тэгов и возможностей, специфичных для отдельных браузеров, однако Консорциум W3C не нашел возможности разработать хорошую спецификацию для такого большого числа инструкций.

После долгих размышлений в мае 1996 г. был выпущен проект HTML 3.2. Проект основывался на части тэгов, имеющихся в версии 3.0, которые показывали стабильность в работе. В сентябре 1996 г. после нескольких месяцев обсуждения версия 3.2 стала предлагаемой спецификацией, а в январе 1997 г. — официальной рекомендацией.

Июль 1997 года ознаменовался выходом предлагаемой спецификации HTML 4.0, которая в декабре 1997 г. стала официальной рекомендацией. На сегодняшний день это последняя из принятых спецификаций.

В приводимом здесь кратком обзоре истории развития языка HTML вряд ли стоит детально описывать особенности различных спецификаций, тем более, что в реальной жизни разработчики далеко не всегда следуют рекомендациям Консорциума. Отметим лишь некоторые идеи, заложенные в основу последней спецификации.

В спецификации HTML 4.0 ключевой идеей стало отделение описания *структуры* документа от описания его *представления* на экране монитора. Опыт показывает, что разделение структуры и представления документа уменьшает затраты на поддержку широкого спектра платформ, сред и т. п., а также облегчает внесение исправлений в документы. В соответствии с этой идеей следует шире пользоваться методами описания представления

документа с помощью *таблиц стилей*, вместо того, чтобы задавать конкретные данные о форме представления вперемешку с содержанием документа. Для реализации этой идеи в спецификации HTML 4.0 ряд тэгов, используемых для непосредственного задания формы представления HTML-элементов, отменены. К отмененным по этой причине тэгам относятся `<CENTER>`, ``, `<BASEFONT>`, `<S>`, `<STRIKE>`, `<U>`. Среди других отмененных тэгов отметим `<ISINDEX>`, `<APPLET>`, `<DIR>`, `<MENU>`. Вместо отмененных тэгов предлагаются альтернативные варианты реализации соответствующих возможностей, на что мы обращаем особое внимание в этой книге.

Понятие *отмененного* (deprecate) тэга состоит в следующем. Если в данной спецификации языка тэг назван отмененным, то это означает, что браузеры должны пока продолжать поддержку таких тэгов, но их использование не рекомендуется. В следующих спецификациях эти тэги, возможно, будут переведены в разряд устаревших (obsolete). Устаревшие тэги могут более не поддерживаться браузерами. В спецификации HTML 4.0 устаревшими названы всего три тэга: `<XMP>`, `<PLAINTEXT>` и `<LISTING>`. Информацию о том, какие из тэгов включены в спецификацию, можно получить из таблицы, приводимой в приложении П1.

Официальные сведения о спецификации HTML всегда можно получить с Web-сайта Консорциума W3C по адресу <http://www.w3.org/TR/>. Спецификация 4.0 находится по адресу <http://www.w3.org/TR/REC-html40-971218>.

Заметим, что по логике вещей официальная спецификация должна играть роль руководящей и направляющей силы, обеспечивая одинаковую форму представления информации различными браузерами. Это идеальный вариант, к которому следует стремиться. На деле все обстоит не так хорошо. Постоянно появляются новые идеи, реализуемые компаниями-разработчиками в своих браузерах и пропагандируемые ими. Удачные идеи приживаются, а затем подхватываются другими разработчиками. Часть возможностей так и остается специфическими особенностями отдельного браузера. Удачные разработки в итоге попадают в спецификацию и становятся общепринятыми. Таким образом, процесс усовершенствования возможностей браузеров и уточнения спецификации идет непрерывно, оказывая взаимное влияние друг на друга.

Структура документа

Первым тэгом, с которого следует начинать описание документов HTML, является тэг `<HTML>`. Он должен всегда начинать описание документа, а завершать описание документа должен тэг `</HTML>`. Эти тэги обозначают, что находящиеся между ними строки представляют единый HTML-документ. Сам по себе документ является обыкновенным текстовым ASCII-файлом. Без этих тэгов браузер или другая программа просмотра, возможно, будет не

в состоянии идентифицировать формат документа и правильно его интерпретировать.

Чаще всего тэг `<HTML>` используется без параметров. В предыдущих версиях использовался параметр `VERSION`, отмененный спецификацией HTML 4.0. На смену этому параметру пришел тэг `<!DOCTYPE>`.

