



Министерство образования и молодёжной  
политики Свердловской области

**ЕКАТЕРИНБУРГСКИЙ КОЛЛЕДЖ  
ТРАНСПОРТНОГО СТРОИТЕЛЬСТВА**

**МЕТОДИЧЕСКОЕ ПОСОБИЕ  
ПО ТЕМЕ «ГИПЕРТЕКСТОВОЕ ПРЕДСТАВЛЕНИЕ  
ИНФОРМАЦИИ. ЯЗЫК РАЗМЕТКИ ГИПЕРТЕКСТА HTML»**

**Разработала**

**Е.М. Александрова**

**2023г.**



## Оглавление

<b>ВВЕДЕНИЕ .....</b>	<b>4</b>
<b>Практическая работа № 1. Основные понятия языка HTML. Структура документа HTML .....</b>	<b>6</b>
<b>Задания к практической работе № 1 .....</b>	<b>10</b>
<b>Практическая работа № 2. Форматирование текста.....</b>	<b>11</b>
<b>Задания к практической работе № 2. Форматирование текста на Web-странице по образцу .....</b>	<b>21</b>
<b>Практическая работа № 3. Управление цветом.....</b>	<b>25</b>
<b>Задания к практической работе № 3.....</b>	<b>28</b>
<b>Практическая работа № 4. Работа со списками .....</b>	<b>29</b>
<b>Задания к практической работе № 4. Работа со списками .....</b>	<b>36</b>
<b>Практическая работа № 5. Вставка графических изображений .....</b>	<b>41</b>
<b>Задания к практической работе № 6. Вставка графических изображений .....</b>	<b>45</b>
<b>Практическая работа № 7. Таблицы HTML-документов.....</b>	<b>48</b>
<b>Задания к практической работе № 7. Таблицы HTML-документов.....</b>	<b>64</b>
<b>Использованные источники.....</b>	<b>68</b>

## ВВЕДЕНИЕ

### Язык разметки гипертекста HTML. Назначение HTML

*World Wide Web (Web)* - это сеть информационных ресурсов. Для того, чтобы сделать эти ресурсы доступными наиболее широкой аудитории, в Web используются три механизма.

1. Единая схема наименования для поиска ресурсов в Web (например, URI).
2. Протоколы для доступа к именованным ресурсам через Web (например, HTTP).
3. Гипертекст для простого перемещения по ресурсам (например, HTML).

Каждый ресурс в Web – документ HTML, изображение, видеоклип, программа и т.д. – имеет адрес, который может быть закодирован с помощью *универсального идентификатора ресурсов (Universal Resource Identifier)*, или URI.

URI обычно состоят из трех частей:

1. Схема наименования механизма, используемого для доступа к ресурсу.
2. Имя машины, на которой располагается ресурс.
3. Имя собственно ресурса, заданное в виде пути.

### Язык разметки гипертекста HTML

Всемирная паутина (*World Wide Web — WWW*) состоит из множества связанных между собой электронных документов, представляющих кладёз информационных данных, описанных с помощью специальных технологических правил.

Эти правила составляются на языке гипертекстовой разметки HTML (*Hypertext Markup Language*).

Можно с уверенностью сказать, что сегодня язык разметки HTML является **основой всех размещенных в Интернете электронных документов. Он выступает в роли некоего фундамента, на базе которого реализуются прочие сетевые программные технологии**, призванные в конечном итоге повысить общую привлекательность, эффективность и интерактивность носителей информационных данных в Сети.

Термин *HyperText Markup Language (HTML)* - означает «язык маркировки гипертекстов» и включает в себя различные способы оформления гипертекстовых документов, дизайн, гипертекстовые редакторы, браузеры и многое другое. Человек, изучивший этот язык, обретает возможность делать сложные вещи простыми способами и, главное, быстро. HTML как основа создания Web-страниц имеет прямое отношение к такому направлению изобразительного искусства, как Web-дизайн.

. HTML дает авторам средства для:

- публикации электронных документов с заголовками, текстом, таблицами, списками, фотографиями и т.д.
- загрузки электронной информации с помощью щелчка мыши на гипертекстовой ссылке.
- разработки форм для выполнения транзакций с удаленными службами, для использования в поиске информации, резервировании,

заказе продуктов и т.д. • включения электронных таблиц, видеоклипов, звуковых фрагментов и других приложений непосредственно в документы.

Наиболее распространенными браузерами (программами) для просмотра HTML-файлов являются Microsoft Internet Explorer и Netscape Navigator. Открывая в браузере любую Web страницу, Вы видите текст, картинки, кнопки, таблицы и многое другое. Как создается Web страница? Для того чтобы создать Web-страницу, можно использовать текстовый редактор Notepad (**Блокнот**).

Откройте любую Web-страничку. То, что вы видите в браузере, это ее "лицо". Чтобы увидеть "изнанку" Web-страницы, выполните команду **View | HTML Code (Вид | Просмотр HTML кода)**. Откроется текстовый редактор Блокнот, в котором вы увидите "устройство» этой страницы.

## Практическая работа № 1. Основные понятия языка HTML. Структура документа HTML

**Цель работы:** знакомство студентов с основными понятиями языка HTML, структурой HTML-документа, обязательными метками, комментариями, способами форматирования текста, физическими и логическими стилями, приобретение навыков создания простейших статических Web-документов.

### Основные понятия

**Гипертекст** - информационная структура, позволяющая устанавливать смысловые связи между элементами текста на экране компьютера таким образом, чтобы можно было легко осуществлять переходы от одного элемента к другому. На практике в гипертексте некоторые слова выделяют путем подчеркивания или окрашивания в другой цвет (гиперссылки). Выделение слова говорит о наличии связи этого слова с некоторым документом, в котором тема, связанная с выделенным словом, рассматривается более подробно.

Отдельный документ, выполненный в формате HTML, называется:

- HTML-документом;
- Web-документом;
- Web-страницей.

Такие страницы, как правило, имеют расширение **HTM** или **HTML**.

**Гиперссылка** - фрагмент текста, который является указателем на другой файл или объект. **Гиперссылки** необходимы для того, чтобы обеспечить возможность перехода от одного документа к другому.

Группа Web-страниц, принадлежащих одному автору или одному издателю и взаимосвязанных общими гиперссылками, образует структуру, которая называется *Web узлом*, или *Web-сайтом*.

Каждая HTML-страница имеет свой уникальный *URL-адрес* в Интернете. **Фрейм (Frame)** - этот термин имеет два значения. Первое – область документа со своими полосами прокрутки. Второе значение – одиночное изображение в анимационном графическом файле (кадр).

**Апплет (Applet)** - программа, передаваемая на компьютер клиента в виде отдельного файла и запускаемая при просмотре Web-страницы.

**Скрипт, или сценарий (Script)**, - программа, включенная в состав Web-страницы для расширения ее возможностей. Браузер Internet Explorer в определенных ситуациях выводит сообщение: "Разрешить выполнение сценариев на странице?" В этом случае имеются в виду скрипты.

**CGI (Common Gateway Interface)** - общее название программ, которые, работая на сервере, позволяют расширять возможности Web-страниц. *Например*, без таких программ невозможно создание интерактивных Web-страниц.

**Браузер (Browser)** - программа для просмотра Web-страниц.

**Элемент** - конструкция языка HTML. Можно представить его себе как контейнер, содержащий данные и позволяющий отформатировать их определенным образом. Любая Web-страница представляет собой набор элементов. Одна из основных идей гипертекста возможность вложения элементов.

**<Начало элемента>** Содержание элемента, данные, которые форматирует элемент **</Конец элемента>**

**Тэг** (по-английски — tag-метка, дескриптор, ярлык) - начальный или конечный маркеры элемента. Тэги определяют границы действия элементов и отделяют элементы друг от друга. В тексте Web-страницы тэги заключаются в угловые скобки <>, а конечный тэг всегда снабжается косой чертой. Текст, не находящийся между такими скобками <> – весь виден, при просмотре в браузере.

**<Начальный тэг>** Содержание элемента, данные, которые форматирует элемент **</Конечный тэг>**

**Пример**

**<P>** Этот текст будет расположен в отдельном абзаце **</P>**

Элемент, содержащий некоторый текст, ограничен начальным тэгом (маркером) <p> и конечным тэгом (маркером) </p>, т. е. текст помещен между тэгами, как в контейнер, а тэги <p> и </p> размечают начало и конец абзаца соответственно.

Любая Web-страница представляет собой набор элементов.

Один из основных принципов HTML возможность вложения элементов, они могут вкладываться один в другой.

**Атрибут** – параметр или свойство элемента. Атрибуты располагаются внутри начального тэга и отделяются друг от друга пробелами. Если элемент содержит текст, то атрибуты могут задавать цвет и размер шрифта, выравнивание текстового абзаца и т. п. Если элемент содержит рисунок, то атрибуты могут задавать размер рисунка, наличие и размер рамки вокруг рисунка и пр.

**Пример**

**<P align="center">** Этот текст будет выровнен по центру экрана **</p>**

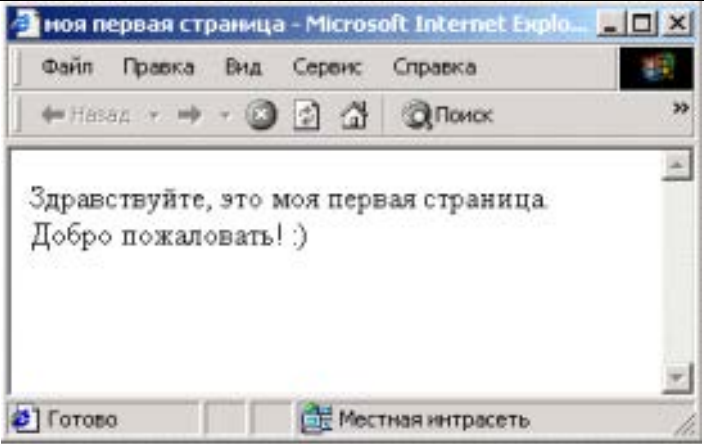
В этом примере опять встречается тэг, определяющий начало и конец абзаца. Однако в начальном тэге находится атрибут **align**, который задает выравнивание текста по центру экрана.

