Министерство образования и науки Украины
Донбасская государственная машиностроительная академия

МЕТОДИЧЕСКИЕ УКАЗАНИЯ

к практическим роботам
по дисциплине
«Анализ, синтез и оптимизация информационных сетей»
(для студентов специальности 123 и 151)

Краматорск, 2018

СОДЕРЖАНИЕ

Практическая Работа 1 3
Язык Гипертекстовой Разметки HTML 4.01 3
Общие Сведения 3
Пример 1. Создание Простейшего HTML-Документа 3
Пример 2. Использование Простейших Элементов Форматирования Текста 6
Пример 3. Форматирование Текста. Сохранение Авторского Форматирования Текста 9
Пример 4. Использование Таблиц В HTML-Документах 11
Пример 5. Списки 15
Пример 6. Включение Изображений В HTML-Документы 17
Пример 7. Использование Таблиц Для Форматирования Документа 19
Порядок Выполнения Работы 21
Контрольные Вопросы 21
Литература 22
Практическая Работа 2 23
Таблицы В HTML-Документах 23
Общие Сведения 23
Основные Теги Таблиц 24
Применение Цвета 25
Примеры Таблиц 26
Порядок Выполнения Работы 33
Содержание Отчета 35
Контрольные Вопросы 35
Литература 35

Практическая работа 1
Язык гипертекстовой разметки HTML 4.01

Цель работы: Изучить основные сведения о языке гипертекстовой разметки HTML 4.01. Освоить простейшие приемы форматирования текстовых документов с помощью текстового редактора

Общие сведения

В последние годы прошлого века появился и приобрел большую популярность новый класс приложений для работы в Интернете – так называемые Web-приложения. Они представляют собой способ организации информации, который, при условии соблюдения определенных правил, позволяет обращаться к ней с любого компьютера. Правила, регламентирующие доступ к информации и передачу ее по сети, носят название HTTP (Hypertext Transfer Protocol – протокол передачи гипертекстовой информации). Поскольку информация в сети между компьютерами зачастую передается в форме документов, Тим Бернерс-Ли на базе существующего языка SGML (Standard Generalized Markup Language – стандартный обобщенный язык разметки) разработал язык HTML (HyperText Markup Language – гипертекстовый язык разметки) для создания электронных документов и отображения их на экране дисплея. Первая программа для работы с HTML-документами – графический броузер Mosaic - была разработана в 1993 году. Фирма Microsoft с опозданием включилась в гонку разработчиков программ-броузеров, однако ее мощь позволила в конце концов выиграть ее и занять лидирующее положение на рынке со своим броузером Internet Explorer.

Хотя способ создания HTML-документов с помощью простого текстового редактора является элементарным и самым трудоемким, его придерживается значительное число профессионалов (так называемые Iron Man HTML). Обусловлено это тем, что, вручную написав HTML-код документа, разработчик уже досконально его знает. Кроме того, некоторые специальные HTML-редакторы (например, FrontPage) привносят в документ много служебной информации. Это никак не отражается на внешнем виде документа, но затрудняет чтение кода документа, засоряя его.

Поэтому осваивать HTML лучше не с помощью специализированных программных пакетов, а исключительно основываясь на ручном создании HTML-кода документа. Научившись вручную писать HTML-код документ, можно без труда освоить любой HTML-редактор.

XHTML – это новый язык разметки, рекомендованный к использованию организацией W3C. W3C также рекомендует производителям броузеров включать в новые версии своих продуктов поддержку XHTML.

Появление XHTML было закономерно, поскольку язык HTML в современном виде не совместим с XML, а именно с XML многие связывают будущее Web. XHTML разработан как язык, который, с одной стороны, обеспечивает совместимость с XML, а с другой стороны, может обрабатываться существующими в настоящее время броузерами. Если авторы Web-страниц рекомендациям W3C и перейдут к использованию XHTML в своих новых разработках, то к тому времени, когда будут реализованы средства поддержки XML, для большинства Web-страниц не возникнут проблемы совместимости.

Слово *расширяемый* в названии языка XHTML означает, что разработчику предоставляются дополнительные возможности по сравнению с HTML.

В лабораторной работе изучаются правила построения несложных документов с использованием версии языка HTML 4.01, просмотр документов организуется с помощью броузера Internet Explorer версии 6.0, подготовка документов производится в среде текстового редактора Aditor Pro 3.10.

Пример 1. Создание простейшего HTML-документа

Структура HTML-документа очень проста. Любой документ HTML представляет собой текстовый файл и содержит элементы, называемые тегами, которые могут занимать одну или несколько строк. Теги ограничены угловыми скобками (<>). Внутри угловых скобок определены атрибуты тега. Корректно составленный документ начинается с тега <html>.

Открывающий тег <head> и закрывающий тег </head> формируют раздел заголовка. В заголовке размещается описание документа в целом. Содержательная часть документа

заключена между парой тегов <body> и </body>. Эта часть документа отображается в окне броузера.

В языке HTML определены элементы, теги и атрибуты. Элемент состоит из открывающего и закрывающего тегов, между которыми расположены данные. Тег идентифицируется именем, которое указывается между открывающей и закрывающей скобками (символоами “<” и “>”); имя определяет, как должна отображаться информация, содержащаяся в составе элемента. Атрибуты уточняют действие тега.

При использовании тегов необходимо учитывать следующие правила.

- Все стандартные теги начинаются с символа “<” и заканчиваются символом “>” (например, <html>).
- Между символами, составляющими имя тега, пробелы недопустимы. Также нежелательно включать пробелы между угловыми скобками и именем тега.
- Каждому открывающему тегу должен соответствовать закрывающий. Закрывающий тег отличается от открывающего тем, что перед именем тега стоит косая черта (</html>).
- При создании HTML-документа следует использовать в именах тегов только символы нижнего регистра (для совместимости с новой версией стандарта XHTML). В лабораторной работе это правило не выдерживается – в ней речь идет о стандарте HTML 4.01.
- Для разделения атрибутов, а также для того, чтобы отделить первый атрибут от имени тега, используются пробелы. Наличие пробелов в составе значения атрибута может привести к непредсказуемым последствиямю.
- Следует придерживаться требуемой структуры HTML-документа и задавать основные теги в правильном порядке.

На рисунке 1 показан внешний вид простейшего документа в окне броузера Internet Explorer.

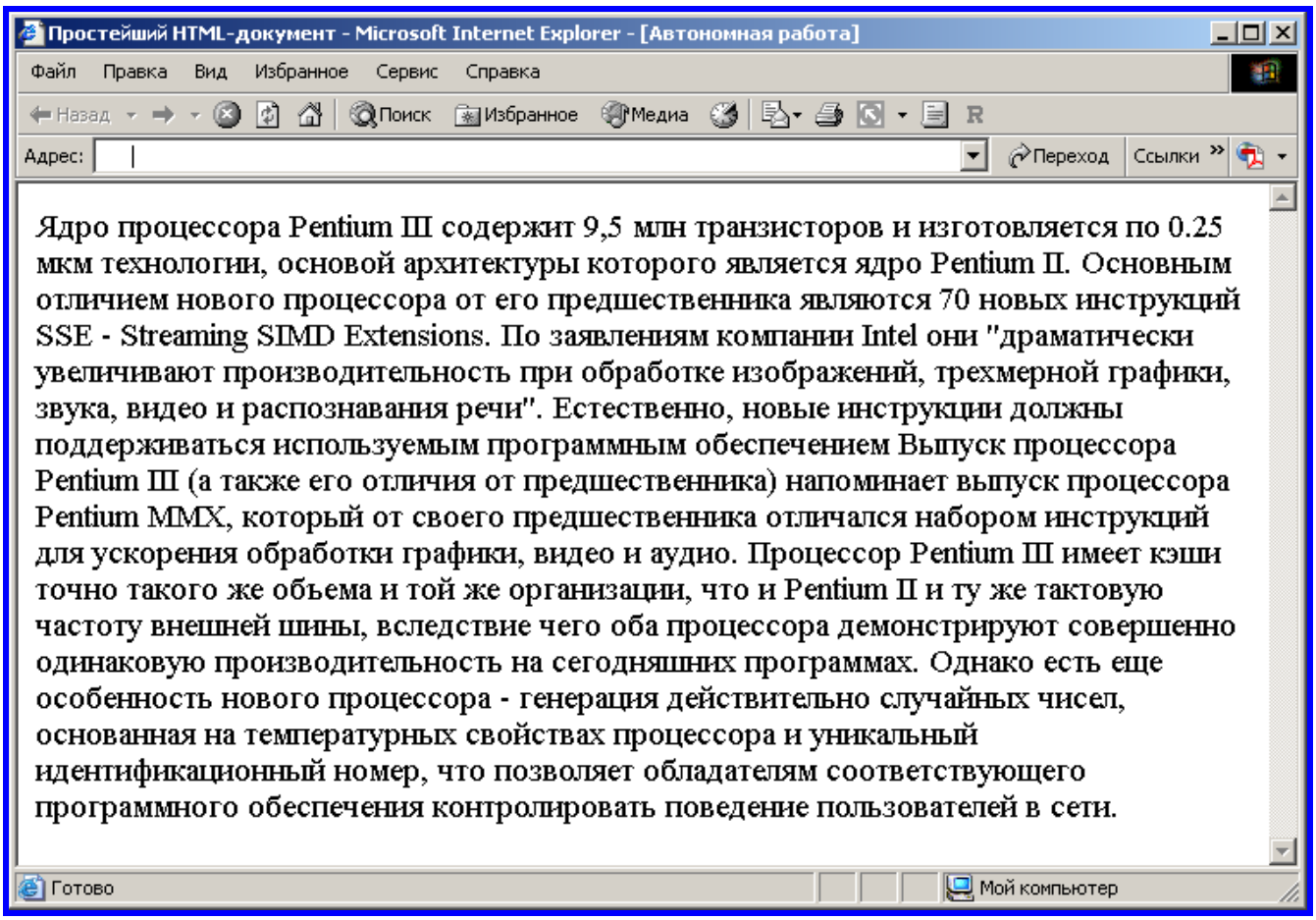


Рисунок 1- Внешний вид простейшего документа

Под простейшим документом имеется в виду документ, в котором присутствует только один неотформатированный текст. "Неотформатированный" значит, что в нем отсутствует, к примеру, выделения жирным шрифтом или курсивом, разделение текста на абзацы и тому подобное. Для создания такого документа достаточно иметь файл с

текстом. Такой файл открывается в редакторе Aditor, а после добавления немного HTML-разметки он сохраняеся качестве HTML-файла. Для этого в меню «Файл» редактора после выбора пункта "Сохранить как..." открывается диалоговое окно, в котором нужно напечатать имя файла с расширением .html. Если текстового файла не имеется, то его можно создать в том же самом редакторе.

При вводе текста можно не соблюдать длину строки, не использовать повторяющихся пробелов и табуляций для лучшего вида текста, так как все это игнорируется браузером при отображении документа. Для этих целей предназначены специальные HTML-элементы. При отображении текста многократные пробелы и табуляции заменяются одним пробелом. Авторские переходы на следующую строку игнорируются и строка текста имеет максимально возможную длину, исходя из размеров окна браузера.

На рисунке 2 приведен код простейшего HTML-документа.

```
<HTML>
<HEAD>
<TITLE> Простейший HTML-документ </TITLE>
</HEAD>
<BODY>

Ядро процессора Pentium III содержит 9,5 млн транзисторов и изготовляется
по 0.25 мкм технологии, основой архитектуры которого является ядро Pentium II.
Основным отличием нового процессора от его предшественника являются 70 новых
 инструкций SSE - Streaming SIMD Extensions. По заявлениям компании Intel
 они "драматически увеличивают производительность при обработке изображений,
трехмерной графики, звука, видео и распознавания речи". Естественно, новые
инструкции должны поддерживаться используемым программным обеспечением
Выпуск процессора Pentium III  (а также его отличия от предшественника)
напоминает выпуск процессора Pentium MMX, который от своего предшественника
отличался набором инструкций для ускорения обработки графики, видео и аудио.
 Процессор Pentium III имеет кэши точно такого же объема и той же организации,
 что и Pentium II и ту же тактовую частоту внешней шины, вследствие чего
оба процессора демонстрируют совершенно одинаковую производительность на
сегодняшних программах.
Однако есть еще особенность нового процессора - генерация действительно
случайных чисел, основанная на температурных свойствах процессора и уникальный
 идентификационный номер, что позволяет обладателям соответствующего
программного  обеспечения контролировать поведение пользователей в сети.

</BODY>
</HTML>
```

Рисунок 2 – Код простейшего документа

На рисунке 3 показан разбор кода. Поскольку пример является очень простым, то комментарии к нему не рассматриваются.

При составлении текста желательно соблюдение следующих принципов, существенно улучшающих его восприятие.

- В каждом абзаце выражается одна мысль.
- Начинать материал следует с краткого изложения его сути.
- Не следует использовать большие непрерывные текстовые блоки.
- Текст нужно разбивать на разделы и подразделы. Заголовки текста должны быть простыми и выразительными. Для упорядочения данных полезно использовать маркированные списки. Важная информация должна выделяться цветом или полужирным шрифтом.
- Текст должен быть кратким. По возможности следует избегать необходимости в его прокрутке.
- Большой объем текста следует разбивать на части, связанные между собой гипертекстовыми ссылками. Части должны быть организованы по пирамидальной схеме. Вначале должно быть расположено краткое изложение материала. После этого следует уточнение деталей.