Большинство современных браузеров могут опознать документ и не содержащий тэгов `<HTML>` и `</HTML>`, все же их употребление крайне желательно.

Между парой тэгов `<HTML>` и `</HTML>` располагается сам документ. Документ может состоять из двух разделов — раздела заголовка (начинающийся тэгом `<HEAD>`) и раздела содержательной части документа (начинающийся тэгом `<BODY>`). Для документов, описывающих фреймворки, вместо раздела `BODY` используется раздел `FRAMESET` (с тэгом `<FRAMESET>`). Далее будут рассмотрены правила составления разделов документа `HEAD` и `BODY`. Построение документов, содержащих фреймы, рассматривается в главе 5.

Раздел документа HEAD

Раздел документа `HEAD` определяет его заголовок и не является обязательным тэгом, однако хорошо составленный заголовок может быть весьма полезен. Задачей заголовка является представление необходимой информации для программы, интерпретирующей документ. Тэги, находящиеся внутри раздела `HEAD` (кроме названия документа, описываемого с помощью тэга `<TITLE>`), не отображаются на экране.

Раздел заголовка открывается тэгом `<HEAD>`. Обычно этот тэг следует сразу же за тэгом `<HTML>`. Закрывающий тэг `</HEAD>` показывает конец этого раздела. Между упомянутыми тэгами располагаются остальные тэги раздела заголовка.

Название документа

Тэг-контейнер `<TITLE>` является единственным обязательным тэгом заголовка и служит для того, чтобы дать документу название. Оно обычно показывается в заголовке окна браузера. Тэг `<TITLE>` нельзя путать с названием файла документа; напротив, он представляет собой текстовую строку, совершенно независимую от имени и местоположения файла, что делает его весьма полезным. Имя же файла жестко определяется операционной системой компьютера, на котором он хранится. Также следует отличать название документа (с тэгом `<TITLE>`) от заголовков внутри документа, обычно размечаемых тэгами `<Hx>`.

Примечание

Обязательность названия документа, вообще говоря, носит характер настоятельной рекомендации. Документ без тэга `<TITLE>` также будет отображаться

браузерами. При этом различные браузеры в качестве заголовка окна будут выдавать различную информацию. Так ранние версии браузера Netscape выдавали строчку "No title". Другие браузеры либо не показывают ничего, либо отображают адрес загруженного файла, повторяя информацию панели **Location** браузера.

Название документа записывается между тэгами `<TITLE>` и `</TITLE>` и представляет собой строку текста. В принципе, название может иметь неограниченную длину и содержать любые символы, кроме некоторых зарезервированных. На практике следует ограничиться одной строкой, имея в виду, что название появляется в заголовке окна браузера. Также следует помнить о том, что останется от названия документа при минимизации окна браузера. Можно рекомендовать ограничивать длину названия документа 60 символами. Увидеть, как отображается название в окне браузера, можно на любом рисунке в данной книге, где приводится пример отображения какого-либо документа.

По умолчанию текст, содержащийся в названии документа, используется при создании закладки (bookmark) для документа. Поэтому, для большей информативности, избегайте безликих названий (Home Page, Index и т. д.). Подобные слова, используемые в качестве названия закладки, обычно совершенно бесполезны. Название документа должно кратко характеризовать его содержание. Заметим, что при отображении на экране документов с фреймовой структурой, когда в каждый из фреймов загружается отдельный документ, имеющий свое название, на экране будет видно только название главного документа. Тем не менее, задавать название отдельных документов, предназначенных для загрузки во фреймы, также настоятельно рекомендуется. Более подробно этот вопрос рассматривается в главе 5.

Важность названия документа определяет следующий факт. Поскольку тэг `<TITLE>` располагается практически в самом начале HTML-файла, то после начала загрузки документа первым делом отображается именно оно. Далее выполняется загрузка основного содержания документа, при этом браузер начинает форматирование документа в окне. Этот процесс, вообще говоря, в зависимости от содержания и структуры документа, а также скорости соединения, может затянуться. В течение достаточно продолжительного времени пользователь будет созерцать пустой экран, единственной информативной строчкой которого будет являться название документа. Весьма часто (при обрыве соединения или, если пользователь не желает дожидаться окончания загрузки документа) вся информация о документе на этом и заканчивается.

