

Ю. Е. ЕДОМСКИЙ

Техника

# WEB- ДИЗАЙНА

ДЛЯ  
СТУДЕНТА



Юрий Едомский

Техника  
**WEB-  
ДИЗАЙНА**  
**ДЛЯ СТУДЕНТА**

Санкт-Петербург

«БХВ-Петербург»

2005

УДК 681.3.06  
ББК 32.973.26-018.2  
Е32

## **Едомский Ю.**

Е32 Техника Web-дизайна для студента. — СПб.: БХВ-Петербург, 2005. — 400 с.: ил.

ISBN 5-94157-742-7

Рассмотрен широкий круг вопросов, связанных с разработкой web-сайтов с использованием HTML и CSS, начиная от простых страниц до создания интерактивных и динамических сайтов. Даны основы компьютерной графики, описана работа с программами Photoshop и ImageReady в плане подготовки изображений к публикации на сайте. Рассмотрены вопросы редактирования сайтов в программе Dreamweaver. Для создания динамических страниц приведены необходимые сведения о программировании на JavaScript.

Книга будет полезна студентам при изучении интернет-технологий, написании курсовых и дипломных проектов, а также преподавателям и всем желающим создавать самостоятельно интерактивные и динамические страницы.

*Для широкого круга пользователей*

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>Елена Сухогозова</i>
Компьютерная верстка	<i>Екатерины Трубниковой</i>
Корректор	<i>Татьяна Кошелева</i>
Дизайн обложки	<i>Игоря Цырульниковой</i>
Зав. производством	<i>Николай Тверских</i>

Лицензия ИД № 02429 от 24.07.00. Подписано в печать 06.10.05.

Формат 60×90<sup>1/16</sup>. Печать офсетная. Усл. печ. л. 32,25.

Тираж 3000 экз. Заказ №

"БХВ-Петербург", 190005, Санкт-Петербург, ул. Есенина, 5Б.

Санитарно-эпидемиологическое заключение на продукцию № 77.99.02.953.Д.006421.11.04 от 11.11.2004 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

ISBN 5-94157-742-7

© Едомский Ю. Е.  
© Оформление, издательство "БХВ-Петербург", 2005

# Содержание

<b>ВВЕДЕНИЕ</b> .....	<b>1</b>
<b>ГЛАВА 1. СОЗДАНИЕ ПРОСТЫХ WEB-СТРАНИЦ</b> .....	<b>5</b>
1.1. Основные понятия и определения.....	5
1.1.1. Задание цвета .....	7
1.1.2. Задание размера шрифта .....	8
1.2. Создание HTML-документа в Блокноте .....	8
1.3. Метки форматирования шрифта .....	10
1.3.1. Метки логического форматирования шрифта.....	10
1.3.2. Метки физического форматирования шрифта .....	12
1.3.3. Вставка специальных символов.....	14
1.4. Метки форматирования текста.....	17
1.5. Создание гипертекстовых ссылок .....	24
1.5.1. Ссылки на внешние ресурсы Интернета.....	26
1.5.2. Адресация внутри одного документа.....	27
1.5.3. Ссылки на документы других типов .....	28
1.5.4. Изменение цвета гипертекста .....	28
1.6. Включение дополнительной информации о документе .....	29
1.6.1. Обновление содержимого страницы.....	29
1.6.2. Переадресация документа.....	29
1.6.3. Включение информации для поисковых систем .....	30
1.7. Метки-контейнеры <DIV> и <SPAN>.....	30
<b>ГЛАВА 2. СОЗДАНИЕ ПРОСТЫХ СТРАНИЦ В РЕДАКТОРЕ MACROMEDIA DREAMWEAVER MX 2004</b> .....	<b>31</b>
2.1. Интерфейс программы Macromedia Dreamweaver MX 2004.....	32
2.2. Определение нового сайта .....	38

2.3. Создание отдельных страниц .....	42
2.4. Работа с текстом .....	45
2.4.1. Форматирование шрифта и текста .....	45
2.4.2. Вставка специальных символов.....	52
2.5. Создание гипертекстовых ссылок.....	54
2.5.1. Создание гиперссылки по фрагменту уже имеющегося текста..	54
2.5.2. Создание ссылки вместе с гипертекстом.....	55
2.5.3. Создание гиперссылки внутри одного документа .....	57
2.5.4. Создание гиперссылки на почтовый адрес .....	58

## **ГЛАВА 3. ПОДГОТОВКА И РАЗМЕЩЕНИЕ ГРАФИЧЕСКИХ ИЗОБРАЖЕНИЙ .....**

**59**

3.1. Форматы графических файлов, используемых в web-сайтах.....	59
3.2. Размещение графики в HTML-документах.....	61
3.3. Работа с графикой в программе Dreamweaver.....	65
3.3.1. Размещение графики .....	65
3.3.2. Изменение свойств изображения.....	68
3.4. Подготовка изображений к публикации на сайте .....	71
3.4.1. Интерфейс программы Adobe Photoshop.....	72
3.4.2. Настройка интерфейса программы .....	74
3.4.3. Отмена и возврат выполненных действий.....	74
3.4.4. Сохранение изображений.....	75
3.4.5. Изменение размеров изображения .....	81
3.4.6. Кадрирование (обрезка) изображений .....	83
3.4.7. Выбор цвета .....	84
3.4.8. Работа с текстом .....	86
3.4.9. Создание простых фигур.....	91
3.4.10. Инструменты рисования .....	94
3.4.11. Клонирование изображений .....	95
3.4.12. Выделение фрагментов изображения.....	96
3.4.13. Копирование и перемещение выделенного фрагмента .....	102
3.4.14. Использование слоев .....	107
3.4.15. Эффекты слоев.....	110
3.4.16. Объединение слоев .....	117
3.4.17. Заливка изображений .....	119
3.4.18. Создание прозрачного фона изображения.....	122
3.4.19. Использование фильтров .....	126
3.4.20. Подготовка фоновых изображений.....	128
3.4.21. Создание GIF-анимации .....	130
3.4.22. Фрагментация изображений .....	145
3.4.23. Создание изображений-карт .....	154