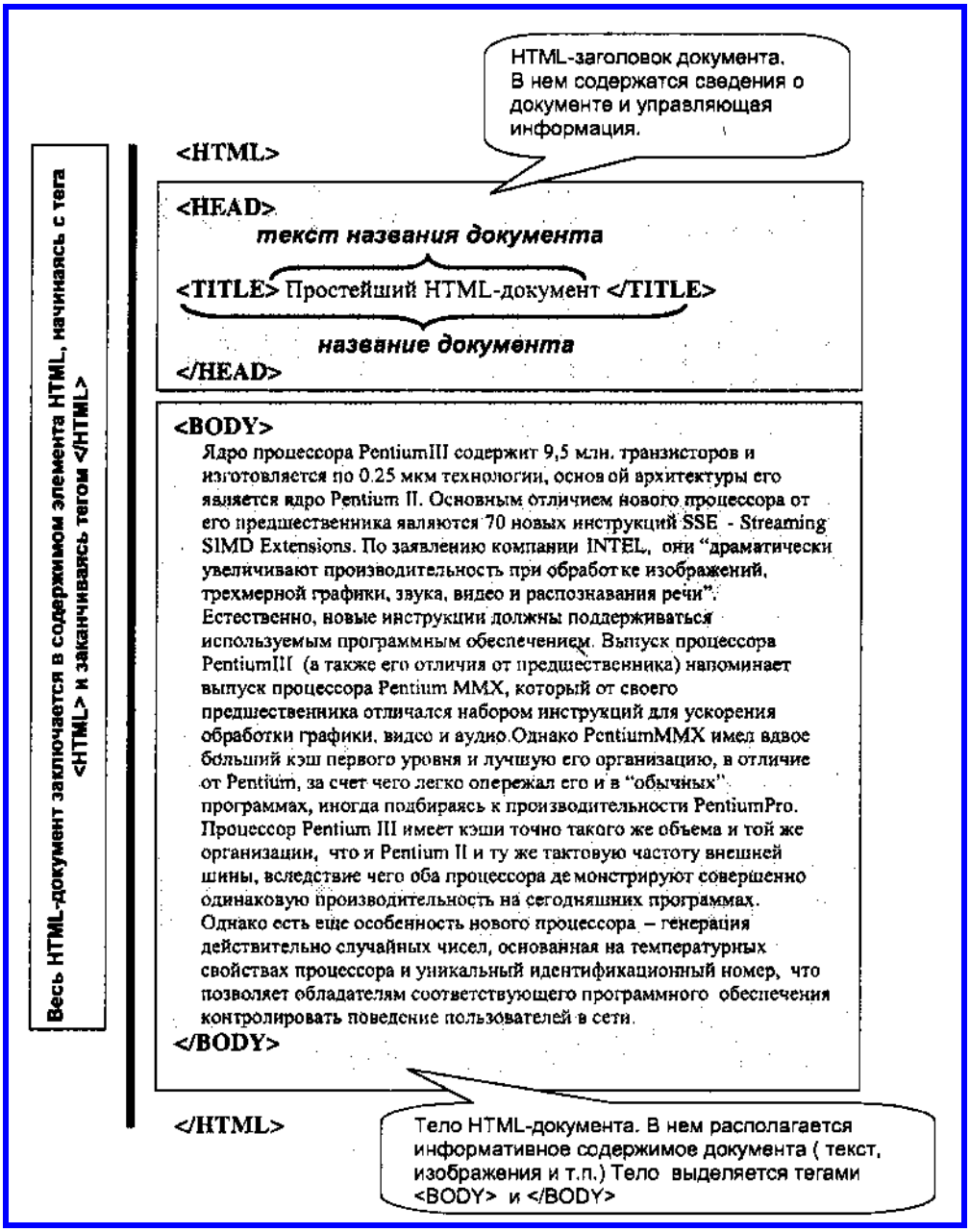


Рисунок 3 – Разбор кода простейшего документа из примера 1

Пример 2. Использование простейших элементов форматирования текста

В следующем примере показаны элементарные возможности форматирования текста, предоставляемые языком HTML. Рассмотрим применение некоторые из них в тексте, использованном в первом примере. Получившуюся разницу можно наблюдать, сравнив рисунок 1 с рисунком 4.

Для того, чтобы разбить текст на абзацы, используется элемент <P>. При этом абзацем считается все, что заключено между начальным и конечным тегами этого элемента. Текст абзаца в окне броузера всегда начинается с новой строки и выделяется сверху и снизу пустыми строками.

Чтобы изменить вид шрифта, которым выводится текст, применяется элемент FONT (в настоящее время он считается устаревшим не рекомендован к применению). С помощью атрибутов color, size и face можно указать цвет шрифта, его размер и вид. Попробуйте поменять значения этих атрибутов и посмотрите, что из этого получится.

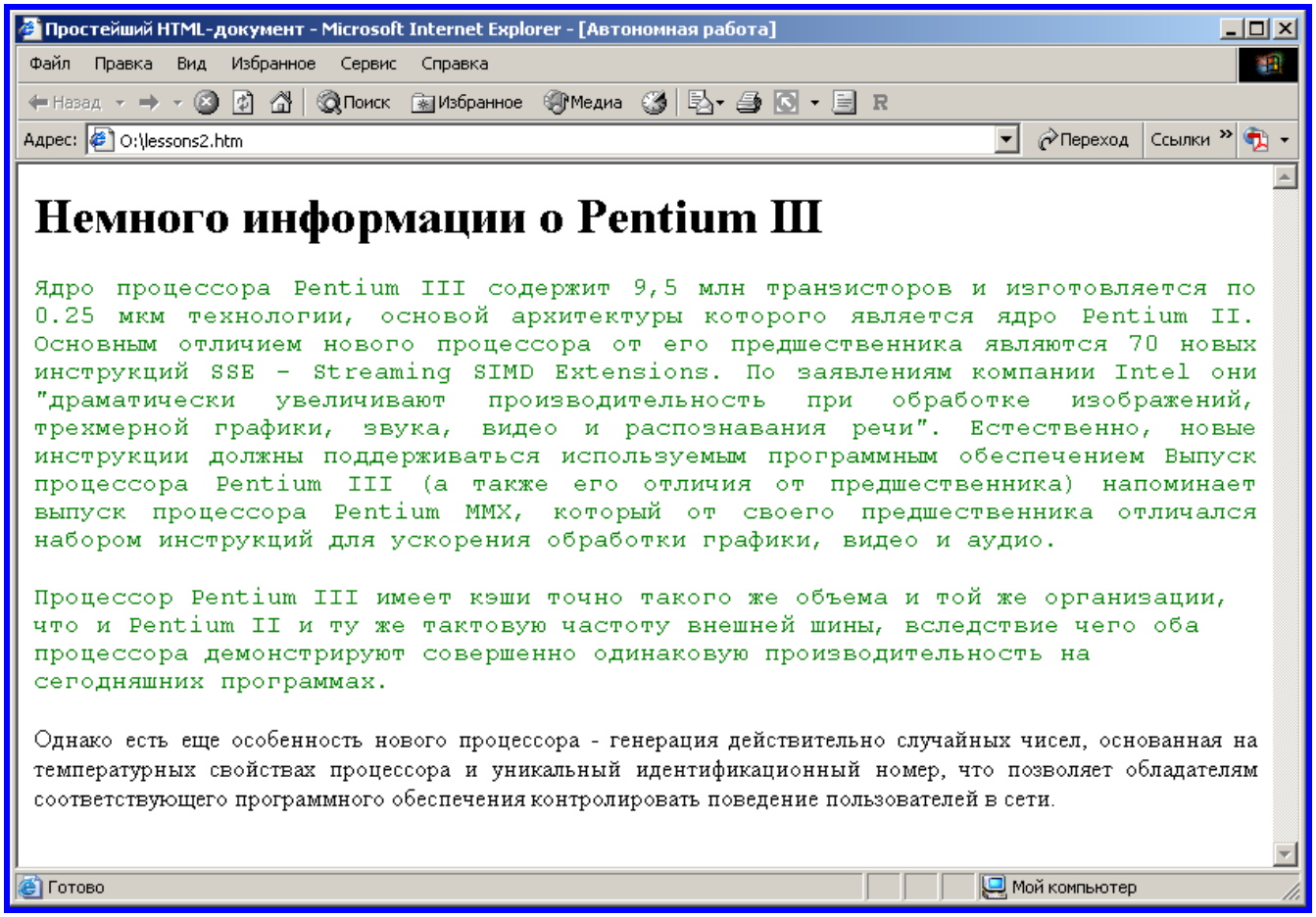


Рисунок 4 – Внешний вид документа с простым форматированием текста

В HTML, так же как и в текстовом процессоре Word, имеется возможность задания заголовков. Особенность заголовков заключается в том, что строка, на которой расположен заголовок, содержит только этот заголовок и ничего более. Кроме этого, все заголовки отображаются жирным шрифтом и выделяются сверху и снизу пустыми строками. Задание заголовков производится с помощью элементов H1, H2, H3, H4, H5 и H6. Отличаются они только размером: самым большим заголовком является текст, заключенный между начальным и конечным тегами элемента H1, самым маленьким — H6. В приведенном примере используется самый большой заголовок H1. Чтобы почувствовать разницу между ним и остальными заголовками, попробуйте в этом же документе заменить H1 на H2, H3 ... H6. Не забывайте только, что заменять надо оба тега — начальный и конечный.

Элемент font определяет вид, размер и цвет шрифта, помещаемого на страницу. Его синтаксис:

<font *атрибуты***>***содержимое*****

Для управления размером шрифта используется атрибут **size**=”*данные*”. Размер шрифта указывается в абсолютных единицах (от 1 до 7) или в виде относительного приращения (например, +3). Если базовый шрифт не указан, по умолчанию используется значение 3.

Цвет шрифта (атрибут **color**=”*data*”) задается как шестнадцатиричное RGB-значение (*#RRGGBB*), или с помощью специальных имен.

Атрибут **face**=”*data*” задает шрифт, который должен использоваться при отображении документа (если этот шрифт доступен). В качестве значения данного свойства может быть задано несколько шрифтов, разделенных запятыми. Наиболее предпочтительные шрифты должны быть указаны первыми в списке.

На рисунке 5 представлен код документа для примера 2.

```
<HTML>
<HEAD>
<TITLE> Простейший HTML-документ </TITLE>
</HEAD>
<BODY>
<H1>Немного информации о Pentium III </H1>
<FONT COLOR=GREEN FACE="COURIER NEW">
<P Align=Justify>
Ядро процессора Pentium III содержит 9,5 млн транзисторов и изготовляется
по 0.25 мкм технологии, основой архитектуры которого является ядро Pentium II.
Основным отличием нового процессора от его предшественника являются 70 новых
 инструкций SSE - Streaming SIMD Extensions. По заявлениям компании Intel
 они "драматически увеличивают производительность при обработке изображений,
трехмерной графики, звука, видео и распознавания речи". Естественно, новые
инструкции должны поддерживаться используемым программным обеспечением
Выпуск процессора Pentium III  (а также его отличия от предшественника)
напоминает выпуск процессора Pentium MMX, который от своего предшественника
отличался набором инструкций для ускорения обработки графики, видео и аудио.
</P>

Процессор Pentium III имеет кэши точно такого же объема и той же организации,
 что и Pentium II и ту же тактовую частоту внешней шины, вследствие чего
оба процессора демонстрируют совершенно одинаковую производительность на
сегодняшних программах.
</FONT>
<P Align=Justify>
Однако есть еще особенность нового процессора - генерация действительно
случайных чисел, основанная на температурных свойствах процессора и уникальный
 идентификационный номер, что позволяет обладателям соответствующего
программного  обеспечения контролировать поведение пользователей в сети.
</P>
</BODY>
</HTML>
```

Рисунок 5 - Внешний вид документа с простым форматированием текста

Разбор кода показан на рисунке 6.

Большое внимание на восприятие содержимого HTML-документа оказывает цвет. Если дизайнер использовал для отображения фона и текста (т.е. переднего плана) цвета, не контрастирующие друг с другом, читать такой текст трудно; особенно большие проблемы возникнут у пользователей с ухудшенным зрением. Некоторые сочетания цвета фона и переднего плана сделают текст нечитаемым для пользователей, плохо различающих цвета. Для того, чтобы сделать эту проблему менее острой, необходимо выбирать для фона и переднего плана цвета, контрастирующие между собой. По возможности, для проверки следует привлекать пользователей, плохо различающих цвета.

По умолчанию фон документов HTML выглядит белым, поэтому белые буквы на таком фоне будут невидимы в Internet Explorer. Средствами атрибута bgcolor тега body можно изменить цвет фона документа HTML, например, сделать его желтым:

```
<body bgcolor="#FFFF00">
```

Здесь цвет определен в формате RGB, первые две шестнадцатеричные цифры определяют красную компоненту, далее следует значение зеленой и голубой компонент соответственно.

По умолчанию параграф, ограниченный тегом <P>, выравнивается по левому краю окна броузера. Выравнивание можно изменить, применив атрибут align. Допустимыми значениями для align являются left, right, center и justify. Если задано значение justify, то содержимое абзаца, состоящее из нескольких строк, распределяется по всей ширине окна броузера.

На рисунке 6 показан разбор кода для примера 3.