Связь с другими документами

Часто HTML-документы связаны между собой, то есть имеют ссылки друг на друга. Ссылки могут быть как абсолютные, так и относительные. И те и другие имеют недостатки. Абсолютные ссылки могут быть слишком гро-

моздкими и переставать работать, если перемещен младший по иерархии документ. Относительные ссылки легче вводить и обновлять, но и эта связь обрывается, если перемещен старший по иерархии документ. Оба вида связей могут нарушиться при переносе документа с одного компьютера на другой.

Часто случается, что пользователь загрузил на свою машину большой документ и отключился от сети для его подробного изучения. Все ссылки в локальной копии документа перестанут работать. Для их "реанимации" придется вновь обратиться к оригиналу документа, находящемуся на удаленном компьютере.

К счастью, разработчики HTML предусмотрели эту проблему и добавили два тэга, `<BASE>` и `<LINK>`, которые включаются в заголовок для того, чтобы связь между документами не нарушалась.

Тэг `<BASE>`

Тэг `<BASE>` служит для указания полного базового URL-адреса документа. С его помощью относительная ссылка продолжает работать, если документ переносится в другой каталог или даже на другой компьютер. Тэг `<BASE>` работает аналогично команде `path` MS-DOS, что позволяет программе просмотра определить ссылку на искомый документ, даже если она находится в старшем по иерархии документе, расположенном на другом компьютере.

Тэг `<BASE>` имеет один обязательный параметр `href`, после которого указывается полный URL-адрес документа. Ниже показан пример использования тэга `<BASE>`.

```
<HTML>
<HEAD>
<TITLE>Указание базового адреса</TITLE>
<BASE href="//www.my_host.ru/~sergeev">
</HEAD>
<BODY>
<IMG SRC=/gifs/news.gif ALT="News">
</BODY>
</HTML>
```

Тэг `<BASE>` указывает браузеру, где искать файл. В случае, если пользователь работает с локальной копией файла и его машина не отключена от сети, изображение пиктограммы **News** будет найдено и показано в окне браузера.

Тэг `<LINK>`

Даже если тэг `<BASE>` позволяет найти файл, остается открытым вопрос о взаимоотношениях документов. Важность этих отношений возрастает пропорционально росту сложности ваших документов. Для того чтобы поддерживать логическую связь между ними, в HTML введен тэг `<LINK>`.

Тэг `<LINK>` указывает на связь документа, содержащего данный тэг и другого документа или объекта. Он состоит из URL-адреса и параметров, конкретизирующих отношения документов. Заголовок документа может содержать любое количество тэгов `<LINK>`. Табл. 1.1 описывает параметры тэга `<LINK>` и их функции.

Таблица 1.1. Параметры тэга `<LINK>`

Параметр	Назначение
href	Указывает на URL-адрес другого документа
rel	Определяет отношение между текущим и другим документом
rev	Определяет отношение между другим документом и текущим (отношение, обратное REL)
type	Указывает тип и параметры присоединенной таблицы стилей

Приведем примеры тэга `<LINK>` с параметрами:

```
<LINK REL="contents" HREF = "../toc.html">
<LINK HREF="mailto:sergeev@mail.ifmo.ru" REV="made">
```

Первая строка указывает на связь с файлом оглавления документа (toc.html — table of contents) с прямым отношением `contents`. Вторая строка описывает связь с URL-адресом автора документа (с обратным отношением `made`).

Между документами может существовать множество различных отношений. Примеры других значений параметра `REL`: `bookmark`, `copyright`, `glossary`, `help`, `home`, `index`, `toc`, `next`, `previous`. Параметр `REV` может также принимать значения: `author`, `editor`, `publisher`, `owner`.

Тэг `<META>`

Разработка новых спецификаций языка разметки гипертекста занимает немалый срок, и за это время компании, производящие браузеры, успевают выпустить несколько версий своих продуктов. Поэтому в раздел заголовка может быть добавлен еще один тэг `<META>`, позволяющий авторам документа определять информацию, не имеющую отношения к HTML.

Эта информация используется браузером для действий, которые не предусмотрены текущей спецификацией HTML. Тэг `<META>` не потребуется вам для создания первых HTML-документов, но он вам наверняка понадобится, когда ваши страницы станут более сложными.