<b>ГЛАВА 4. ИСПОЛЬЗОВАНИЕ ТАБЛИЦ ДЛЯ РАЗМЕЩЕНИЯ ДАННЫХ И КОМПОНОВКИ СТРАНИЦ</b> .....	<b>159</b>
4.1. Создание таблиц средствами HTML .....	159
4.2. Работа с таблицами в программе Dreamweaver.....	167
4.2.1. Добавление таблиц .....	167
4.2.2. Изменение свойств таблицы .....	169
4.2.3. Изменение свойств ячеек таблицы .....	172
4.2.4. Создание таблицы в режиме компоновки страницы .....	174
<b>ГЛАВА 5. КОМПОНОВКА СТРАНИЦ С ИСПОЛЬЗОВАНИЕМ ФРЕЙМОВ</b> .....	<b>181</b>
5.1. Создание фреймов средствами HTML .....	181
5.1.1. Деление окна браузера на фреймы .....	181
5.1.2. Описание фреймов.....	183
5.1.3. Дополнительные параметры меток <code>&lt;FRAMESET&gt;</code> и <code>&lt;FRAME&gt;</code> .....	187
5.1.4. Ссылки из документов, находящихся во фреймах .....	188
5.2. Создание фреймов в программе Dreamweaver .....	189
5.2.1. Создание нового документа с фреймовой структурой .....	189
5.2.2. Создание документа с фреймовой структурой на базе имеющегося HTML-документа.....	190
5.2.3. Изменение фреймовой структуры документа.....	190
5.2.4. Изменение параметров документа с фреймовой структурой.....	192
5.2.5. Изменение содержимого отдельных фреймов .....	195
5.2.6. Создание HTML-документа для каждого фрейма.....	197
5.2.7. Сохранение документа с фреймовой структурой и связанных с ним документов .....	197
<b>ГЛАВА 6. СОЗДАНИЕ ИНТЕРАКТИВНЫХ СТРАНИЦ С ИСПОЛЬЗОВАНИЕМ ФОРМ</b> .....	<b>199</b>
6.1. Создание форм средствами HTML.....	199
6.2. Пересылка содержимого формы .....	204
6.3. Создание форм в программе Dreamweaver .....	208
<b>ГЛАВА 7. КАСКАДНЫЕ ТАБЛИЦЫ СТИЛЕЙ — РАСШИРЕНИЕ ВОЗМОЖНОСТЕЙ ФОРМАТИРОВАНИЯ</b> .....	<b>219</b>
7.1. Назначение каскадных таблиц стилей .....	219
7.2. Связь каскадной таблицы стилей с HTML-документом.....	220
7.3. Задание одинаковым меткам разных параметров форматирования.....	223
7.4. Создание стиля форматирования, применимого к различным меткам .....	225

7.5. Вложение меток .....	226
7.6. Единицы измерения в каскадных таблицах стилей .....	229
7.7. Основные свойства каскадной таблицы стилей .....	231
7.7.1. Свойства форматирования шрифтов.....	231
7.7.2. Свойства цвета и фона.....	233
7.7.3. Свойства форматирования текста .....	235
7.7.4. Свойства блоков.....	239
7.7.5. Позиционирование объектов .....	244
7.7.6. Использование слоев.....	248
7.7.7. Видимость объектов .....	249
7.7.8. Изменение внешнего вида указателя мыши .....	251
7.7.9. Изменение цвета элементов полосы прокрутки.....	252
7.8. Работа с каскадными таблицами стилей в программе Dreamweaver .....	253
7.8.1. Создание новой таблицы стилей .....	254
7.8.2. Редактирование таблицы стилей .....	264
7.8.3. Присоединение внешней таблицы стилей .....	266
7.8.4. Применение стиля форматирования или отказ от его применения.....	267
7.8.5. Работа со слоями.....	270
<b>ГЛАВА 8. СОЗДАНИЕ ДИНАМИЧЕСКИХ СТРАНИЦ .....</b>	<b>277</b>
8.1. Структура и размещение программы.....	277
8.2. Типы данных .....	278
8.3. Литералы и переменные. Оператор присваивания .....	279
8.4. Выражения .....	282
8.5. Подпрограммы-функции.....	288
8.6. Локальные и глобальные переменные .....	291
8.7. Основные понятия объектного программирования.....	295
8.8. Подпрограммы-функции с параметрами .....	298
8.9. Оператор условного перехода .....	303
8.10. Окно браузера как объект программирования .....	314
8.11. Перетаскивание объектов .....	319
8.12. Создание раскрывающегося графического меню .....	323
8.12.1. Горизонтальное меню.....	323
8.12.2. Вертикальное меню .....	327
8.12.3. Компактное меню .....	330
8.13. Поиск информации на странице .....	331
8.14. Дата и время в HTML-документах .....	333
8.15. Понятие массива .....	336
8.15.1. Электронный календарь .....	337
8.15.2. Электронные часы .....	339
8.15.3. Обновление содержимого в зависимости от времени суток ....	340

---

8.16. Создание всплывающей анимированной подсказки.....	342
8.17. Вывод текста в строку состояния браузера .....	343
8.18. Вывод в строку состояния браузера бегущего текста.....	344
8.19. Автоматическая компоновка страницы в зависимости от разрешения экрана .....	346
<b>ГЛАВА 9. ДОБАВЛЕНИЕ СТАТИЧЕСКИХ И ДИНАМИЧЕСКИХ ЭФФЕКТОВ.....</b>	<b>351</b>
9.1. Статические фильтры.....	352
9.2. Динамические фильтры (переходы).....	367
9.2.1. Описание динамических фильтров .....	368
9.2.2. Примеры использования динамических фильтров .....	373
<b>ПРЕДМЕТНЫЙ УКАЗАТЕЛЬ .....</b>	<b>385</b>



# Введение

Книга предназначена для широкого круга читателей — от начинающих создателей web-страниц до имеющих определенный опыт, но желающих расширить свои знания. Прежде всего, она адресована студентам, работающим над курсовыми и дипломными проектами. Полезную информацию найдут для себя и школьники, делающие первые шаги в освоении сети Интернет, а также специалисты, желающие заявить о себе и своих достижениях в Интернете, но не имеющие достаточных знаний в области компьютерных технологий.

