

ФЕДЕРАЛЬНОЕ АГЕНТСТВО ПО ОБРАЗОВАНИЮ  
ГОУ ВПО «УРАЛЬСКИЙ ГОСУДАРСТВЕННЫЙ ЛЕСОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»  
Кафедра информационных технологий и моделирования

Е.В. Кох

# ОСНОВЫ HTML

Методические указания  
по выполнению лабораторно-практического цикла  
для студентов очной и заочной формы обучения  
направления «Прикладная информатика»  
специальности 080801 «Прикладная информатика в экономике»;  
дисциплина «Информационные технологии».

Екатеринбург  
2010

Печатается по рекомендации методической комиссии ФЭУ  
Протокол № 25 от 25.09.09

Рецензент канд. техн. наук, доцент кафедры МиВЭДП Воронов М.П.

Редактор Е.А. Назаренко  
Компьютерная верстка Г.И. Романовой

---

Подписано в печать 16.08.10		Поз. 44
Плоская печать	Формат 60x84 1/16	Тираж 50 экз.
Заказ №	Печ. л. 2,56	Цена 13 руб. 52 коп.

---

Редакционно-издательский отдел УГЛТУ  
Отдел оперативной полиграфии УГЛТУ

## 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 вэб-странице.

*Название сервера (доменное имя)* - обозначает компьютер, содержащий данную вэб-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 вэб-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.

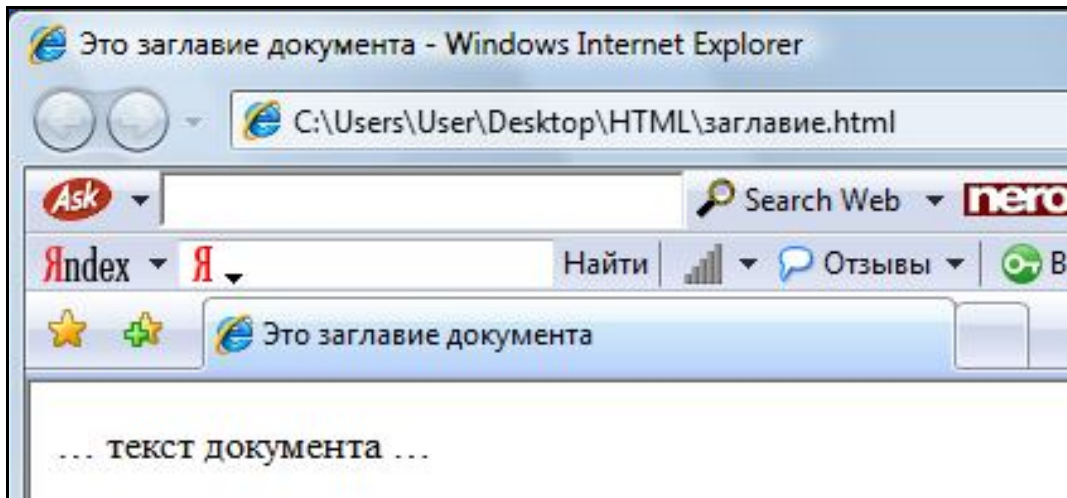


Рис. 1. Простейший HTML-документ

### 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 predetermined 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 `<BR>` 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 <FONT> и </FONT>.

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.

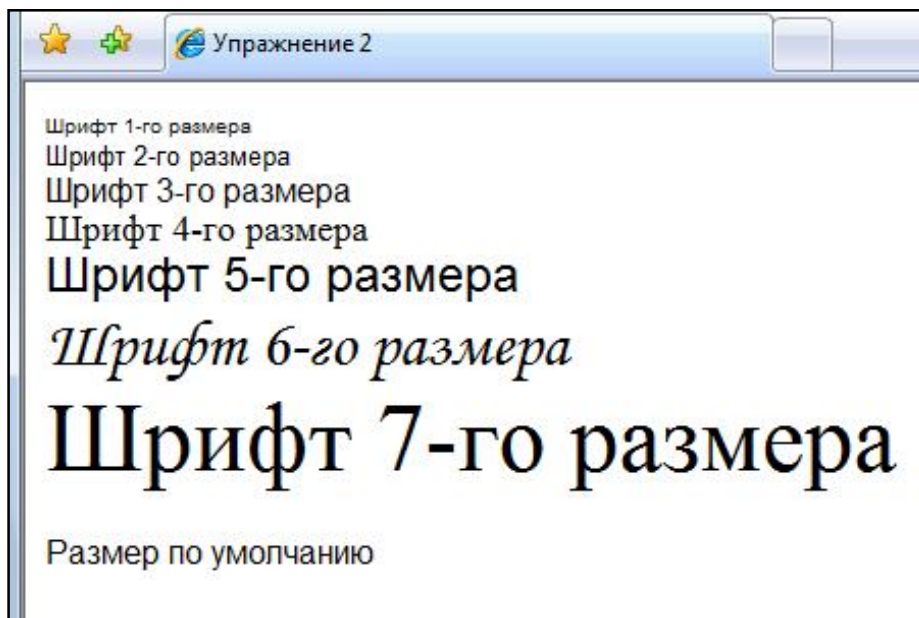


Рис. 2. Задание шрифта текста

## 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.

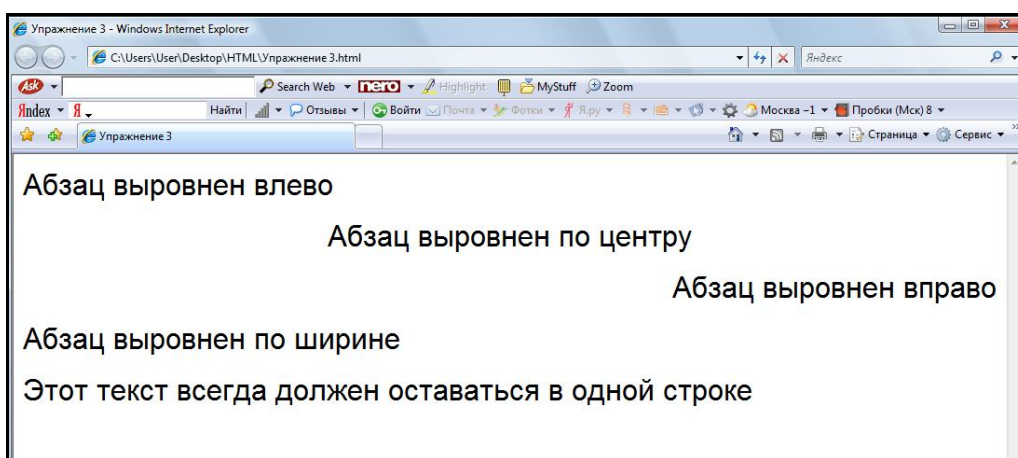


Рис. 3. Абзацы Web-страницы с установкой различных параметров выравнивания текста

## 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.

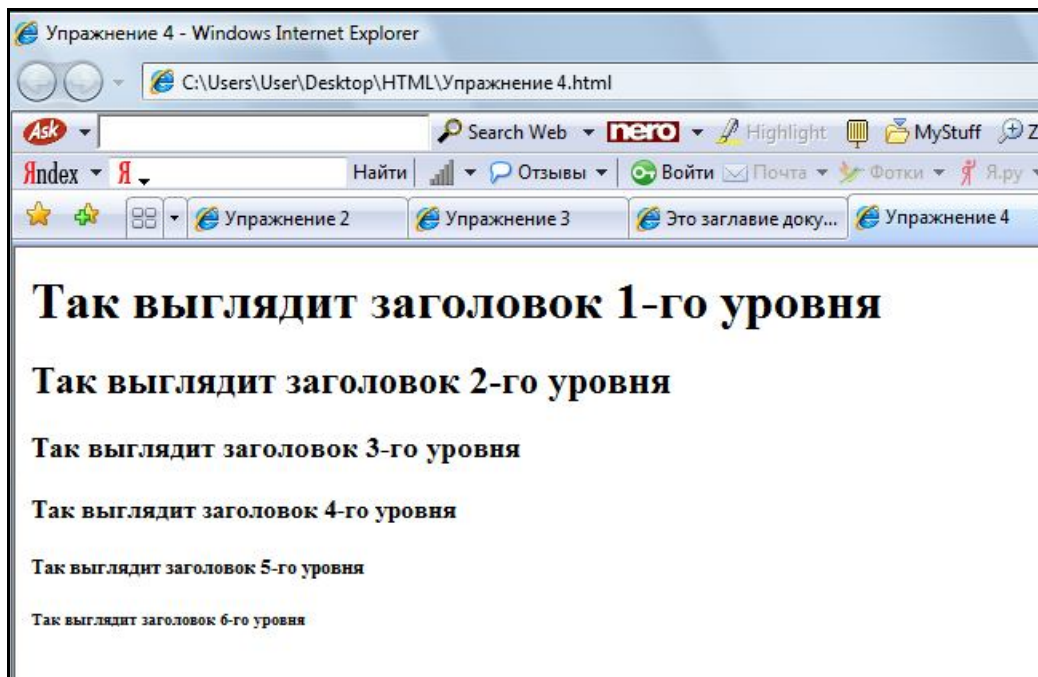


Рис. 4. Использование различных типов заголовков

## 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).

&#34(числовой код)&quot (именной код)	-	прямые двойные кавычки
&#60	&lt	- меньше
&#62	&gt	- больше
&#177	&plusmn	- 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.