**Примечание:**

- любая полезная информация должна находиться между начальным и конечным тэгами, указывающими ее формат;
- все атрибуты располагаются в начальном тэге;
- для удобства работы начальный тэг вы можете писать с прописной (заглавной) буквы (P), а конечный - со строчной (маленькой) буквы (/p) хотя это и не обязательно;
- не для всех элементов требуется ставить конечный (закрывающий) тэг.
- написание каждого нового элемента начинайте с новой строки. Вложенные элементы выделяйте отступом (табуляцией). Это опять - таки не обязательно, но значительно облегчит вашу работу.

## Структура документа HTML

Для того чтобы понять структуру Web-страницы, нам необходимо подробно рассмотреть все элементы, входящие в приведенный листинг.

Пример. 1	Образец полученной страницы
<pre>&lt;HTML&gt; &lt;HEAD&gt; &lt;TITLE&gt; моя первая страница &lt;/title&gt; &lt;/head&gt; &lt;BODY&gt; Здравствуйте, это моя первая страница. &lt;BR&gt; Добро пожаловать! :) &lt;/body&gt; &lt;/html&gt;</pre>	

### **<COMMENT> </comment>**

Текст комментария. В любом языке программирования есть конструкции, позволяющие создавать произвольные ремарки.

HTML в этом смысле — не исключение. Текст, помещенный внутри COMMENT, игнорируется браузером. COMMENT может располагаться в любом месте кода Web-страницы. Комментарий должен быть отделен от основного текста.

Существует, правда, одно ограничение: внутри комментария не должны располагаться другие элементы. Так должно быть, разумеется, только в том случае, когда необходимо, чтобы все содержимое элемента COMMENT не отображалось на экране монитора. Если в комментарии будет присутствовать другой элемент, то его содержимое будет выведено на экран, отформатированное соответствующим образом.

Существует еще один способ обозначения комментария. Он заключается в использовании восклицательного знака и обрамлении текста комментария двойными тире. Например: **<!-- Строка комментария -->**

**<!-- Комментарий -- Не комментарий -- Снова комментарий -->**

### **<HTML> </html>**

Отличительный признак HTML-документа. Одним из принципов языка является многоуровневое вложение элементов. HTML является самым внешним, так как между его стартовым и конечным тегами должна находиться вся Web-страница. В принципе, этот элемент можно рассматривать как формальность. Он имеет атрибуты version, lang и dir, которыми в данном случае мало кто пользуется, и допускает вложение элементов HEAD, BODY и PLAINTEXT, определяющих общую структуру Web-страницы. Естественно, что конечным тегом </html> заканчиваются все гипертекстовые документы.

### **<HEAD> </head>**

Информация о документе, которая не выводится на экран, называется заголовком. Так же, как и HTML, HEAD служит только для формирования общей



структуры документа. Этот элемент может иметь атрибуты lang и dir и допускает вложение элементов TITLE, ISINDEX, BASE, META, LINK, NEXTID.

### **<TITLE> </title>**

Элемент для размещения заголовка Web-страницы. Строка текста, расположенная внутри, отображается не в документе, а в заголовке окна браузера. Эта особенность часто используется для организации поиска в WWW. Поэтому авторы, создающие Web-страницы, должны позаботиться о том, чтобы строка внутри TITLE, не будучи слишком длинной (не более 64 символов), достаточно точно отражала назначение документа.

### **<BODY> </body>**

Этот элемент включает в себе гипертекст, который определяет собственно Web страницу. Эта та часть документа, которую разрабатывает автор страницы и которая отображается браузером. Соответственно, конечный тег этого элемента надо искать в конце HTML-файла. Внутри BODY можно использовать все элементы, предназначенные для дизайна Web-страницы. *Внутри стартового тега* элемента BODY можно расположить ряд атрибутов, обеспечивающих установки для всей страницы в целом.

<b>Параметр</b>	<b>Функция</b>
bgcolor = "#RRGGBB"	<b>Определение цвета фона.</b> Цвет фона задается тремя двухразрядными шестнадцатеричными числами, которые определяют интенсивность красного, зеленого и синего цветов соответственно. Более подробно об определении цветов будет рассказано ниже.
background = "Путь к файлу фона"	<b>Указание фонового рисунка.</b> Оба вышеприведенных атрибута не являются альтернативными и часто используются совместно.
text = "#RRGGBB"	<b>Определение цвета основного текста.</b>
link = "#RRGGBB"	<b>Определение цвета текста гиперссылок.</b>
vlink = "#RRGGBB"	<b>Определение цвета для использованных гиперссылок.</b>
alink = "#RRGGBB"	<b>Определение цвета для последней выбранной пользователем гиперссылки.</b>
bgproperties	Изменение свойств фона (например, фиксирование фонового рисунка)
topmargin="10", bottommargin="10", leftmargin, rightmargin	Определение размера отступа от верхнего, нижнего, левого и правого краев документа. Значение задается в пикселях
marginwidth="10", marginheight="10"	Netscape объединяет упомянутые параметры в две группы: горизонтальные и вертикальные отступы. Для учета особенностей всех браузеров, надо поставить и те и другие параметры

**Пример.**

```
<body topmargin="5", bottommargin="5", leftmargin="10", rightmargin="10",  
marginwidth="10", marginheight="5"
```

**Задания к практической работе № 1**

**Задание 1. Создание простейшей Web-страницы.**

**Цель:** научиться создавать Web-страницы в редакторе Notepad ++

**Указания к выполнению**

1. Создайте на диске отдельную папку для будущей страницы. **Первые шаги \_ФИО**

2. Откройте редактор Notepad ++и наберите в нем структуру HTML документа, которая приведена в примере 1:

3. Сохраните этот документ, присвоив ему **ФИО\*.html**

4. Откройте с помощью Internet Explorer ваш документ (не закрывайте редактор, он нам еще пригодится).

Если вы что-то изменили в вашем \*.html документе (в редакторе), то, не забудьте сохранить изменения, а, чтобы посмотреть, как это выглядит в Internet Explorer, надо нажать в Internet Explorer кнопку **ОБНОВИТЬ** (тоже самое касается других браузеров).

Если изменений не видно, то это значит, что вы где-то что-то неправильно написали, или забыли сохранить документ.

5. Вернитесь к сохраненному в редакторе файлу.

6. Внесите в него следующие изменения: пусть это будет ваша первая страничка. **Укажите в ней ваши фамилию, имя, ФИО родителей, братьев и сестер, свои увлечения.** Используйте для этого форматирование абзацев.

7. В строке <TITLE> укажите: **"Домашняя страничка (ваше имя и фамилия)".** 8. Сохраните файл как **ФИО2.htm.**

9. Просмотрите результат в браузере, при необходимости отредактируйте файл при помощи Блокнота.

**Результат выполнения практической №1**

папка	<b>Первые шаги _ФИО</b>	
файл	1. <b>ФИО*.html</b> 2. <b>ФИО2.htm.</b>	

## Практическая работа № 2. Форматирование текста

**Цель:** познакомиться с основами HTML, обязательными метками, &-последовательностями, комментариями, способами форматирования текста (шрифта и абзаца), физическими и логическими стилями, горизонтальными линиями.

Текст – единственный объект Web-страницы, который не требует специального определения. Иными словами, произвольные символы интерпретируются по умолчанию как текстовые данные.

Для форматирования текста существует большое количество элементов. Форматировать текст можно с помощью традиционных элементов: выделять фрагменты курсивом, полужирным шрифтом, выбирать шрифт, размер и цвет шрифта, выравнивать текстовые фрагменты. Если автору не хватает простых вариантов форматирования текста, он может прибегнуть к таблицам стилей, которые существенно расширят возможности языка HTML по форматированию.

Рассмотрим стандартные элементы языка HTML, позволяющие форматировать текст.

### Форматирование абзаца

`<P> ... </p>`

Элемент абзаца разделяет фрагменты текста вертикальным отступом. Он позволяет использовать только начальный тэг, так как следующий элемент `</p>` обозначает не только начало следующего абзаца, но и конец предыдущего. В тех случаях, когда по смыслу необходимо обозначить завершение абзаца, можно использовать и конечный тэг.

### Вместе с элементом абзаца можно использовать и атрибут выравнивания align

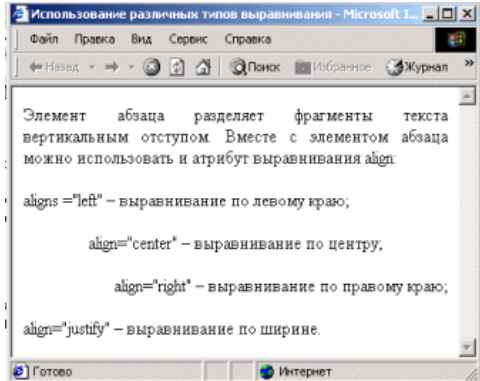
`align="left"` – выравнивание по левому краю;

`align="right"` – выравнивание по правому краю;

`align="center"` – выравнивание по центру;

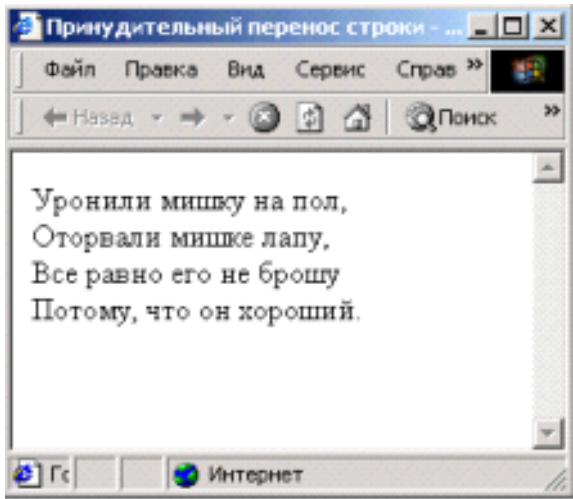
`align="justify"` – выравнивание по ширине.