Требования к новичкам не слишком велики:

- владение элементарными приемами работы с операционной системой Windows, а именно умение открыть нужную программу, создать папку, сохранить в ней файл или найти ранее сохраненный файл;
- элементарные навыки практической работы в Интернете: наличие представления о том, что для просмотра документов, расположенных в Интернете или создаваемых для этих целей, предназначена программа браузер и хотя бы небольшой опыт работы с такой программой.

Главное достоинство книги состоит в том, что в ней можно найти ответы на многие вопросы, которые обычно рассредоточены по разным источникам. Например, для подготовки изображений к публикации в Интернете необходима литература по компьютерной графике, с целью ускорения работы потребуется знакомство с редактором web-сайтов, а при создании динамических страниц понадобится литература по программированию. Обычно специальная литература по каждому из этих направлений имеет

большой объем и достаточно сложна для начинающих. Несмотря на широкий спектр излагаемого материала и небольшой объем книги, важные вопросы излагаются достаточно подробно, а сокращение достигается за счет материалов, без которых вполне можно обойтись.

Прочитав первые две главы и хотя бы часть третьей, новичок уже сможет приступить к созданию собственных web-страниц. Дочитав третью главу, можно дополнить свои страницы рисунками причудливых форм, анимированными изображениями и специально подготовленными фоновыми рисунками. В целом главы 2 и 3 могут быть полезны разработчикам разного уровня.

В главе 2 приводится первая часть описания программы создания сайтов Macromedia Dreamweaver MX 2004. Знакомство с этой программой продолжится и в последующих главах, когда ее использование может существенно ускорить работу над сайтом.

Материалы главы 3 будут полезны всем, кто раньше не занимался компьютерной графикой. В ней достаточно подробно рассмотрены вопросы подготовки изображений к публикации на сайте как в программе Adobe Photoshop, так и в программе ImageReady, поставляемой фирмой Adobe вместе с программой Photoshop и существенно расширяющей ее возможности именно по созданию изображений для web-страниц.

Новичку не следует торопиться с переходом к последующим главам. Получив навыки практической работы по созданию простых страниц с иллюстрациями и поняв, что не удастся расположить рисунки и фрагменты текста в произвольных местах страницы, можно переходить к созданию таблиц (глава 4).

Назначение таблиц состоит не только в привычном расположении структурированных данных в определенном порядке, но и в осуществлении компоновки элементов страницы. Используя для табличной компоновки страницы программу Dreamweaver, можно значительно облегчить эту работу.

В главе 5 предлагается познакомиться с еще одним методом компоновки — фреймами. Здесь необходимо понять, в каких случаях лучше использовать таблицы, а в каких фреймы и в чем их принципиальное отличие.

Глава 6 имеет достаточно обособленный характер. Основное назначение форм, рассматриваемых в этой главе, — создание об-

ратной связи между пользователями и владельцем сайта. После прочтения материалов *главы 6* можно не только заявить о себе в сети Интернет, но и облегчить посетителю сайта отправку отзывов на размещенные материалы. Даже если в ближайшее время вы не планируете организацию такого диалога, пропускать этот материал не следует, так как формы используются в качестве объектов программирования в *главе 8*.

*Глава 7* знакомит со свойствами каскадных таблиц стилей (CSS), которые позволяют существенно дополнить и расширить возможности языка HTML и в случае совместного использования с программой-сценарием на языке JavaScript создавать динамические страницы. По всей видимости, на изучение этой главы придется потратить больше времени, чем на изучение некоторых предыдущих, но это, безусловно, окупится теми практическими возможностями, которые станут доступными. В конце главы можно будет познакомиться с еще одним методом компоновки страницы с использованием слоев.

*Главу 8* следует рассматривать как введение в программирование на языке JavaScript. Программирование на любом из алгоритмических языков, в том числе и на JavaScript, является достаточно сложной задачей особенно на ранней стадии изучения. Называть себя программистом после прочтения главы, и даже после практического выполнения приведенных там примеров, будет преждевременным. Тем не менее содержащиеся в *главе 8* сведения позволяют получить практические навыки по созданию простых, но достаточно полезных программ, познакомиться с основными понятиями программирования. Тот, кому изложенный в главе материал покажется слишком сложным, сможет воспользоваться приведенными примерами, не вдаваясь в тонкости их реализации. Тот же, кто почувствует интерес к программированию, прежде всего, должен попытаться решить несколько собственных задач, базирующихся на предлагаемых примерах. В случае успеха в дальнейшем можно приступить к изучению специальной литературы по программированию, а полученные начальные знания позволят более осмысленно выбрать специальную литературу и облегчат процесс ее освоения.

*Глава 9* посвящена применению мультимедийных эффектов, разработанных фирмой Microsoft и поддерживаемых браузером

MS Internet Explorer. Начиная с версии 5.5 браузера, фирма Microsoft изменила синтаксис задания фильтров, принцип функционирования динамических фильтров, расширила перечень фильтров. Все эти изменения нашли отражение в *главе 9*.

Каждая глава содержит множество примеров, иллюстрирующих пройденный материал. Все примеры работоспособны в браузере Microsoft Internet Explorer 6.0. Этот браузер выбран не случайно. Дело в том, что он является не только самым популярным среди посетителей Интернета, но и, как правило, превосходит конкурентов по своим возможностям. При попытке просмотра созданных документов в браузерах более ранних версий, а в особенности других производителей, желаемого результата можно не увидеть. Это, прежде всего, относится к материалам *глав 8 и 9*.