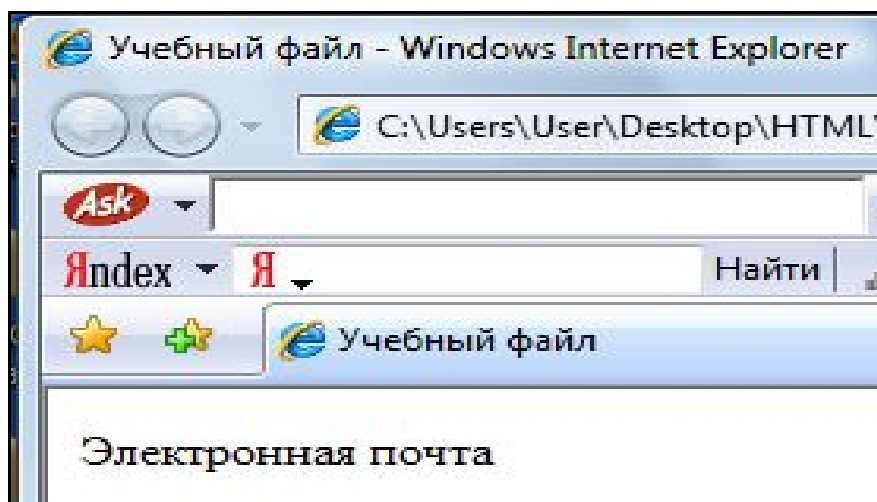


Рис.5. Создание простейшего HTML – документа

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 <BR>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.

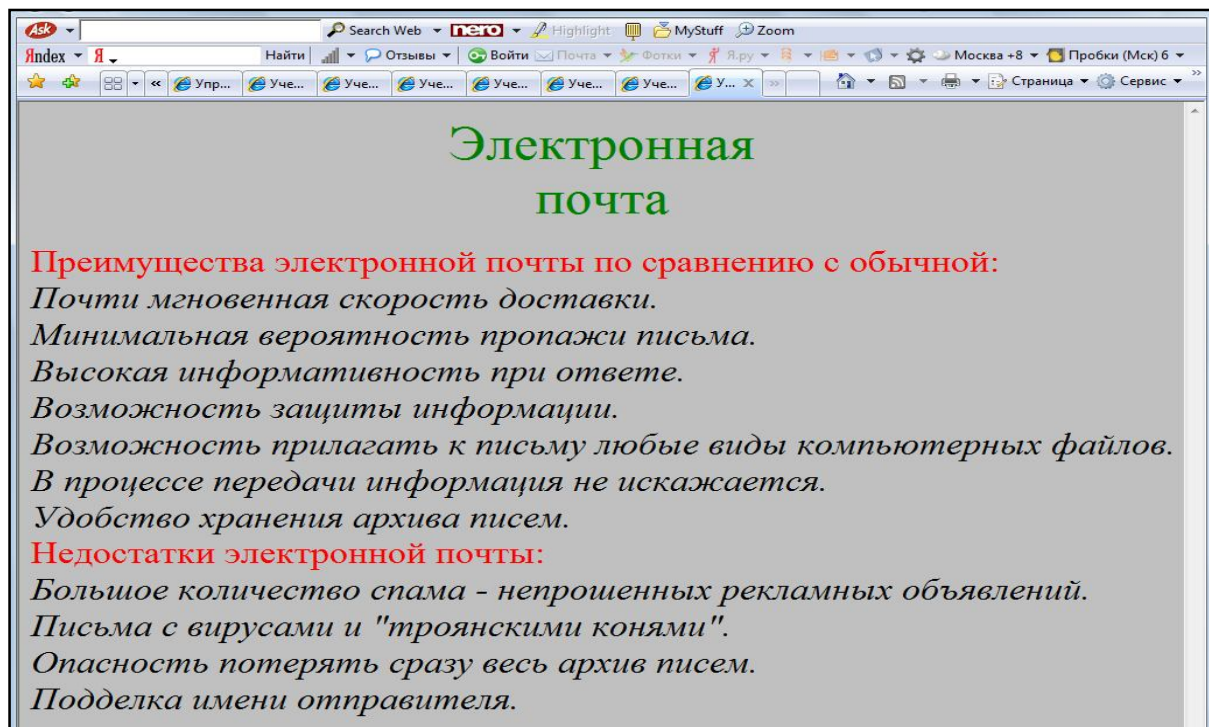


Рис. 6. Учебный файл

## 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 `<OL>...</OL>` устанавливают начало и конец

нумерованного списка, а дескрипторы <LI>...</LI> отмечают отдельные пункты списка. Тип нумерованного списка устанавливается с помощью атрибута TYPE дескриптора <OL>.

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.

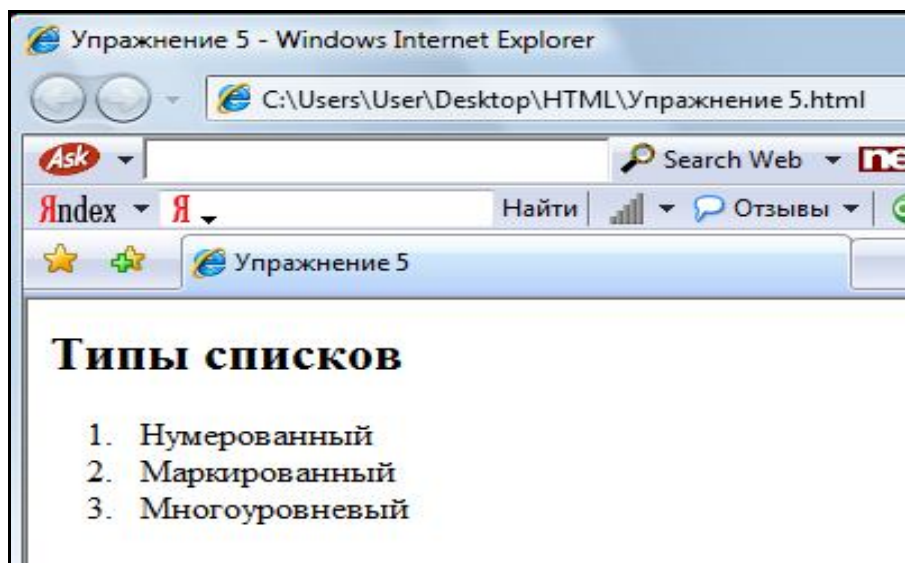


Рис.7. Нумерованный список

## Маркированные списки

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

Элементы `<UL>` и `<LI>` имеют ряд атрибутов. Специфичными атрибутами элемента `<UL>` 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.



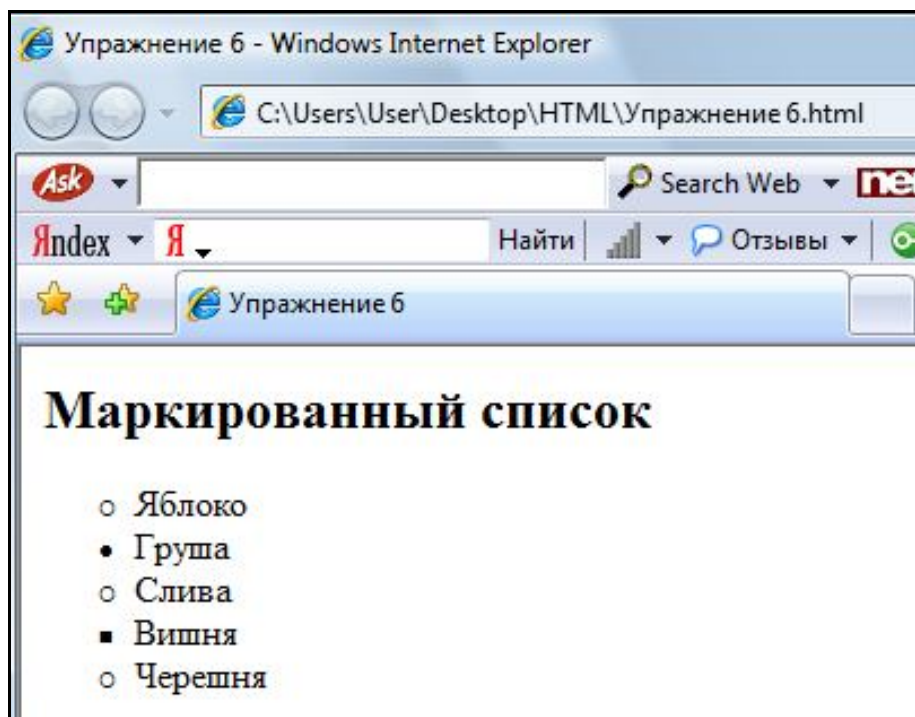


Рис.8. Маркированный список

### 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>
</BODY>
```

```
<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.

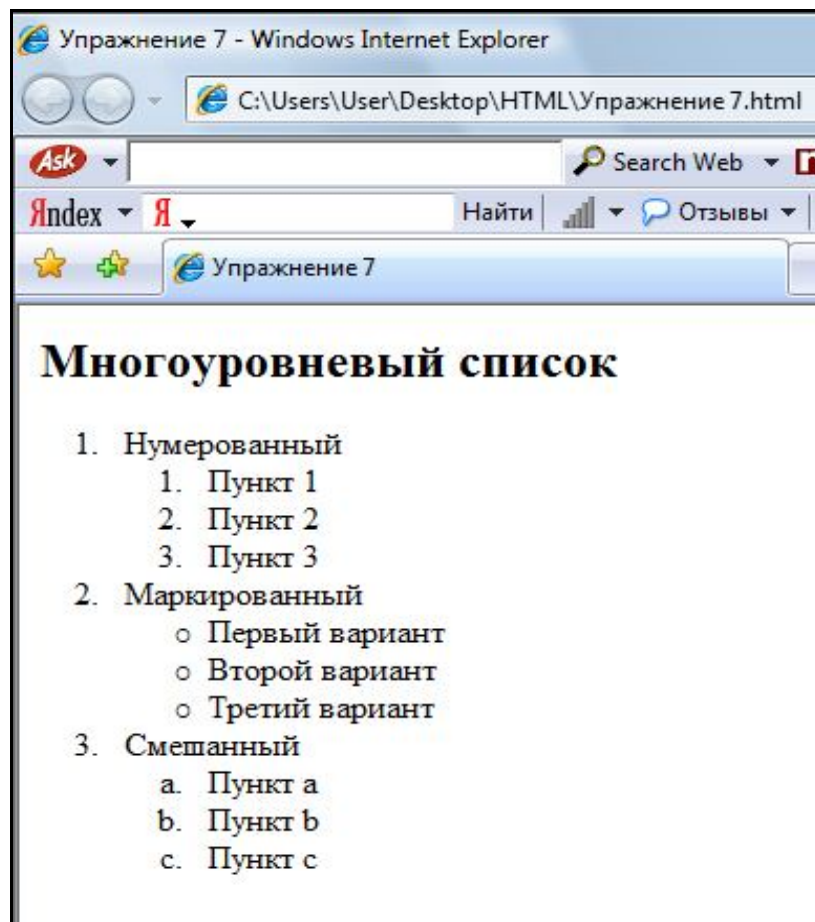


Рис. 9. Многоуровневый список

## ЛАБОРАТОРНАЯ РАБОТА № 2

Цель работы: использование списков.