Пример:

```
<META HTTP-EQUIV=refresh" CONTENT="60" RL="www.my_host.ru/homepage.html">
```

Браузеры Netscape Navigator и Internet Explorer поймут эту запись как инструкцию ожидать 60 секунд, а затем загрузить новый документ. Такая инструкция часто используется при изменении местоположения документов. Небольшой документ с приведенной строкой может быть оставлен на старом месторасположении документа для автоматической ссылки на его новое месторасположение.

Следующая строка:

```
<META HTTP-EQUIV=refresh" CONTENT="60">
```

инструктирует браузер перезагружать страницу каждые 60 секунд. Это может быть полезно, если данные на странице часто обновляются, например, в случае отслеживания котировок акций.

Стало весьма популярным применение элемента <META> для решения некоторых типичных задач. В качестве примера можно привести указание ключевых слов, используемых поисковыми системами. Этот способ позволяет включать в индекс документа дополнительные слова, которые могут явно не входить в его содержание. Для этого в тэге <META> в качестве значения параметра NAME указывается имя некоторого свойства. А при помощи параметра CONTENT указывается значение данного свойства, например:

```
<META NAME="author" CONTENT="Александр Сергеев">
```

Спецификация HTML не определяет каких-либо конкретных имен свойств, записываемых в тэге <META>. Однако есть несколько часто применяемых свойств, например, description, keywords, author, robots и др.:

```
<META NAME="description" CONTENT="Описание возможностей языка HTML 4.0">
<META NAME="keywords" CONTENT="тэг, гипертекст, HTML, браузер">
```

Приведенные тэги <META> вполне могли бы быть указаны, например, для электронного варианта данной книги.

Тэг <META> может иметь параметры, указанные в табл. 1.2.

Таблица 1.2. Параметры тэга <META>

Параметр	Назначение
HTTP-EQUIV	Определяет свойство для тэга
NAME	Обеспечивает дополнительное описание тэга. Если этот параметр опущен, он считается эквивалентным параметру HTTP-EQUIV
URL	Определяет адрес документа для свойства
CONTENT	Определяет возвращаемое значение для свойства

Еще одно важное предназначение тэга `<META>` — это указание кодировки текста. Так, для текста на русском языке в кодировке Windows нужно записать следующую строчку:

```
<META HTTP-EQUIV="Content-Type" CONTENT="text/html; charset=Windows-1251">
```

Другие элементы заголовка

В разделе заголовка документа могут присутствовать еще два тэга — `<STYLE>` и `<SCRIPT>`. Их назначение связано с использованием таблиц стилей в документе и записью скриптов. Эти вопросы подробно рассматриваются во второй части книги.

Раздел документа BODY

В этом разделе документа располагается его содержательная часть. Большинство тэгов, рассматриваемых далее в этой главе и последующих, должно располагаться в данном разделе документа. Здесь мы рассмотрим лишь некоторые общие вопросы.

Раздел документа `BODY` должен начинаться тэгом `<BODY>` и завершаться тэгом `</BODY>`, между которыми располагается все содержимое данного раздела. Строго говоря, наличие этих тэгов не является обязательным, поскольку браузеры могут определить начало содержательной части документа по контексту. Однако их употребление рекомендуется.

Тэг `<BODY>` имеет ряд параметров, ни один из которых не является обязательным. Перечень параметров приведен в табл. 1.3.

Таблица 1.3. Перечень параметров тэга `<BODY>`

Параметр	Назначение
<code>ALINK</code>	Определяет цвет активной ссылки
<code>BACKGROUND</code>	Указывает на URL-адрес изображения, которое используется в качестве фонового
<code>BOTTOMMARGIN</code>	Устанавливает границу нижнего поля документа в пикселах
<code>BGCOLOR</code>	Определяет цвет фона документа
<code>BGPROPERTIES</code>	Если установлено значение <code>FIXED</code> , фоновое изображение не прокручивается
<code>LEFTMARGIN</code>	Устанавливает границу левого поля документа в пикселах
<code>LINK</code>	Определяет цвет еще не просмотренной ссылки
<code>RIGHTMARGIN</code>	Устанавливает границу правого поля документа в пикселах

Таблица 1.3 (окончание)

Параметр	Назначение
SCROLL	Устанавливает наличие или отсутствие полос прокрутки окна браузера
TEXT	Определяет цвет текста
TOPMARGIN	Устанавливает границу верхнего поля документа в пикселах
VLINK	Определяет цвет уже просмотренной ссылки

Использование параметров `BACKGROUND` и `BGCOLOR`, определяющих фон документа, подробно рассмотрено в главе 3.