Для успешного освоения материалов книги читателю следует установить на своем компьютере программы Macromedia Dreamweaver MX 2004 и Adobe Photoshop 8.0 CS. Это последние на момент издания книги версии программ. Конечно, отличия последних версий программ от предыдущих не столь значительны, чтобы не суметь разобраться в их работе по приведенному описанию. Тем не менее определенные отличия есть, что потребует дополнительного времени для освоения программ. Кроме того, со временем все равно придется осваивать более новые версии, а делать это, перепрыгивая несколько ступеней, сложнее.

В книге не рассматриваются вопросы применения современной и весьма перспективной технологии Flash фирмы Macromedia. Возможно, что это ее недостаток, который со временем придется исправить. По всей видимости, читатель, получивший определенный опыт и уверенность в своих силах, сам заинтересуется технологией Flash и воспользуется для этого специальной литературой.



# Глава 1

## Создание простых web-страниц

### 1.1. Основные понятия и определения

Документы, которыми осуществляется обмен информацией в Интернете, обычно представляют собой текстовые файлы, написанные с использованием специального языка HTML (HyperText Markup Language, язык разметки гипертекста). Язык HTML — это набор меток, которые расставляются по тексту и определяют внешний вид документа в окне браузера.

*Браузеры* — специальные программы, предназначенные для просмотра документов, размеченных HTML-метками. В настоящее время самым распространенным браузером является Microsoft Internet Explorer.

*Метки* — небольшие последовательности символов, ограниченные угловыми скобками и предлагающие браузеру определенные действия (команды). Например, встретив в тексте HTML-документа метку, предписывающую сделать фрагмент текста полужирным, браузер выполнит это требование. Точно также, используя соответствующие метки, можно предложить браузеру изменить размер и цвет шрифта, выровнять текст, вставить рисунки и т. д. Некоторые метки применяются попарно, одна из них называется *открывающей*, другая — *закрывающей*. Действие метки будет распространяться на фрагмент документа, заключенный между открывающей и закрывающей меткой. В англоязычной литературе метки именуется словом *tag* (ярлык, бирка, признак).

Этот термин механически перешел и в отечественную литературу, причем часть авторов пишет *тэг*, а не менее значительная часть — *тег*. Слово "метка" более точно отражает существо вопроса, поэтому мы будем использовать именно этот термин.

Открывающая метка может иметь следующую структуру:

```
<имя [параметр] [= "значение"] [параметр] [= "значение"] ...>
```

Элементы, написанные в квадратных скобках, являются необязательными составляющими метки. Это означает, что метки могут не содержать *параметры* или содержать параметры без значений. Но если параметры есть, квадратные скобки писать не нужно. Имя метки, а также параметры отделяются друг от друга пробелами. После символа "<" и перед символом ">" пробелы не ставятся. Многоточие означает, что параметров может быть много. *Значения параметров* записываются в прямых кавычках, однако, если значение параметра не содержит пробелов, то кавычки можно не писать.

Закрывающая метка содержит только имя, перед которым ставится правый слеш:

```
</имя>
```

Как видим, закрывающая метка не содержит никаких параметров. Ее назначение — ограничение действия открывающей метки. Не всякая метка имеет в паре закрывающую метку.

Для написания меток используются только латинские буквы. Допускается использование как строчных, так и заглавных букв.

Структура HTML-документа:

```
<HTML>
```

```
<HEAD>
```

```
<TITLE>
```

Заголовок окна документа

```
</TITLE>
```

```
</HEAD>
```

```
<BODY>
```

Тело документа

```
</BODY>
```

```
</HTML>
```

Такую структуру должен иметь любой HTML-документ. Она включает три пары меток с именами HTML, HEAD, BODY, которые в данном случае не имеют параметров.

В головной части документа, то есть внутри пары меток <HEAD>...</HEAD>, могут размещаться другие метки. На первых порах будем использовать только единственную и обязательную пару меток <TITLE>...</TITLE>, предназначенную для размещения заголовка окна браузера. После открытия документа текст заголовка будет воспроизводиться в строке заголовка окна браузера (синяя верхняя полоса окна).

В теле документа размещается текст и метки, определяющие его внешний вид в окне браузера.

### 1.1.1. Задание цвета

В HTML для задания цвета используется 6-разрядное шестнадцатеричное число, перед которым ставится символ #. В качестве цветовой модели используется RGB (Red — красный, Green — зеленый, Blue — синий). Шестнадцатеричная система счисления имеет 16 цифр (0, 1, ..., 9, A, B, C, D, E, F). Для задания каждой из трех составляющих цвета отводится два разряда шестнадцатеричного числа, что позволяет задать значения от 00 до FF, соответствующие значениям 0–255 в десятичной системе счисления. Например, запись #FF0000 означает, что красная составляющая цвета имеет максимальное значение FF (255 в десятичной системе), т. е. максимальную яркость, а остальные составляющие цвета отсутствуют. Очевидно, что таким образом задан ярко-красный цвет. Для задания темно-красного цвета используется запись #800000. В этом случае красная составляющая имеет значение 80 (128 в десятичной системе), остальные составляющие по-прежнему отсутствуют. Изменяя значения отдельных составляющих, можно задать 16777216 различных цветов. Например: #0000FF — синий, #00FFFF — голубой, #FFFF00 — желтый, #808080 — серый, #FFFFFF — белый, #000000 — черный. Современные браузеры позволяют не писать перед числом символ #, однако лучше этой возможностью не пользоваться.

Для задания цвета также можно использовать английские названия цветов. Например: red — красный, cyan — голубой, magenta — пурпурный, yellow — желтый, white — белый, black — черный.

### 1.1.2. Задание размера шрифта

В HTML размер шрифта задается в условных единицах от 1 до 7. Разными браузерами шрифт одного и того же размера может воспроизводиться по-разному. На рис. 1.1 показано воспроизведение шрифтов браузером Microsoft Internet Explorer.