1. В своей папке HTML сделайте копию файла Учебный файл.
2. Переименуйте копию в Списки.
3. Откройте файл Списки.

4. Внесите изменения в текст файла HTML, таким образом, в результате которых должен получиться документ, приведенный на рис. 10.

## ГИПЕРССЫЛКИ

Гиперссылки – ключевой, практически самый главный элемент гипертекста (текста, обладающего возможностями навигации). Гиперссылки позволяют быстро переходить к другим Web-страницам. Гиперссылки исключают необходимость ввода URL необходимой страницы, к которой обращается пользователь. Гиперссылки, как правило, выделяются цветом и подчеркиванием. В качестве гиперссылки может выступать и графическое изображение, в таком случае оно обычно выделяется рамкой.

### Простой переход к ресурсам

Для создания гиперссылки в тексте HTML-документа используется элемент **A**, который задается при помощи парных тегов `<A>` и `</A>`. Текст, изображение или другой элемент HTML-документа, заключенный между этими тегами, становится представлением гиперссылки в тексте.

### Навигация между HTML-документами

Для создания простейшей гиперссылки, обеспечивающей переход к нужному HTML-документу, достаточно задать в качестве значения атрибута `href` элемента **A** URI нужного ресурса (файла HTML-документа). Допустим, что сайт, по которому нужно реализовать навигацию, состоит из пяти страниц. Файлы имеют свои имена. У сайта есть главная страница (`index.html`), на которой помещено оглавление. Элементы оглавления позволяют перейти к соответствующим страницам, то есть являются гиперссылками.

#### *Упражнение 8*

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

```
<HTML>
  <HEAD>
    <TITLE>Главная страница сайта</TITLE>
  </HEAD>
<BODY>
  <H2>Оглавление</H2>
```

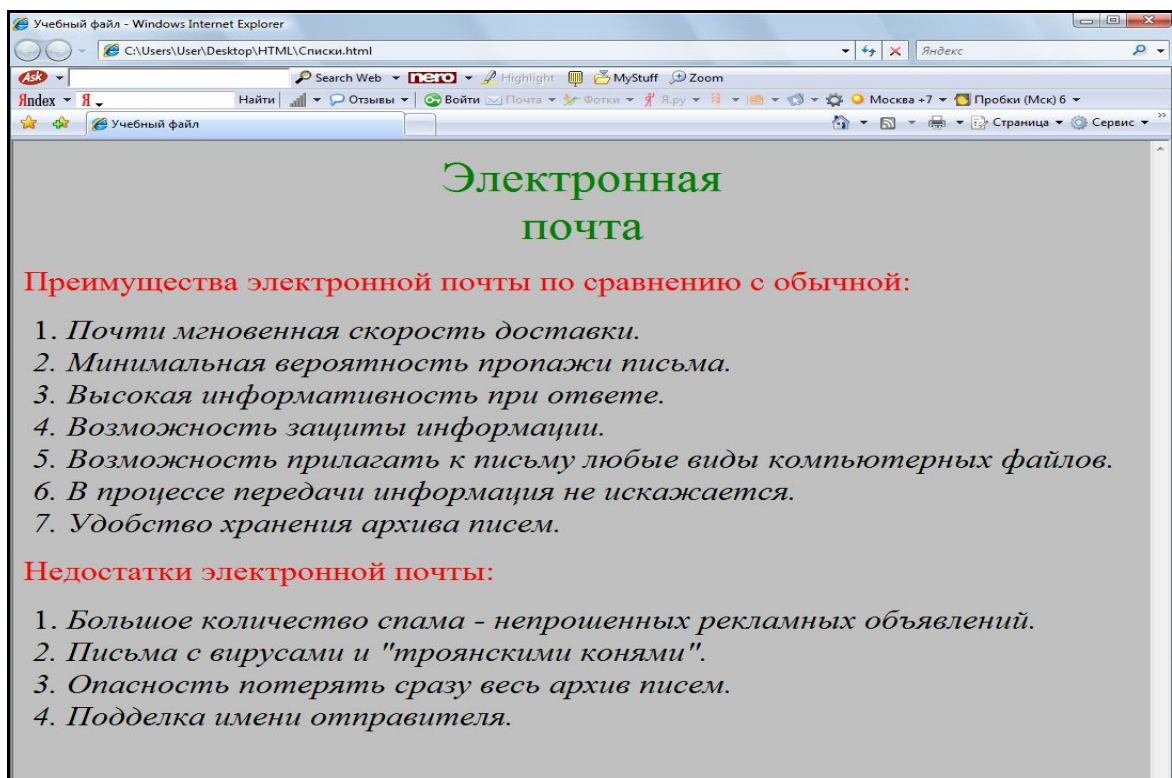


Рис.10. Списки

```

<A href='Упражнение 1.html'>Упражнение 1 </A><BR>
<A href='Упражнение 2.html'>Упражнение 2</A><BR>
<A href='Упражнение 3.html'>Упражнение 3</A><BR>
<A href='Упражнение 4.html'>Упражнение 4</A><BR>
<A href='Упражнение 5.html'>Упражнение 5 </A><BR>
<A href='Упражнение 6.html'>Упражнение 6 </A><BR><A href
='Упражнение 7.html'>Упражнение 7</A><BR>
</BODY>
</HTML>

```

3. Сохраните файл под именем «Оглавление.html».
4. Загрузите свой документ. На экране вы увидите результат своей работы, изображенный на рис. 11.

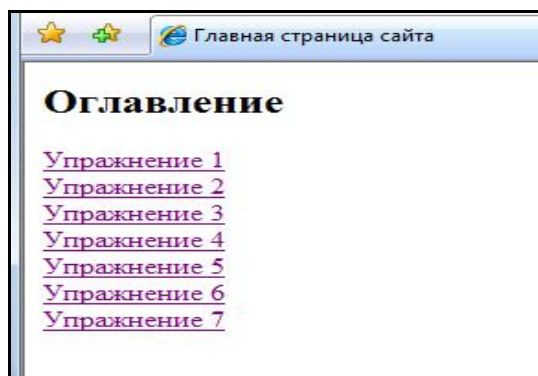


Рис.11. Использование гиперссылок

## ГРАФИКА НА WEB-СТРАНИЦАХ

Для вставки изображения в текст HTML-документа используется элемент. Для задания HTML-документа применяется одиночный тэг `<IMG >`. Список наиболее используемых атрибутов:

- атрибуту `src` присваивается путь к файлу на диске или URL – адрес файла рисунка в Интернете;
- `alt` – альтернативный текст, который отображается на месте изображения, если по каким-либо причинам само изображение не может быть показано;
- `border` – задает толщину границы вокруг рисунка в пикселях;
- `align` – определяет выравнивание изображения;
- `height` – задает высоту изображения в пикселях;
- `width` – определяет ширину изображения в пикселях;
- `vspace` – задает величину свободного пространства между изображением и текстом сверху и снизу;
- `hspace` – задает величину свободного пространства между изображением и текстом справа и слева;
- `name` – позволяет идентифицировать изображение так, что на него могут ссылаться различные сценарии.

### *Упражнение 9*

1. Откройте программу Paint.
2. Создайте простой рисунок (например, опознавательный знак транспортного средства).
3. Установите необходимые размеры для рисунка, сохраните его в своей папке HTML под именем **рисунок 1.jpeg**.
4. Откройте программу Блокнот.
5. Наберите в окне редактора текст файла HTML:

```
<HTML>
  <HEAD>
    <TITLE>Вставка изображений</TITLE>
  </HEAD>
  <BODY>
    <IMG src = 'рисунок1.jpeg' border=2> - небольшое изображение с рамкой
  </BODY>
</HTML>
```

6. Сохраните файл под именем «Упражнение 9.html».
7. Загрузите свой документ.
8. На экране вы увидите результат своей работы, изображенный на рис. 12.



Рис.12. Web – страница с рисунком

### Линейки

Линейка – горизонтальная линия в окне браузера. Для вставки линейки в HTML-документ используется элемент `HR`. Этот HTML – документ задается при помощи одинарного тэга `<HR>` и имеет следующие атрибуты:

- **align** – задает выравнивание линейки в окне браузера, может принимать значения **left**, **right** или **center**;
- **noshade** – булев атрибут, указывает браузеру, что линейку следует отображать плоской (без традиционной тени);
- **size** – численное значение, определяющее толщину линии;
- **width** – численное значение, определяющее ширину линейки; может задаваться как абсолютное (в пикселях) значение, так и относительное (в процентах от ширины окна браузера).

#### Упражнение 10

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

```
<HTML>
  <HEAD>
    <TITLE>Линейки</TITLE>
  </HEAD>
<BODY>
  Линейка с настройками по умолчанию
  <HR>
  Линейки различной толщины
  <HR size=4>
  <HR size=10>
```

Плоская линейка

```
<HR noshade size=10>
```

Линейки с различным выравниванием

```
<HR width =”50%” align=left>
```

```
<HR width =”50%” >
```

```
<HR width =”50%” align=right>
```

```
</BODY>
```

```
</HTML>
```

3. Сохраните файл под именем «Упражнение 10.html».