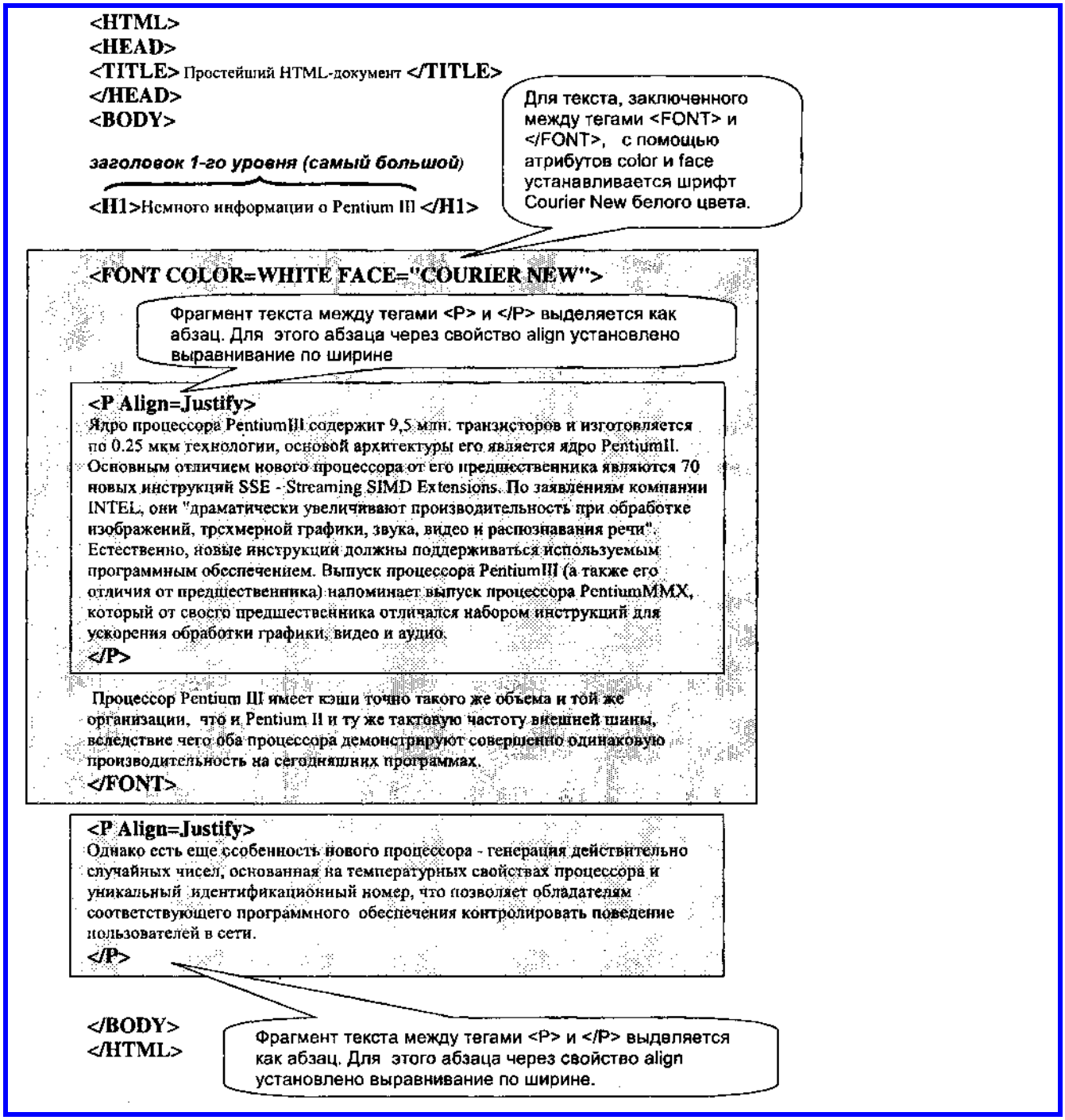


Рисунок 6 – Разбор кода простейшего документа из примера 2

Пример 3. Форматирование текста. Сохранение авторского форматирования текста

В обычном случае текст при отображении в окне браузера теряет авторское форматирование и отображается сплошным потоком. Однако форматирование можно сохранить, если заключить текст между начальным и конечным тегами элемента PRE (между <PRE> и </PRE>). Такой текст воспринимается браузером как отформатированный и выводится в том виде, в котором он задан в HTML-коде документа. С помощью элемента PRE сохраняется авторская расстановка слов, многочисленные пробелы, табуляции. Наиболее часто таким образом выводятся стихотворения, для которых расстановка слов имеет особое значение. Пример подобного форматирования показан на рисунке 7.

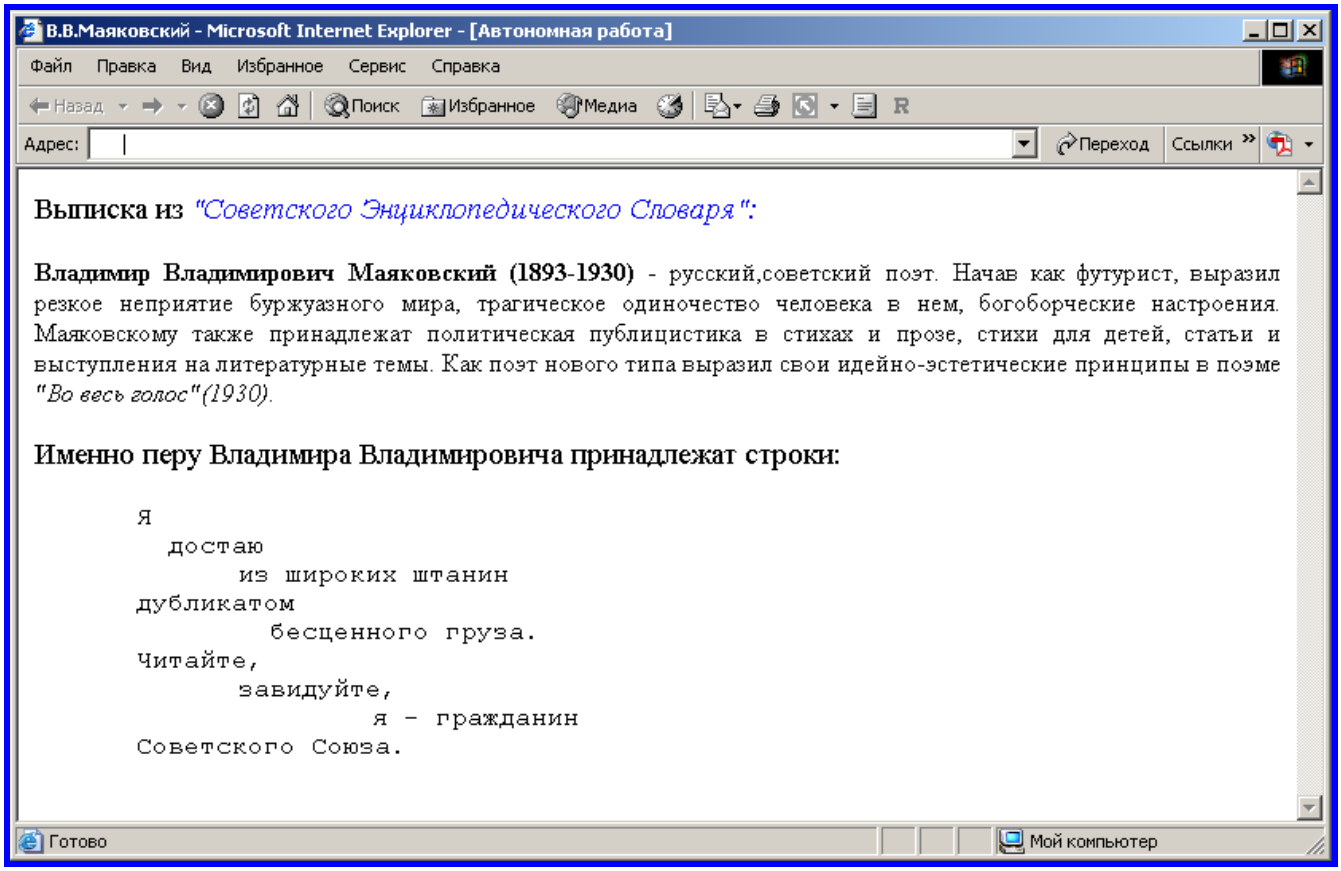


Рисунок 7 - Пример документа, с сохраненным авторским форматированием текста

Для выделения текста жирным шрифтом или курсивом используются элементы B и I соответственно. Они выделяют текст, расположенный между своим начальным и конечным тегами. Причем к одному и тому же фрагменту текста может быть одновременно применено и выделение курсивом и выделение жирным шрифтом.

Код документа

```
<HTML>
<HEAD>
<TITLE>В.В.Маяковский </TITLE>
</HEAD>
<BODY>
Выписка из <font color=blue>
<I>"Советского Энциклопедического Словаря":</I>
</Font>
<Font size=-1>
<P align=justify>
<B>Владимир Владимирович Маяковский (1893-1930)</B> -
русский,советский поэт.
Начав как футурист, выразил резкое неприятие буржуазного мира, трагическое
одиночество человека в нем, богоборческие настроения. Маяковскому также
принадлежат политическая публицистика в  стихах и прозе, стихи для детей,
статьи и выступления на литературные темы. Как поэт нового типа выразил свои
идейно-эстетические принципы в поэме <EM>"Во весь голос"(1930)</EM>.
</P>
</Font>
Именно перу Владимира Владимировича принадлежат строки:

<Pre>
      Я
        достаю
            из широких штанин
      дубликатом
              бесценного груза.
```

```
        Читайте,
              завидуйте,
                    я - гражданин
        Советского Союза.
</Pre>
</BODY>
</HTML>
```

На рисунке 8 выполнен разбор кода.

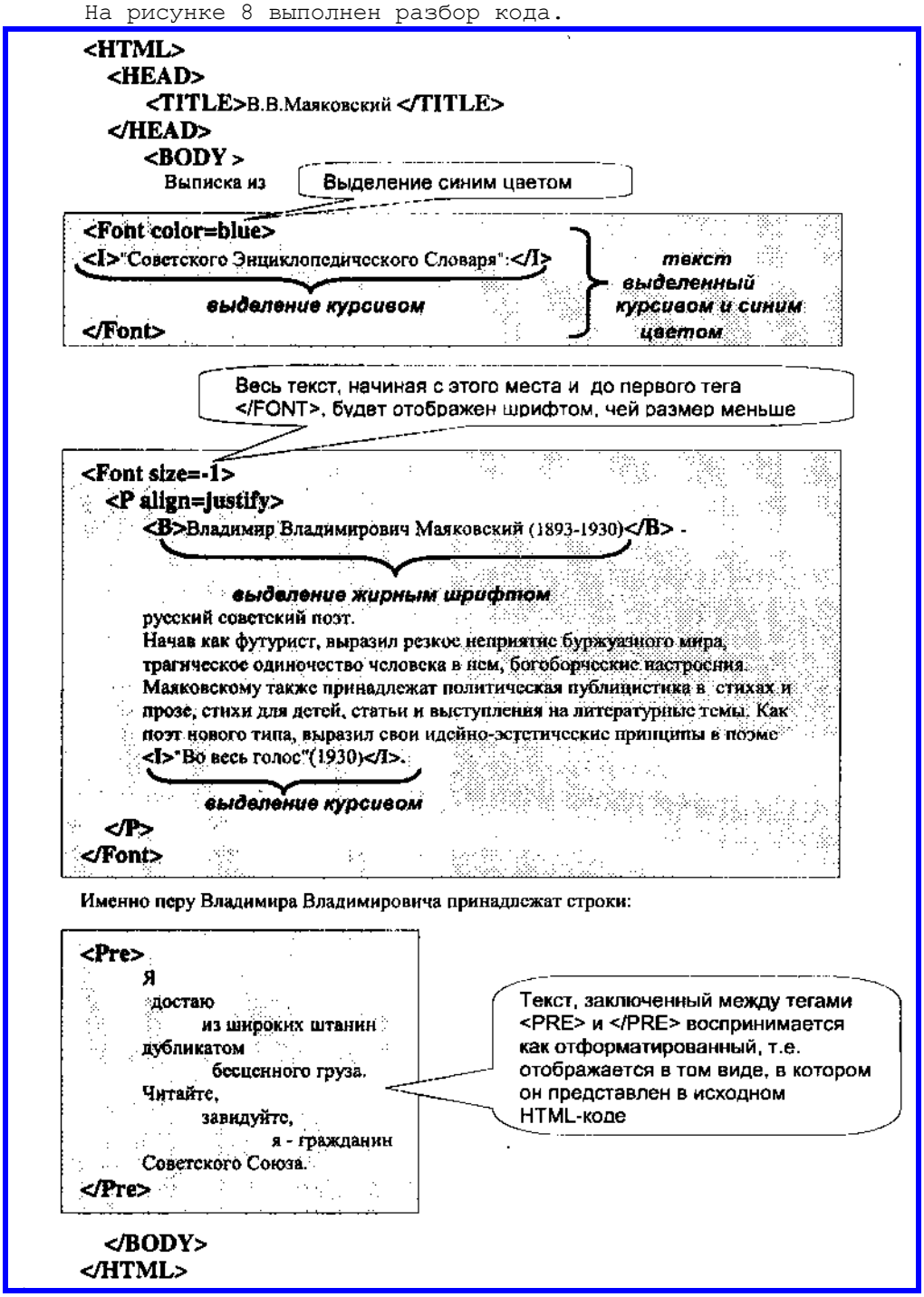
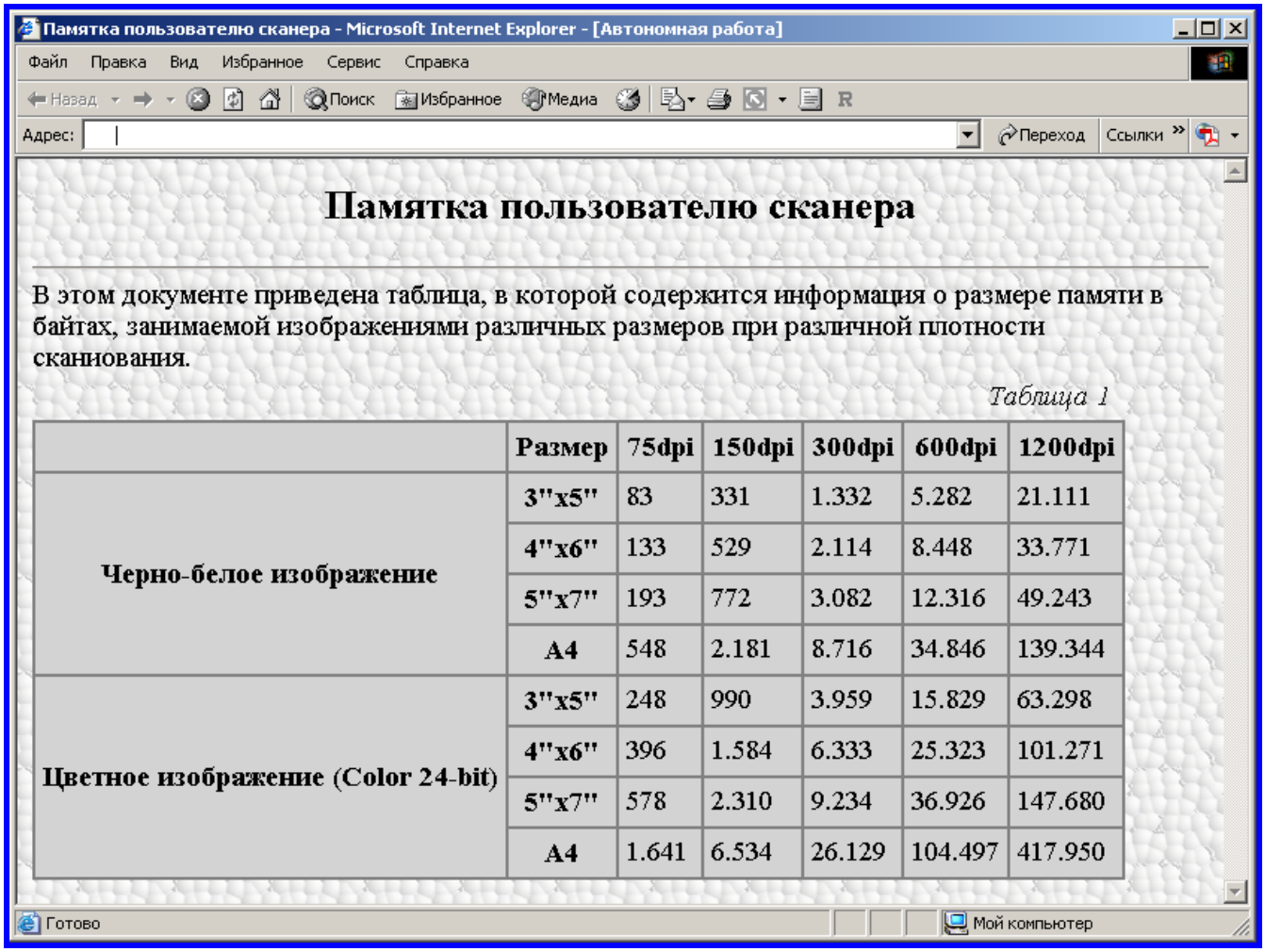


Рисунок 8 – Разбор кода простейшего документа из примера 3

Пример 4. Использование таблиц в HTML-документах

Таблицы в HTML-документах задаются с помощью элемента TABLE. Один элемент TABLE определяет одну таблицу, задаваемую между его начальным и конечным тегами. На рисунке 9 показан пример отображения таблицы в окне броузера.