Выравнивание по ширине – опасный параграф. Он не работает в старых версиях браузеров (т.е. текст выравнивается в таком случае по левому краю документа).

Пример. 1	Образец полученной страницы
<pre>&lt;HTML&gt; &lt;HEAD&gt; &lt;TITLE&gt; Использование различных типов выравнивания &lt;/title&gt; &lt;/head&gt; &lt;BODY&gt; &lt;P align="justify"&gt; Элемент абзаца разделяет фрагменты текста вертикальным отступом. Вместе с элементом абзаца можно</pre>	

<p>использовать и атрибут выравнивания align: &lt;/p&gt;          &lt;P align="left"&gt; aligns ="left" –          выравнивание по левому краю;          &lt;/p&gt;          &lt;P align="center"&gt; align="center" –          выравнивание по центру; &lt;/p&gt;          &lt;P align="right"&gt; align="right" –          выравнивание по правому краю;          &lt;/p&gt;          &lt;P align="justify"&gt; align="justify" –          выравнивание по ширине.&lt;/p&gt;          &lt;/body&gt;          &lt;/html&gt;</p>	
--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--

**<BR>** Элемент, обеспечивающий принудительный переход на новую строку. Он имеет только начальный тэг. В месте его размещения строка заканчивается, а оставшийся текст печатается с новой строки.

Пример _3	Образец полученной страницы
<pre>&lt;HTML&gt; &lt;HEAD&gt; &lt;TITLE&gt; Принудительный перенос строки &lt;/title&gt; &lt;/head&gt; &lt;BODY&gt; Уронили мишку на пол, &lt;BR&gt; Оторвали мишке лапу, &lt;BR&gt; Все равно его не брошу &lt;BR&gt; Потому, что он хороший. &lt;/body&gt; &lt;/html&gt;</pre>	

Если таким образом расставить элемент <br> в этом стихотворении, то в экране браузера мы увидим стандартным образом написанное четверостишие на четырех строках.

### **<NOBR> </nobr>**

Этот элемент по своему действию является прямой противоположностью предыдущему. Текст, заключенный между его тегами, будет выведен в одну строку. Длинная строка не уместится на экране, и для ее просмотра придется использовать горизонтальную полосу прокрутки.

### **<PRE> </pre>**

Элемент для обозначения текста, отформатированного заранее (preformatted). Подразумевается, что текст будет выведен в том виде, в каком он был подготовлен

пользователем. Текст, заключенный между метками <PRE> и </pre> (от английского preformatted — предварительно форматированный), выводится браузером на экран как есть — со всеми пробелами, символами табуляции и конца строки. Во всех других случаях браузер игнорирует эти символы. Возможен и обратный эффект: если пользователь введет текст как одну длинную строку, то она не будет разорвана браузером, а уйдет за границы окна программы. В этом смысле элемент PRE работает так же, как элемент NOBR. По умолчанию для отформатированного заранее текста выбирается моноширинный шрифт. Этот элемент удобно использовать для демонстрации листингов программ или для вывода текстовых документов, переформатирование которых может привести к искажению их смысла.

#### <BLOCKQUOTE> ... </blockquote>

Текст, заключенный между метками <BLOCKQUOTE> и </blockquote>, выводится браузером на экран с увеличенным левым полем (с отступом).

#### <CENTER> </center>

Элемент для центрирования текста, а точнее, любого содержимого. Не является общеупотребительным. В тех случаях, когда это возможно, вместо него используют атрибут align = "center".

#### <DIV> </div>

Элемент, похожий на предыдущий. Он позволяет выравнивать содержимое по левому краю, по центру или по правому краю. Для этого стартовый тег должен содержать соответствующий атрибут:

align = "left"

align = "center"

align = "right"

### **Форматирование шрифта**

HTML допускает два подхода к шрифтовому выделению фрагментов текста — логическое и физическое форматирование.

В группу тегов логического форматирования входят теги, отображающие на экране монитора элементы документа, таким образом, как установлено по умолчанию в спецификации языка разметки HTML. Переопределить их параметры или свойства нельзя, за исключением ситуаций использования стилевых шаблонов CSS и обособления тегам физическое форматирования. Результат действия разных тегов логического форматирования визуально может совпадать, ибо основное их предназначение заключается в логическом выделении отдельных элементов HTML.

Теги физического форматирования позволяют разработчику визуально изменять вид текста, делая его **жирным** или *наклонным*, варьируя его параметры и значения.

### **Физические стили**

*Под физическим стилем принято понимать прямое указание браузеру на модификацию текущего шрифта.*

#### <B> </b>

Выделение текста **полужирным шрифтом** (от слова **Bold**), очень популярный элемент. Использование полужирного шрифта — прием, позаимствованный из текстовых редакторов.

### Пример

Этот текст имеет обычное начертание - <B>, а этот будет выделен полужирным шрифтом </B>.

<I> </i>

Выделение текста *курсивом* (от слова *Italic*).

### Пример

Этот текст имеет обычное начертание <I>, а этот выделен курсивом </i>.

<TT> </tt>

Элемент, обозначающий текст телетайпа (teletype). Его особенность заключается в использовании моноширинного шрифта, имитирующего пишущую машинку, то есть имеющего фиксированную ширину символа.

<STRIKE> </strike>

Элемент, создающий перечеркнутое начертание текста. В настоящее время его заменяют тегом: <DEL> </del>.

<U> </u>

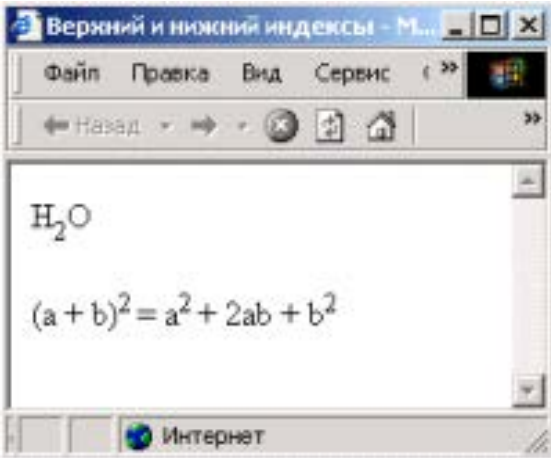
Подчеркнутое начертание текста.

<SUB> </sub>

Элемент, создающий эффект нижнего индекса (subscript).

<SUP> </sup>

Элемент, создающий эффект верхнего индекса (superscript).

Пример. _4	Образец полученной страницы
<pre>&lt;HTML&gt; &lt;HEAD&gt; &lt;TITLE&gt; Верхний и нижний индексы &lt;/title&gt; &lt;/head&gt; &lt;BODY&gt; H&lt;SUB&gt;2&lt;/sub&gt;O &lt;BR&gt;&lt;BR&gt; (a + b)&lt;SUP&gt;2 &lt;/sup&gt; = a&lt;SUP&gt;2 &lt;/sup&gt; + 2ab + b&lt;SUP&gt;2 &lt;/sup&gt; &lt;/body&gt; &lt;/html&gt;</pre>	

### Размер шрифта

<BASEFONT>

Элемент, определяющий базовый (основной) размер шрифта. Внутри элемента необходимо указать атрибут:

size = Базовый размер шрифта

Величина атрибута может лежать в пределах от 1 до 7. По умолчанию используется величина 3.

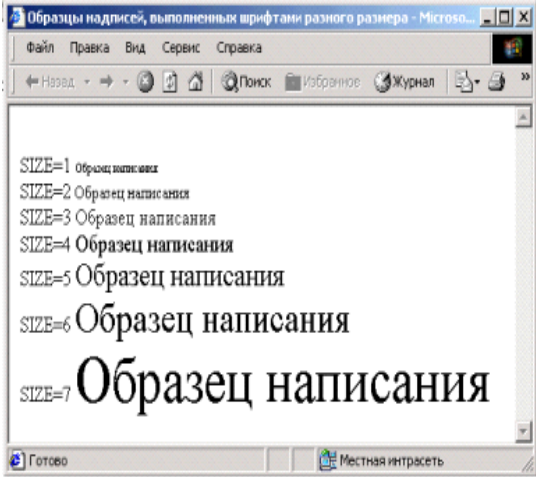
Установка, выполняемая этим элементом, имеет значение для элемента FONT (см. ниже), который позволяет задавать относительный размер шрифта.

## <FONT> </font>

Определение типа, размера и цвета шрифта. Все эти характеристики определяются при помощи соответствующих атрибутов.

Абсолютный размер шрифта задается при помощи атрибута size (размер): size=Абсолютный размер шрифта

Этот атрибут может принимать значения от 1 до 7. В таблице показано несколько образцов надписей, выполненных шрифтами разного размера.

Пример. _5	Образец полученной страницы
<pre>&lt;HTML&gt; &lt;HEAD&gt; &lt;TITLE&gt;      Образцы      надписей, выполненных  шрифтами  разного размера &lt;/title&gt; &lt;/head&gt; &lt;BODY&gt; &lt;BR&gt; SIZE=1 &lt;FONT size=1&gt; Образец написания &lt;/font&gt; &lt;BR&gt; SIZE=2 &lt;FONT size=2&gt; Образец написания &lt;/font&gt; &lt;BR&gt; SIZE=3 &lt;FONT size=3&gt; Образец написания &lt;/font&gt; &lt;BR&gt; SIZE=4 &lt;FONT size=4&gt; Образец написания &lt;/font&gt; &lt;BR&gt; SIZE=5 &lt;FONT size=5&gt; Образец написания &lt;/font&gt; &lt;BR&gt; SIZE=6 &lt;FONT size=6&gt; Образец написания &lt;/font&gt; &lt;BR&gt; SIZE=7 &lt;FONT size=7&gt; Образец написания &lt;/font&gt; &lt;/body&gt; &lt;/html&gt;</pre>	

### Пример

<FONT size=3> Тише едешь - дальше будешь </font>

Размер шрифта может задаваться относительно базового:

size=+Число

size=-Число

При назначении величины для size необходимо учитывать величину базового размера. Обе они в сумме должны соответствовать одному из абсолютных размеров. Так, для базового размера, равного 3, относительный размер может находиться в пределах от - 2 до + 4. Если величина выходит за допустимый предел, то используется или шрифт размера 7, или шрифт размера 1.