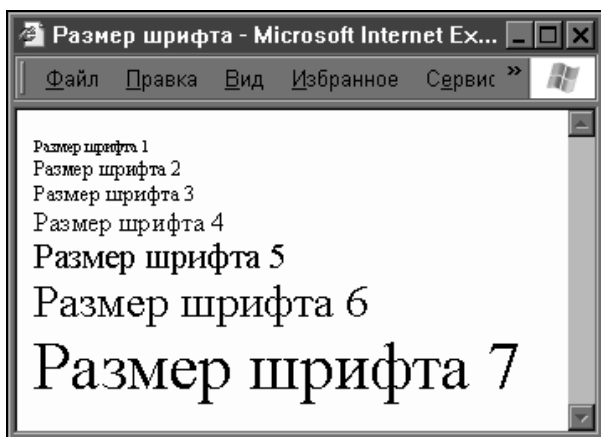


Рис. 1.1. Воспроизведение шрифтов разного размера браузером Microsoft Internet Explorer

Язык HTML не позволяет задать размеры шрифта, отличающиеся от приведенных выше. Для задания произвольных размеров необходимо использовать свойства *каскадных таблиц стилей (CSS)* (см. главу 7).

## 1.2. Создание HTML-документа в Блокноте

Блокнот (Notepad) — простой текстовый редактор, входящий в стандартный набор компонентов MS Windows. Откроем Блокнот и наберем следующий текст, представленный в листинге 1.1.



**Листинг 1.1. Содержимое простого HTML-документа**

```
<HTML>
<HEAD>
<TITLE>Моя первая страница</TITLE>
</HEAD>
<BODY>
Мой первый HTML-документ
Здесь помещается текст моего первого HTML-документа, который
пока не содержит форматирования и воспроизводится браузером с
параметрами шрифта, принятыми по умолчанию.
</BODY>
</HTML>
```

После завершения набора документа выполним команду **Сохранить** меню **Файл**. В открывшемся диалоговом окне **Сохранить как** выберем папку, в которую хотим сохранить документ. В строке **Имя файла** введем имя с обязательным расширением `htm` или `html` (например `mydoc.htm`). В раскрывающемся списке **Тип файла** выберем пункт **Все файлы** и нажмем кнопку **Сохранить**. После этого можно закрыть блокнот, отыскать созданный документ в папке и дважды щелкнуть по имени файла. Благодаря расширению `htm` документ откроется в браузере, и вы увидите заголовок окна, набранный после метки `<TITLE>`, и текст, набранный после метки `<BODY>`.

Дальнейшую работу с документом можно продолжать, не закрывая окно браузера. Для этого нужно выбрать команду **Просмотр HTML-кода** меню **Вид**. Откроется окно блокнота с текстом создаваемого документа. Дополним текст документа фразой: Продолжаю редактировать мой первый документ. Дадим команду **Сохранить** и, свернув блокнот, перейдем к окну браузера. Выполним команду **Обновить** меню **Вид** (дублируется кнопкой **Обновить** на панели инструментов) и убедимся в том, что в тексте документа произошли изменения. После этого примера дальнейшая работа с документом становится очевидной. Она будет заключаться в открытии Блокнота, внесении в текст документа нужных изменений, сохранении документа и обновлении содержимого окна браузера.

## 1.3. Метки форматирования шрифта

Чтобы отформатировать шрифт нашего документа, то есть изменить его внешний вид, нужно расставить по тексту *метки форматирования шрифта*, которые и будут рассмотрены в этом разделе.

Метки форматирования шрифта принято разделять на метки *физического* и *логического* форматирования. Метки физического форматирования позволяют задать конкретные параметры шрифта, например, размер, цвет, начертание и так далее, которые будут реализованы браузером. Метки логического форматирования определяют лишь характер помечаемого текста, например: заголовок, аббревиатура, цитата и так далее, а браузер отобразит текст с теми физическими характеристиками, которые предусмотрены для подобного текста. Если одного и того же результата можно достичь как с помощью меток физического форматирования, так и меток логического форматирования, следует отдать предпочтение меткам логического форматирования.

### 1.3.1. Метки логического форматирования шрифта

В данном разделе рассматриваются метки логического форматирования, их запись и результат отображения в браузере.

- Увеличение размера шрифта: `<BIG>...</BIG>`.

Пример записи: `<BIG>Текст</BIG>`.

Размер шрифта увеличивается на единицу относительно текущего уровня.

- Уменьшение размера шрифта: `<SMALL>...</SMALL>`.

Пример записи: `<SMALL>Текст</SMALL>`.

Размер шрифта уменьшается на единицу относительно текущего уровня.

- Разметка аббревиатур: `<ACRONYM>...</ACRONYM>`.

Пример записи: `<ACRONYM TITLE="Hypertext Markup Language">HTML</ACRONYM>`.

В метке можно использовать параметр `TITLE`, который позволяет задать полную форму записи аббревиатуры, появ-

ляющуюся в браузере в виде всплывающей подсказки, при наведении указателя мыши на текст.

- Разметка цитат `<CITE>...</CITE>`.

Пример записи: `<CITE>Текст</CITE>`.

Текст выводится курсивом.

- Разметка удаленного текста: `<DEL>...</DEL>`.

Пример записи: `<DEL>Текст</DEL>`.

Текст перечеркивается горизонтальной линией.

- Разметка терминов или определений: `<DFN>...</DFN>`.

Пример записи: `<DFN>Текст</DFN>`.

Текст выводится курсивом.

- Разметка текста вставляемого в документ и изменяющего его по отношению к предыдущей версии: `<INS>...</INS>`.

Пример записи: `<INS>Текст</INS>`.

Текст подчеркивается.

- Разметка важных фрагментов текста: `<EM>...</EM>`.

Пример записи: `<EM>Текст</EM>`.

Текст выводится курсивом.

- Разметка текста, вводимого пользователем с клавиатуры: `<KBD>...</KBD>`.

Пример записи: `<KBD>Текст</KBD>`.

Текст выводится моноширинным шрифтом (все символы одинаковой ширины).

- Разметка важных фрагментов текста: `<STRONG>...</STRONG>`.

Пример записи: `<STRONG>Текст</STRONG>`.