Памятка пользователю сканера

В этом документе приведена таблица, в которой содержится информация о размере памяти в байтах, занимаемой изображениями различных размеров при различной плотности сканирования.

Таблица 1

	Размер	75dpi	150dpi	300dpi	600dpi	1200dpi
Черно-белое изображение	**3"x5"**	83	331	1.332	5.282	21.111
	4"x6"	133	529	2.114	8.448	33.771
	5"x7"	193	772	3.082	12.316	49.243
	A4	548	2.181	8.716	34.846	139.344
Цветное изображение (Color 24-bit)	**3"x5"**	248	990	3.959	15.829	63.298
	4"x6"	396	1.584	6.333	25.323	101.271
	5"x7"	578	2.310	9.234	36.926	147.680
	A4	1.641	6.534	26.129	104.497	417.950

Рисунок 9 - Внешний вид документа с таблицей

Каждая строка таблицы задается элементом TR. Указание конечного тега этого элемента является необязательным: строкой считается все, что расположено после начального тега и до следующего элемента TR, или до конечного тега элемента TABLE.

Ячейки задаются с помощью элементов TH и TD. TH — ячейки заголовка, TD — ячейки данных. Различие между ними заключается в том, что информация в ячейках заголовка выводится жирным шрифтом и выровненной по центру. Содержимое ячеек данных отображается обычным шрифтом и выровненным по левому краю.

Роль HTML-таблиц в формировании документа очень велика, и она не ограничивается просто рисованием табличек. Основное использование HTML-таблиц заключается в форматировании всего документа (либо значительных его частей). При этом используются таблицы с невидимыми границами.

По умолчанию информация в HTML-документе выводится последовательно непрерывным потоком. И чтобы отобразить ее, например, в виде нескольких колонок (как в газете) - и используются таблицы с невидимыми границами, в ячейках которых помещаются отдельные фрагменты документа.

Достаточно часто таблицы используются для отображения больших анимированных изображений. Поскольку при анимации обычно изменяется не все изображение целиком, а только его отдельные части, гораздо выгоднее разбить изображение на несколько прямоугольных частей, анимируя некоторые из них, чем анимировать все изображение полностью (большая экономия в размере, а значит, и в скорости загрузки такого изображения). В этом случае отдельные части изображения как раз и размещаются в ячейках таблицы с невидимыми границами и нулевыми внутриячеечными отступами.

Код документа

```
<HTML>
<HEAD>
<TITLE>Памятка пользователю сканера</TITLE>
</HEAD>
```

```
<BODY background="font/165.gif">
<H2 align=center>Памятка пользователю сканера</H2>
<HR>
В этом документе приведена таблица, в которой содержится информация о размере памяти
в байтах, занимаемой изображениями различных размеров при различной плотности
сканиования.

<TABLE frame=border  Cellpadding=5 bgcolor=lightgrey Cellspacing=0
bordercolor=gray>
<CAPTION align=right><EM> Таблица 1 </EM></CAPTION>
<TR>
<TH>
<TH>Размер
<TH>75dpi
<TH>150dpi
<TH>300dpi
<TH>600dpi
<TH>1200dpi
<TR>
<TH rowspan=4>Черно-белое изображение
<TH>3"x5"
<TD>83
<TD>331
<TD>1.332
<TD>5.282
<TD>21.111
<TR>
<TH>4"x6"
<TD>133
<TD>529
<TD>2.114
<TD>8.448
<TD>33.771

<TR>
<TH>5"x7"
<TD>193
<TD>772
<TD>3.082
<TD>12.316
<TD>49.243

<TR>
<TH>A4
<TD>548
<TD>2.181
<TD>8.716
<TD>34.846
<TD>139.344
<TR>
<TH rowspan=4>Цветное изображение (Color 24-bit)
<TH>3"x5"
<TD>248
<TD>990
<TD>3.959
<TD>15.829
<TD>63.298
<TR>
<TH>4"x6"
<TD>396
<TD>1.584
<TD>6.333
<TD>25.323
<TD>101.271
```

```
<TR>
<TH>5"x7"
<TD>578
<TD>2.310
<TD>9.234
<TD>36.926
<TD>147.680

<TR>
<TH>A4
<TD>1.641
<TD>6.534
<TD>26.129
<TD>104.497
<TD>417.950

</TABLE>
</BODY>
</HTML>
```

На рисунке 11 выполнен разбор кода.

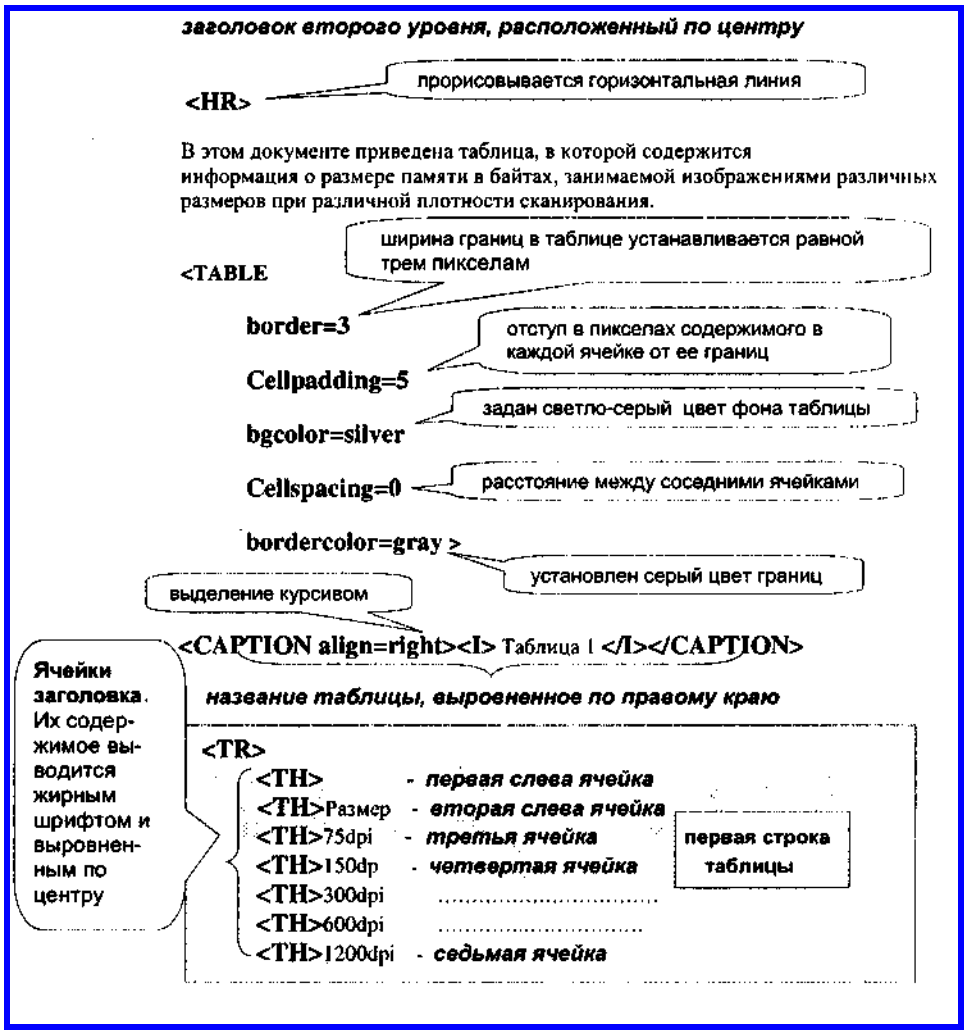
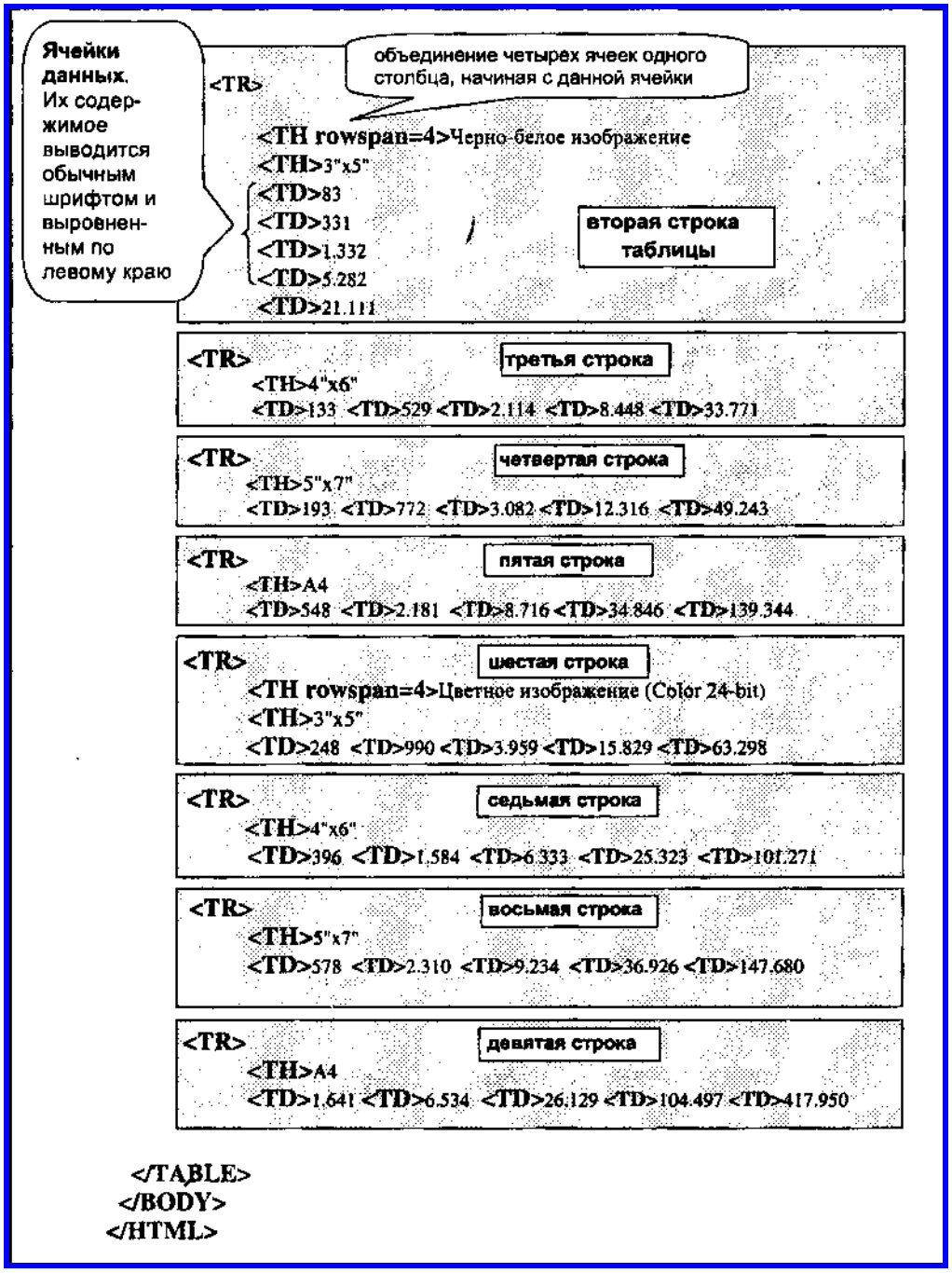


Рисунок 11 – Разбор кода простейшего документа из примера 3



Продолжение рисунка 11

Пример 5. Списки

Язык HTML располагает возможностями по отображению списков. При этом выделяют упорядоченные и неупорядоченные списки. Упорядоченные списки предполагают использование какой-либо нумерации элементов списка (арабскими цифрами, римскими цифрами, латинским буквами и т.п.). Неупорядоченные списки никакой нумерации по своему определению не имеют, а выделяют каждый свой элемент специальным значком (пулькой): кружочком, квадратиком и т.п. На рисунке 12 показано окно броузера с текстом, содержащим список.