При помощи относительных величин тоже можно получить семь градаций размера шрифта



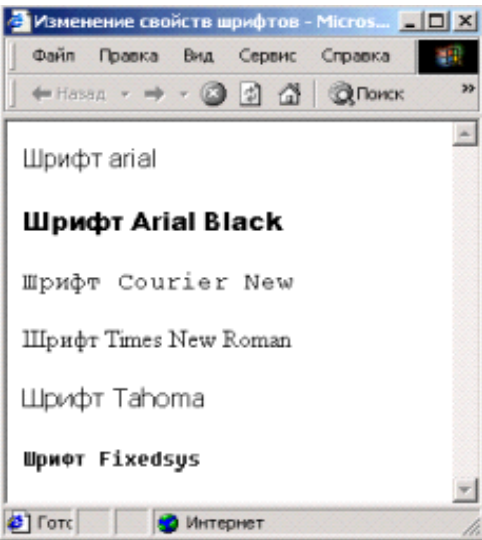
Для элемента FONT можно использовать атрибут цвета:  
color="Цвет"

Название цвета задается либо числом, написанным в шестнадцатеричной системе, например, color="#ff0000", либо просто его названием, написанным на английском языке, например, color="red".

#### Пример

```
<FONT color="blue"> Это шрифт синего цвета </font>
```

Атрибут face (вид) позволяет задавать тип шрифта:

Пример _6	Образец полученной страницы
<pre>&lt;HTML&gt; &lt;HEAD&gt; &lt;TITLE&gt; Изменение свойств шрифтов &lt;/title&gt; &lt;/head&gt; &lt;BODY&gt; &lt;FONT face="arial"&gt;Шрифт arial&lt;/font&gt; &lt;BR&gt;&lt;BR&gt; &lt;FONT face="Arial Black"&gt; Шрифт Arial Black&lt;/font&gt; &lt;BR&gt;&lt;BR&gt; &lt;FONT face="Courier New"&gt; Шрифт Courier New &lt;/font&gt; &lt;BR&gt;&lt;BR&gt; &lt;FONT face="Times New Roman"&gt; Шрифт Times New Roman&lt;/font&gt; &lt;BR&gt;&lt;BR&gt; &lt;FONT face="Tahoma"&gt; Шрифт Tahoma&lt;/font&gt; &lt;BR&gt;&lt;BR&gt; &lt;FONT face="Fixedsys"&gt; Шрифт Fixedsys&lt;/font&gt; &lt;/body&gt; &lt;/html&gt;</pre>	

face="Название шрифта"

Правда, есть одна проблема. Web-страницы просматривает множество людей, и нет гарантии, что у каждого из них окажется нужный шрифт. Если в системе не установлен шрифт точно с таким же названием, то браузер использует свой стандартный. Он имеет два назначенных по умолчанию шрифта: один пропорциональный, другой моноширинный.

Все эти атрибуты могут быть использованы совместно внутри тэга <font>.

#### Пример

```
<FONT face="Arial" size=3 color="blue" > Это шрифт arial размером 3,
цвет синий </font>
```

Элемент FONT может с успехом заменять элементы заголовка H1...H6. Для последних, например, не предусмотрена возможность указания цвета букв.



Чтобы заголовок, созданный на основе элемента FONT, хорошо смотрелся, этот элемент необходимо комбинировать с другими: CENTER, B, I, P и т. д.

**<BIG> </big>**

Этот тег используется, если необходимо выделить часть текста небольшим увеличением размера шрифта относительно остальных слов.

**<SMALL> </small>**

Этот тег используется, если необходимо выделить часть текста небольшим уменьшением размера шрифта относительно остальных слов.

### **Логические стили**

При использовании логических стилей автор документа не может знать заранее, что увидит на экране читатель. Разные браузеры толкуют одни и те же метки логических стилей по-разному. Некоторые браузеры игнорируют некоторые метки вообще и показывают нормальный текст вместо выделенного логическим стилем. Вот самые распространенные логические стили.

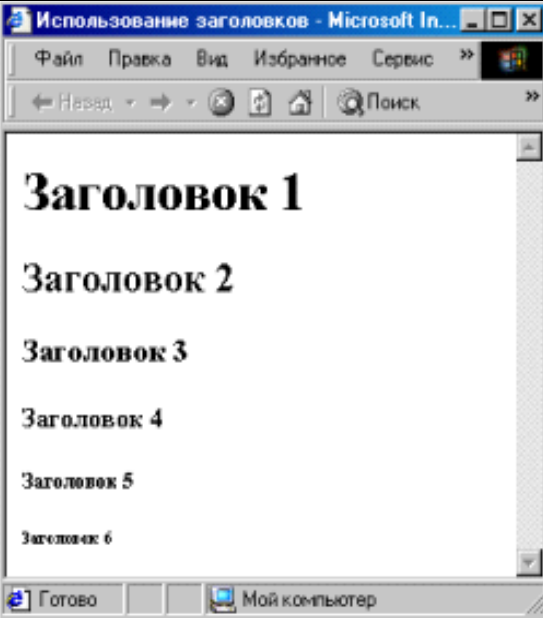
**<H1> </h1>**

Элемент заголовка. Существует шесть уровней заголовков, которые обозначаются H1. . H6. Заголовок уровня 1 самый крупный, а уровень 6 обеспечивает самый маленький заголовок. Представление об относительных размерах букв в них дает рис.1. Для заголовков можно использовать атрибут, задающий выравнивание влево, по центру или вправо:

align = "left"

align = "center"

align = "right"

Пример _7	Образец полученной страницы
<pre> &lt;HTML&gt; &lt;HEAD&gt; &lt;TITLE&gt;Использование заголовков&lt;/title&gt; &lt;/head&gt; &lt;BODY&gt; &lt;H1&gt;Заголовок 1&lt;/H1&gt; &lt;H2&gt;Заголовок 2&lt;/H2&gt; &lt;H3&gt;Заголовок 3&lt;/H3&gt; &lt;H4&gt;Заголовок 4&lt;/H4&gt; &lt;H5&gt;Заголовок 5&lt;/H5&gt; &lt;H6&gt;Заголовок 6&lt;/H6&gt; &lt;/body&gt; &lt;/html&gt; </pre>	

## Горизонтальные линии

Горизонтальная линия (horizontal rule) или, на типографский манер, горизонтальные линейки – очень часто используемый элемент.

**Во-первых**, потому, что с его помощью очень удобно делить страницу на части, использовать для отделения одного блока текста от другого. Небольшой по размеру текст, сверху и снизу которого располагаются горизонтальные линии, привлекает больше, чем обычно, внимание читателя.

**Во-вторых**, выбор таких элементов оформления у автора страницы очень небольшой. Действительно, в HTML практически отсутствуют подобные конструкции, а вот для горизонтальной линии почему-то было сделано исключение. Правда, несмотря на некоторую скупость языка в этой области, можно придумать немало стандартных графических образов, которые разнообразили бы вид страниц.

Если нужна простая, без затей линия, можно воспользоваться тегом

**<HR>**

**<HR>**

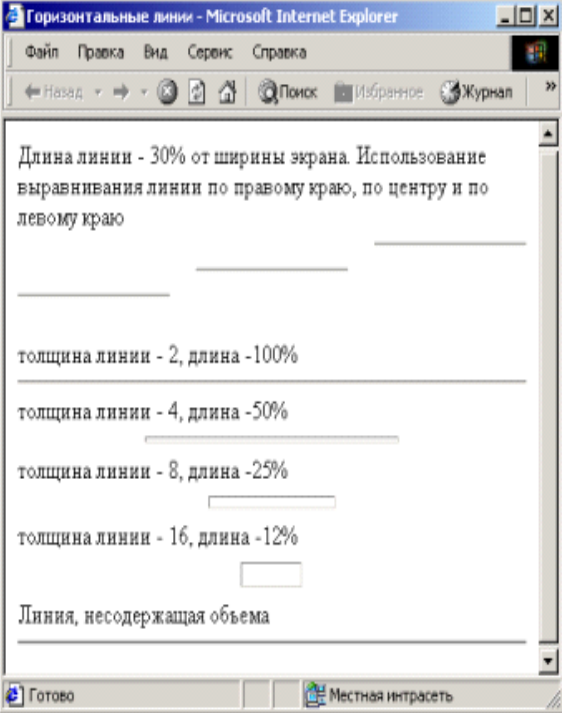
Пример создания горизонтальных линий с помощью тега HR

**<HR>**

Пример создания горизонтальных линий с помощью тега HR

Элемент не имеет конечного тега, но допускает ряд атрибутов:

- |    |                                       |                                                                                                                                                     |
|----|---------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------|
| 1. | <code>&lt;HR align="right"&gt;</code> | выравнивание влево, по центру, вправо, по ширине:<br><code>align = "left" align = "center"</code><br><code>align = "right" align = "justify"</code> |
| 2. | <code>&lt;HR width="30%"&gt;</code>   | длина линии в процентах и пикселях<br><code>width =</code> Длина в пикселях<br><code>width =</code> Длина в процентах % от ширины экрана            |
| 3. | <code>&lt;HR size="6"&gt;</code>      | Толщина линии                                                                                                                                       |
| 4. | <code>&lt;HR NoShade&gt;</code>       | Отмена объемности                                                                                                                                   |
| 5. | <code>&lt;HR color="red"&gt;</code>   | цвет линии, только в IE                                                                                                                             |

Пример. _8	Образец полученной страницы
<pre> &lt;HTML&gt; &lt;HEAD&gt; &lt;TITLE&gt; Горизонтальные линии &lt;/title&gt; &lt;/head&gt; &lt;BODY&gt; Длина линии - 30% от ширины экрана. Использование выравнивания линии по правому краю, по центру и по левому краю &lt;HR width="30%" align="right"&gt; &lt;HR width="30%" align = "center"&gt; &lt;HR width="30%" align = "left"&gt; &lt;BR&gt; толщина линии - 2, длина - 100%      &lt;HR      SIZE=2 WIDTH=100%&gt;   толщина линии - 4, длина -50% &lt;HR SIZE=4 WIDTH=50%&gt; толщина линии - 8, длина -25% &lt;HR SIZE=8 WIDTH=25%&gt; толщина линии - 16, длина - 12%      &lt;HR      SIZE=16 WIDTH=12%&gt;   Линия, несодержащая объема &lt;HR NoShade&gt; &lt;/body&gt; &lt;/html&gt; </pre>	

### Escape последовательности (символьные объекты)

Некоторые символы, такие как "<" и ">" воспринимаются браузерами как начало и конец HTML-меток, поэтому должен существовать способ их выражения, как символьных данных внутри самого документа или в URL.