4. Загрузите свой документ.

5. На экране вы увидите результат своей работы, изображенный на рис. 13.

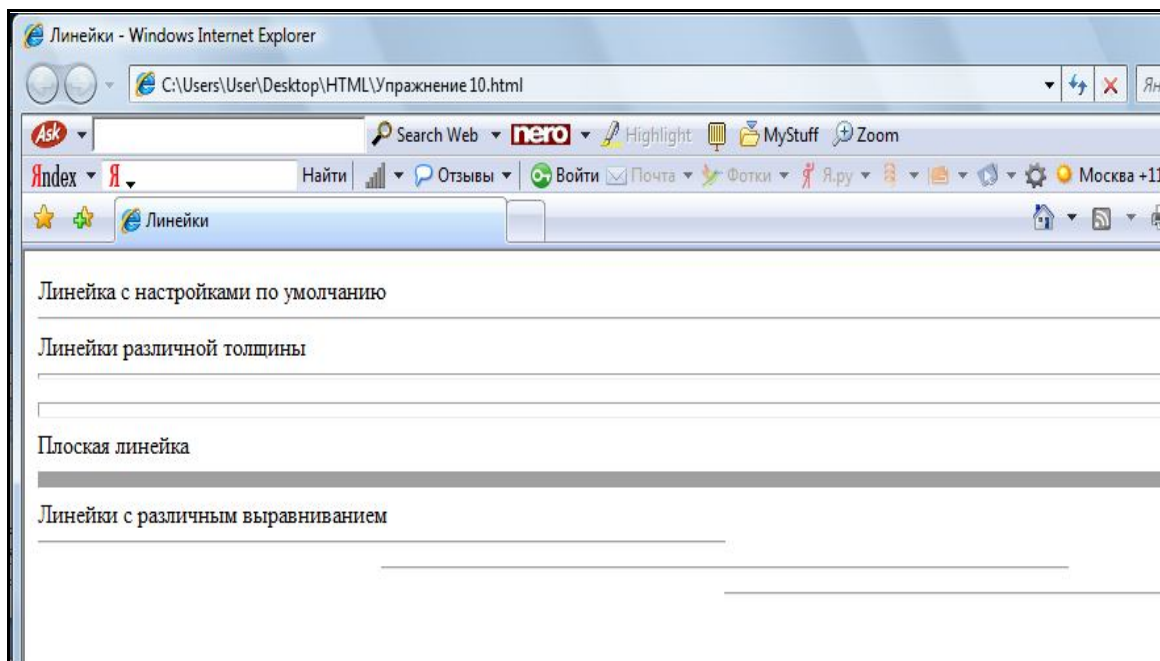


Рис.13. Использование линеек

## Изображения-гиперссылки

Изображения, особенно небольшие пиктограммы и значки, часто используются в качестве графического представления гиперссылок. Создание изображения-гиперссылки выполняется очень просто: достаточно заключить HTML-элемент **IMG** внутрь элемента **A**.

### Упражнение 11

1. Запустите программу Блокнот.

2. Наберите в окне редактора текст файла HTML:

```
<HTML>
```

```
<HEAD>
```

```
<TITLE>Описание</TITLE>
```

```
</HEAD>
```

```
<BODY>
```

Это опознавательный знак транспортного средства. Устанавливается на транспортное средство, на котором проводится обучение вождению.

```
</BODY>
```

```
</HTML>
```

3. Сохраните документ под именем «Упражнение 11.html».

4. Скопируйте файл «Упражнение 9». Переименуйте копию в «Упражнение 11a.html». Внесите изменения в текст HTML-документа:

```
<HTML>
```

```
<HEAD>
```

```
<TITLE>Графическая ссылка</TITLE>
```

```
</HEAD>
```

```
<BODY>
```

```
<A href = 'Упражнение 11.html'><IMG src='рисунок 1.jpeg' border=2></A>
```

```
</BODY>
```

```
</HTML>
```

5. Сохраните документ под именем «Упражнение 11a.html».

6. Загрузите свой документ.

7. На экране вы увидите результат своей работы, изображенный на рис. 14.

8. Наведите курсор мыши на рисунок и используйте графическую ссылку.

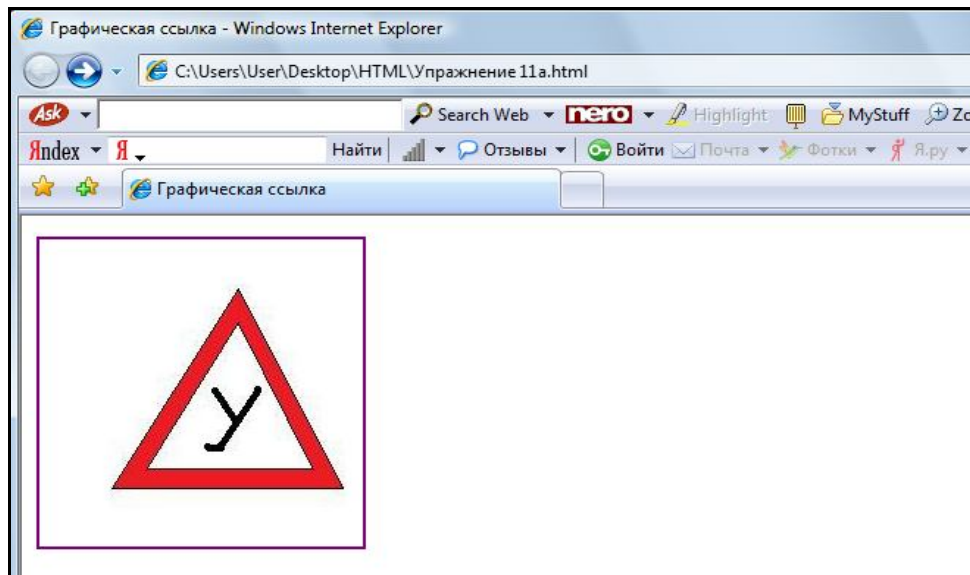


Рис.14. Графическая ссылка

## ТАБЛИЦЫ

Таблица создается с помощью дескрипторов `<TABLE>...</TABLE>`. Эти дескрипторы создают объект таблицы в том месте текста, где они до-



бавлены в коде HTML. Теперь в таблицу нужно добавить строки и столбцы. Для этого используются дескрипторы:

<TR>...</TR> - новая строка таблицы;

<TH>...</TH> - ячейка заголовка;

<TD>...</TD> - обычная ячейка таблицы.

### Заголовок таблицы

Заголовок таблицы задается парными тегами <CAPTION> </CAPTION> и помещается после тега <TABLE>. Для элемента CAPTION можно задать атрибут, определяющий положение заголовка относительно таблицы – align. Этот атрибут может принимать следующие значения:

– top – заголовок показывается сверху таблицы (используется по умолчанию);

– bottom – под таблицей;

– left – слева от таблицы;

– right – справа от таблицы.

### Параметры отображения таблицы

Наиболее часто используемые атрибуты HTML-элемента TABLE:

1) align – задает положение таблицы в окне браузера;

2) bgcolor – задает цвет фона таблицы;

3) border – задает толщину внешней границы таблицы;

4) bordercolor – цвет границы таблицы;

5) cellpadding – размер пустого пространства между границами и содержимым ячеек таблицы;

6) cellspacing – размер пустого пространства между ячейками таблицы;

7) frame – задает отображаемые части внешней рамки таблицы, может принимать одно из перечисленных значений:

– void – рамка не отображается (по умолчанию);

– above – отображается только верхняя граница;

– below – показывается только нижняя граница;

– hspace – отображается верхняя и нижняя границы;

– vspace – показывается правая и левая границы;

– lhs – отображается только левая граница

– rhs – отображается только правая граница;

– box – рамка отображается полностью;

– rules – задает, какие именно границы между ячейками должны отображаться, может принимать одно из следующих значений:

• none – границы между ячейками не отображаются;

• rows – отображаются только границы между строками;

• cols – отображаются только границы между столбцами;

• all – отображаются все границы между ячейками;

• height – задает рекомендуемую высоту таблицы;

• width – задает рекомендуемую ширину таблицы.

### Слияние ячеек

Для элемента TD существует два атрибута, которые используются для объединения нескольких ячеек при создании более сложных таблиц:

- **colspan** – задает количество столбцов для слияния;
- **rowspan** – определяет количество строк для слияния.

### Параметры отображения строк таблицы

Для настройки особого отображения отдельных строк таблицы используется элемент TR (этот элемент объединяет ячейки в строки таблицы).

## ЛАБОРАТОРНАЯ РАБОТА № 3

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

```
<HTML>
  <HEAD>
    <TITLE>Таблица с объединенными ячейками</TITLE>
  </HEAD>
  <TABLE align=center border=3 bordercolor=black>
    <CAPTION align=top><B>Доходы от продаж за второе полугодие
2007 года</B></CAPTION>
    <!--формирование первой строки шапки таблицы-->
    <TR align=center >
      <TD rowspan=2><B>Филиал\Период </B>
      <TD colspan=3><B>3квартал </B>
      <TD colspan=3><B>4квартал </B>
    <!--формирование второй строки шапки таблицы (название месяцев)-->
    <TR align=center >
      <TD><B>Июль</B><TD><B>Август</B><TD><B>Сентябрь
      </B><TD><B>Октябрь</B><TD><B>Ноябрь</B><TD><B>
      Декабрь</B>
    <!--далее следуют строки с данными (первая ячейка каждой строки –
название филиала) -->
    <TR align=right><TD align=left>Филиал1<TD>123123<TD>
      323233<TD>323453<TD>231423<TD>323212<TD>243673
    <TR align=right ><TD align=left > Филиал2<TD>223523<TD>
      225243<TD>314423<TD>212445<TD>373812<TD>274673
    <TR align=right ><TD align=left > Филиал3<TD>183123<TD>
      186834<TD>323453<TD>231423<TD>323212<TD>243673
    <TR align=right ><TD align=left > Филиал4<TD>125163<TD>
      334343<TD>123553<TD>167423<TD>254412<TD>132367
    </TABLE>
</HTML>
```

3. Сохраните документ под именем `Таблицы.html`».
4. Загрузите свой документ.
5. На экране вы увидите результат своей работы, изображенный на рис. 15.