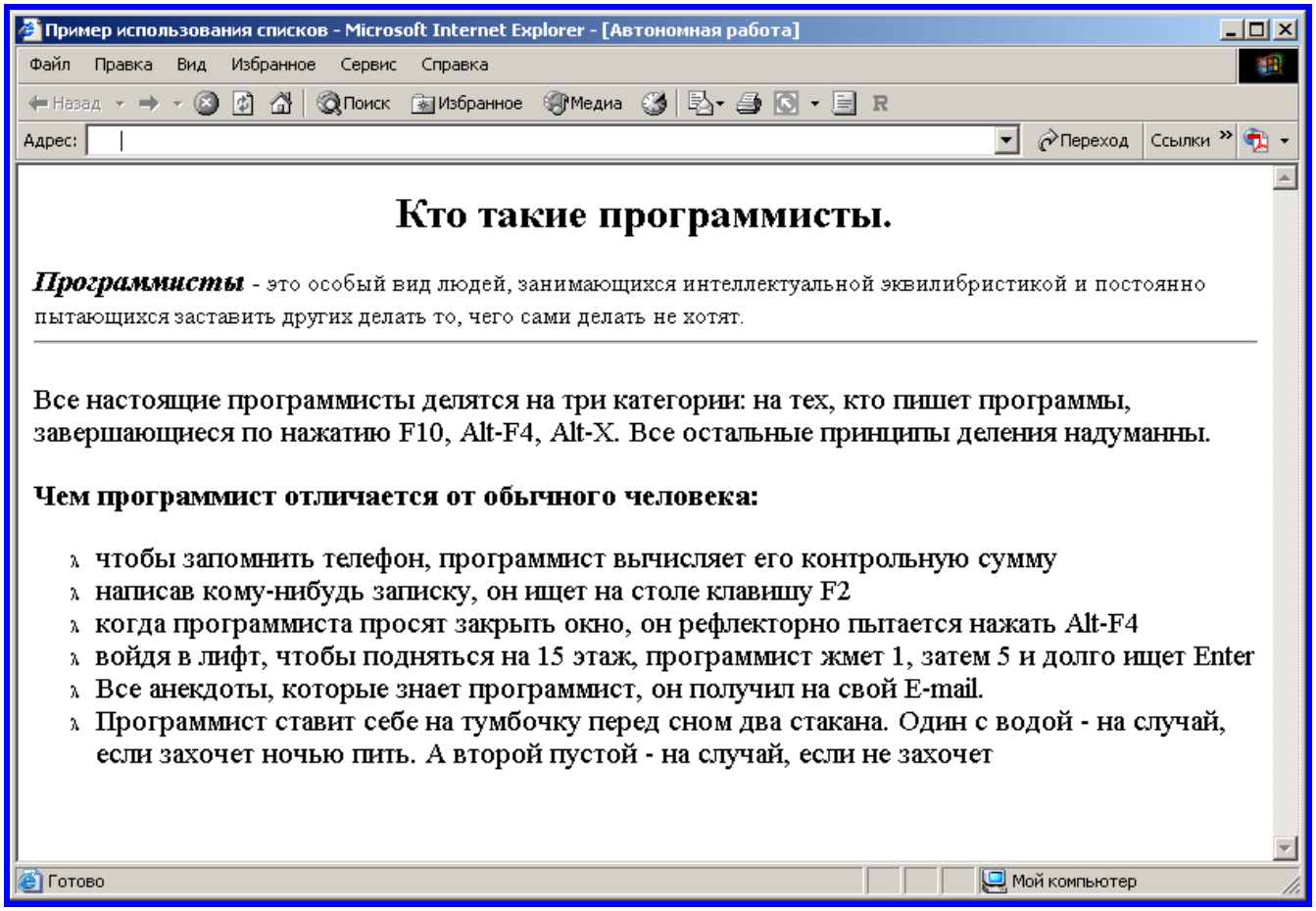


Рисунок 12 - Внешний вид документа со списком

В документе, представленном в данном примере, продемонстрировано использование неупорядоченного списка. Списки такого типа задаются элементом UL. Каждый отдельный элемент списка задается элементом LL При этом задание конечного тега этого элемента является необязательным — элементом списка считается все, что расположено после одного начального тега (<LI>) до следующего начального тега элемента LI или до конечного тега элемента UL. В качестве маркеров (пулек) могут быть использованы заштрихованные кружочки, незаштрихованные кружочки или заштрихованные квадратики. По умолчанию используются заштрихованные кружочки (это видно на примере).

Неупорядоченные списки задаются HTML-элементом OL. Стоит в приведенном примере заменить <UL> и </UL> на <OL> и </OL>, как элементы списка будут отображены пронумерованными арабскими цифрами.

Кроме упорядоченных и неупорядоченных списков, в HTML определены еще такие списки, как списки определений. Информация в таких списках представляется в виде словарных статей: термин — определение термина.

Код документа

```
<HTML>
<HEAD>
<TITLE>Пример использования списков</TITLE>
</HEAD>
<BODY background="FONT\18.gif">
<H2 align=center>Кто такие программисты.</H2>
<P>
<STRONG><EM> Программисты </EM></STRONG>
<FONT size=-1>
- это особый вид людей, занимающихся интеллектуальной эквилибристикой и  постоянно пытающихся заставить других  делать то, чего сами делать не хотят.
</FONT>
<HR>
<P>
```

Все настоящие программисты делятся на три категории: на тех, кто пишет программы, завершающиеся по нажатию F10, Alt-F4, Alt-X. Все остальные принципы деления надуманны.

```
<H4>Чем программист отличается от обычного человека:</H4>
<UL align=center>
 <LI>чтобы запомнить телефон, программист вычисляет его контрольную  сумму
  <LI>написав кому-нибудь записку, он ищет на столе  клавишу F2
  <LI>когда программиста просят закрыть окно, он рефлекторно пытается нажать Alt-F4
  <LI>войдя в лифт, чтобы подняться на 15 этаж, программист жмет 1, затем 5 и долго
ищет Enter
  <LI>Все анекдоты, которые знает программист, он получил на свой E-mail.
<LI>
Программист ставит себе на тумбочку перед сном два стакана.
Один с водой - на случай, если захочет ночью пить. А второй
пустой - на случай, если не захочет</UL>
</BODY>
</HTML>
```

Пример 6. Включение изображений в HTML-документы

На сегодняшний момент времени абсолютное большинство HTML-документов содержат в себе изображения. Да оно и понятно: с использованием изображений резко улучшается внешний вид документа и возрастает наглядность представленной в документе информации.

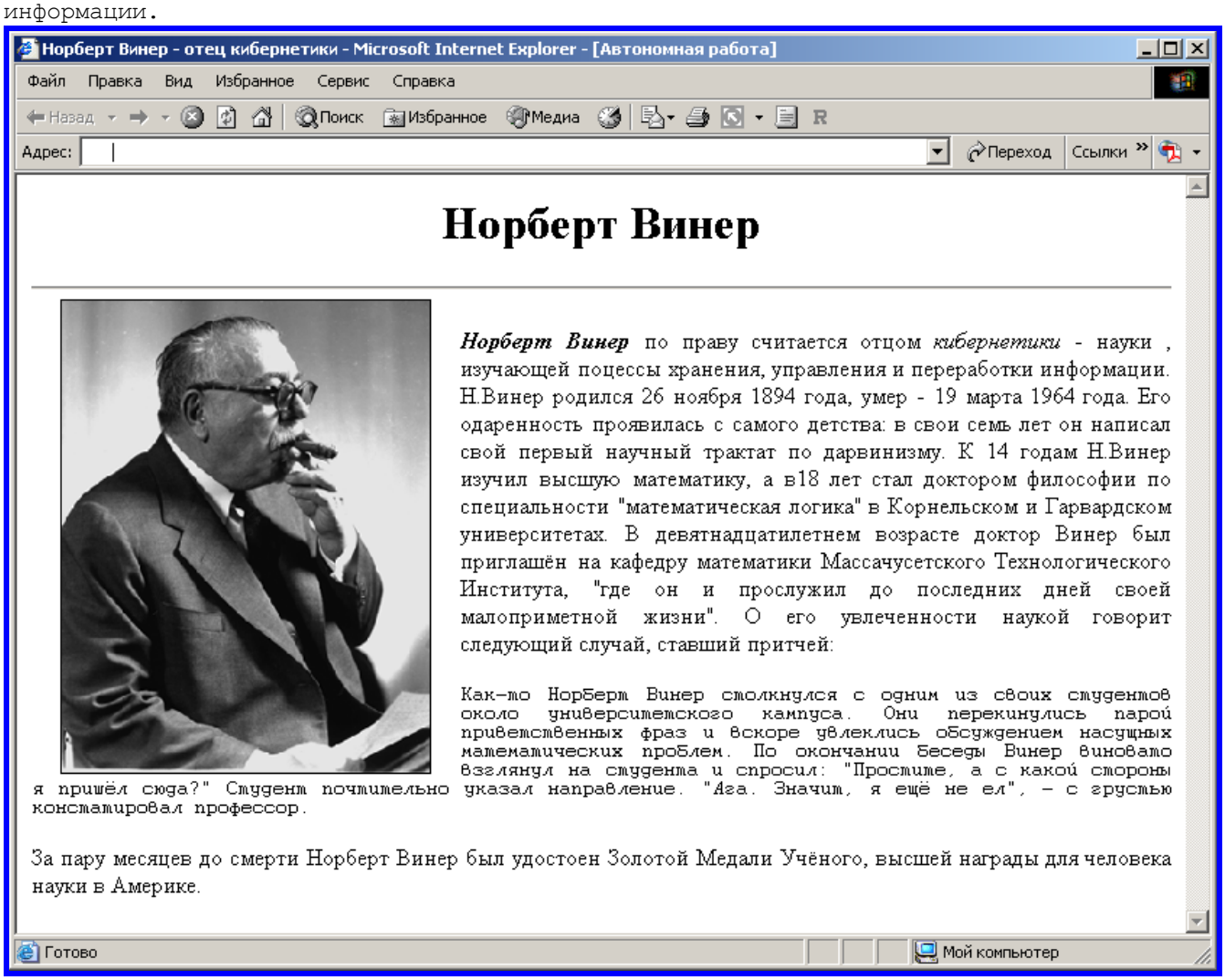


Рисунок 13 - Документ с изображением

Включение изображений в HTML-документ производится с помощью элемента IMG, в качестве значения атрибута src которого указывается месторасположение графического

файла, в котором содержится изображение.

С изображениями связана одна небольшая деталь: для того, чтобы изображение обтекалось текстом, необходимо прикрепить его к одному из краев документа — к правому или к левому, Осуществляется это путем задания значения right или left для атрибута align элемента IMG. В противном случае изображение будет восприниматься браузером просто как очень большая буква, содержащаяся в строке текста. Наглядно это можно увидеть, если стереть запись align=left в коде приведенного здесь примера,

То же самое, кстати говоря, относится и к таблицам: чтобы они обтекались текстом, необходимо прикрепить их к одному из краев документа.

Элемент IMG поддерживает многие известные форматы графических файлов: GIF, JPEG, BMP и др. Что касается каких-либо экзотических форматов, то такие изображения тоже можно внедрять в HTML-документ, но уже с помощью универсального элемента OBJECT.

Код документа

```
<HTML>
<HEAD>
<TITLE>Норберт Винер - отец кибернетики</TITLE>
</HEAD>
<BODY>
<H1 align=center>Норберт Винер</H1>
<HR>
<IMG src="IMAGES\wiener2.gif" border=1 align=left hspace=20>
<P align=justify>
<STRONG><EM>Норберт Винер</EM></STRONG> по праву считается отцом
<EM>кибернетики</EM> - науки , изучающей поцессы хранения, управления и переработки
информации. Н.Винер родился 26 ноября 1894 года, умер - 19 марта 1964 года. Его
одаренность проявилась с самого детства: в свои семь лет он написал
свой первый научный трактат по дарвинизму. К 14 годам Н.Винер изучил
высшую математику, а в 18 лет стал доктором философии по специальности
"математическая логика" в Корнельском и Гарвардском университетах.
В девятнадцатилетнем возрасте доктор Винер был приглашён на кафедру
математики Массачусетского Технологического Института, "где он и прослужил
до последних дней своей малоприметной жизни". О его увлеченности наукой
говорит следующий случай, ставший притчей:
<BR><BR>
<FONT size=-1 face="courier">
Как-то Норберт Винер столкнулся с одним из своих студентов около
университетского кампуса. Они перекинулись парой приветственных
фраз и вскоре увлеклись обсуждением насущных математических проблем.
По окончании беседы Винер виновато взглянул на студента и спросил:
"Простите, а с какой стороны я пришёл сюда?" Студент почтительно
указал направление. "Ага. Значит, я ещё не ел", - с грустью констатировал
профессор.
</FONT>
<BR><BR>
За пару месяцев до смерти Норберт Винер был удостоен Золотой Медали
Учёного, высшей награды для человека науки в Америке.</P>
</BODY>
</HTML>
```

На рисунке 14 выполнен разбор кода документа, содержащего изображение.

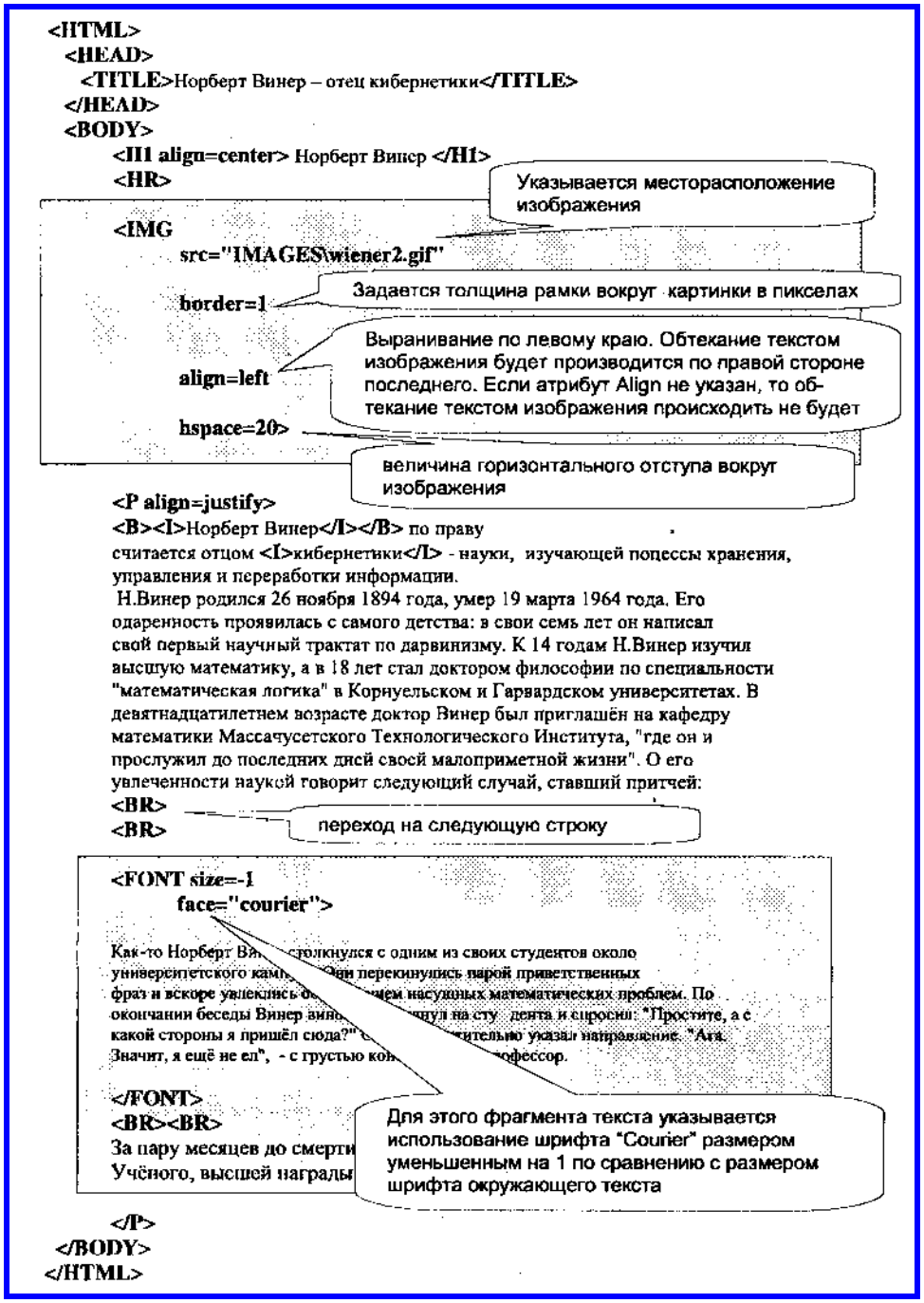


Рисунок 14 - Документ с изображением

Пример 7. Использование таблиц для форматирования документа

Как уже упоминалось немного раньше, таблицы являются и используются в качестве мощнейшего средства по форматированию HTML-документа, или, иначе говоря, по его разметке, причем для этого используется таблицы с невидимыми границами. На рисунке 19 выполнен разбор кода HTML-документа, содержащего таблицу с невидимыми границами.