В HTML это делается с помощью &- последовательностей (их еще называют символьными объектами или escape (эскейп) – последовательностями).

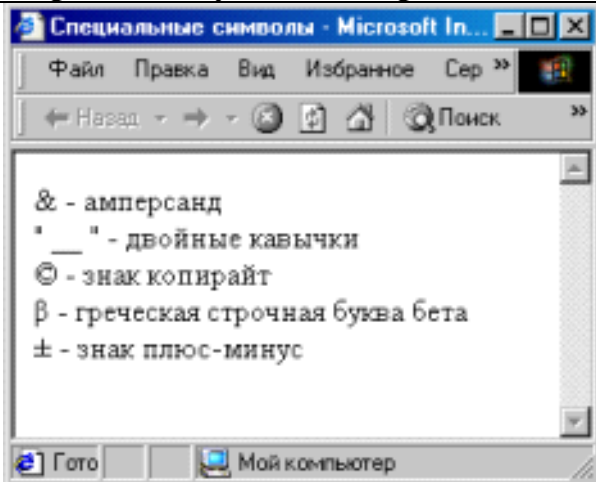
Браузер показывает на экране символ "<", когда встречается в тексте последовательность &lt; (по первым буквам английских слов less than — меньше, чем). Знак ">" кодируется последовательностью &gt; (по первым буквам английских слов greater than — больше, чем).

Принято использовать следующие нотации (соответствия) показанные в таблице 1

Таблица 1 - Символьные объекты

символ	обычное имя символа	HTML запись символа (escape последовательность)
<	символ "меньше чем", левая угловая скобка	&lt;
>	символ "больше чем", правая угловая скобка	&gt;
&	амперсанд	&amp;
"	двойные кавычки	&quot;
©	знак копирайт	&copy
®	знак зарегистрированной торговой марки	&reg
	непрерывный пробел	&nbspsp

Точка с запятой – обязательный элемент &-последовательности. Кроме того, все буквы, составляющие последовательность, должны быть в нижнем регистре (т.е., маленькие). Использование меток типа &QUOT; или &AMP не допускается.

Пример. _9	Образец полученной страницы
<pre> &lt;HTML&gt; &lt;HEAD&gt; &lt;TITLE&gt; Специальные символы &lt;/title&gt; &lt;/head&gt; &lt;BODY&gt; &amp;amp - амперсанд &lt;BR&gt; &amp;quot; __ &amp;quot; - двойные кавычки &lt;BR&gt; &amp;copy - знак копирайт &lt;BR&gt; &amp;beta; - греческая строчная буква бета &lt;BR&gt; &amp;plusmn; - знак плюс-минус &lt;/body&gt; &lt;/html&gt; </pre>	

## Задания к практической работе № 2. Форматирование текста на Web-странице по образцу

### Задание 1.

1. Создайте новую Web-страницу в текстовом редакторе Notepad ++ В элементе <TITLE> укажите название странички «Задание 1. Форматирование текста на Web-странице по образцу».
2. Отформатируйте следующий текст согласно указаниям (рисунок 1),

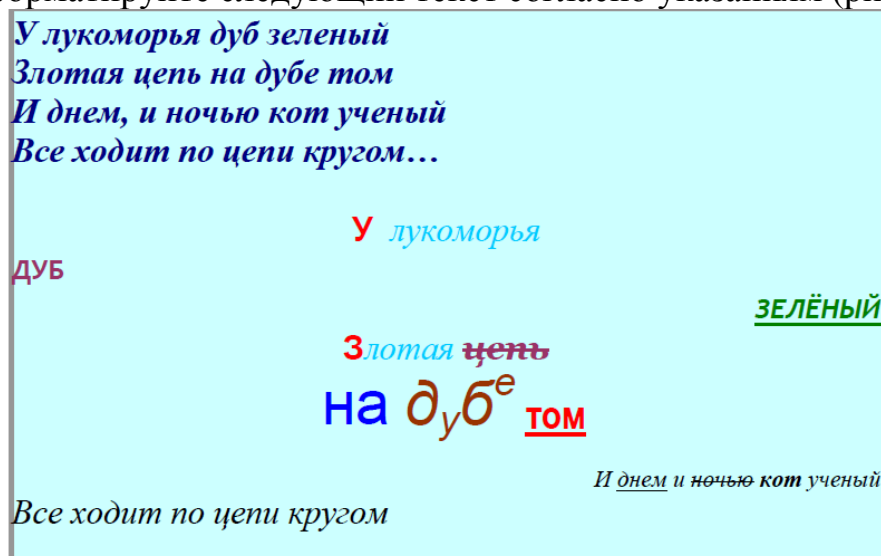


Рисунок 1

Поставить цвет фона страницы в соответствии с номером компьютера

Номер компьютера	Русское название	Английское название	RGB-код
1	Белый	white	#FFFFFF
2	Желтый	yellow	#FFFF00
3	Золотой	Gold	#FFD700
4	Оранжевый	orange	#FFA500
5	Красный	red	#FF0000
6	Пурпурный	purple	#800080
7	Каштановый	maroon	#800000
8	Светло-зеленый	lime	#00FF00
9	Зеленый	green	#008000
10	Оливковый	olive	#808000
11	Аквамарин	aqua	#00FFFF
12	Синий	blue	#0000FF

Поставить цвет четверостишья в соответствии с номером компьютера

Номер компьютера	Русское название	Английское название	RGB-код
1	Черный	black	#000000
2	Ультрамарин	navy	#000080
3	Индиго	Indigo	#4B0080

4	Черный	black	#000000
5	Серебристый	silver	#C0C0C0
6	Оливковый	olive	#808000
7	Белый	white	#FFFFFF
8	Фиолетовый	violet	#EE80EE
9	Фуксиновый	fuchsia	#FF00FF
10	Серый	gray	#808080
11	Фиолетовый	violet	#EE80EE
12	Сизый	teal	#008080

## Задание 2.

1. Создайте новую Web-страницу в текстовом редакторе Notepad ++

2. В элементе <TITLE> укажите название странички «Задание 2.

**Форматирование текста на Web-странице по образцу».**

3. Напишите HTML-файл, чтобы при его выполнении получился следующего формата текст: (*Элемент BR обеспечивает разрыв строки*)

### **ВЫБОР МИССИИ ПРЕДПРИЯТИЯ**

### **ПРЕДСТАВЛЕНИЕ О СОБСТВЕННОЙ РОЛИ**

Исходным пунктом стратегического управления предприятием является понимание своей роли перед обществом, потребителями и сотрудниками.

**Миссия** – общая цель предприятия, выражающая его отношение к своим клиентам.

Опыт показывает, что одним из существующих подходов к формулированию миссии является следующий.

*Миссия – это декларированные ведущие принципы и стандарты компании.*

### **ПРИМЕРЫ.**

*Миссия “KODAK” – “доверьте свои воспоминания Кодаку”.*

*Миссия ®“Coca-Cola” – просто протяните руку, чтобы Кола стала доступна Вам.*

## **Задание 3. Использование линий.**

- в редакторе Notepad ++

- Создайте новую Web-страницу В элементе <TITLE> укажите название странички "Задание 3. Использование линий".

- Изобразите радугу, используя горизонтальные линии различного цвета (толщина 10 пикселей, длина 50% от ширины экрана, и выровняйте ее по центру) (рисунок 2)..

## **Образец полученной страницы**

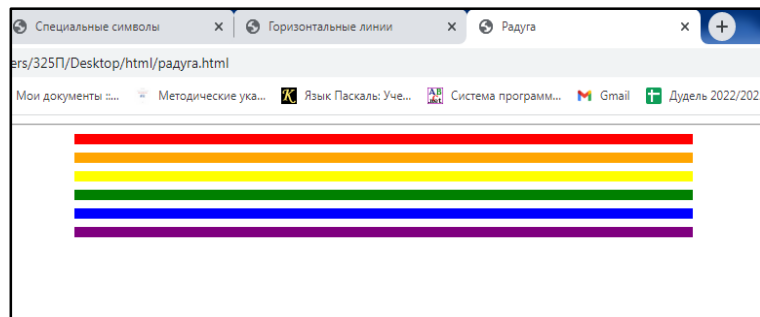


Рисунок 2

#### Задание 4.

1. Создайте новую Web-страницу в текстовом редакторе Notepad ++
2. В элементе <TITLE> укажите название странички **"Задание 4. Домашняя страничка (ваше имя и фамилия)"**.
3. Напишите HTML-файл, чтобы при его выполнении получился текст следующего содержания: укажите ваши фамилию, имя, отчество, год рождения, название вашего учебного заведения, название специальности, группу. Отрадите в нем также следующие сведения: *мой город и что мне нравится в нем, мои друзья*. (рисунок 3)

Для оформления используйте различные шрифты, стили и способы выравнивания, а также горизонтальные линии.

