

ПРАКТИЧЕСКАЯ РАБОТА 1

Цель работы: изучение основных концепций HTML, необходимых для создания простейших файлов HTML, содержащий форматированный текст

ВВЕДЕНИЕ

Информационные технологии (ИТ, от англ. information technology, IT) — широкий класс дисциплин и областей деятельности, относящихся к технологиям управления и обработки данных, в том числе с применением вычислительной техники. В настоящее время под информационными технологиями (ИТ) чаще всего понимают компьютерные технологии. В частности, ИТ имеют дело с использованием компьютеров и программного обеспечения для хранения, преобразования, защиты, обработки, передачи и получения информации. Важнейшими историческими этапами развития ИТ является письменность, изобретение книгопечатания, использование почты, телефона, телеграфа, телевидения. Особое место в развитии ИТ занимают компьютеры, электронная почта и широкое использование компьютерных сетей (локальных и глобальных), которые обеспечивают не только содержательную обработку информации, но и передачу текстовых, мультимедийных (графика, видео и звук) и других материалов практически на любые расстояния в реальном масштабе времени.

ИТ (Информационные технологии) являются наиболее важной составляющей процесса использования информационных ресурсов общества. К настоящему времени она прошла несколько эволюционных этапов, смена которых определялась, главным образом, развитием научно-технического прогресса, появлением новых технических средств переработки информации.

Сегодня Интернет является одним из самых современных и революционных средств передачи информации. Интернет состоит из многочисленных компьютерных сетей, объединенных в одну глобальную сеть и обменивающихся между собой информацией. Все разнообразие оформления и возможностей веб-страниц реализовано на основе одного средства разметки текста – HTML. Язык разметки гипертекста HTML (Hyper Text Markup Language) является стандартным средством представления информации в среде Word Wide Web (WWW) в виде веб-страниц.

WWW (World Wide Web) - это часть Internet, представляющая собой гигантский набор документов, хранящихся на компьютерах по всему миру. Сокращенно WWW называют просто Вэб. Веб-страница - это отдельный документ Web, который способен содержать информацию различного вида - текст, рисунки, фото, видео, аудиозаписи. Такие веб-страницы хранятся на специальных компьютерах, которые называются серверами.

Будучи сохраненной на веб-сервере, страница становится доступной для использования другими пользователями. Коллекция веб-страниц, объединенная по какому-либо признаку, называется веб-сайтом или просто сайтом.

URL (Uniform Resource Locator) - это уникальный адрес, которым обладает каждая вэб-страница в сети. Если пользователю известен URL

страницы, то он может ее отобразить у себя в браузере, набрав ее адрес в строке браузера. В общем случае URL состоит из наименования используемого протокола, названия сервера и обозначения пути доступа к странице.

Протокол определяет правила обращения к веб-странице.

Название сервера (доменное имя) - обозначает компьютер, содержащий данную веб-страницу.

Путь доступа - указывает местоположение страницы на диске сервера.

Например, адрес данной страницы `http://on-line-teaching.com/html/index.html`,

где *http* - наименование протокола;

on-line-teaching.com - наименование сервера или доменное имя;

/html/index.html - путь доступа к странице на самом сервере.

Web-браузер - это программа, установленная на компьютере пользователя, и служащая для поиска и отображения информации в сети. Браузер считывает веб-страницы и другие файлы с диска сервера и отображает их содержимое на мониторе компьютера пользователя. Среди наиболее популярных браузеров следует назвать Microsoft Internet Explorer, Netscape Navigator, Opera.

Язык HTML в том виде, в котором он существует сейчас, обладает большим потенциалом представления информации с расчетом на пользователей персональных компьютеров. Документы HTML можно просматривать на очень большом количестве различных по своим возможностям устройств: от черно-белого экрана мобильного телефона до телетайпа или терминала. Также в последние версии HTML включен ряд возможностей, облегчающих представление документов неграфическими средствами (например, речевыми синтезаторами), позволяющими пользоваться услугами WWW людям с ограниченными физическими возможностями.

БАЗОВЫЕ ПОНЯТИЯ

Web-страницы - это документы в формате HTML, содержащие текст и специальные тэги (дескрипторы) HTML. По большому счету, тэги HTML необходимы для форматирования текста (т.е. придания ему нужного вида), который "понимает" браузер. Документы HTML хранятся в виде файлов с расширением «.htm» или «.html».

Тэги HTML сообщают браузеру информацию о структуре и особенностях форматирования Веб-страницы. Каждый тэг содержит определенную инструкцию и заключается в угловые скобки < >. Большинство тэгов состоит из открывающей и закрывающей частей и воздействует на текст, заключенный внутри.

Теги бывают одиночными и контейнерными. Контейнером называется пара: открывающий <ТЕГ> и закрывающий </ТЕГ>.

<ТЕГ> Контейнер </ТЕГ>

Атрибуты тега задают значения свойств данного объекта или объектов, помещенных в контейнер. Значения свойств, содержащих пробелы, берутся в кавычки, в остальных случаях кавычки можно опустить.