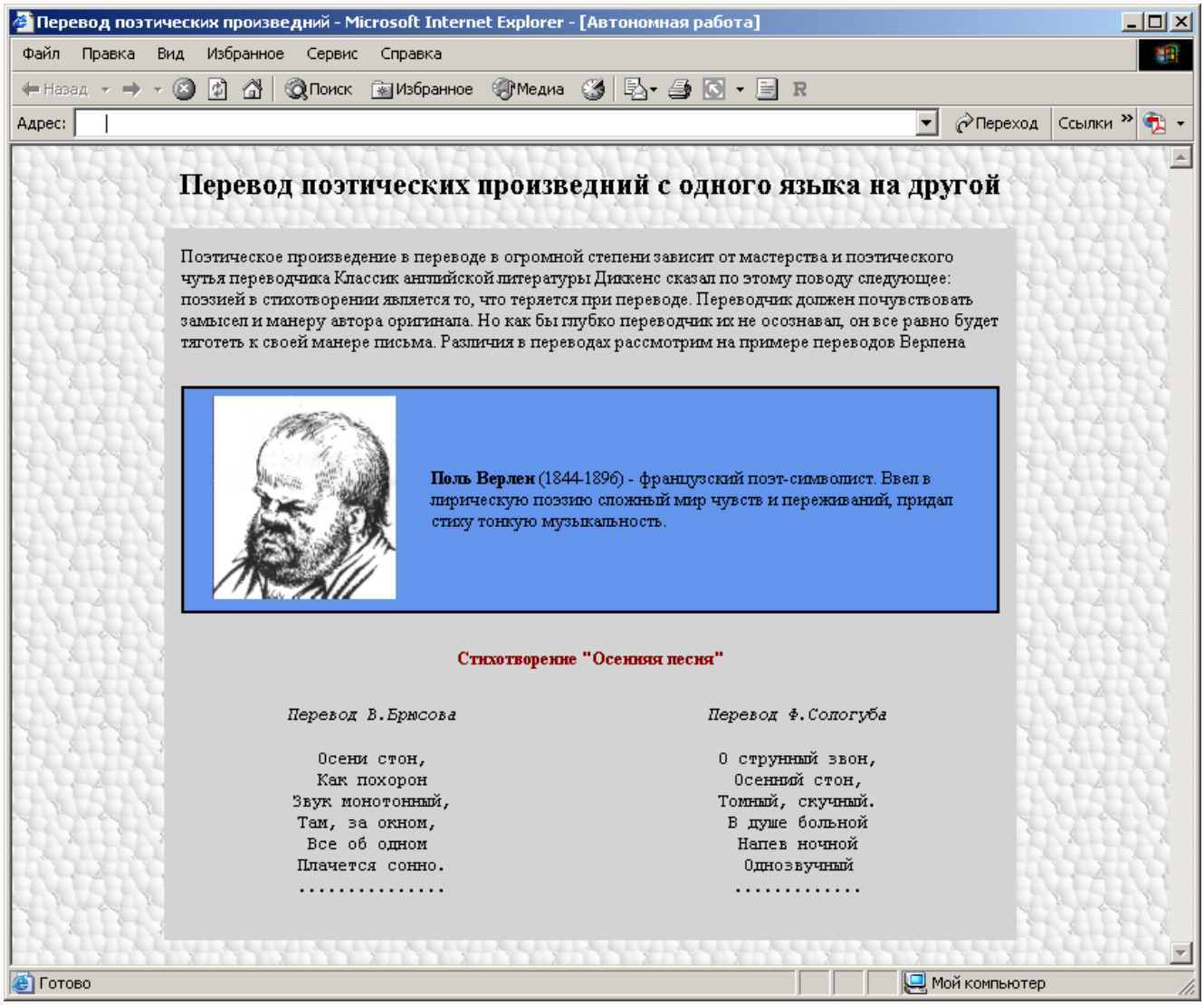


Рисунок 15 - Внешний вид документа, отформатированного с помощью таблицы с невидимыми границами

Изначально содержимое документа отображается последовательно в виде непрерывного потока в том порядке, в котором оно задано. Использование таблиц позволяет разбить документ на отдельные прямоугольные области (ячейки), в каждую из которых можно помещать свою определенную информацию. В результате информация на странице может быть представлена, например, в виде газетных колонок.

Благодаря этому значительно возрастает привлекательность документа и удобство работы с ним.

Код документа

```
<HTML>
<HEAD>
<TITLE>Перевод поэтических произведний</TITLE>
</HEAD>
<BODY background="FONT\165.gif">
<H2 align=center>
    Перевод поэтических произведний с одного языка на другой <H2>
<TABLE width=600 bgcolor=LIGHTGREY Cellpadding=12 Cellspacing=0>
<TR>
<TD colspan=2>Поэтическое произведение в переводе в огромной степени
зависит от мастерства и поэтического чутья переводчика
Классик английской литературы Диккенс сказал по этому поводу следующее:
поэзией в стихотворении является то, что теряется при переводе.
Переводчик должен почувствовать замысел и манеру автора оригинала.
Но как бы глубко переводчик  их не осознавал, он все равно будет
```

```
тяготеть к своей манере письма.
Различия в переводах рассмотрим на примере переводов Верлена
<TR>
<TD colspan=2>
<Table bgcolor=cornflowerblue frame=border rules=none
Cellpadding=5 Cellspacing=0 bordercolor=black>
<TR>
<TD><IMG src="IMAGES\verlen.gif" hspace=15>
<TD><STRONG>Поль Верлен</STRONG> (1844-1896) - французский поэт-символист. Ввел в лирическую
поэзию сложный мир чувств и переживаний, придал стиху тонкую музыкальность.
</TABLE>
<TR>
<TD colspan=2 align=center><FONT color=darkred>
<STRONG>Стихотворение "Осенняя песня"</STRONG></FONT>
<TR>
<TD align=center>
<PRE>
<EM>Перевод В.Брюсова</EM>

Осени стон,
Как похорон
Звук монотонный,
Там, за окном,
Все об одном
Плачется сонно.
...............
</PRE>
<TD align=center>
<PRE>
<EM>Перевод Ф.Сологуба</EM>

О струнный звон,
Осенний стон,
Томный, скучный.
В душе больной
Напев ночной
Однозвучный
.............
</PRE>
</BODY>
<HTML>
```

Порядок выполнения работы

1. В личной папке создать папку Лаб_1_ЛССУ, скопировать в нее папки FONT и IMAGES со всеми файлами, расположенные в папке с описанием лабораторной работы.

2. Открыть текстовый файл Sample1.txt с помощью редактора AditorPro. Добавить элементы форматирования, сохранить полученный файл, присвоив ему имя Sample1.htm. Отладить HTML=документ, запустив его на выполнение клавишей F5.

3. Для каждого из примеров 2-7 создать соответствующие текстовые файлы, преобразовать их в HTML-документы. Отлаженные примеры показать преподавателю.

Контрольные вопросы

1) В каком году была создана первая графическая программа-броузер?

2) Почему рекомендуется создавать HTML-документ вручную, не прибегая к помощи специализированных программных продуктов?.

3) Почему имена тегов должны состоять из букв нижнего регистра?.

4) Каких правил необходимо придерживаться при составлении текста HTML-документа?

5) Какой элемент используется в HTML-документе для разбивки его на параграфы?

6) Поясните назначение атрибутов, определяющих шрифт HTML-документа?

7) Перечислите элементы таблиц, применяемые в HTML-документах.

8) Как задаются в HTML-документе нумерованные списки?

9) Как производится включение изображений в HTML-документ?

Литература

1. Финков М.В. Интернет. Шаг второй: от пользователя к профессионалу. – СПб.: Наука и Техника, 2002. – 768 с.

2. Фролов А.В., Фролов Г.В. Создание Web-приложений: Практическое руководство. – М.: Издательско-торговый дом «Руская редакция», 2001. – 1040 с.

3. Гиз Ксения, Холмс Александра. Основы Web-дизайна: вспомогательное руководство. – М.: Издательский дом «Вильямс», 2002. – 640 с.

Практическая работа 2
Таблицы в HTML-документах

Цель работы: Изучить основные теги элемента TABLE; правила структурирования текста с помощью таблиц. Освоить простейшие приемы форматирования текстовых документов с помощью таблиц.

Общие сведения

Модель таблиц HTML позволяет авторам Web-страниц упорядочивать данные – простой и форматированный текст, изображения, ссылки, экранные формы, поля форм, другие таблицы и т.д. - в строки и столбцы ячеек. Широкое применение таблиц в HTML-документах нашли в силу того, что возможности форматирования без них достаточно бедны, например, невозможно задать абсолютные позиции тех или иных элементов оформления документа.

HTML-таблица создается с помощью элемента **TABLE**, который содержит необязательный элемент **CAPTION** (предоставляет краткое описание таблицы), и описания строк, сформированные с помощью элемента **TR**. Ячейки таблицы могут содержать "заголовок" (элемент **TH**) или "данные" (элемент **TD**). Ячейки могут распространяться на несколько строк или столбцов, причем авторы могут помечать каждую ячейку.

Модель таблиц HTML разработана так, чтобы броузеры могли генерировать таблицы *последовательно* (т.е. по мере получения строк таблицы), а не ждали получения всей таблицы до начала генерации.

Заполнение таблицы происходит построчно; заголовки выводятся полужирным шрифтом и располагаются по центру ячейки. Данные имеют обычный шрифт и выравниваются по левой стороне ячейки.

Пример простой таблицы (рисунок 1).

```
<TABLE>
<CAPTION>Заголовок таблицы</CAPTION>
<TR><TH>Заголовок 1</TH><TH>Заголовок 2</TH></TR>
<TR><TD>Ячейка 1</TD><TD>Ячейка 2</TD></TR>
<TR><TD>Ячейка 3</TD><TD>Ячейка 4</TD></TR>
</TABLE>
```

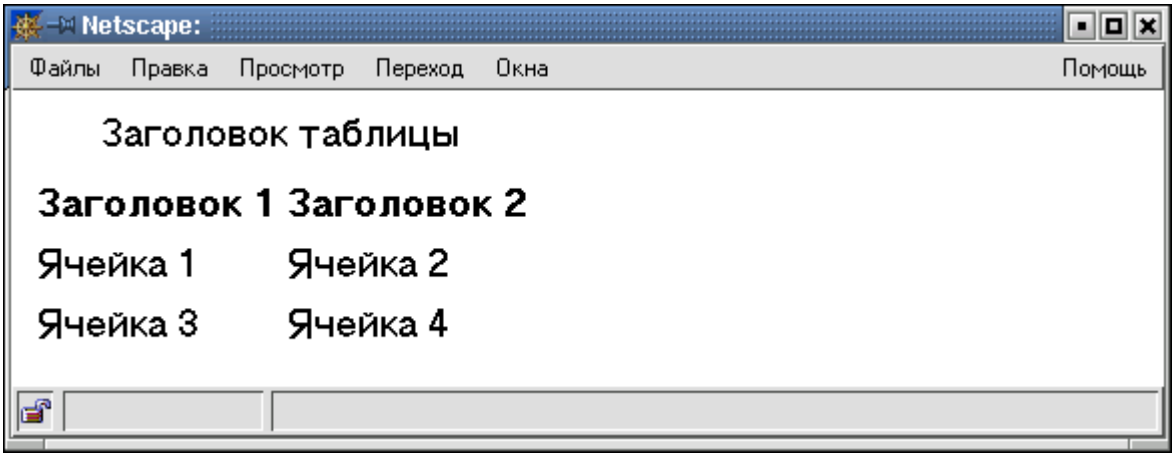


Рисунок 1 – Пример простой таблицы

Наличие в ячейках данных не обязательно. Создать пустую ячейку можно двумя способами: ничем не заполнять ее контейнер (**<TD></TD>**), либо поместить в нее символ неразрывного пробела, задаваемого специальной последовательностью символов --   (т.е. создать ячейку вида **<TD> </TD>**).

Строки таблицы могут группироваться с помощью элементов **THEAD**, **TFOOT** и **TBODY** в логические разделы. При использовании с атрибутом **RULES="GROUPS"** элемента **TABLE** эти фрагменты выделяются в наборы строк. В составе таблицы допустим один раздел заголовков (*header section*), один раздел сносок (*footer section*) и любое количество разделов тела таблицы (*body section*). Каждый раздел таблицы должен

содержать, по крайней мере, одну строку, сформированную посредством тега **TR**. Все строки, входящие в группу, форматируются одновременно.

Элемент **TFOOT** должен располагаться до элемента **TBODY** в определении **TABLE**, чтобы броузеры могли генерировать нижний заголовок до получения всех (возможно, многочисленных) строк данных:

```
<TABLE>
<THEAD>
     <TR> ...заголовок...
</THEAD>
<TFOOT>
     <TR> ...нижний заголовок...
</TFOOT>
<TBODY>
     <TR> ...первая строка данных блока 1...
     <TR> ...вторая строка данных блока 1...
</TBODY>
<TBODY>
     <TR> ...первая строка данных блока 2...
     <TR> ...вторая строка данных блока 2...
</TBODY>
</TABLE>
```

Разделы **THEAD**, **TFOOT** и **TBODY** должны содержать одинаковое число столбцов.

Авторы могут группировать столбцы для предоставления дополнительной структурной информации, которая может использоваться броузерами. Более того, авторы могут объявлять свойства столбцов в начале определения таблицы (с помощью элементов **COLGROUP** и **COL**) таким образом, который позволяет броузеру генерировать таблицу последовательно, а не ждать считывания всех данных таблицы перед тем, как начать генерацию.