#### Образец полученной страницы

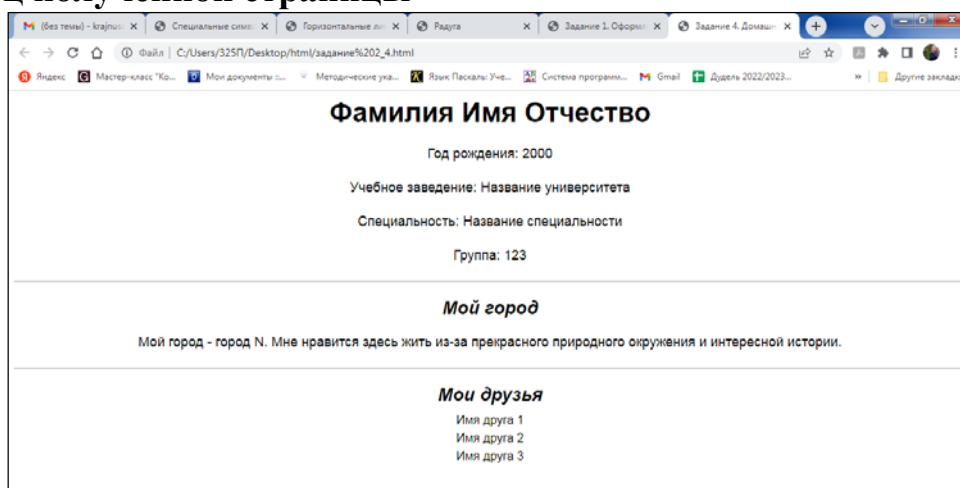


Рисунок 3

## Результат выполнения практической № 2

папка	Первые шаги _ФИО
файл	1. Задание2_1.html 2. Задание2_2.htm. 3. Задание3_3.htm. 4. Задание3_4.htm.

### Контрольные вопросы

1. Дайте характеристику форматированию абзаца.
2. Дайте характеристику форматированию шрифта.
3. Логические стили в HTML - документах.
4. Комментарии в HTML-документах.
5. Горизонтальные линии.
6. Escape последовательности.



### Практическая работа № 3. Управление цветом

**Цель:** научиться создавать цветной фон Web-страницы, использовать шрифт различного цвета.

С помощью цвета вы можете оживить любую Web-страницу, сделать ее более выразительной и удобной для просмотра. Кодирование цвета используется для раскрашивания шрифтов, горизонтальных линий и фона, других элементов странички. Цвета обозначаются английскими названиями или числовыми шестнадцатеричными кодами.

Самый простой способ определить цвет написать его название на английском языке. Так, например, задается красный цвет шрифта в элементе **<FONT>**:

**Пример.**

`<FONT color="red"> Текст красного цвета </font>`

В таблице 2 представлены все допустимые названия цветов. На самом же деле цвет определяется не названием, а так называемым RGB-кодом. Любой цвет представляется в этом случае как комбинация красного (R), зеленого (G) и синего (B) цветов, взятых в определенной пропорции. Цвет задается в шестнадцатеричной системе счисления шестью цифрами (от 0 до F) первые две цифры - красный цвет следующие две цифры - зеленый цвет последние две цифры - синий цвет. Доля каждой цветовой составляющей определяется интенсивностью цвета.

Таблица 2 – Названия цветов и RGB-код

Русское название	Английское название	RGB-код
Белый	white	#FFFFFF
Желтый	yellow	#FFFF00
Золотой	Gold	#FFD700
Оранжевый	orange	#FFA500
Красный	red	#FF0000
Пурпурный	purple	#800080
Каштановый	maroon	#800000
Светло-зеленый	lime	#00FF00
Зеленый	green	#008000
Оливковый	olive	#808000
Аквамарин	aqua	#00FFFF
Синий	blue	#0000FF
Сизый	teal	#008080
Ультрамарин	navy	#000080
Индиго	Indigo	#4B0080
Фуксиновый	fuchsia	#FF00FF
Фиолетовый	violet	#EE80EE
Серебристый	silver	#C0C0C0
Серый	gray	#808080
Черный	black	#000000

Если вы внимательно посмотрите на коды, приведенные в таблице, то обнаружите, что для формирования стандартных цветов используются или крайние значения интенсивности базового цвета 00 и FF, или среднее значение 80. Многие современные приложения имеют средства для работы с цветом, предоставляя пользователю возможность, выбрав в палитре цвет, увидеть его численные характеристики. И наоборот, задав численные значения, получить новый оттенок. Нельзя, правда, сказать, что все подобные программы совместимы между собой в смысле генерации цвета. Создав некоторый оттенок в одной программе, а потом, задав его RGB-код в другой, вы не обязательно получите тот же результат. Цветовые нюансы для Web-страниц лучше всего проверять на самих страницах.

Еще один аспект применения цвета. Выше упоминалось, что элемент HR, создающий горизонтальную линию, допускает использование ряда атрибутов. С их помощью линию можно превратить в цветной прямоугольник. Вот, например, прямоугольник светло-зеленого цвета, выровненный влево, высотой 20 и шириной 18 пикселей:

**Пример.**

```
<HR color="lime" size=20 width=18 align="left">
```

Подобные изображения можно применять для разделения частей страницы или в качестве маркеров для списков. Теоретически их можно использовать и в гиперссылках (как деталь, на которой надо щелкать мышью), но это не совсем удобно: элемент HR всегда размещается на отдельной строке.

**Цветовая гамма HTML-документа**

Цветовая гамма HTML-документа определяется атрибутами, размещенными внутри метки <BODY>. Вот список этих атрибутов:

**bgcolor** – определяет цвет фона документа;

**Пример**

```
<BODY bgcolor="#FFFFFF"> ...</body>
```

Насыщенность красным, зеленым и синим – одинакова – FF (это шестнадцатеричное представление числа 255). Результат — белый цвет.

**text** – определяет цвет текста документа;

**Пример.**

```
<BODY text="#0000FF ">
```

Это значит, что весь текст страницы будет синим, кроме текста, для которого мы специально прописали <FONT></font> (если цвет в <BODY> не задавать, то по умолчанию он будет черным).

**link** – определяет цвет выделенного элемента текста, при нажатии на который происходит переход по гипертекстовой ссылке.

**vlink** – определяет цвет ссылки на документ, который уже был просмотрен ранее.

**alink** – определяет цвет ссылки в момент, когда на нее указывает курсор мыши и нажата ее правая кнопка, то есть непосредственно перед переходом по ссылке.

**link** = “#FF0000” - цвет гипертекстовой ссылки. Насыщенность красным — FF (255), зеленым и синим — 00 (ноль). Результат — красный цвет.

Кроме того, метка <BODY> может включать атрибут **background** = "[имя файла]", который задает изображение, служащее фоном для текста и других

изображений. Как и любое другое изображение, фон должен быть представлен в формате GIF (файл с расширением \*.gif) или JPEG (файл с расширением \*.jpg или \*.jpeg).

Браузеры заполняют множественными копиями изображения-фона всё пространство окна, в котором открыт документ, подобно тому, как при строительстве большие пространства стен покрывают маленькими (и одинаковыми) плитками.

Важно отметить, что цвет фона и изображение-фон никак не отображаются на бумаге при выводе HTML-документа на печать. Из этого есть одно важное практическое следствие: **старайтесь не использовать текст белого цвета.**

### **Сочетание цветов**

Сочетать можно любой цвет с любым. Ну, кроме тех, которые нельзя сочетать. Для большинства людей всегда очевидно – удачен ли подбор оттенков, однако многие страдают страшной, почти неизлечимой болезнью – они рисуют ярко-желтым по ярко-бирюзовому (рисунок 4):

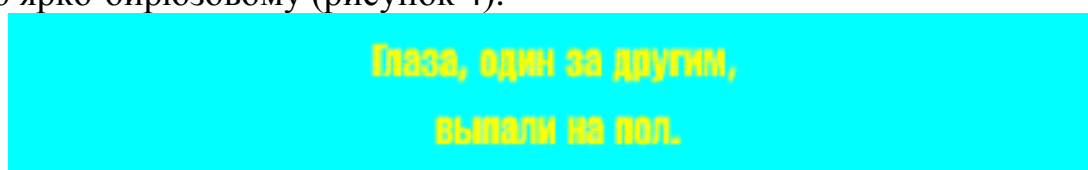


Рисунок 4

Можно прописать несложное лекарство – больному рекомендуется проверять цветосочетание в монохромном режиме (рисунок 5):

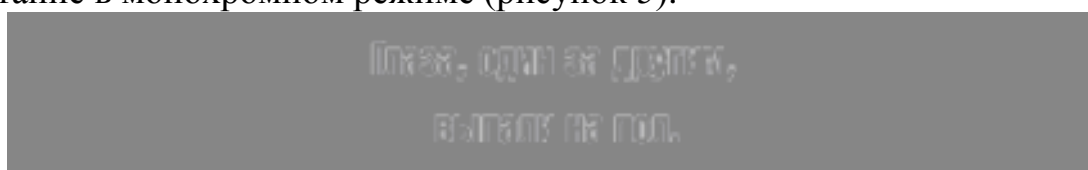


Рисунок 5

Не читается. Цвета одинаковой интенсивности вообще плохо читаются друг на друге.

Подбор цветовой гаммы всегда должен начинаться с вопроса – «для чего?». Если вы оформляете сайт, то, вероятно, у вас есть некий объем текста, который на сайте нужно разместить.

**Если есть текст, то нужно позаботиться о его читаемости.**

Исследования показывают, что человеческий глаз лучше всего воспринимает черным по белому. Настолько хорошо воспринимает, что в случае, если кто-то все-таки не увидел, ему незамедлительно сообщают: «Написано же черным по белому!!!».

Белый на черном тоже имеет право на существование, но это худший вариант. Для больших текстов такое сочетание точно не подходит.

С каждым из этих прекрасных цветосочетаний (рисунок 6) можно сделать отличный дизайн. Точно с таким же успехом можно сделать и отвратительный. Оказывается, не все зависит от цветов.



Рисунок 6

### Задания к практической работе № 3

#### Управление цветом

##### Задание 1. Оформление цветом Web-страницы

###### Указания к выполнению

- Создайте новую Web-страницу в редакторе Notepad ++
- В элементе <TITLE> укажите название странички «Задание.