Параметр `BGPROPERTIES`, принимающий единственное значение `FIXED`, поддерживается только браузером Microsoft Internet Explorer.

Параметры `BOTTOMMARGIN`, `LEFTMARGIN`, `RIGHTMARGIN` и `TOPMARGIN`, задающие расстояния в пикселах между краями текста и соответствующими краями окна, а также параметр `SCROLL`, распознаются только браузером Microsoft Internet Explorer, начиная с версии 4.0.

Параметры `ALINK`, `LINK`, `TEXT` и `VLINK` определяют цвета ссылок и текста документа.

В языке HTML цвета определяются цифрами в шестнадцатеричном коде. Цветовая система базируется на трех основных цветах — красном, зеленом и синем — и обозначается RGB. Для каждого цвета задается шестнадцатеричное значение в пределах от 00 до FF, что соответствует диапазону 0–255 в десятичном исчислении. Затем эти значения объединяются в одно число, перед которым ставится символ #. Например, число #800080 обозначает фиолетовый цвет. Чтобы не запоминать совокупности цифр, вместо них можно пользоваться названиями цветов, которые приводятся в приложении П2. Ранние версии браузеров распознавали только названия 16 стандартных цветов, отмеченных в приложении звездочками. Современные версии браузеров распознают все 140 названий цветов.

Форматирование текста

В данном разделе будут рассмотрены возможности форматирования отдельных символов текста документа.

Тэги уровня блока и последовательные тэги

Некоторые HTML-тэги, которые могут появляться в разделе BODY, называют *тэгами уровня блока* (block level), в то время как другие *последователь-*

ными (inline) тэгами или, называя по-другому, *тэгами уровня текста* (text level), хотя такое разделение тэгов по уровням в известной степени условно.

Различие уровней HTML-тэгов заключается в следующем: тэги уровня блока могут содержать последовательные тэги и другие тэги уровня блока, тогда как последовательные тэги содержат только данные и другие последовательные тэги. Блочные тэги описывают более крупные структуры документов, по сравнению с последовательными тэгами.

По умолчанию тэги этих видов размещаются в тексте описания различным образом: тэги уровня блока начинаются с новой строки, в то время как последовательные — нет.

Логическое и физическое форматирование

Для форматирования текста HTML-документов предусмотрена целая группа тэгов, которую можно условно разделить на тэги *логического* и *физического* форматирования.

Тэги логического форматирования обозначают (своими именами) структурные типы своих текстовых фрагментов, такие, например, как *программный код* (тэг <CODE>), *цитата* (тэг <CITE>), *аббревиатура* (тэг <ABBR>) и т. д. (см. описания этих и других подобных тэгов в следующем разделе). С помощью тэгов и можно, например, отметить отдельные фрагменты как *выделенные*, или *сильно выделенные*. Заметим, что речь идет о *структурной разметке*, которая не влияет на конкретное экранное представление фрагмента браузером. Поэтому такая разметка и называется *логической*. Фрагменты с логическим форматированием браузеры отображают на экране определенным образом, заданным по умолчанию. Вид отображения никак не связан со структурным типом фрагмента (т. е. именем тэга логического форматирования), но может быть легко переопределен.

Тэги физического форматирования определяют *формат отображения* указанного в них фрагмента текста в окне браузера (согласно предпочтениям автора документа). Например, для отображения фрагмента *курсивом* можно использовать тэг курсива <I>. Этот и другие, часто используемые тэги физического форматирования описаны ниже в разделе "*Тэги физического форматирования текста*" этой главы.

Между разработчиками HTML-документов долгое время шли споры о преимуществах и недостатках того или иного подхода. С выходом спецификации HTML 4.0 эти споры завершились в пользу применения логического форматирования, поскольку был провозглашен принцип отделения структуры документа от его представления. Действительно, только на базе логического форматирования можно гибко управлять представлением документа, используя современные методы (основанные на таблицах стилей, динамически изменяющихся документах и т. д.).