Нет надобности отдельно создавать пустые ячейки, если планируется, что все оставшиеся в строке ячейки не будут заполнены. Так как тег **<TR>** сигнализирует о начале новой строки, пустые ячейки будут добавлены броузером автоматически.

Основные теги таблиц

Назначение основных атрибутов тега **<TABLE>** и значения, которые они могут принимать перечислены в таблице 1.

Таблица 1 – Назначение основных тегов элемента **TABLE**

Атрибут	**Назначение**
BORDER=n	Определяет ширину рамки таблицы (в пикселах), например, BORDER=1; значение, равное нулю, означает отсутствие рамки
WIDTH=n	Определяет ширину всей таблицы в пикселах, либо в процентах от ширины окна браузера
HEIGHT=n	Определяет высоту всей таблицы в пикселах, либо в процентах от высоты окна браузера
ALIGN	Задает горизонтальное выравнивание таблицы в окне браузера; имеет значения left, center и right (по умолчанию -- left)
CELLPADDING=n	Добавляет свободное пространство между данными внутри ячейки и ее границами; по умолчанию значение равно 2
CELLSPACING=n	Добавляет свободное пространство между ячейками внутри всей таблицы; по умолчанию значение равно 2
HSPACE=n	Задает области свободного пространства указанной ширины (в пикселах) слева и справа от таблицы
VSPACE=n	Задает области свободного пространства заданной высоты (в пикселах) сверху и снизу от таблицы
BGCOLOR=цвет	Устанавливает цвет фона всей таблицы
BACKGROUND=URL	Указывает фоновое изображение для таблицы, где URL - адрес источника (имя файла с изображением)

Пример. Изменим содержимое документа, созданного в предыдущем примере,

добавив атрибуты **BORDER** и **ALIGN** в тег **<TABLE>**:

```
<TABLE BORDER=1 ALIGN=center>
```

Теперь ячейки таблицы будут обрамлены рамкой, а таблица выровнена по центру окна броузера (рисунок 2).

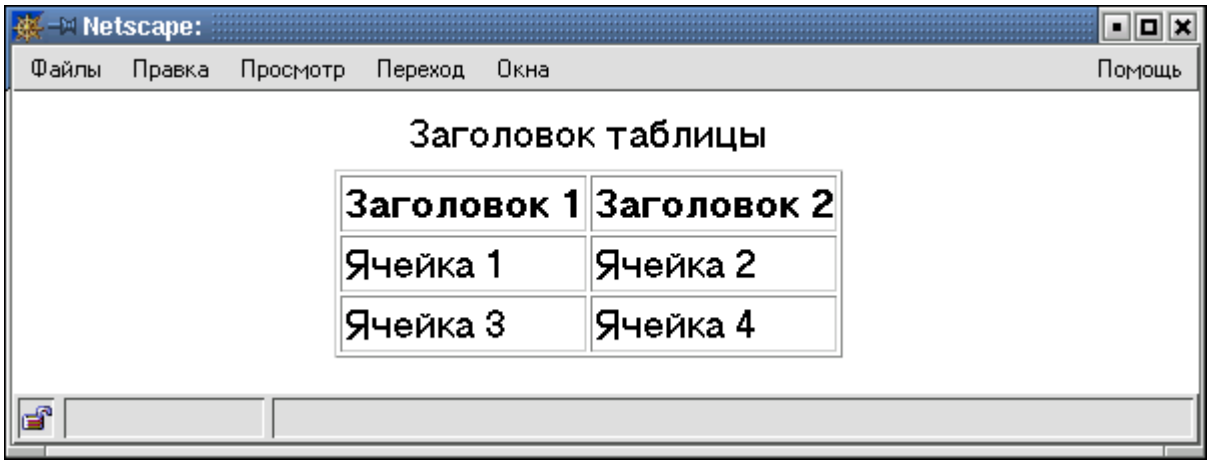


Рисунок 2 – Выравнивание таблицы по центру и добавление рамок

Смежные ячейки таблицы могут объединяться. Например, в таблице из нескольких столбцов все ячейки первой строки можно объединить и поместить в этой строке красивый заголовок таблицы. Возможно также объединение нескольких строк или создание пустой прямоугольной области.

Для соединения двух смежных ячеек в одном столбце нужно использовать атрибут **ROWSPAN** тега <**TH**> или **<TD>**, например,

```
<TD ROWSPAN=2>
```

Для объединения двух смежных ячеек в одной строке нужно использовать атрибут COLSPAN тех же тегов, например,

```
<TD COLSPAN=2>
```

Если требуется создать таблицу с объединениями столбцов и в то же время точно контролировать ширину каждого столбца, необходимо задать ширину по крайней мере одной ячейки в каждом столбце.

Авторы могут указывать ширину столбцов тремя способами:

Фиксированная

Указание фиксированной ширины дается в пикселах (например, width="30"). Использование фиксированной ширины позволяет использовать последовательную генерацию.

В процентах

Указание ширины в процентах (например, width="20%") означает процент горизонтального пространства, доступного для таблицы (между текущим левым и правым полями, включая floats). Это пространство не зависит от самой таблицы, поэтому указание ширины в процентах позволяет использовать последовательную генерацию.

Пропорциональная

Указание пропорциональной ширины (например, width="3*") означает число частей горизонтального пространства, *необходимого* для таблицы. Если ширина таблицы определяется как фиксированное значение (с помощью атрибута width элемента **TABLE**), броузеры могут генерировать таблицу последовательно и с указанием пропорциональной ширины.

Применение цвета

В HTML не предусмотрено специальных средств раскрашивания таблиц. Однако как Netscape Navigator, так и Microsoft Internet Explorer имеют расширения, позволяющие изменять цвет ячеек и рамок. Можно изменить цвет фона ячейки при помощи атрибута **BGCOLOR** перед размещением в ней текста или изображения, а также использовать атрибут **BORDERCOLOR** для изменения цвета рамки ячейки. Теги <**TABLE**>, <**TD**>, <**TH**> и <**TR**> также допускают использование в них указанных атрибутов. Таким образом, вы можете изменить цвет всей таблицы, отдельной ячейки или строки таблицы.

Значения цветов, установленные на уровне ячейки, будут перекрывать значения, установленные на уровне строки, которые в свою очередь, будут перекрывать

значения, заданные на уровне всей таблицы.

Пример. Создадим таблицу, удовлетворяющую следующим требованиям:

- цвет таблицы -- белый;
- вторая строка светло-серого цвета, в ячейке 2 -- цвет "teal";
- цвет надписи в ячейке 5 -- красный;
- первый столбец составляет 20% от ширины таблицы, два другие -- по 40%;
- ячейка 3 центрирована, а ячейка 4 -- выровнена вправо;
- ячейки 6 и 8 объединены в одну, центрированы и выровнены по нижнему краю ячейки;
- поля внутри ячеек -- 10 пикселей.

Фрагмент документа с таблицей примера.

```
<TABLE BORDER=8 WIDTH=90% BGCOLOR="white"
   CELLPADDING=10 ALIGN=center>
   <TR>
      <TH WIDTH=20%>Простой заголовок</TH>
      <TH WIDTH=80% COLSPAN=2>Заголовок на 2 столбца</TH>
   </TR>
   <TR BGCOLOR="#CCCCCC">
      <TH WIDTH=20%>Заголовок на 1 строку</TH>
      <TD WIDTH=40%>Ячейка 1</TD>
      <TD WIDTH=40% BGCOLOR="teal">Ячейка 2</TD>
   </TR>
   <TR>
      <TH WIDTH=20% ROWSPAN=3>Заголовок на 3 строки</TH>
      <TD ALIGN=center>Ячейка 3</TD>
      <TD ALIGN=right>Ячейка 4</TD>
   </TR>
   <TR>
      <TD><FONT COLOR="red">Ячейка 5</FONT></TD>
      <TD ROWSPAN=2 ALIGN=center VALIGN=bottom>Ячейка 6</TD>
   </TR>
   <TR>
      <TD>Ячейка 7</TD>
   </TR>
</TABLE>
```

Примеры таблиц

Простая таблица

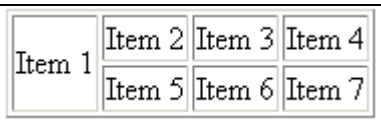
Вид таблицы	Фрагмент HTML-документа

A B C
D E F
	`<TABLE BORDER>`
	`<TR>`
	`<TD>A</TD> <TD>B</TD> <TD>C</TD>`
	`</TR>`
	`<TR>`
	`<TD>D</TD> <TD>E</TD> <TD>F</TD>`
	`</TR>`
	`</TABLE>`

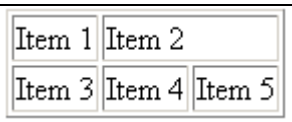
Два примера с объединением столбцов (с использованием атрибута ROWSPAN)

Вид таблицы	Фрагмент HTML-документа

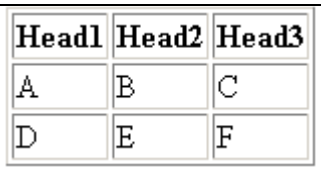
Item 1 / Item 2 / Item 3 / Item 4 / Item 5	`<TABLE BORDER>`
	`<TR>`
	`<TD>Item 1</TD>`
	`<TD ROWSPAN=2>Item 2</TD>`
	`<TD>Item 3</TD>`
	`</TR>`
	`<TR>`
	`<TD>Item 4</TD> <TD>Item 5</TD>`
	`</TR>`
	`</TABLE>`

	<TABLE BORDER> <TR> <TD ROWSPAN=2>Item 1</TD> <TD>Item 2</TD> <TD>Item 3</TD> <TD>Item 4</TD> </TR> <TR> <TD>Item 5</TD> <TD>Item 6</TD> <TD>Item 7</TD> </TR> </TABLE>

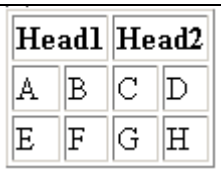
Пример с объединением смежных строк (при помощи атрибута COLSPAN)

Вид таблицы	**Фрагмент HTML-документа**
	<TABLE BORDER> <TR> <TD>Item 1</TD> <TD COLSPAN=2>Item 2</TD> </TR> <TR> <TD>Item 3</TD> <TD>Item 4</TD> <TD>Item 5</TD> </TR> </TABLE>

Демонстрация заголовков (используется тег <TH>)

Вид таблицы	**Фрагмент HTML-документа**
	<TABLE BORDER> <TR> <TH>Head1</TH> <TH>Head2</TH> <TH>Head3</TH> </TR> <TR> <TD>A</TD> <TD>B</TD> <TD>C</TD> </TR> <TR> <TD>D</TD> <TD>E</TD> <TD>F</TD> </TR> </TABLE>

Демонстрация использования атрибута COLSPAN в заголовках

Вид таблицы	**Фрагмент HTML-документа**
	<TABLE BORDER> <TR> <TH COLSPAN=2>Head1</TH> <TH COLSPAN=2>Head2</TH> </TR> <TR> <TD>A</TD> <TD>B</TD> <TD>C</TD> <TD>D</TD> </TR> <TR> <TD>E</TD> <TD>F</TD> <TD>G</TD> <TD>H</TD> </TR> </TABLE>

Демонстрация множественных заголовков (используя атрибут COLSPAN)

Вид таблицы

Head1		Head2	
Head 3	Head 4	Head 5	Head 6
A	B	C	D
E	F	G	H

Фрагмент HTML-документа

```
<TABLE BORDER>
        <TR>
                <TH COLSPAN=2>Head1</TH>
                <TH COLSPAN=2>Head2</TH>
        </TR>
        <TR>
                <TH>Head 3</TH> <TH>Head 4</TH>
                <TH>Head 5</TH> <TH>Head 6</TH>
        </TR>
        <TR>
                <TD>A</TD> <TD>B</TD> <TD>C</TD> <TD>D</TD>
        </TR>
        <TR>
                <TD>E</TD> <TD>F</TD> <TD>G</TD> <TD>H</TD>
        </TR>
</TABLE>
```

Демонстрация боковых заголовков

Вид таблицы

Head1	Item 1	Item 2	Item 3
Head2	Item 4	Item 5	Item 6
Head3	Item 7	Item 8	Item 9

Фрагмент HTML-документа

```
<TABLE BORDER>
   <TR><TH>Head1</TH>
      <TD>Item 1</TD> <TD>Item 2</TD>
      <TD>Item 3</TD></TR>
   <TR><TH>Head2</TH>
       <TD>Item 4</TD> <TD>Item 5</TD>
       <TD>Item 6</TD></TR>
    <TR><TH>Head3</TH>
       <TD>Item 7</TD> <TD>Item 8</TD>
       <TD>Item 9</TD></TR>
</TABLE>
```

Демонстрация боковых заголовков (используя атрибут ROWSPAN)

Вид таблицы

Head1	Item 1	Item 2	Item 3	Item 4
	Item 5	Item 6	Item 7	Item 8
Head2	Item 9	Item 10	Item 3	Item 11

Фрагмент HTML-документа

```
<TABLE BORDER>
        <TR><TH ROWSPAN=2>Head1</TH>
            <TD>Item 1</TD> <TD>Item 2</TD>
<TD>Item 3</TD> <TD>Item 4</TD>
        </TR>
        <TR><TD>Item 5</TD> <TD>Item 6</TD>
<TD>Item 7</TD> <TD>Item 8</TD>
        </TR>
        <TR><TH>Head2</TH>
            <TD>Item 9</TD> <TD>Item 10</TD>
<TD>Item 3</TD> <TD>Item 11</TD>
        </TR>
</TABLE>
```

Пример таблицы использующей все эти тэги

Вид таблицы

		Average	
		Height	Weight
Gender	Males	1.9	0.003
	Females	1.7	0.002

Фрагмент HTML-документа

```
<TABLE BORDER>
    <TR>  <TD><TH ROWSPAN=2></TH>
          <TH COLSPAN=2>Average</TH></TD>
    </TR>
    <TR>  <TD><TH>Height</TH>
          <TH>Weight</TH></TD>
    </TR>
    <TR>  <TH ROWSPAN=2>Gender</TH>
          <TH>Males</TH><TD>1.9</TD>
          <TD>0.003</TD>
```