Филиал\Период	3квартал			4квартал		
	Июль	Август	Сентябрь	Октябрь	Ноябрь	Декабрь
Филиал1	123123	323233	323453	231423	323212	243673
Филиал2	223523	225243	314423	212445	373812	274673
Филиал3	183123	186834	323453	231423	323212	243673
Филиал4	125163	334343	123553	167423	254412	132367

Рис.15. Таблица с объединенными ячейками

## ФРЕЙМЫ

Фреймы позволяют разбить окно просмотра на несколько прямоугольных областей, располагающихся рядом друг с другом. В каждую из областей можно загрузить отдельный HTML-документ, просмотр которого осуществляется отдельно от других документов. Между фреймами, также как и между отдельными окнами браузера, можно организовать взаимодействие, которое заключается в том, что выбор ссылки в одном из фреймов может привести к загрузке нужного документа в другой фрейм или окно браузера.

При использовании фреймов, файл, описывающий структуру фреймов (обычно это - `index.html`) не может содержать в себе никакой другой информации, кроме этого описания. Вместо раздела `BODY` такой файл содержит раздел `FRAMESET`. Внутри тэга-контейнера `<FRAMESET>... <FRAMESET>` могут содержаться только вложенные тэги `FRAMESET` и тэги `FRAME`.

`FRAMESET` используется для разбиения окна браузера или окна родительского фрейма на отдельные фреймы. При разбиении окна на фреймы используются следующие атрибуты элемента `FRAMESET`:

- `cols` – список значений ширины создаваемых фреймов (могут задаваться как абсолютные значения в пикселях, так и процентные);
- `rows` – список значений высоты создаваемых фреймов.

Между тэгами <FRAMESET> </FRAMESET> должны содержаться либо описания каждого фрейма с использованием HTML-элемента FRAME. Этот HTML-элемент задается одиночным тэгом <FRAME>. Наиболее часто используемые атрибуты HTML-элемента <FRAME>:

- **name** – задает имя фрейма, позволяя использовать его в качестве целевого фрейма в гиперссылках;
- **src** – URI документа, содержимое которого отображается в окне фрейма;
- **noresize** – булев атрибут, блокирует возможность изменения размера окна фрейма;
- **scrolling** – позволяет показывать или убирать полосы прокрутки в окне фрейма, может принимать значение **yes** (показать полосы прокрутки), **no** (не отображать) или **auto** (используется по умолчанию: показать полосы прокрутки только в том случае, если содержимое не умещается в окне фрейма);
- **frameborder** – задает, должен ли браузер показывать рамку окна фрейма, может принимать значение 1 (рамка показывается) или 0 (рамка не показывается);
- **marginwidth** – задает расстояние между границами фрейма и его содержимым по горизонтали (в пикселях);
- **marginheight** – задает расстояние между границами фрейма и его содержимым по вертикали.

### Целевой фрейм отдельных гиперссылок

Для указания целевого фрейма каждой гиперссылки, то есть фрейма, в котором будет показано содержимое открываемого по ссылке HTML-документа, используется атрибут **target**. Значением этого атрибута может быть либо имя нужного фрейма (значений атрибута **name** этого фрейма), либо одного из следующих predefined значений:

- \_blank** – открыть документ в новом окне;
- \_self** – открыть документ в том же окне, где находится гиперссылка;
- \_parent** – открыть документ в окне родительского фрейма (аналогично **\_self**, если фрейм, в котором находится гиперссылка, не имеет родителя);
- \_top** – заменить содержимое окна браузера на содержимое открываемого документа, отменяя все созданные ранее фреймы.

Атрибут **target** поддерживается для всех HTML-элементов, позволяющих создавать гиперссылки: **A**, **LINK**, **AREA**. Пример:

**FRAMESET ROWS="15%, 70%, 15%"** </FRAMESET> - создается фреймовая структура (по умолчанию с сеткой шириной 5 пикселей) из трех строк: первая строка 15 % высоты окна браузера, вторая 70 % и третья 15 %;

**<FRAMESET COLS="\*, 3\*" BORDER=3 BORDERCOLOR=GRAY> </FRAMESET>** - создается фреймовая структура из двух столбцов: первый  $\frac{1}{4}$  ширины окна браузера, второй  $\frac{3}{4}$  ширины, с сеткой шириной 3 пикселя, цвет сетки - серый;

**<FRAMESET ROWS="50%, 50%" COLS="50%, 50%" FRAMEBORDER=NO> </FRAMESET>** - создается фреймовая структура без сетки из двух столбцов и двух строк, делящих окно браузера на 4 равные части.

### *Упражнение 12*

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

```
<HTML>
  <HEAD>
    <TITLE>Сайт с фреймами</TITLE>
  </HEAD>
  <FRAMESET cols="200,*">
    <FRAMESET rows="70%,*">
      <FRAME SRC="Страница 1.html">
      <FRAME SRC="Страница 2.html">
    </FRAMESET>
    <FRAME SRC="Страница 3.html">
  </FRAMESET>
</HTML>
```

3. Сохраните документ под именем `index.html`.
4. Создайте самостоятельно три простых HTML-документа, под именами Страница 1, Страница 2, Страница 3. Первый документ должен содержать текст: «Содержимое первой страницы» соответственно два следующих: «Содержимое второй страницы», «Содержимое третьей страницы».
5. Загрузите установочный файл (`index`).
6. На экране вы увидите результат своей работы, изображенный на рис. 16.
7. Изучите внимательно текст HTML-документ (`index`). Здесь создается внешний набор из двух фреймов, занимающих всю высоту окна браузера. При этом в левом фрейме внешнего набора создается еще один набор фреймов. Нужно заметить, что блок **<FRAMESET> </FRAMESET>** аналогичен одному элементу **FRAME** (описывает содержимое одного фрейма из внешнего набора). Поэтому, анализируя приведенный пример, можно однозначно сказать, что разбивается левый фрейм внешнего набора, а в правый фрейм загружается содержимое документа Страница 3.html.

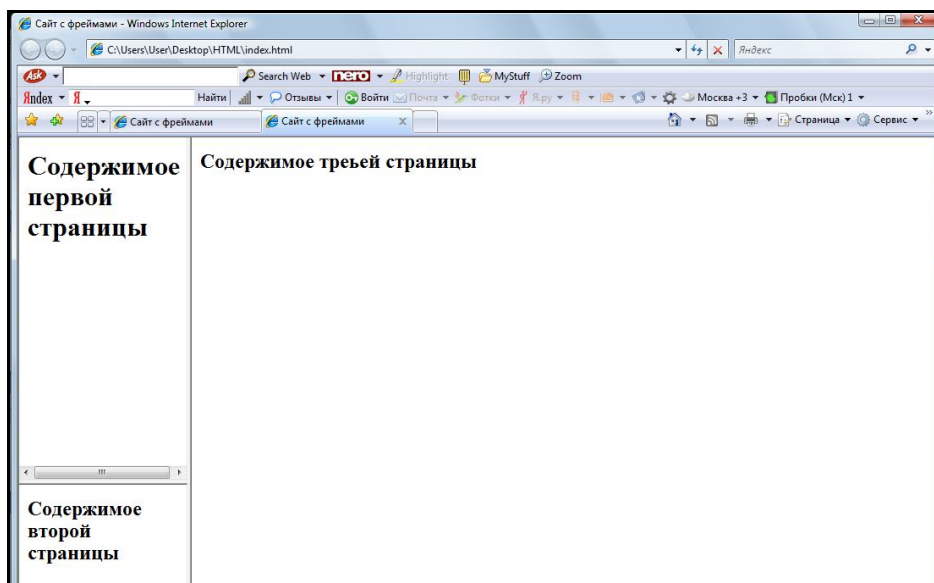


Рис.16. Сайт с фреймами

## ЛАБОРАТОРНАЯ РАБОТА № 4

1. Запустите программу Paint.
2. Создайте рисунок, который вы в дальнейшем будете использовать в качестве фонового изображения.
3. Сохраните этот рисунок под именем «Фон.gif».
4. Создайте домашнюю страничку, состоящую из нескольких фреймов (структуру фреймов выбрать самостоятельно).
5. Оформите, Web-страницу используя полученные знания.
6. Пример создания домашней странички приведен на рис. 17.
7. Переход по ссылке «Моё хобби» на рис. 18.