Тем не менее, на настоящий момент может свободно использоваться и физическое форматирование. В спецификации HTML 4.0 некоторые тэги физического форматирования не рекомендуются для применения, однако, пока они все еще поддерживаются всеми браузерами. Заметим, что некоторые тэги логического форматирования, призванные заменить отдельные тэги физического форматирования, распознаются не всеми браузерами, что делает их применение крайне неудобным. Примером может служить логический тэг ``, который рекомендуется использовать вместо физического тэга `<STRIKE>`.

Рассматриваемые ниже тэги относятся к тэгам уровня текста, т. е. призваны, в основном, размечать небольшие группы символов. Некоторые тэги могут задавать разметку и на уровне блока.

Тэги логического форматирования текста

Тэг `<ABBR>`

Тэг `<ABBR>` отмечает свой текст как *аббревиатуру* (ABBREVIATION). Несмотря на то, что этот тэг включен в спецификацию HTML 4.0, он до настоящего времени не поддерживается ни одним браузером.

Тэг `<ACRONYM>`

Тэг `<ACRONYM>`. Так же, как и тэг `<ABBR>`, используется для отметки аббревиатур. Этим тэгом рекомендуется отмечать так называемые *акронимы*, т. е. произносимые слова, состоящие из аббревиатур. Тэг `<ACRONYM>` возможно в будущем станет использоваться для невизуального отображения элементов, например при речевом синтезе.

Данный тэг удобно использовать в сочетании с параметром `TITLE`, в качестве значения которого можно указать полную форму записи аббревиатуры. Тогда визуальные браузеры при наведении курсора на текст, размеченный тэгом `<ACRONYM>`, будут выдавать полное наименование в виде появляющейся подсказки.

Заметим, что тэг `<ACRONYM>` распознается только браузером Microsoft Internet Explorer. Пример:

```
<ACRONYM TITLE="Санкт-Петербургский государственный институт  
точной механики и оптики">
```

СПБГИТМО</ACRONYM> – один из ведущих технических вузов Санкт-Петербурга

Тэг `<CITE>`

Тэг `<CITE>` используется для отметки цитат или названий книг и статей, ссылок на другие источники и т. д. Браузерами такой текст обычно выводится курсивом. Пример:

```
<CITE>Невское время</CITE> является одной из наиболее популярных  
городских газет Санкт-Петербурга
```

Тэг <CODE>

Тэг <CODE> отмечает свой текст как *небольшой фрагмент программного кода*. Как правило, отображается моноширинным шрифтом. Этот тэг не следует путать с тэгом <PRE>, являющимся элементом уровня блока, который следует использовать для отметки *больших фрагментов (листингов) кода*.

Например:

```
Пример простейшего оператора языка программирования C:<BR>
<CODE>puts("Hello, World!");</CODE>
```

Есть еще одно различие в использовании тэгов <CODE> и <PRE>. В коде программ часто бывает важно наличие нескольких идущих подряд пробелов. Их отображение будет сохранено только при использовании тэга <PRE>.

Тэг

Тэг отмечает свой текст как *удаленный*. Этот элемент полезно использовать для отметки изменений, вносимых в документ от версии к версии. Тэг может использоваться как элемент уровня текста и как элемент уровня блока.

Тэг имеет два необязательных параметра: SITE и DATETIME. Значение параметра SITE должно представлять собой URL-адрес документа, поясняющего причины удаления данного фрагмента.

Параметр DATETIME указывает дату удаления в формате: YYYY-MM-DDThh:mm:ssTZD, определяющем год, месяц, число, часы, минуты и секунды удаления, а также часовой пояс (Time Zone). Например:

```
Последней принятой спецификацией языка разметки HTML является версия
<DEL SITE="http://www.w3.org/TR/1999/10-29T16:12:53+0.00" 3.2</DEL> 4.0
```

Текст, помеченный тэгом обычно отображается перечеркнутым текстом. В спецификации HTML 4.0 этому тэгу отдается предпочтение перед тэгом физического форматирования <STRIKE> или <S>, обозначающих перечеркнутый текст. Однако тэг в настоящее время распознается только браузером Microsoft Internet Explorer.

Тэг <DFN>

Тэг <DFN> отмечает свой текстовый фрагмент как *определение* (DeFinitioN). Например, этим тэгом можно отметить какой-либо термин, когда он встречается в тексте в первый раз. Пример:

```
<DFN>Internet Explorer</DFN> – это популярный Web-браузер
```

Тэг <DFN> поддерживается только браузером Microsoft Internet Explorer. Отображается по умолчанию курсивом.