##### Оформление цветом Web-страницы.» (рисунок 7).

- В начальном элементе <BODY> используйте соответствующий атрибут, чтобы сделать цвет фона странички черным.

- Сделайте надпись белым цветом «Спокойной ночи». Выровняйте ее по середине страницы. Размер шрифта 4.

- Вставьте горизонтальную линию красного цвета.

- Сделайте надпись желтого цвета «Приятных сновидений». Выровняйте ее по середине страницы. Размер шрифта 7.

- Вставьте горизонтальную линию синего цвета толщиной 10 пикселей, длиной 50% от ширины экрана, и выровняйте ее по центру.

Спокойной

##### Образец полученной страницы

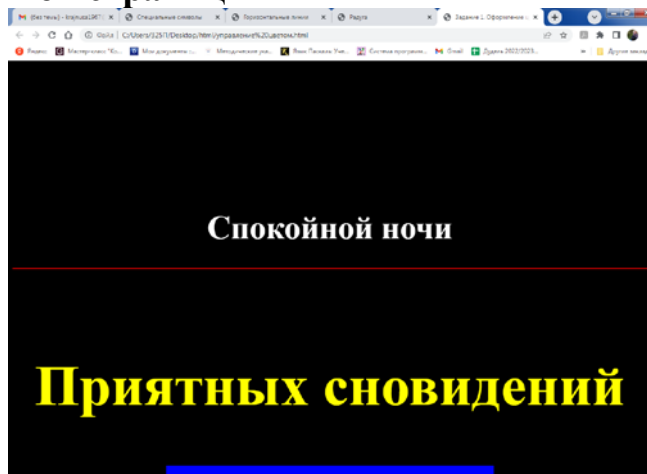


Рисунок 7

##### Результат выполнения практической № 3

папка	Первые шаги_ФИО
файл	1. Управление цветом.html

## Практическая работа № 4. Работа со списками

**Цель работы:** познакомиться с разными видами списков в HTML-документах, научиться создавать списки на Web-странице.

Очень важный элемент Web-страничек – списки.

Использование списков в качестве средства представления информации объясняется следующим.

1. Информация в виде списка позволяет большие массивы данных на отдельные четкие фрагменты, которые человеку гораздо удобнее воспринимать, чем весь поток целиком.

2. Списки позволяют создавать понятную и логичную внутреннюю структуру информационных данных ориентироваться в которой просто и удобно.

3. Использование списков удобно для отображения определенных пошаговых и прочих последовательных процессов.

Список отличается от обычного текста тем, что пользователю не надо думать о нумерации его пунктов: эту задачу программа берет на себя. Если список дополняется новыми пунктами или укорачивается, нумерация корректируется автоматически. В случае нумерованных списков программа ставит перед каждым пунктом маркеры: кружки, прямоугольники, ромбы или другие изображения. В результате список принимает удобочитаемый вид.

Спецификация HTML предусматривает три основных типа списков:

- маркированные списки;
- нумерованные списки;
- списки определений.

### **Маркированный (нумерованный - Unordered List) список**

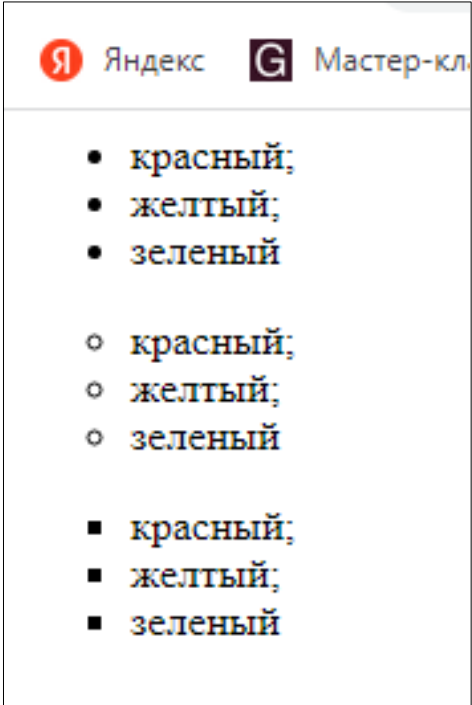
Маркированный список представляет собой нумерованный или неупорядоченный перечень элементов, для заголовка которых используются специальные маркеры. В качестве маркеров выступают специальные символы, называемые буллетами (**bullets**).

Текст, расположенный между тегами `<UL>` и `</ul>`, воспринимается как нумерованный список. Каждому элементу списка предшествует символ bullet (пуля) или графическое изображение.

Элемент `<UL>` `</ul>` является своеобразным обрамлением списка. Он позволяет отделять один список от другого.

Каждый новый элемент списка следует начинать с метки `<LI>` (List Item – пункт списка или элемент списка). У метки `<LI>` нет парной закрывающей метки. Для элемента `<UL>` возможными значениями атрибута `type` являются `disc`, `square` и `circle`, где

- маркер в виде закрашенного круга – для значения "disc"
- маркер в виде незакрашенного круга – для значения "circle"
- маркер в виде закрашенного квадрата – для значения "square"

Пример. 10	Образец полученной страницы
<pre> &lt;HTML&gt; &lt;HEAD&gt; &lt;TITLE&gt; маркерovaný список &lt;/title&gt; &lt;/head&gt; &lt;BODY&gt; &lt;UL&gt; &lt;LI&gt; красный; &lt;LI&gt; желтый; &lt;LI&gt; зеленый &lt;/ul&gt; &lt;UL type=circle&gt; &lt;LI&gt; красный; &lt;LI&gt; желтый; &lt;LI&gt; зеленый &lt;/ul&gt; &lt;UL type=square&gt; &lt;LI&gt; красный; &lt;LI&gt; желтый; &lt;LI&gt; зеленый &lt;/ul&gt; &lt;/body&gt; &lt;/html&gt; </pre>	

**После каждого пункта списка осуществляется перенос строки, причем без указания каких-либо структурных тегов.**

Если заголовок списка размещается между тегом `<UL>` и `<LI>`, то он будет отделено от начала самого списка одним переносом строки и отступом от левого края документа (как и все пункты списка). При указании заголовка перед тегом `<UL>` браузер создает двойной перенос и размещает текст заголовка левее пунктов списка.

Отличительной особенностью маркированных списков является возможность использования вместо стандартных HTML-маркеров свои собственные графические изображения. Для указания браузеру того, что в качестве маркера будет использовано графическое изображение, внутри тега `<UL>` вместо указателя элемента `<LI>` размещается обыкновенная HTML-конструкция описания графических объектов.

**Пример.**

```

<UL>
  <IMG src = "marker.gif"> красный
  <IMG src = "marker.gif"> желтый
  <IMG src = "marker.gif"> зеленый
</UL>

```

## Нумерованные списки

Нумерованный список часто называется упорядоченным и представляет собой определенную последовательность элементов.

Нумерованные списки (Ordered List) отмечают тегами `<OL>... </ol>` внутри которых располагаются пункты перечня информационных данных, за указание которых отвечает специальный тег `<LI>`. Чтобы программа правильно расставила номера вместо символов, выделяющих новый элемент, используются арабские или римские числа или буквы латинского алфавита.

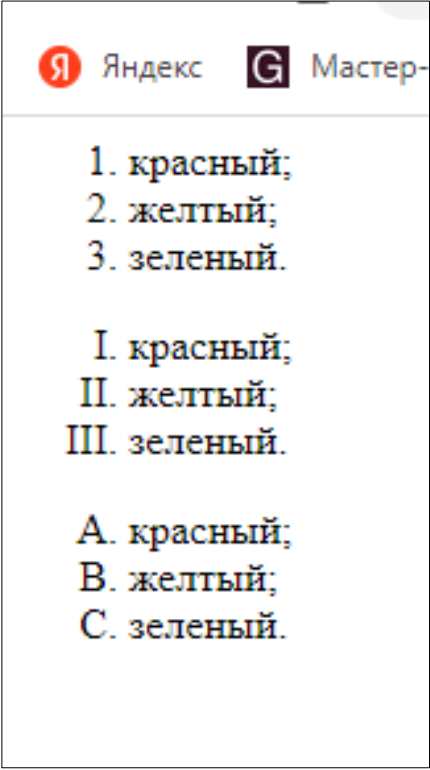
Тег составления нумерованных списков `<OL>` может включать атрибуты `type`, `start` и `compact`.

Способ нумерации задается при помощи атрибута `type`.

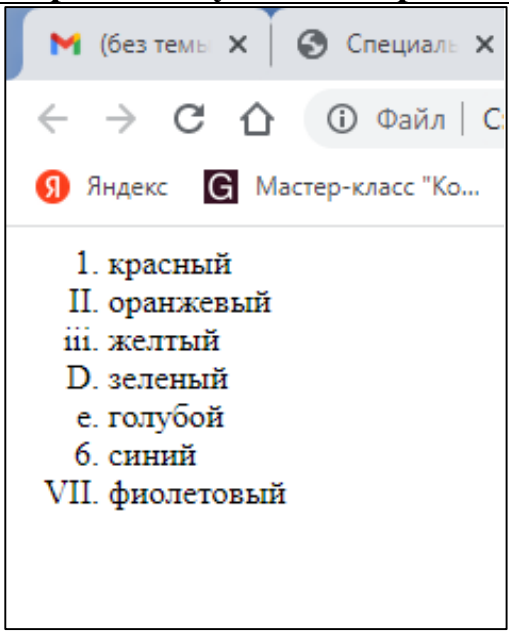
### Атрибут Стиль нумерации

`type="1"` арабские цифры 1, 2, 3, ... `type="i"` римские цифры в нижнем регистре i, ii, iii, ... `type="I"` римские цифры в верхнем регистре I, II, III, ... `type="a"` латинские буквы нижнего регистра a, b, c, ... `type="A"` латинские буквы верхнего регистра A, B, C, ...