Текст выводится полужирным шрифтом.

- Разметка заголовков (6 уровней):

`<H1>...</H1>`

`<H2>...</H2>`

...

`<H6>...</H6>`

Пример записи: `<H4>Заголовок</H4>`.

Текст выводится полужирным шрифтом. Размер шрифта зависит от выбранного уровня заголовка (`<H1>` — самый крупный, `<H6>` — самый мелкий). Метки могут иметь параметр `ALIGN` со значениями: `LEFT` — выравнивание заголовка по левому краю (действует по умолчанию), `RIGHT` — выравнивание заголовка по правому краю, `CENTER` — центрирование заголовка.

Пример записи: `<H1 ALIGN=CENTER>Заголовок</H1>`.

### 1.3.2. Метки физического форматирования шрифта

Наряду с метками логического форматирования, используются и метки физического форматирования, которые позволяют задать конкретные параметры шрифта не только для всего текста, но и для отдельных его элементов.

□ Задание гарнитуры, размера и цвета шрифта всего текста: `<BASEFONT>`.

Пример записи: `<BASEFONT FACE="Times New Roman" SIZE=5 COLOR=Green>Текст.`

Метка не имеет закрывающей метки и действует на весь текст, расположенный ниже.

Описание параметров:

- `FACE="значение"` — позволяет указать гарнитуру (рисунок) шрифта;
- `SIZE="значение"` — размер шрифта (целое число от 1 до 7);
- `COLOR="значение"` — цвет шрифта.

В данном примере запись значения параметра `FACE` без кавычек не допустима, так как состоит из нескольких слов, а значения параметров `SIZE` и `COLOR` можно писать без кавычек. Задавая гарнитуру шрифта в параметре `FACE`, следует помнить, что такой гарнитуры на компьютере пользователя может не оказаться. В этом случае текст будет воспроизводиться гарнитурой, предусмотренной по умолчанию, что может существенно изменить вид документа. В связи с этим иногда в

параметре `FACE` указывают через запятую в порядке убывания приоритета несколько гарнитур, относящихся к одному семейству, заканчивая список именем семейства, к которому они принадлежат. Например:

```
<BASEFONT FACE="Arial, Helvetica, sans-serif">Текст.
```

- Задание гарнитуры, размера и цвета шрифта фрагмента текста: `<FONT>...</FONT>`.

Пример записи: `<FONT FACE="Helvetica" SIZE=-1 COLOR=Red>Текст</FONT>`.

Параметры аналогичны параметрам метки `<BASEFONT>`, за исключением параметра `SIZE`. Запись значения параметра `SIZE` со знаком минус означает, что размер шрифта уменьшается на соответствующую величину относительно заданного по умолчанию или заданного в метке `<BASEFONT>`. Запись без знака означает, что размер задан абсолютно.

- Разметка полужирного шрифта: `<B>...</B>`.

Пример записи: `<B>Текст</B>`.

Рекомендуемый аналог `<STRONG>`.

- Разметка курсивного шрифта: `<I>...</I>`.

Пример записи: `<I>Текст</I>`.

Рекомендуемые аналоги `<EM>`, `<DFN>`.

- Разметка подчеркнутого шрифта: `<U>...</U>`.

Пример записи: `<U>Текст</U>`.

Рекомендуемый аналог: `<INS>`.

- Разметка моноширинного шрифта: `<TT>...</TT>`.

Пример записи: `<TT>Текст</TT>`.

Рекомендуемый аналог: `<KBD>`.

- Разметка шрифта, перечеркнутого горизонтальной линией: `<STRIKE>...</STRIKE>` или `<S>...</S>`.

Пример записи: `<STRIKE>Текст</STRIKE>`.

Рекомендуемый аналог: `<DEL>`.

- Создание подстрочного индекса: `<SUB>...</SUB>`.

Пример записи:  $X_{2}$ .

Результат действия:  $X_2$ .

□ Создание надстрочного индекса:  $Y^3$ .

Пример записи:  $Y^3$ .

Результат действия:  $Y^3$ .

## Пример форматирования шрифта

Созданный ранее в блокноте первый HTML-документ не содержал форматирования и воспроизводился браузером шрифтом, принятым по умолчанию. Пришла пора создать второй документ, похожий по содержанию, но использующий некоторые из рассмотренных выше меток для форматирования шрифта (листинг 1.2).

### Листинг 1.2. HTML-документ, содержащий метки форматирования шрифта

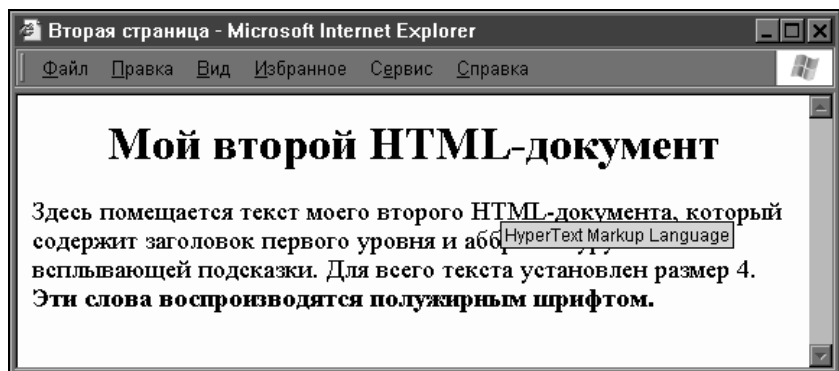
```
<HTML>
<HEAD>
<TITLE>Вторая страница</TITLE>
</HEAD>
<BODY>
<BASEFONT SIZE=4>
<H1 ALIGN=CENTER>Мой второй HTML-документ</H1>
Здесь помещается текст моего второго <ACRONYM TITLE="Hyper Text
Markup Language">HTML</ACRONYM>-документа, который содержит
заголовок первого уровня и аббревиатуру с текстом всплывающей
подсказки. Для всего текста установлен размер 4. <STRONG>Эти
слова воспроизводятся полужирным шрифтом.</STRONG>
</BODY>
</HTML>
```

Вид документа в окне браузера показан на рис. 1.2.