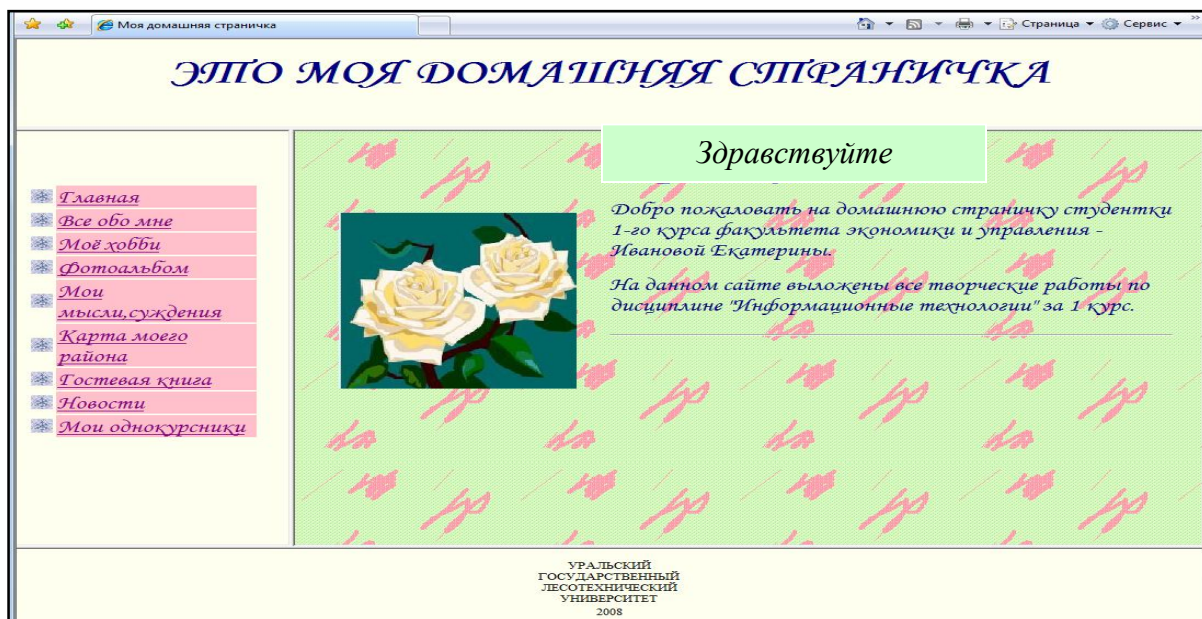


Рис.17. Пример оформления домашней странички



Рис. 18. Переход по ссылке «Моё хобби»

## Элемент NOFRAMES

Фреймы являются весьма привлекательным средством оформления сайтов. Однако фреймы появились не сразу. У достаточно большого количества пользователей до сих пор могут стоять старые версии браузеров, НЕ поддерживающие фреймов вообще.

Если важно, чтобы при просмотре сайта, использующего фреймы, пользователь хоть что-то увидел, то следует применять HTML-элемент NOFRAMES. Этот элемент задается парными тегами <NOFRAMES> </NOFRAMES>. Он помещается в один файл с описанием набора фреймов. Между этими тегами помещается текст, который браузер показывает в том случае, если он не поддерживает фреймы (или если пользователь отключил в браузере поддержку фреймов).

## ФОРМЫ

Назначение форм состоит в том, чтобы дать возможность посетителю управлять содержимым Web-страницы, вызывать на выполнение сценарии, а также вводить собственные данные и отправлять их на Web-сервер или по адресу электронной почты.

### Создание формы

Для вставки формы в HTML-документ используется элемент FORM. Он задается парными тегами < FORM > </ FORM >. Между этими тегами помещаются описания элементов управления формы. При создании формы используются следующие атрибуты:

- **action** – обязательный для каждой формы параметр, URI программы-обработчика данных формы;
- **method** – задает способ отправки данных, введенных в форму, может принимать значения **get** (используется по умолчанию) или **post**;
- **enctype** – задает тип данных формы, если используется метод отправки **post**; по умолчанию имеет значение **application/x-www-form-urlencoded**; при необходимости передачи файлов используется значение **multipart/form-data**;
- **accept-charset** – применяется при передаче файлов, позволяет указать, какие кодировки используются для каждого из файлов (список строковых значений – названий кодировок), по умолчанию устанавливается значение **UNKNOWN** (приложение на сервере должно само определять кодировку);
- **accept** – описывает типы файлов (**MIME** – типы), передаваемые серверу; если этот параметр не использовать, то серверное приложение должно уметь само определять типы передаваемых ему файлов.

### Стандартные элементы управления

Элементы управления, которые используются чаще всего – однострочное текстовое поле, поле для ввода пароля, флажки, переключатели, кнопки (как пользовательские, так и выполняющие стандартные действия), поля имен файлов. Все упомянутые элементы управления отображаются браузером **Internet Explorer**. Для обозначения всех этих элементов управления используется один **HTML**-элемент – **INPUT**. Этот элемент задается одиночным тэгом **<INPUT>** и имеет следующие атрибуты:

- **type** – принимает строку, задающую тип элемента управления (по умолчанию используется строка **text** и создается соответственно поле для ввода текста. Значения атрибута **type** приведены в табл. 3.

Таблица 3

Значения атрибута **type**

Значение атрибута	Особенности
text	Создается однострочное текстовое поле. Значение атрибута <b>value</b> отображается в качестве текста по умолчанию. Атрибут <b>size</b> воспринимается как количество символов, которое может отображаться в поле без необходимости горизонтальной прокрутки текста. Атрибут <b>maxlength</b> задает максимальное количество символов, которые могут быть введены в поле
password	Действие этого значения аналогично действию значения <b>text</b> , но дополнительно скрывается вводимый пользователем текст (заменяется символами «*» или другими). Используется для ввода конфиденциальной информации типа паролей



Значение атрибута	Особенности
checkbox	Создает элемент управления флажок. Задавать значение атрибута value нужно обязательно, так как именно это значение отправляется серверу, если флажок установлен. Можно использовать несколько элементов управления checkbox с одинаковым значением атрибута name для обеспечения возможности задания нескольких значений одного свойства
radio	Создает переключатель. Имеет смысл использовать как минимум два этих элемента управления вместе, объединенных в группу, с заданием одного и того же значения атрибута name. Тогда у пользователя появляется возможность выбора одного из нескольких вариантов (одновременно можно установить только один переключатель в группе). Значение атрибута value задавать также обязательно, оно используется аналогично элементу управления checkbox
file	Создает поле для ввода имени файла с возможностью выбора файла с помощью диалогового окна открытия файла или без него, что зависит от браузера. Содержимое выбранного файла или файлов пересылается вместе с формой. Для корректной работы необходима установка значения атрибута method формы в post, а enctype — в multipart/form-data. Значение атрибута value используется как имя файла по умолчанию. Значение атрибута size задается и работает аналогично элементу управления text
submit	Создает кнопку, при нажатии которой браузер отправит форму. Значение атрибута value задает надпись на кнопке
image	Создает изображение, при щелчке на котором произойдет отправка формы браузером. При этом на сервер передаются также координаты указателя мыши относительно левого верхнего угла изображения. Атрибут src задает расположение используемого изображения
reset	Создает кнопку, при нажатии которой значения всех элементов управления будут заменены значениями по умолчанию. Значение value задает надпись на кнопке
button	Создает пользовательскую кнопку. При нажатии или другом действии с кнопкой может выполняться ассоциированный сценарий. Значение атрибута value используется как надпись на кнопке. Подробно создание сценариев, ассоциирование их с элементами управления и не только будет рассмотрено далее
hidden	Создает скрытый элемент управления. Для него задаются атрибуты name и value. Пользователь не видит и не может изменить содержимое этого элемента управления. Однако данные скрытого элемента управления отправляются серверу вместе с остальными данными формы

- **name** – используется для задания имени элементу управления (строка, которая помимо идентификации элемента управления добавляется в данные, отсылаемые серверу);
- **checked** – булев атрибут, если он установлен, то флажок или переключатель считается (и отображается браузером);
- **size** – задает размер элемента управления (единицы измерения и действие специфичны для разных элементов управления);
- **value** – начальное значение для полей ввода текста и полей для указания имен файлов, также используется как надпись таких элементов управления, как кнопки; необязательно для всех элементов управления, кроме флажков и переключателей;
- **disabled** – булев атрибут, установка которого не позволяет пользователю работать с элементом управления;
- **readonly** – булев атрибут, позволяет запретить изменение состояния элемента управления (работает только для текстовых полей и поля выбора файла, так что для остальных элементов управления лучше использовать атрибут **disabled**, однако при этом данные деактивированных элементов управления не отправляются серверу);
- **maxlength** – задает максимальную длину текста, который может быть введен в текстовые поля (положительное численное значение);
- **src** – для элемента управления **image** задает расположение используемого изображения;
- **title** – описание элемента управления (может отображаться браузерами как всплывающая подсказка);
- **align** – задает горизонтальное выравнивание элемента управления, работает так же, как и для любого другого HTML-элемента, поддерживающего этот атрибут;
- **tabindex** – номер элемента управления при навигации при помощи табуляции;
- **accesskey** – «горячая» клавиша для элемента управления (для перехода к элементу управления нужно нажать **Alt** и заданную клавишу).

### Многострочное текстовое поле

Обозначается этот элемент управления HTML-элементом **TEXTAREA** и задается парными тэгами **<TEXTAREA>** и **</TEXTAREA>**.

### Списки

Другой способ дать возможность пользователю выбрать то или иное значение состоит в использовании списков. Различают два вида списков: обычные с полосой прокрутки и раскрывающиеся. Раскрывающийся список удобен в тех случаях, когда нужно сэкономить место на странице и уместить много элементов управления в пределах окна обозревателя.

Несмотря на различия во внешнем виде и функциональности, списки обоих типов создаются с помощью одних и тех же дескрипторов. `<SELECT>...</SELECT>` — создает пустой объект списка.

Установка атрибутов в дескрипторе `<SELECT>` определяет вид и функциональность списка.

**SIZE** — устанавливает число строк в поле списка. Если этому атрибуту присвоено значение 1, обозреватель показывает раскрывающийся список, при любом другом положительном значении — обычный список с полосой прокрутки.

**MULTIPLE** — позволяет выбирать в списке сразу несколько элементов. По умолчанию атрибут не установлен.

При выборе какого-либо элемента снимается выделение с элемента, который был выбран ранее.

Выбор нескольких элементов невозможен в раскрывающемся списке, поэтому установка атрибута **multiple** преобразует список в обычный даже в том случае, если атрибуту **size** присвоено значение 1.

**NAME** — имя объекта списка для обращения к списку в коде сценария.

`<OPTION>` — создает элемент списка. Данные дескрипторы добавляются в список между парой дескрипторов `<SELECT>...</SELECT>`. Любой текст за дескриптором `<OPTION>` отображается обозревателем как текст элемента списка. В данном дескрипторе устанавливаются следующие атрибуты:

– **VALUE** — значение элемента списка. Данное значение никак не отображается в окне обозревателя, но именно оно, а не подпись выбранного элемента списка, добавляется в данные формы;

– **SELECTED** — установка этого атрибута выделяет элемент списка по умолчанию.