	</TR> <TR><TH>Females</TH> <TD>1.7</TD><TD>0.002</TD> </TR> </TABLE>

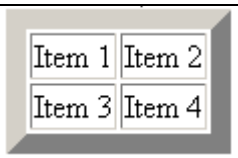
Использование ROWSPAN/COLSPAN

Вид таблицы	Фрагмент HTML-документа
	<TABLE BORDER> <TR> <TD ALIGN=center ROWSPAN=2 COLSPAN=2>A</TD> <TD>1</TD> <TD>2</TD> </TR> <TR> <TD>3</TD> <TD>4</TD> </TR> <TR> <TD ALIGN=center ROWSPAN=2 COLSPAN=2>C</TD> <TD ALIGN=center ROWSPAN=2 COLSPAN=2>D</TD> </TR> <TR> </TR> </TABLE>

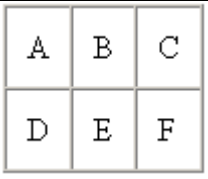
Таблица без рамки

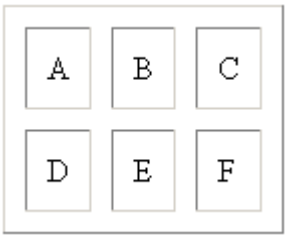
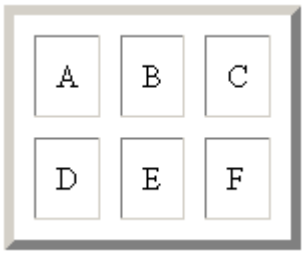
Вид таблицы	Фрагмент HTML-документа
Item 1 Item 2 Item 3 Item 4 Item 5	<TABLE> <TR> <TD>Item 1</TD> <TD ROWSPAN=2>Item 2</TD> <TD>Item 3</TD> </TR> <TR> <TD>Item 4</TD> <TD>Item 5</TD> </TR> </TABLE>

Таблица с рамкой шириной 10

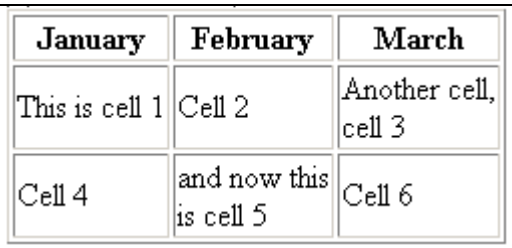
Вид таблицы	Фрагмент HTML-документа
	<TABLE BORDER=10> <TR> <TD>Item 1</TD> <TD> Item 2</TD> </TR> <TR> <TD>Item 3</TD> <TD>Item 4</TD> </TR> </TABLE>

CELLPADDING И CELLSPACING

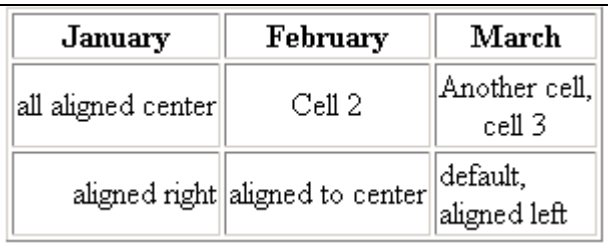
Вид таблицы	Фрагмент HTML-документа
	<TABLE BORDER CELLPADDING=10 CELLSPACING=0> <TR> <TD>A</TD> <TD>B</TD> <TD>C</TD> </TR> <TR> <TD>D</TD> <TD>E</TD> <TD>F</TD> </TR> </TABLE>
	<TABLE BORDER CELLPADDING=0 CELLSPACING=10> <TR> <TD>A</TD> <TD>B</TD> <TD>C</TD> </TR> <TR> <TD>D</TD> <TD>E</TD> <TD>F</TD>

	</TR> </TABLE>
	<TABLE BORDER CELLPADDING=10 CELLSPACING=10> <TR> <TD>A</TD> <TD>B</TD> <TD>C</TD> </TR> <TR> <TD>D</TD> <TD>E</TD> <TD>F</TD> </TR> </TABLE>
	<TABLE BORDER=5 CELLPADDING=10 CELLSPACING=10> <TR> <TD>A</TD> <TD>B</TD> <TD>C</TD> </TR> <TR> <TD>D</TD> <TD>E</TD> <TD>F</TD> </TR> </TABLE>

Демонстрация множественных строк в таблицы строк в таблице

Вид таблицы	**Фрагмент HTML-документа**
	<TABLE BORDER> <TR> <TH>January</TH> <TH>February</TH> <TH>March</TH> </TR> <TR> <TD>This is cell 1</TD> <TD>Cell 2</TD> <TD>Another cell, cell 3</TD> </TR> <TR> <TD>Cell 4</TD> <TD>and now this is cell 5</TD> <TD>Cell 6</TD> </TR> </TABLE>

Управление горизонтальным выравниванием внутри таблицы. Атрибут ALIGN=LEFT|RIGHT|CENTER может применяться к отдельным ячейкам или всей строке

Вид таблицы	**Фрагмент HTML-документа**
	<TABLE BORDER> <TR> <TH>January</TH> <TH>February</TH> <TH>March</TH> </TR> <TR ALIGN=center> <TD>all aligned center</TD> <TD>Cell 2</TD> <TD>Another cell, cell 3</TD> </TR> <TR> <TD ALIGN=right>aligned right</TD> <TD ALIGN=center>aligned to center</TD> <TD>default, aligned left</TD> </TR> </TABLE>

Управление горизонтальным выравниванием внутри таблицы. Атрибут *VALIGN=TOP|BOTTOM|MIDDLE* может применяться к отдельным ячейкам или всей строке

Вид таблицы

January	February	March
all aligned to top	and now this is cell 2	Cell 3
aligned to the top	aligned to the bottom	default alignment, center

Фрагмент HTML-документа

```
<TABLE BORDER>
   <TR>
        <TH>January</TH>
        <TH>February</TH>
        <TH>March</TH>
   </TR>
   <TR VALIGN=top>
        <TD>all aligned to top</TD>
        <TD>and now this<br>is cell
2</TD>
        <TD>Cell 3</TD>
   </TR>
   <TR>
        <TD VALIGN=top>aligned to
the top</TD>
         <TD VALIGN=bottom>aligned
to the bottom</TD>
         <TD>default
alignment,<br>center</TD>
   </TR>
</TABLE>
```

Управление выравниванием тематического заголовка таблицы **CAPTION=TOP|BOTTOM**

Вид таблицы

Верхний заголовок

January	February	March
This is cell 1	Cell 2	Another cell, cell 3

Фрагмент HTML-документа

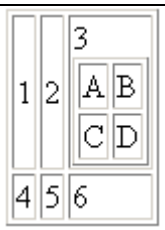
```
<TABLE BORDER>
<CAPTION ALIGN=top>Верхний заголовок</CAPTION>
   <TR>
        <TH>January</TH>
        <TH>February</TH>
        <TH>March</TH>
   </TR>
   <TR>
        <TD>This is cell 1</TD>
        <TD>Cell 2</TD>
        <TD>Another cell,<br> cell 3</TD>
   </TR>
</TABLE>
```

January	February	March
This is cell 1	Cell 2	Another cell, cell 3

Нижний заголовок

```
<CAPTION ALIGN=bottom>Нижний заголовок</CAPTION>
   <TR>
        <TH>January</TH>
        <TH>February</TH>
        <TH>March</TH>
   </TR>
   <TR>
        <TD>This is cell 1</TD>
        <TD>Cell 2</TD>
        <TD>Another cell,<br> cell 3</TD>
   </TR>
</TABLE>
```

Вложенные таблицы. Тег <!-- ROW 1, TABLE 1 --> представляет пример включения комментария

Вид таблицы	**Фрагмент HTML-документа**
	<TABLE BORDER> <TR> <!-- ROW 1, TABLE 1 --> <TD>1</TD> <TD>2</TD> <TD>3 <TABLE BORDER> <TR> <!-- ROW 1, TABLE 2 --> <TD>A</TD> <TD>B</TD> </TR> <TR> <!-- ROW 2, TABLE 2 --> <TD>C</TD> <TD>D</TD> </TR> </TABLE> </TD> </TR> <TR> <!-- ROW 2, TABLE 1 --> <TD>4</TD> <TD>5</TD> <TD>6</TD> </TR> </TABLE>

Задание ширины таблицы

Ширина таблицы и каждой ячейки 50%

Width=50%	Width=50%
3	4

```
<TABLE BORDER WIDTH="50%">
     <TR><TD>Width=50%</TD><TD>Width=50%</TD>
     </TR>
     <TR><TD>3</TD><TD>4</TD>
     </TR>
</TABLE>
```

Указана ширина таблицы, для ячеек ширина не задана

Item width affects cell size	2
3	4

```
<TABLE BORDER WIDTH="50%">
     <TR><TD>Item width affects cell size</TD><TD>2</TD>
     </TR>
     <TR><TD>3</TD><TD>4</TD>
     </TR>
</TABLE>
```

Общая ширина таблицы превышает 100%

WIDTH=100%	This is item 2
3	4

```
<TABLE BORDER WIDTH="100%">
     <TR><TD>WIDTH=100%</TD><TD>This is item 2</TD>
     </TR>
     <TR><TD>3</TD><TD>4</TD>
     </TR>
</TABLE>
```

Центрирование таблицы на странице

A	B	C
D	E	F

```
<CENTER>
<TABLE BORDER WIDTH="50%">
   <TR>
       <TD>A</TD> <TD>B</TD> <TD>C</TD>
   </TR>
   <TR>
       <TD>D</TD> <TD>E</TD> <TD>F</TD>
   </TR>
</TABLE>
</CENTER>
```

Указана ширина вложенной таблицы

Item 1		Item 2
Item A	Item B	Item 4

```
<TABLE BORDER WIDTH="50%">
    <TR><TD>Item 1</TD><TD>Item 2</TD>
    </TR>
    <TR><TD>
      <TABLE BORDER WIDTH=100%>
        <TR><TD>Item A</TD><TD>Item B</TD>
        </TR>
      </TABLE>
      </TD>
      <TD>Item 4</TD>
    </TR>
</TABLE>
```

Указана высота ячеек таблицы

HEIGHT=15%	Item 2
3	4

```
<TABLE BORDER WIDTH="50%" HEIGHT="15%">
    <TR><TD>HEIGHT=15%</TD> <TD>Item 2</TD>
    </TR>
    <TR><TD>3</TD><TD>4</TD>
    </TR>
</TABLE>
```

Порядок выполнения работы

1. В личной папке создайте папку Лаб_2_ЛССУ, в которой должны располагаться файлы HTNL-документов лабораторной работы. Последовательно подготовьте тексты примеров (рисунки 3 – 11).

2. Используя тег TH, создайте таблицу, у которой заголовки столбцов расположены сверху (рисунок 3).

Столбец 1	Столбец 2	Столбец 3
000	001	002
010	011	012
100	101	102

Рисунок 3 – Таблица, заголовки столбцов которой расположены сверху

3. Создайте таблицу, внешний вид которой показан на рисунке 4.

Строка 1	000	001	002
Строка 2	010	011	012
Строка 3	100	101	102

Рисунок 4 – Таблица, заголовки столбцов которой расположены сбоку

4. Путем объединения ячеек получите таблицу, у которой для двух столбцов имеется общий заголовок с помощью атрибута COLSPAN (рисунок 5).

Столбец 1	Столбец 2 и 3	
000	001	002
010	011	012
100	101	102

Рисунок 5 – Таблица с общим заголовком для двух столбцов

5. С помощью атрибута ROWSPAN создайте таблицу с общим заголовком для двух строк, как показано на рисунке 6.

Строка 1 и 2	000	001	002
	010	011	012
Строка 3	100	101	102

Рисунок 6 – Таблица с общим заголовком для двух строк

6. «Шапка» таблицы может состоять из заголовков, расположенных в несколько ярусов. Создайте таблицу, внешний вид которой приведен на рисунке 7.

Заголовок для всех столбцов		
Стобец 1 и 2		**Столбец 3**
000	001	002
010	011	012
100	101	102

Рисунок 7 – Таблица с «шапкой» в два яруса

7. Создайте таблицу со множественными заголовками и верхним тематическим заголовком (рисунок 8)

Таблица 1.1 - Таблица с двойным заголовоком

Заголовок для всех столбцов		
Стобец 1 и 2		**Столбец 3**
000	001	002
010	011	012
100	101	102

Рисунок 8 – Таблица с верхним тематическим заголовком

8. Создайте таблицу со множественными заголовками и нижним тематическим заголовком (рисунок 9)

Заголовок для всех столбцов		
Стобец 1 и 2		Столбец 3
000	001	002
010	011	012
100	101	102

Таблица 1.2 - Таблица с двойным заголовоком

Рисунок 9 – Таблица с нижним тематическим заголовком

9. Создайте таблицу, множественные заголовки которой располагаются так, как показано на рисунке 10.

		Тип выравнивания		
		По левому краю	Центрирование	По правому краю
Направление	По вертикали	Этот текст выровнен по левому краю	Этот текст отцентрирован	Этот текст выровнен по правому краю
	По горизонтали	Вверху	Посередине	Внизу

Рисунок 10 – Таблица с множественным заголовками для групп столбцов и строк

10. Создайте вложенную таблицу (рисунок 11).

01	02	001	002
11	12		
100		101	102
200		201	202

Рисунок 11 – Вложенная таблица

11. Подготовьте письменный отчет и получите зачет по выполненной лабораторной работе.

Содержание отчета

Отчет должен содержать:

1. Наименование лабораторной работы и ее цель.
2. Тексты HTML-документов, позволяющих получить таблицы, внешний вид которых показан на рисунках 3-11.
3. Выводы.

Контрольные вопросы

1) Для чего используются HTNL-таблицы?
2) Каким образом используется цвет в таблицах?.
3) Почему необходимо задавать ширину столбцов таблицы?.
4) В каких случаях броузер выполняет переформатирование таблицы, если для каждого столбца задана его ширина?
5) Какой элемент используется в HTML-таблице для указания тематического заголовка?
6) Поясните назначение атрибутов, задающих горизонтальное выравнивание в ячейках HTML-таблицы?
7) Каким образом производится вертикальное выравнивание текста в ячейках HTML-таблицы?
8) Как задаются в HTML-таблице высота ячейка?
9) Как организовать в HTML-таблице множественные заголовки для столбцов?

Литература

1. Финков М.В. Интернет. Шаг второй: от пользователя к профессионалу. – СПб.: Наука и Техника, 2002. – 768 с.
2. Фролов А.В., Фролов Г.В. Создание Web-приложений: Практическое руководство. – М.: Издательско-торговый дом «Руская редакция», 2001. – 1040 с.