## 1.3.3. Вставка специальных символов

К *специальным символам* относятся служебные символы, используемые в языке HTML (< — меньше, > — больше, & — амперсанд, " — кавычки), а также некоторые специальные знаки.

В HTML для отображения специальных символов предусмотрены последовательности символов, некоторые из которых приведены в табл. 1.1.



**Рис. 1.2.** Результат форматирования шрифта

**Таблица 1.1.** Последовательность для отображения некоторых символов

Последовательность	Символ
&lt;	< знак меньше
&gt;	> знак больше
&nbsp;	неразрывный пробел (non-breaking space) <sup>1</sup>
&amp;	& амперсанд
&quot;	" прямые кавычки
&copy;	© знак авторского права (Copyright)
&reg;	® знак зарегистрированной торговой марки (Registered)

<sup>1</sup> Неразрывный пробел называется так потому, что слова, между которыми он находится, не могут быть разнесены браузером на разные строки, и если не будут помещаться на одной строке, то будут перенесены вместе. Поэтому его, в частности, можно использовать для разделения величины и единицы измерения. Например: `120&nbsp;км/час`.

Таблица 1.1 (окончание)

Последовательность	Символ
&lquo;	« левая парная кавычка
&rquo;	» правая парная кавычка
&mdash;	— длинное тире

Существует и более универсальный способ отображения символов. Нужные символы вводятся в документ при помощи специального кода, который включает знак амперсанда (&), знак номера (#), десятичный код символа и точку с запятой. Этот способ применяется также для отображения символов, которые невозможно ввести с клавиатуры. Примеры специальных кодов некоторых символов приведены в табл. 1.2.

Таблица 1.2. Специальные коды некоторых символов

Код	Символ
&#160;	неразрывный пробел
&#169;	© — знак авторского права
&#171;	« — левая кавычка
&#187;	» — правая кавычка

Чтобы определить код символа, в **Главном меню Windows** (кнопка **Пуск**) нужно воспользоваться командой **Программы | Стандартные | Служебные | Таблица символов**. В диалоговом окне **Таблица символов** в раскрывающемся списке **Шрифт** выбирается название гарнитуры, после чего появится список содержащихся в данной гарнитуре символов. Выделив щелчком нужный символ, в нижней части диалогового окна прочтем код символа. Символы некоторых гарнитур кодируются кодом ASCII (American Standard Code for Information Interchange), для которого используются два разряда шестнадцатеричного числа (записываются после символов 0x). Символы других гарнитур кодируются Юникодом (Unicode), использующим четырехразрядные шестнадцатеричные числа (записываются после символов U+). В обоих случаях их нужно предварительно перевести в десятичный код, для чего можно использовать **Калькулятор (Программы |**



**Стандартные | Калькулятор**) в режиме инженерных расчетов. Для примера выберите гарнитуру Arial и убедитесь в правильности кодов, приведенных в табл. 1.2.

Если гарнитура шрифта, используемая для основного текста, не содержит нужного символа, то для использования символа в HTML-документе требуется изменить гарнитуру и после этого отобразить нужный символ при помощи кода. Например, пусть в HTML-документе необходимо использовать символ "→". По **Таблице символов** определяем, что такой символ содержится в гарнитуре **Symbol** и кодируется шестнадцатеричным кодом AE. Переведя AE в десятичный, код получим 174. В HTML-документе необходимо написать:

```
<FONT FACE="symbol">&#174;</FONT>
```

## 1.4. Метки форматирования текста

Под форматированием текста подразумевается изменение внешнего вида документа путем разбиения его на абзацы, выравнивания абзацев, проведения горизонтальных разделительных линий, создания списков.

□ Создание абзаца: `<P>...</P>`.

Пример записи: `<P ALIGN=RIGHT>Текст абзаца</P>`.

Параметр `ALIGN` позволяет задать выравнивание строк абзаца и может иметь следующие значения:

- `LEFT` — выравнивание по левому краю (действует по умолчанию);
- `RIGHT` — выравнивание по правому краю;
- `CENTER` — центрирование строк;
- `JUSTIFY` — выравнивание по ширине (по правой и левой сторонам).

### Примечание

Закрывающая метка `</P>` не обязательна.

Абзацы отделяются друг от друга интервалами, равными пустой строке.

Первая строка абзаца не имеет отступа.

- Переход на новую строку в пределах абзаца: `<BR>`.

Текст, располагающийся после метки `<BR>`, будет воспроизводиться с начала новой строки, при этом строки абзаца сохранят выравнивание, заданное для всего абзаца.

- Вставка текста, предварительно отформатированного в текстовом редакторе: `<PRE>...</PRE>`.

Пример записи: `<PRE>Текст</PRE>`

Использованные в предварительно отформатированном тексте переходы на новую строку, пробелы, символы табуляции сохранятся при выводе его браузером. Текст будет выводиться моноширинным шрифтом;

- Отображение фрагмента текста с отступом: `<BLOCKQUOTE>...</BLOCKQUOTE>`.

Пример записи: `<BLOCKQUOTE>Текст</BLOCKQUOTE>`.

Основное назначение `<BLOCKQUOTE>` — выделение фрагментов текста (например цитат) путем изменения их положения относительно остального текста. Текст сдвигается на несколько позиций вправо, перед и после текста устанавливаются интервалы.

- Создание маркированного списка: `<UL>...</UL>`.

Список представляет собой последовательность абзацев, начала которых при выводе списка браузером будут отмечены специальным значком (маркером). До и после списка будут добавлены интервалы.

Пример записи:

```
<UL TYPE=disc>
  <LI>Первый абзац списка
  <LI>Второй абзац списка
  ...
  <LI>Последний абзац списка
</UL>
```

Метка `<LI>` отмечает начало абзаца, что обеспечивает его вывод с новой строки.