Если атрибут **selected** установлен в нескольких дескрипторах `<option>`, тогда как в дескрипторе списка `<select>` атрибут **multiple** не был установлен, выделен будет последний элемент в списке с установленным атрибутом **select**.

### *Упражнение 13*

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

```
<HTML>
  <HEAD>
    <TITLE>Переключатели и флажки</TITLE>
  </HEAD>
  <BODY>
    <FORM>
      <!-- Поле ввода -->
      <BR>Фамилия:
      <BR><INPUT TYPE='text' VALUE='Иванов'><BR>
      <!-- Группа переключателей -->
```

```

<H2>Группа переключателей</H2>
<BR><INPUT TYPE='radio' NAME='color'>Белый
<BR><INPUT TYPE='radio' NAME='color'>Красный
<BR><INPUT TYPE='radio' NAME='color'>Желтый
<BR><INPUT TYPE='radio' = 'color'>Зеленый<BR>
<!-- Группа флажков -->
<H2>Группа флажков</H2>
<BR><INPUT TYPE= 'checkbox' NAME='fruit'>Груша
<BR><INPUT TYPE= 'checkbox' NAME='fruit'>Персик
<BR><INPUT TYPE= 'checkbox' NAME='fruit'>Яблоко<BR>
<!-- Кнопки -->
<BR><INPUT TYPE= 'button' VALUE='Пишите мне' ON-
CLICK=window.open('mailto:MyAddress@mail.ru');>
<BR><INPUT TYPE= 'button' VALUE='Посмотрите мои фотографии'
ONCLICK=window.open('http:\\\\www.MySite.org\\MyFoto.htm');> </FORM>
</BODY>
</HTML>

```

3. Сохраните файл под именем «Упражнение 13. html». Загрузите свой документ. На экране вы увидите результат своей работы, изображенный на рис. 19.

Рис.19. Форма с основными элементами управления

#### Упражнение 14

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

```
<HTML>
  <HEAD>
    <TITLE>Раскрывающиеся и обычные списки</TITLE>

</HEAD>
<BODY>
<FORM>
<TABLE>
<CAPTION> <font size='5'>Выберите страну:</font></CAPTION>
<TR><TD VALIGN='top'>
<!--Раскрывающийся список-->
<SELECT NAME='COUNTRY' SIZE='1'>
<OPTION VALUE='Беларусь'>Беларусь
<OPTION VALUE='Казахстан'>Казахстан
<OPTION VALUE='Россия'>Россия
<OPTION VALUE='Украина'>Украина
</SELECT></TD>
<!--Обычный список-->
<TD VALIGN='top'>
<SELECT NAME='COUNTRIES' SIZE='3' MULTIPLE>
<OPTION VALUE='Беларусь'>Беларусь
<OPTION VALUE='Казахстан'>Казахстан
<OPTION VALUE='Россия' SELECTED>Россия
<OPTION VALUE='Украина'>Украина
</SELECT></TD></TR></TABLE></FORM></BODY></HTML>
```

3. Сохраните файл под именем «Упражнение 14.html». Загрузите свой документ. На экране вы увидите результат своей работы, изображенный на рис. 20.

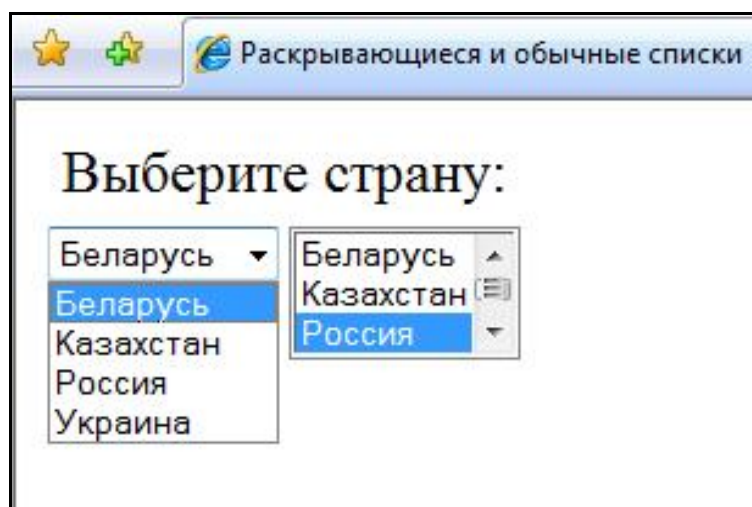
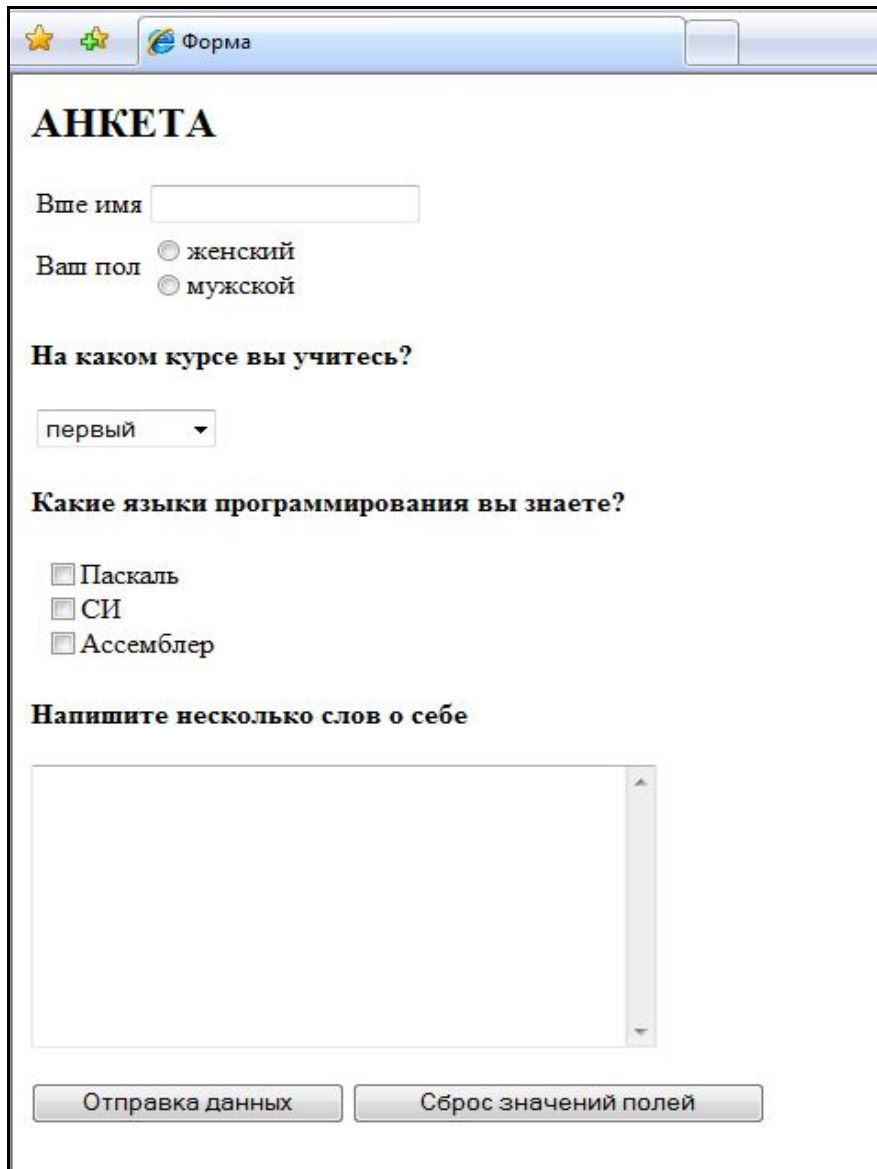


Рис.20. Группы флажков и переключателей

## ЛАБОРАТОРНАЯ РАБОТА № 5

1. Используя полученные вами знания, создайте форму.
2. Форма должна содержать все ранее описанные элементы управления.
3. Тему и оформление задайте самостоятельно.
4. Пример формы приведен на рис. 21.



The image shows a browser window with the title 'Форма' (Form). The main content is a survey form titled 'АНКЕТА'. It contains the following elements:

- A text input field for 'Ваше имя' (Your name).
- Radio buttons for 'Ваш пол' (Your gender): 'женский' (female) and 'мужской' (male).
- A dropdown menu for 'На каком курсе вы учитесь?' (Which course are you studying?) with 'первый' (first) selected.
- Checkboxes for 'Какие языки программирования вы знаете?' (Which programming languages do you know?): 'Паскаль' (Pascal), 'СИ' (C), and 'Ассемблер' (Assembly).
- A large text area for 'Напишите несколько слов о себе' (Write a few words about yourself).
- Two buttons at the bottom: 'Отправка данных' (Send data) and 'Сброс значений полей' (Reset field values).

Рис. 21. Пример формы «Анкета»

## ИЗОБРАЖЕНИЯ-КАРТЫ

Изображение-карта – это обычное изображение с наложенной на него картой активных областей. Для создания карты применяется элемент MAP (задается парными тэгами `< MAP ></ MAP >`). Обычно для элемента MAP задается значения атрибута name – имя, которое используется для ссылок на созданную карту.

Внутри элемента MAP при помощи одиночных тегов <AREA> задаются активные области карты. Для каждого элемента AREA могут задаваться следующие атрибуты:

- **coords** – координаты активной области (количество и значение зависят от типа области, заданного атрибутом **shape**);
- **href** – URL, на который указывает гиперссылка, ассоциированная с областью;
- **nohref** – булев атрибут, создает неактивную область (с областью ассоциируется пустая гиперссылка);
- **shape** – задает тип области, может принимать значения: **rect** (прямоугольник), **circle** (круглая область), **poly** (многоугольник) и **default** (по умолчанию используется вся область карты);
- **alt** – альтернативный текст;
- **title** – текст подсказки (отображается в виде всплывающей подсказки при наведении указателя мыши на область).