СТРУКТУРА HTML – ДОКУМЕНТА

В идеальном случае HTML - документ состоит из трех частей:

- информации о версии используемого HTML;
- заголовка документа;
- тела документа.

Пример простейшего HTML - документа, содержащий все структурные элементы.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN">
<HTML>
  <HEAD>
    <TITLE>HTML-документ</TITLE>
  </HEAD>
  <BODY>
    <H1>Заголовок</H1>
    <P>Первый абзац
    <P>Второй абзац
  </BODY>
</HTML>
```

Первый тэг, который должен находиться в любом HTML-документе, это <HTML> ... </HTML>. Этот тэг указывает на то, что данный документ действительно содержит в себе HTML-текст. Все, что вы напишете в своем документе, должно находиться внутри данного тэга:

```
<HTML>
...
    текст документа
...
</HTML>
```

Заголовок документа

В заголовке (его еще называют «шапкой») HTML – документа содержатся сведения о документе: название (тема документа), ключевые слова (используемые поисковыми машинами), а также ряд других данных, которые не являются содержимым документа.

Определение заголовка должно содержаться внутри тэга **<HEAD> ...</HEAD>**:

```
<HTML>
    <HEAD>
        ...
        описание заголовка
        ...
    </HEAD>
...
текст документа
...
</HTML>
```

В разделе описания заголовка можно указать заглавие документа, для этого используется тэг **<TITLE> ... </TITLE>**. Когда браузер встречает этот тэг, он отображает все, что находится внутри него как заглавие.

Упражнение 1

1. Создайте папку HTML, в которой вы будете сохранять сконструированные Web-страницы.
2. Запустите программу Блокнот.
3. Наберите в окне редактора простейший текст файла HTML:

```
<HTML>
    <HEAD>
        <TITLE>Это заглавие документа</TITLE>
    </HEAD>...
    текст документа.....
</HTML>
```

Сохраните файл под именем «Упражнение 1.html».

Загрузите свой документ. Убедитесь, что название Web-страницы отразилось в верхней, статусной, строке браузера.

4. На экране вы увидите результат своей работы, изображенный на рис. 1.

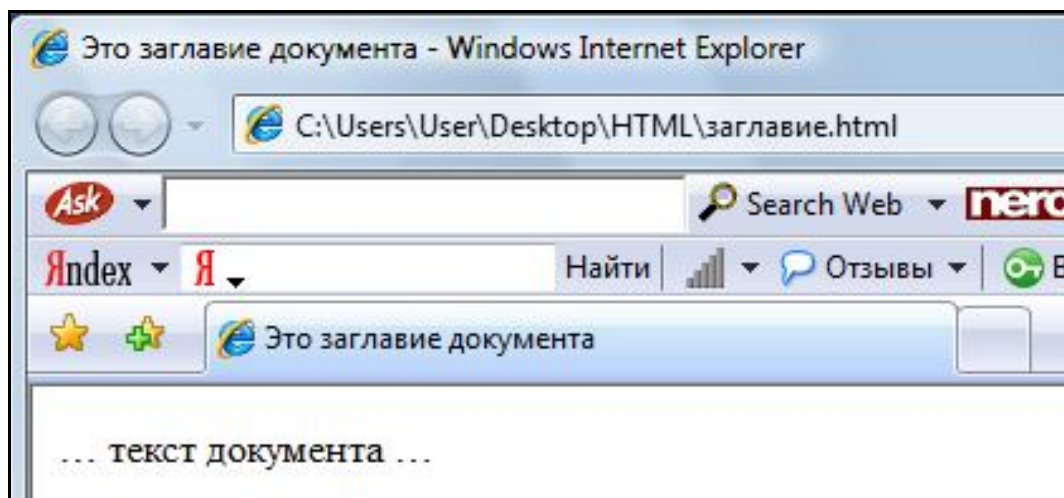


Рис. 1. Простейший HTML-документ

Определение тела документа

Весь остальной HTML-документ, включая весь текст, содержится внутри тэга `<BODY> ...</BODY>` (тело). Теперь наш документ выглядит примерно так:

```
<HTML>
  <HEAD>
    <TITLE>Это заглавие документа</TITLE>
  </HEAD>
  <BODY>
    ...
    текст документа ...
  </BODY>
</HTML>
```

Наиболее часто используемые атрибуты элемента **BODY**:

background – URL, указывающий расположение изображения для фона (обычно берется небольшое изображение, которое размножается для заполнения фона всего документа);

bgcolor – цвет фона HTML-документа;

text – цвет шрифта документа;

link – цвет непосещенных гиперссылок;

vlink – цвет посещенных гиперссылок;

alink – цвет гиперссылок при выборе их пользователем (при нажатии «Enter» произойдет переход по такой ссылке);

contenteditable – позволяет разрешить или запретить пользователю редактирование содержимого HTML-документа при просмотре его браузером (значения true, false, inherit).

Все атрибуты, позволяющие задать цвет, имеют тип %Color. Значения таких атрибутов могут задаваться шестнадцатеричными числами с символом # в начале каждого числа, например:

`bgcolor = "#FF0005"`

Также атрибутам задания цвета можно присваивать предопределенные идентификаторы некоторых наиболее часто употребляемых цветов. Список этих названий приведен в табл. 1.

Таблица 1

Идентификаторы и значения цветов

Название	Значение	Цвет
black	#000000	Черный
silver	#808080	Темно-серый
gray	#C0C0C0	Серый
white	#FFFFFF	Белый
maroon	#800000	Бордовый
red	#FF0000	Красный
purple	#800080	Фиолетовый
fuchsia	#FF00FF	Лиловый
green	#00FF00	Зеленый
lime	#00FF00	Ярко-зеленый
olive	#808000	Оливковый
yellow	#FFFF00	Желтый
navy	#000080	Темно-синий
blue	#0000FF	Синий
teal	#008080	Сине-зеленый
aqua	#00FFFF	бирюзовый

ОСОБЕННОСТИ ВВОДА ТЕКСТА

Для того чтобы поместить простой текст на страницу, его достаточно ввести в тело документа. При этом браузер отобразит текст со шрифтом по умолчанию и цветом, заданным для текста тела страницы. Чтобы чтение информации, содержащейся в HTML-документе, стало приятным занятием, применяется форматирование текста.

Существуют специальные команды, выполняющие перевод строки и задающие начало нового абзаца. Кроме того, имеется команда, которая запрещает программе браузера каким-либо образом изменять форматирование текста и позволяет точно воспроизвести на экране заданный фрагмент текстового файла.

Тэг перевода строки `
` отделяет строку от последующего текста или графики. Тэг абзаца `<P>` тоже отделяет строку, но добавляет еще пустую строку, которая зрительно выделяет абзац. Оба тэга являются одноэлементными.

Задание начертания текста является, возможно, самым простым средством форматирования содержимого документа, которое доступно в HTML. Для изменения начертания текста в HTML-код вводятся элементы, обозначающие текст, написанный с соответствующим начертанием. В табл. 2 приведен список элементов, которые применяются при задании начертания текста.

Таблица 2

Элементы задания начертания текста

Элемент	Описание
B	Полужирное начертание
I	Курсивное начертание
U	Подчеркнутый текст
STRIKE,S	Перечеркнутый текст
BIG	Текст с увеличенным размером шрифта
SMALL	Текст с уменьшенным размером шрифта
SUP	Верхний индекс
SUB	Нижний индекс
TT	Текст, записанный моноширинным шрифтом (все символы имеют одинаковую ширину)
BLINK	Мерцающий текст (редко поддерживается браузерами).

Задание шрифта текста

Если нужно отобразить некоторый текст с использованием определенного шрифта, а не применяемого браузером по умолчанию, то в HTML предусмотрен элемент **FONT**. Он вводится при помощи парных тегов **** и ****.

Параметры шрифта для элемента **FONT** устанавливаются заданием значений следующих его атрибутов:

- **face** — задает название шрифта, например Arial или System;
- **size** — задает размер шрифта (значение от 1 до 7, по умолчанию используется значение 3; для атрибута size могут использоваться только семь значений);
- **color** — задает цвет шрифта.

Упражнение 2

1. Запустите программу Блокнот.
2. Наберите в окне редактора простейший текст файла HTML:

```
<HTML>
  <HEAD>
    <TITLE>Упражнение 2</TITLE>
  </HEAD>
  <P>
```

```

<Font Size='1' Face='Arial'>Шрифт 1-го размера </Font><BR>
<Font Size='2' Face='Arial'>Шрифт 2-го размера </Font><BR>
<Font Size='3' Face='Arial'>Шрифт 3-го размера </Font><BR>
<Font Size='4' Face='Times New Roman'>Шрифт 4-го размера
</Font><BR>
<Font Size='5' Face='Arial'>Шрифт 5-го размера </Font><BR>
<Font Size='6' Face='Monotype Corsiva'>Шрифт 6-го размера
</Font><BR>
<Font Size='7' Face='Botang'>Шрифт 7-го размера </Font>
<P><Font Face='Arial'> Размер по умолчанию </Font>
</HTML>

```

1. Сохраните файл под именем «Упражнение 2.html».
2. Загрузите свой документ.
3. На экране вы увидите результат своей работы, изображенный на рис. 2.

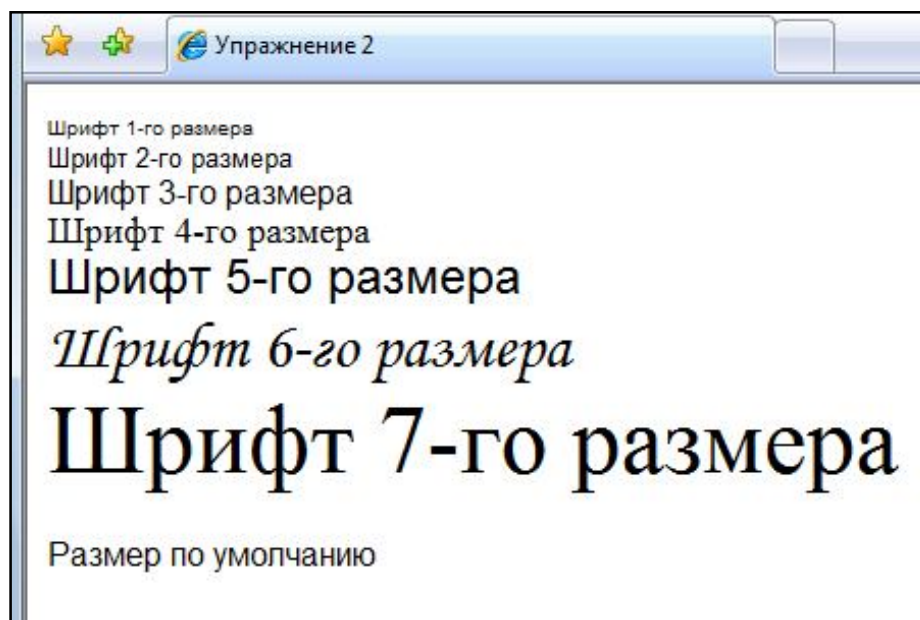


Рис. 2. Задание шрифта текста

ЗАПРЕТ РАЗРЫВА СТРОКИ

Иногда бывает нужно разорвать в определенных местах строки в тексте документа, а наоборот, не допустить разделения некоторых слов в строках. Для этого HTML позволяет использовать неразрывные пробелы и элементы NOBR.

Если текст, слова которого разделены неразрывными пробелами, не помещается в окне браузера, то появится горизонтальная полоса прокрутки.

Когда необходимо выделить большой участок текста, для которого недопустим автоматический перенос слов, то целесообразнее использовать элемент NOBR.

Ему соответствуют парные теги `<NOBR>` и `</NOBR>`. Весь текст, находящийся между этими тегами, будет отображаться браузером в одной строке.

Остерегайтесь создания слишком длинных неразрывных строк, поскольку необходимость горизонтальной прокрутки для прочтения таких строк только ухудшает чтение HTML-документа. Для вставки разрывов строк в тексте, заключенном между `<NOBR>` и `</NOBR>`, можно использовать рассмотренный ранее элемент `BR`.

Упражнение 3

1. Запустите программу Блокнот.
2. Наберите в окне редактора текст файла HTML:

```
<HTML>
```

```
  <HEAD>
```

```
    <TITLE>Упражнение 3</TITLE>
```

```
  </HEAD>
```

```
  <BODY>
```

```
    <Font Size='6' Face='Arial'>
```

```
    <P ALIGN="left"> Абзац выровнен влево
```

```
    <P ALIGN="center"> Абзац выровнен по центру
```

```
    <P ALIGN="right"> Абзац выровнен вправо
```

```
    <P ALIGN="justify"> Абзац выровнен по ширине
```

```
    <P> <NOBR> Этот текст всегда должен оставаться в одной строке </NOBR>
```

```
  </Font>
```

```
  </BODY>
```

```
</HTML>
```

3. Сохраните файл под именем «Упражнение 3.html».
4. Загрузите свой документ.
5. На экране вы увидите результат своей работы, изображенный на рис. 3.

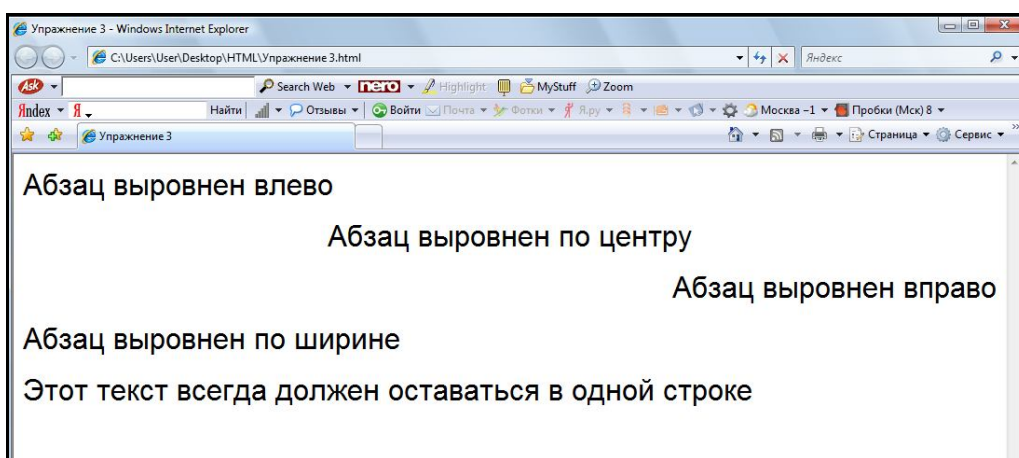


Рис. 3. Абзацы Web-страницы с установкой различных параметров выравнивания текста

ЗАГОЛОВКИ

Важным этапом в структурировании HTML-документа является использование заголовков (в их обычном понимании) для обозначения начала больших фрагментов текста.

В HTML поддерживаются шесть видов заголовков. Им соответствуют элементы H1, H2, H3, H4, H5, H6. Элементы H1 - H6 задаются при помощи соответствующих парных тэгов.

Упражнение 4

1. Запустите программу Блокнот.
2. Наберите в окне редактора текст файла HTML:

```
<HTML>
  <HEAD>
    <TITLE>Упражнение 4</TITLE>
  </HEAD>
  <H1> Так выглядит заголовок 1-го уровня </H1>
  <H2> Так выглядит заголовок 2-го уровня </H2>
  <H3> Так выглядит заголовок 3-го уровня </H3>
  <H4> Так выглядит заголовок 4-го уровня </H4>
  <H5> Так выглядит заголовок 5-го уровня </H5>
  <H6> Так выглядит заголовок 6-го уровня </H6>
</HTML>
```

3. Сохраните файл под именем «Упражнение 4.html».
4. Загрузите свой документ. На экране вы увидите результат своей работы, изображенный на рис. 4.

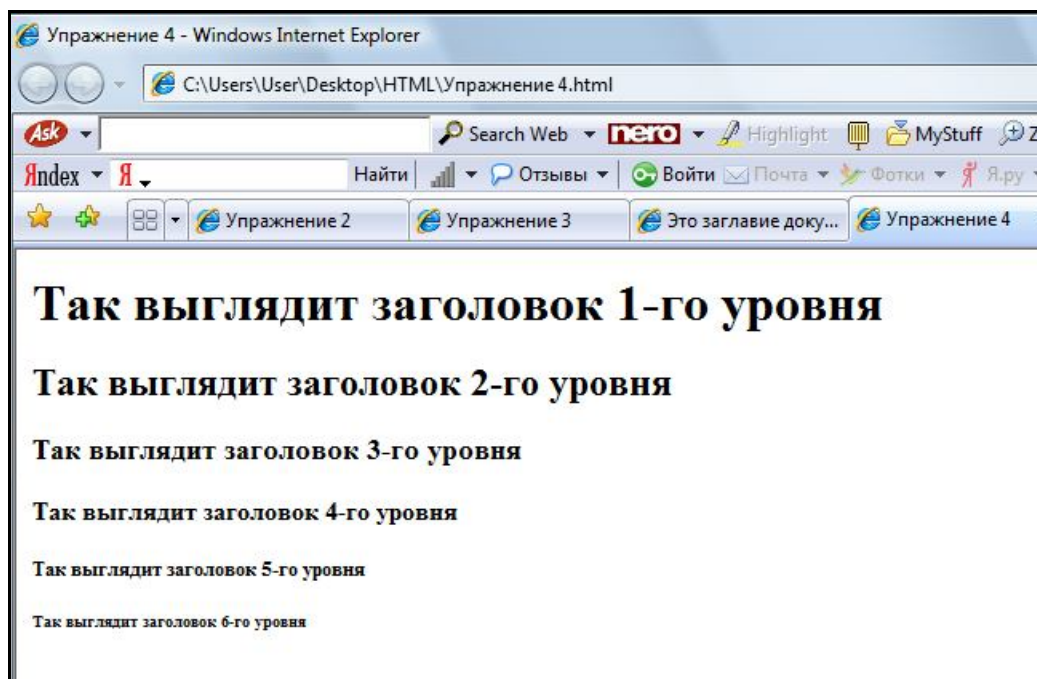


Рис. 4. Использование различных типов заголовков

СПЕЦИАЛЬНЫЕ СИМВОЛЫ

Для добавления в текст специальных символов используется числовой или именной код. В обоих случаях код символа начинается с символа амперсанда (&), за которым следует номер символа (числовой код) или сокращенное имя (именной код).

"(числовой код)" (именной код) - прямые двойные кавычки		
<	<	- меньше
>	>	- больше
±	±	- плюс-минус.

ЛАБОРАТОРНАЯ РАБОТА № 1

Цель работы: изучение основных концепций HTML, необходимых для создания простейших файлов HTML, содержащий форматированный текст

1. Запустите программу Блокнот.

2. Наберите в окне редактора простейший текст файла HTML:

```
<HTML>
  <HEAD>
    <TITLE>Учебный файл</TITLE>
  </HEAD>
  <BODY>
    Электронная почта
  </BODY>
</HTML>
```

3. Сохраните файл под именем «Учебный файл.html».

4. Загрузите свой документ. Убедитесь, что название Web – страницы («Учебный файл») отразилось в верхней, статусной, строке браузера.

5. На экране вы увидите результат своей работы, изображенный на рис. 5.

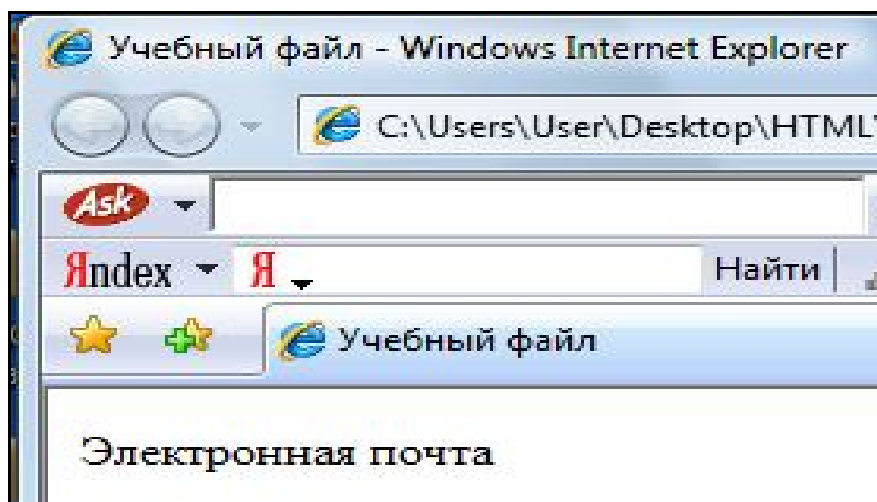


Рис.5. Создание простейшего HTML – документа

6. Откройте «Учебный файл» с помощью программы Блокнот.

7. Внесите изменения в текст файла HTML.

<HTML>

<HEAD>

<TITLE>Учебный файл</TITLE>

</HEAD>

<BODY>

<P>

Электронная
почта

</BODY>

</HTML>

8. Сохраните изменения и просмотрите полученный документ. Что изменилось?

9. В «Учебный файл» внесите изменения таким образом, чтобы в результате получился документ, приведенный на рис. 6.

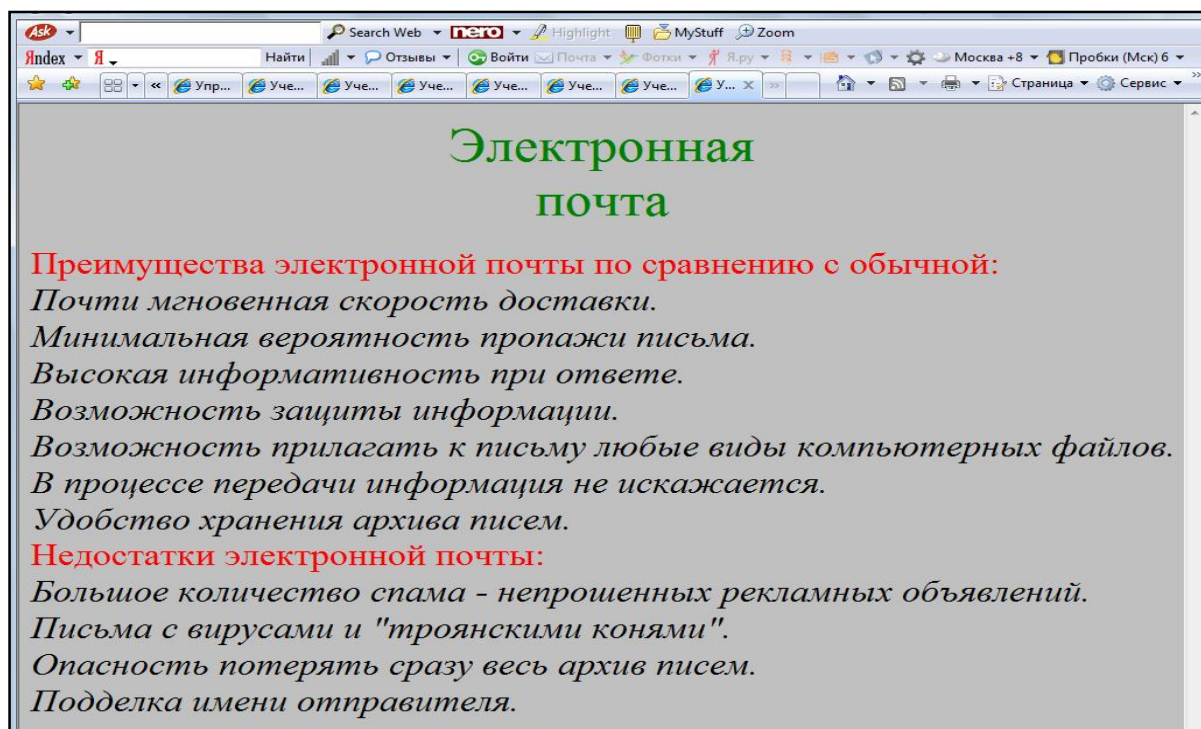


Рис. 6. Учебный файл

СПИСКИ

Нумерованные списки

Нумерованные списки применяются для упорядочения приводимых данных. При нумерации элементов таких списков могут быть использованы как арабские, так и римские цифры, символы латинского алфавита. Для создания нумерованного списка используется комбинация двух пар дескрипторов. Дескрипторы ... устанавливают начало и конец

нумерованного списка, а дескрипторы ... отмечают отдельные пункты списка. Тип нумерованного списка устанавливается с помощью атрибута TYPE дескриптора .

TYPE='1'	1,2,3,4 ... (задан по умолчанию)
TYPE='i'	i,ii,iii,iv,...
TYPE='I'	I,II,III,IV,...
TYPE='a'	a,b,c,d,...
TYPE='A'	A,B,C,D.

Упражнение 5

1. Запустите программу Блокнот.
2. Наберите в окне редактора текст файла HTML:
<HTML>