Параметр TYPE позволяет задать тип маркера и может принимать значения:

- disc — закрашенный кружок;
- circle — не закрашенный кружок (действует по умолчанию);
- square — закрашенный квадрат.

Следует обратить внимание, что вопреки принципу нечувствительности к регистру, характерному для HTML, перечисленные выше значения параметра TYPE, нужно писать только строчными буквами.

Допускается использование параметра TYPE в метке <LI>.

В качестве маркеров можно использовать специальные символы. Для этого можно применить следующий способ:

```
<UL>
<FONT FACE="wingdings">&#74;</FONT>Первая строка списка<BR>
<FONT FACE="wingdings">&#76;</FONT>Вторая строка списка<BR>
<FONT FACE="wingdings">&#79;</FONT>Третья строка списка<BR>
<FONT FACE="wingdings">&#60;</FONT>Четвертая строка списка<BR>
<FONT FACE="wingdings">&#38;</FONT>Пятая строка списка
</UL>
```

Дополнив фрагмент соответствующими метками, в окне браузера увидим результат, приведенный на рис. 1.3.

Чтобы в качестве маркера использовать рисунок, нужно вместо метки <FONT> поставить метку рисунка <IMG>. Например:

```
<UL>
<IMG SRC="имя файла рисунка">Первая строка списка<BR>
<IMG SRC="имя файла рисунка">Вторая строка списка
</UL>
```

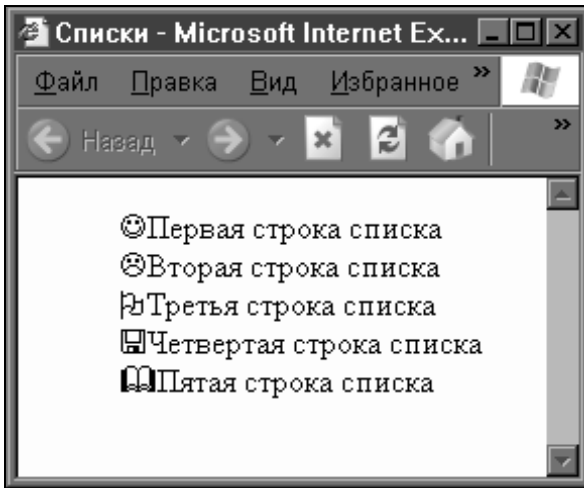
О размещении рисунков см. в главе 3.

- Создание нумерованного списка: <OL>...</OL>.

Пример записи:

```
<OL TYPE=a START=3>
  <LI>Первый элемент списка
```

```
<LI>Второй элемент списка  
...  
<LI>Последний элемент списка  
</OL>
```



**Рис. 1.3.** Маркированный список с различными символами в качестве маркеров

В отличие от маркированного списка, абзацы входящие в список, нумеруются. Вид нумерации задается параметром `TYPE`, который может принимать следующие значения:

- A — нумерация большими латинскими буквами;
- a — нумерация маленькими латинскими буквами;
- I — нумерация большими римскими цифрами;
- i — нумерация маленькими римскими цифрами;
- 1 — нумерация арабскими цифрами (действует по умолчанию).

По умолчанию нумерация списка задается начиная с первого номера. В случае продолжения нумерации параметр `START` позволяет задать начальный номер списка. Для всех видов нумерации номер должен быть указан арабской цифрой.

Метка `<LI>` может иметь параметры: `TYPE` — изменяет вид номера только для данного элемента списка, `VALUE` — изменяет вид номера данного элемента списка с изменением нумерации последующих абзацев.

Пример записи:

```
<OL>
```

```
<LI> Первый элемент списка
```

```
<LI TYPE=a VALUE=1> Первый элемент второго списка
```

```
<LI TYPE=a> Второй элемент второго списка
```

```
<LI VALUE=2> Второй элемент списка
```

```
<LI> Третий элемент списка
```

```
</OL>
```

В окне браузера такой список будет выглядеть как на рис. 1.4.

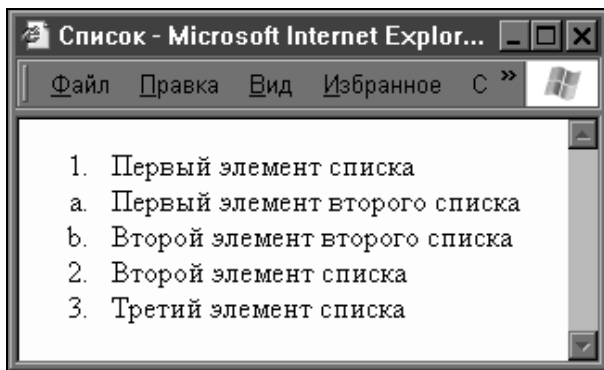


Рис. 1.4. Нумерованный список с двумя видами нумерации

Допускается создание многоуровневого списка, когда один список вкладывается в другой независимо от типа. Список более низкого уровня вложения будет изображаться в браузере смещенным вправо относительно списка предыдущего уровня вложения (листинг 1.3).

#### Листинг 1.3. Создание многоуровневого списка

```
<OL>
```

```
<LI>Холодные закуски
```

```
<UL>
  <LI>Винегреты
    <UL>
      <LI>Винегрет овощной
      <LI>Винегрет с рыбой
    </UL>
  <LI>Салаты
    <UL>
      <LI>Салат зеленый
      <LI>Салат "Весна"
      <LI>Салат витаминный
    </UL>
</UL>
<LI>Супы
  <UL>
    <LI>Щи
      <UL>
        <LI>Щи из свежей капусты
        <LI>Щи из квашеной капусты
        <LI>Щи боярские
      </UL>
    <LI>Борщи
      <UL>
        <LI>Борщ московский
        <LI>Борщ украинский
        <LI>Борщ флотский
      </UL>
    </UL>
</OL>
```

Внешний вид списка в окне браузера представлен на рис. 1.5.

□ Создание списка определений: `<DL>...</DL>`.

Список определений включает определяемый термин, перед которым ставится метка `<DT>` и абзац с его определением, которому предшествует метка `<DD>`.