Тэг <INS>

Тэг <INS> отмечает свой текст как *вставку* (INSeRtion). Этот элемент полезно использовать для отметки изменений, вносимых в документ от версии к версии. Тэг <INS> может использоваться как элемент уровня текста и как элемент уровня блока.

Тэг имеет два необязательных параметра: CITE и DATETIME. Значение параметра CITE должно представлять собой URL-адрес документа, поясняющего подробности внесенных дополнений.

Параметр DATETIME указывает дату вставки в формате: YYYY-MM-DDThh:mm:ssTZD, определяющем год, месяц, число, часы, минуты и секунды вставки, а также часовой пояс (Time Zone).

Текст, помеченный тэгом <INS>, обычно отображается подчеркнутым текстом. Тэг <INS> в настоящее время распознается только браузером Microsoft Internet Explorer.

Тэг

Тэг (EMphasis — выделение, подчеркивание) используется для выделения важных фрагментов текста. Браузеры обычно отображают такой текст курсивом. Пример:

Пример выделения отдельных слов текста

Применение данного тэга предпочтительнее применения тэга физического форматирования <I>.

Тэг <KBD>

Тэг <KBD> отмечает текст как вводимый пользователем с клавиатуры. Обычно отображается моноширинным шрифтом, например:

Чтобы запустить текстовый редактор, напечатайте: <KBD>notepad</KBD>

Применение данного тэга предпочтительнее применения тэга физического форматирования <TT>.

Тэг <Q>

Тэг <Q> отмечает *короткие цитаты* в строке текста. В отличие от тэга уровня блока <BLOCKQUOTE> при отображении не выполняется отделение размеченного текста пустыми строками. Обычно отображается курсивом. Тэг <Q> (в отличие от <BLOCKQUOTE>) в настоящее время распознается только браузером Microsoft Internet Explorer.

Тэг имеет параметр CITE, в качестве значения которого можно указать источник цитаты.

Тэг <SAMP>

Тэг <SAMP> отмечает текст как *образец* (SAMPle). Обычное использование этого тэга — отметка текста, выдаваемого программами (sample output). Ис-

пользуется также для выделения нескольких символов моноширинным шрифтом.

Применение данного тэга предпочтительнее применения тэга физического форматирования <TT>. Например:

В результате работы программы будет напечатано: <SAMP>Hello, World!</SAMP>.

Тэг

Тэг , как правило, используется для *выделения* важных фрагментов текста. Браузеры обычно отображают такой текст полужирным шрифтом. Пример:

Санкт-Петербург расположен в самой восточной оконечности Финского залива в устье реки Невы

Применение данного тэга предпочтительнее применения тэга физического форматирования . Тэгом обычно размечают более важные фрагменты текста, чем те, что размечены тэгом .

Тэг <VAR>

Тэг <VAR> отмечает *имена переменных* программ. Обычно такой текст отображается курсивом. Пример:

Задайте значение переменной <VAR>N</VAR>

Отображение некоторых из примеров, приведенных при описании тэгов логического форматирования текста, показано на рис. 1.1. На рисунке видно, как при наведении указателя мыши на текст, отмеченный тэгом <ACRONYM>, выдается подсказка.

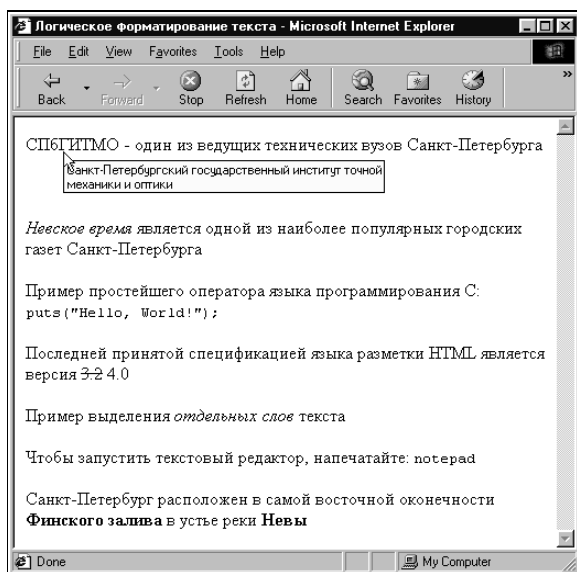


Рис. 1.1. Примеры форматирования текста