```
<HEAD>
<TITLE>Упражнение 5</TITLE>
</HEAD>
<BODY>
<H2>Типы списков </H2>
<OL>
<LI> Нумерованный </LI>
<LI> Маркированный</LI>
<LI> Многоуровневый</LI>
</OL>
</BODY>
```

</HTML>

3. Сохраните файл под именем «Упражнение 5.html».
4. Загрузите свой документ.
5. На экране вы увидите результат своей работы, изображенный на рис. 7.

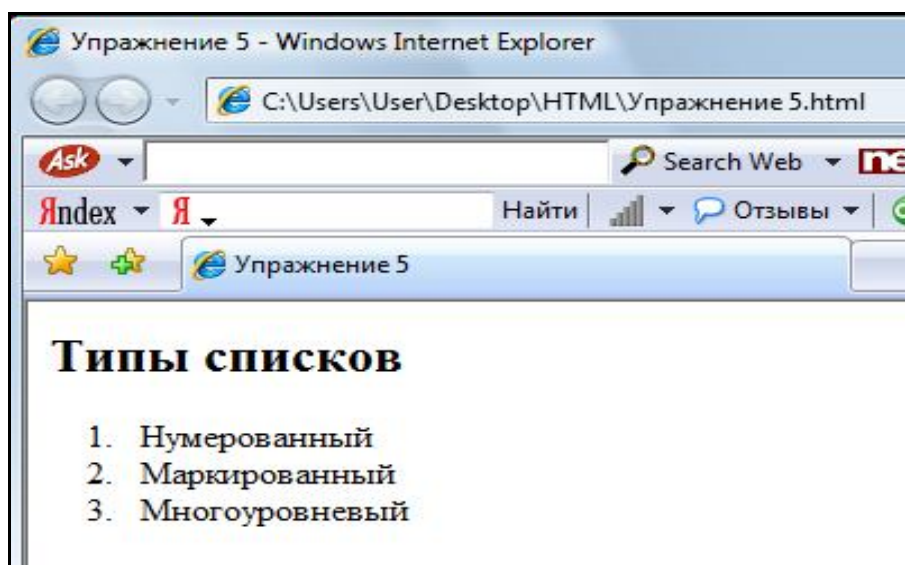


Рис.7. Нумерованный список

Маркированные списки

Маркированные списки применяются для перечисления неупорядоченной информации. Для создания маркированного списка используется комбинация двух пар дескрипторов. Дескрипторы `...` устанавливают начало и конец маркированного списка, а дескрипторы `...` отмечают отдельные пункты списка.

Элементы `` и `` имеют ряд атрибутов. Специфичными атрибутами элемента `` являются следующие:

compact – заставляет браузер показывать список более компактно;

type – позволяет задать тип маркера списка, может принимать значения: `circle` (круг без заливки), `disc` (круг с заливкой) и `square` (квадрат).

Упражнение 6

1. Запустите программу Блокнот.
2. Наберите в окне редактора текст файла HTML:

```
<HTML>
  <HEAD>
    <TITLE>Упражнение 6</TITLE>
  </HEAD>
  <BODY>
    <H2>Маркированный список </H2>
    <UL TYPE='circle'>
      <LI>Яблоко </LI>
      <LI TYPE='disc'> Груша</LI>
      <LI> Слива</LI>
      <LI TYPE='square'> Вишня </LI>
      <LI> Черешня</LI>
    </UL>
  </BODY>
</HTML>
```

3. Сохраните файл под именем «Упражнение 6.html».
4. Загрузите свой документ. На экране вы увидите результат своей работы, изображенный на рис. 8.

Многоуровневый

Важной особенностью списков HTML является та простота, с которой можно создавать списки различной вложенности, используемые для очень подробной и разветвленной классификации. При вложении можно использовать списки различных типов. При создании многоуровневого (вложенного) списка нужно добавить новый список после пункта, требующего детализации, внутри другого списка.

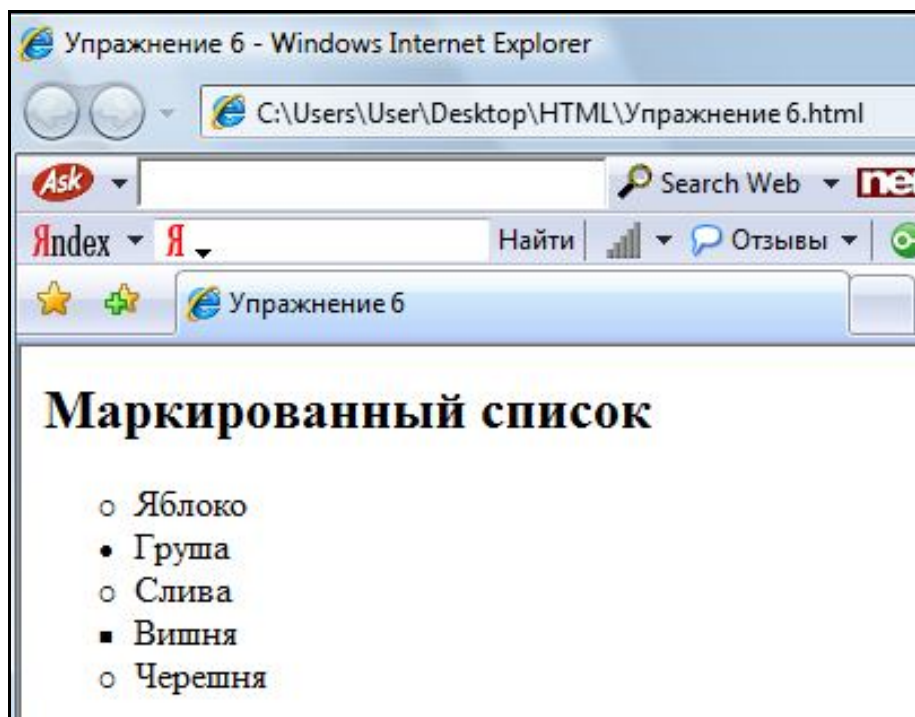


Рис.8. Маркированный список

Упражнение 7

1. Запустите программу Блокнот.
2. Наберите в окне редактора текст файла HTML:

```
<HTML>
  <HEAD>
    <TITLE>Упражнение 7</TITLE>
  </HEAD>
<BODY>
  <H2>Многоуровневый список</H2>
  <OL>
    <LI>Нумерованный</LI>
    <OL>
      <LI>Пункт 1</LI>
      <LI>Пункт 2</LI>
      <LI>Пункт 3</LI>
    </OL>
    <LI>Маркированный</LI>
    <UL>
      <LI>Первый вариант</LI>
      <LI>Второй вариант</LI>
      <LI>Третий вариант</LI>
    </UL>
    <LI>Смешанный</LI>
  </OL>
```

```
<OL type='a'>
  <LI> Пункт a</LI>
  <LI> Пункт b</LI>
  <LI> Пункт c</LI>
</OL>
</OL>
</BODY>
</HTML>
```

3. Сохраните файл под именем «Упражнение 7.html».

4. Загрузите свой документ. На экране вы увидите результат своей работы, изображенный на рис. 9.

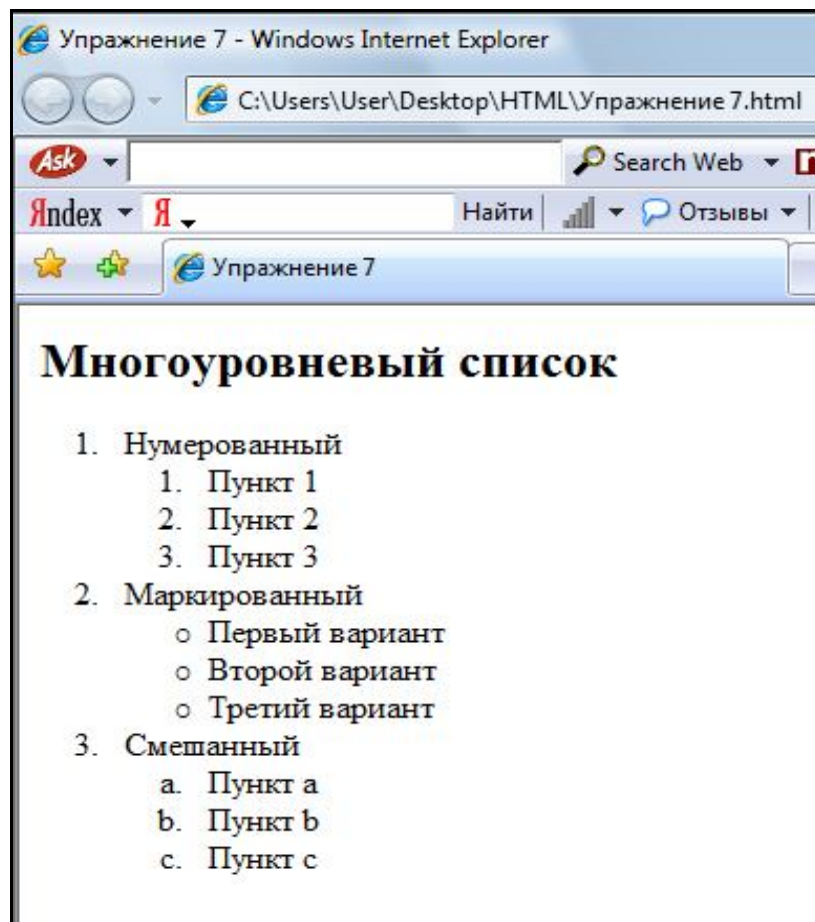


Рис. 9. Многоуровневый список