Если для атрибута **shape** установлено значение **rect**, то атрибуту **coords** задаются координаты верхнего левого и нижнего правого углов прямоугольника (например, **coords** = “10,10,34,45”).

Если для атрибута **shape** установлено значение **poly**, то атрибуту **coords** задаются координаты всех вершин многоугольника. При этом лучше координаты первой вершины указать и в конце, в случае, если разработчики браузера забыли сделать автоматическое закрытие многоугольника (например, **coords** = “10,10,34,45,50,20,10,10” – задание треугольной области).

Если для атрибута **shape** установлено значение **circle**, то атрибуту **coords** задаются координаты центра и радиус окружности (например, **coords** = “100,100,45”).

Задание координат для создания горячих областей приведены на рис. 22.

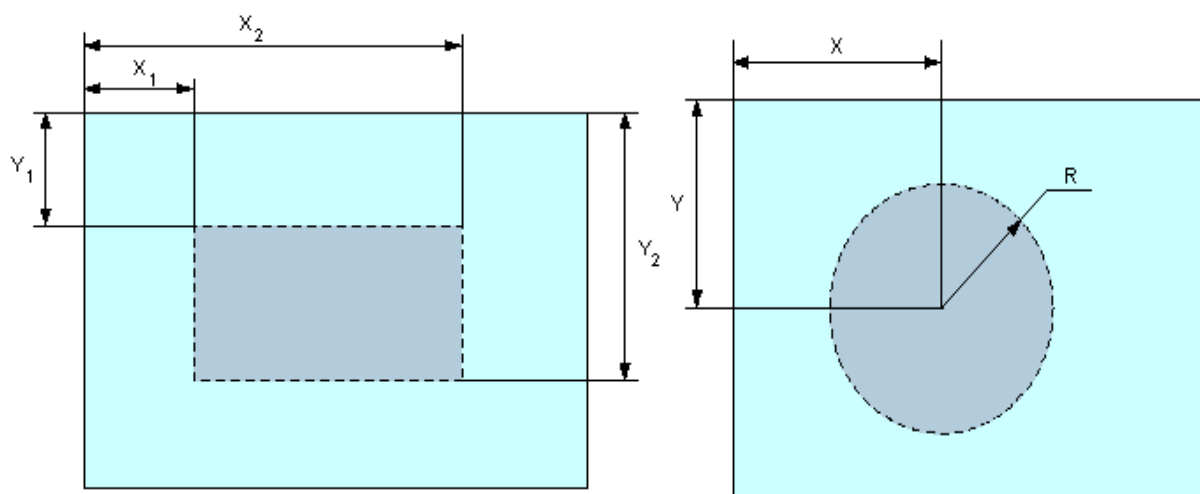


Рис. 22. Изображение координат для создания горячих областей

### Упражнение 15

1. В программе Paint (в левом верхнем углу) создайте прямоугольник, проведите в нем диагонали.

2. Сохраните рисунок под именем **прямоугольник**.

3. В программе Блокнот создайте HTML- документ:

```
<HTML>
    <TITLE>Изображения-карты</TITLE>
<BODY>
<IMG src="прямоугольник.gif" usemap="#map1">
<MAP name=map1>
<!-- Левый треугольник-->
<AREA
coords="50,60,10"
shape=circle
href='Упражнение 1.html'
alt="Переход к упражнению 1"
title="Переход к упражнению 1">
<!-- Верхний треугольник-->
<AREA
coords="100,30,10"
shape=circle
href='Упражнение 2.html'
alt="Переход к упражнению 2"
title="Переход к упражнению 2">
<!-- Правый треугольник-->
<AREA
coords="180,60,10"
shape=circle
href='Упражнение 3.html'
alt="Переход к упражнению 3"
title="Переход к упражнению 3">
<!-- Нижний треугольник-->
<AREA
coords="100,100,10"
shape=circle
href='Упражнение 4.html'
alt="Переход к упражнению 4"
title="Переход к упражнению 4">
</MAP>
</BODY>
</HTML>
```



4. Сохраните файл под именем «Упражнение 15.html». Загрузите свой документ. На экране вы увидите результат своей работы, изображенный на рис. 23.

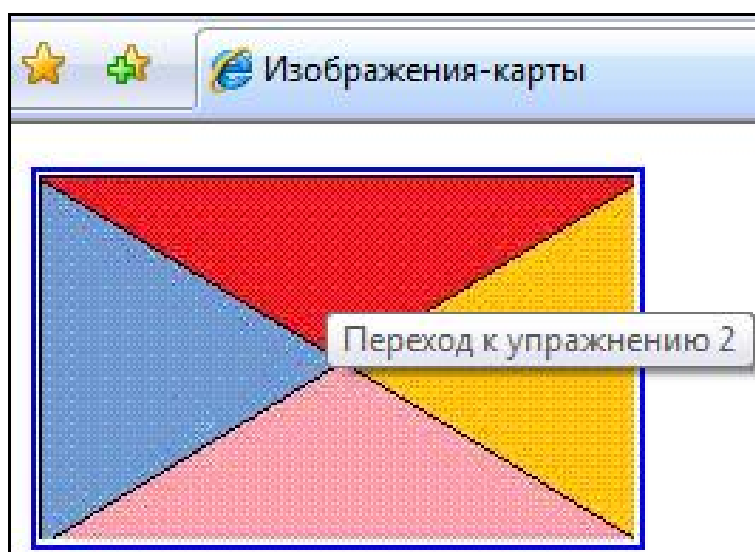


Рис. 23. Изображение – карта

## ЛАБОРАТОРНАЯ РАБОТА № 6

1. Используя полученные вами знания, создайте изображение-карту.
2. Тему и оформление задайте самостоятельно.
3. Пример приведен на рис. 24.

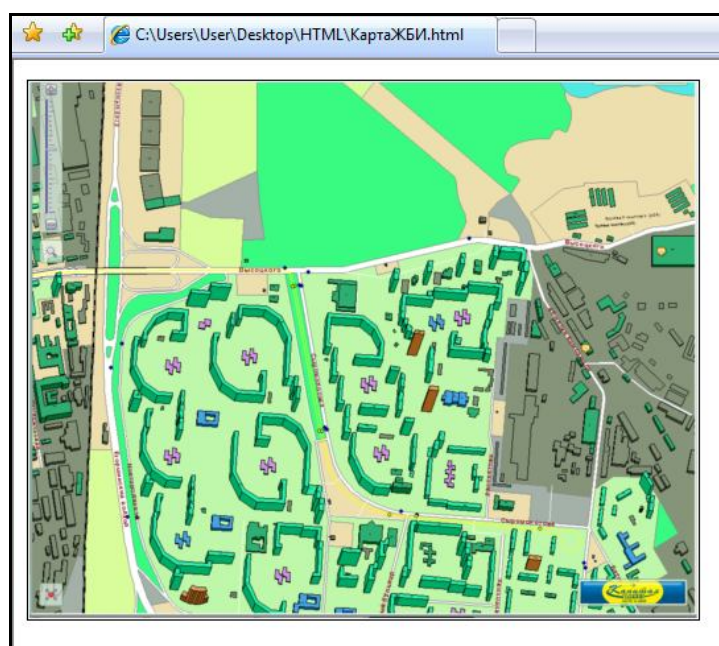


Рис. 24. Пример оформления карты с горячими областями

## Список использованной литературы

1. Информационные технологии [Текст]: учеб. пособие / Е.Л. Румянцева, В.В. Слюсарь. М.: ФОРУМ: ИНФРА-М, 2009. 256 с.: ил.
2. Использование HTML, JavaScript и CSS. Руководство Web – дизайнера [Текст] / О.Н. Рева. М.: Эксмо, 2008. 464 с.
3. HTML. Популярный самоучитель [Текст] / А.А. Чиртик. СПб.: Питер, 2006. 224 с.: ил.
4. HTML. Просто, как дважды два. [Текст] / О.Н. Рева. М.: Изд-во Эксмо, 2007. 256 с.: ил.

## Оглавление

Введение .....	3
Базовые понятия.....	5
Структура HTML-документа.....	5
Заголовок документа.....	6
Упражнение 1 .....	6
Определение тела документа.....	7
Особенности ввода текста .....	8
Задание шрифта текста .....	9
Упражнение 2 .....	9
Запрет разрыва строки .....	10
Упражнение 3 .....	11
Заголовки .....	12
Упражнение 4 .....	12
Специальные символы.....	13
Лабораторная работа № 1 .....	13
Списки .....	14
Нумерованные списки .....	14
Упражнение 5 .....	15
Маркированные списки .....	16
Упражнение 6 .....	16
Многоуровневый.....	16
Упражнение 7 .....	17
Лабораторная работа № 2 .....	18
Гиперссылки.....	19
Простой переход к ресурсам .....	19
Навигация между HTML-документами.....	19
Упражнение 8 .....	19
Графика на Web-страницах .....	21
Упражнение 9 .....	21
Линейки .....	22

Упражнение 10 .....	22
Изображения-гиперссылки.....	23
Упражнение 11 .....	23
Таблицы .....	24
Заголовок таблицы .....	25
Параметры отображения таблицы .....	25
Слияние ячеек .....	26
Параметры отображения строк таблицы .....	26
Лабораторная работа № 3 .....	26
Фреймы .....	27
Целевой фрейм отдельных гиперссылок .....	28
Упражнение 12 .....	29
Лабораторная работа № 4 .....	30
Элемент NOFRAMES.....	31
Формы.....	31
Создание формы.....	31
Стандартные элементы управления .....	32
Многострочное текстовое поле .....	34
Списки .....	34
Упражнение 13 .....	35
Упражнение 14 .....	37
Лабораторная работа № 5 .....	38
Изображения-карты .....	38
Упражнение 15 .....	40
Лабораторная работа № 6 .....	41
Список использованной литературы.....	42



Е.В. Кох

# ОСНОВЫ HTML

Екатеринбург  
2010