Типом нумерации, используемым по умолчанию, является `type="1"`, поэтому при создании нумерованных списков такого типа необязательно создавать значения атрибута `type` (сам параметр также можно опустить).

Пример. 11	Образец полученной страницы
<pre>&lt;HTML&gt; &lt;HEAD&gt; &lt;TITLE&gt; нумерованный список &lt;/title&gt; &lt;/head&gt; &lt;BODY&gt; &lt;OL&gt; &lt;LI&gt; красный; &lt;LI&gt; желтый; &lt;LI&gt; зеленый. &lt;/ol&gt; &lt;OL type="I"&gt; &lt;LI&gt; красный; &lt;LI&gt; желтый; &lt;LI&gt; зеленый. &lt;/ol&gt; &lt;OL type="A"&gt; &lt;LI&gt; красный; &lt;LI&gt; желтый; &lt;LI&gt; зеленый. &lt;/ol&gt; &lt;/body&gt; &lt;/html&gt;</pre>	

Возможно смешанное использование различных типов нумерации в пределах одного списка. Однако браузеры по-разному реагируют на такое смешение стилей.

Пример. 12	Образец полученной страницы
<pre> &lt;HTML&gt; &lt;HEAD&gt; &lt;TITLE&gt;  Использование  разных типов  нумерации  в  одном списке&lt;/title&gt; &lt;/head&gt; &lt;BODY&gt; &lt;OL type="1"&gt; &lt;LI&gt; красный &lt;LI type="I"&gt; оранжевый &lt;LI type="i"&gt; желтый &lt;LI type="A"&gt; зеленый &lt;LI type="a"&gt; голубой &lt;LI type="1"&gt; синий &lt;LI type="I"&gt; фиолетовый &lt;/ol&gt; &lt;/body&gt; &lt;/html&gt; </pre>	

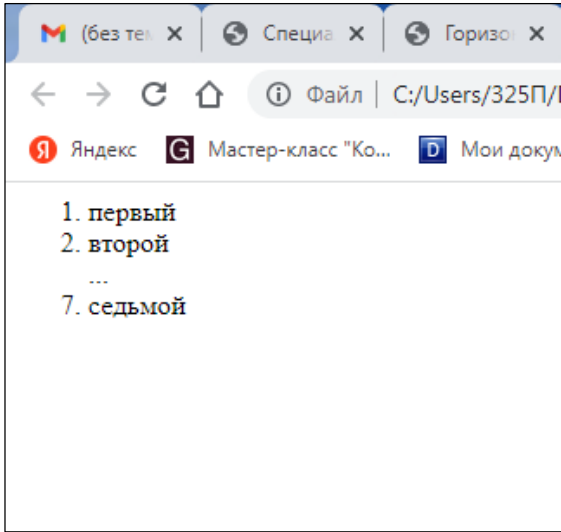
Параметр `start` позволяет начинать нумерованный список не с первой позиции. При этом значение этого параметра может быть только числовой эквивалент конкретного типа списка. Например, чтобы начать список, созданный на основе нумерации большими латинскими буквами, с буквы F, надо указать следующую HTML-конструкцию:

`<OL type="A" start="6">`

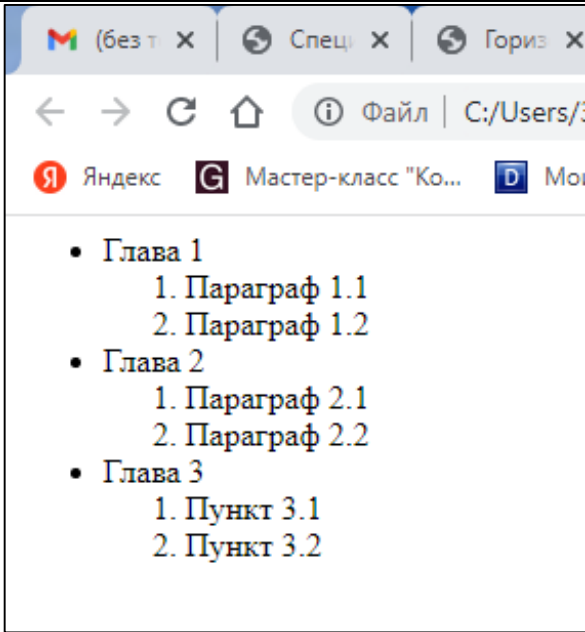
Параметр **compact** изначально был предусмотрен для списков, отображение которых должно производиться в компактном виде, например, с использованием меньшего размера шрифта, меньшего отступа от края страницы или меньшего расстояния между пунктами списка. Однако на практике этот параметр почти не используется, поскольку наиболее распространенные браузеры никак не реагируют на его применение.

Иногда может возникнуть ситуация, когда список содержит определенный разрыв в последовательности представления пунктов. Для этой цели используется специальный параметр тега `<LI>` – `value`.



Пример. 13	Образец полученной страницы
<pre> &lt;HTML&gt; &lt;HEAD&gt; &lt;TITLE&gt; нумерованный список с разрывом последовательности&lt;/title&gt; &lt;/head&gt; &lt;BODY&gt; &lt;OL&gt; &lt;LI&gt; первый &lt;LI&gt; второй &lt;BR&gt; ... &lt;LI value=7&gt; седьмой &lt;/ol&gt; &lt;/body&gt; &lt;/html&gt; </pre>	

**Вложенный список (Nested Lists)** может содержать в себе элемент или целый список любого вида. Вложенные списки очень удобны при подготовке разного рода планов и оглавлений. Число уровней вложенности в принципе не ограничено, однако злоупотреблять вложенными списками все же не следует. Необходимо быть особенно внимательным при расположении флагов начала и конца каждого списка! Чтобы не ошибиться, удобно записывать элементы вложенной конструкции с небольшим отступом.

Пример. 14	Образец полученной страницы
<pre> &lt;HTML&gt; &lt;HEAD&gt; &lt;TITLE&gt;Вложенные списки&lt;/title&gt; &lt;/head&gt; &lt;BODY&gt; &lt;UL&gt; &lt;LI&gt; Глава 1 &lt;OL&gt; &lt;LI&gt; Параграф 1.1 &lt;LI&gt; Параграф 1.2 &lt;/ol&gt; &lt;LI&gt; Глава 2 &lt;OL&gt; &lt;LI&gt; Параграф 2.1 &lt;LI&gt; Параграф 2.2 &lt;/ol&gt; &lt;LI&gt; Глава 3 &lt;OL&gt; &lt;LI&gt;Пункт 3.1 &lt;LI&gt;Пункт 3.2 &lt;/ol&gt; &lt;/ul&gt; </pre>	

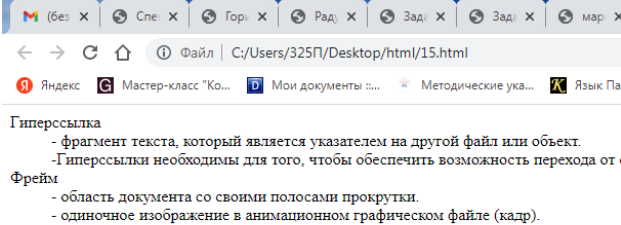
</body> </html>	
--------------------	--

**Списки определений** – это особый тип структуризации информационных данных, идеально подходящий для отображения терминологических и толковых словарей, а также различных справочников средствами HTML.

**<DL> ... </DL>**

Список определений несколько отличается от других видов списков. Вместо меток <LI> в списках определений используются метки <DT> (от английского definition term — определяемый термин) и <DD> (от английского definition — определение определения).

Каждый пункт списка может быть дополнен одним или несколькими блоками текста при помощи тега (тегов) DD. Каждый блок автоматически размещается с новой строки. Термин «определение» носит условный характер. Абзацы, размещенные в списке, могут быть определениями, дополнениями, разъяснениями пунктов. По сути, пункт представляет собой заголовок, а определение — произвольный текст под заголовком.

Пример. 15	Образец полученной страницы
<pre> &lt;HTML&gt; &lt;HEAD&gt; &lt;TITLE&gt;Списки определений&lt;/title&gt; &lt;/head&gt; &lt;BODY&gt; &lt;BASEFONT SIZE=3&gt; &lt;DL&gt; &lt;DT&gt;Гиперссылка &lt;DD&gt;- фрагмент текста, который является указателем на другой файл или объект. &lt;DD&gt;- Гиперссылки необходимы для того, чтобы обеспечить возможность перехода от одного документа к другому. &lt;DT&gt;Фрейм &lt;DD&gt;- область документа со своими полосами прокрутки. &lt;DD&gt;- одиночное изображение в анимационном графическом файле (кадр). &lt;/dl&gt; &lt;/body&gt; &lt;/html&gt; </pre>	

Если определяемые термины достаточно коротки, можно использовать модифицированную открывающую метку **<DL compact>**.

Пример. 16	Образец полученной страницы
<pre> &lt;HTML&gt; &lt;HEAD&gt; &lt;TITLE&gt;Списки определений&lt;/title&gt; &lt;/head&gt; &lt;BODY&gt; &lt;BR&gt; &lt;DL compact&gt; &lt;DT&gt;Москва &lt;DD&gt; – столица нашей Родины &lt;DT&gt;Екатеринбург &lt;DD&gt; – город, в котором я живу. &lt;/dl&gt; &lt;/body&gt; &lt;/html&gt; </pre>	

### Список-меню

**<MENU>** (Menu List) используется для представления информации в виде списка меню, в котором каждая запись занимает одну строку. Это позволяет сделать список более компактным по сравнению с конструкцией **<UL>...</UL>**. Формат такого списка зависит от используемой программы просмотра HTML-документа, однако многие программы просмотра используют **<MENU>** как "синоним" **<UL>**. Каждому элементу списка на экране обычно предшествует символ "пуля".

Пример 17	Образец полученной страницы
<pre> &lt;HTML&gt; &lt;HEAD&gt; &lt;TITLE&gt;Список-меню&lt;/title&gt; &lt;/head&gt; &lt;BODY&gt; &lt;BR&gt; &lt;MENU&gt; &lt;LI&gt;Это - первый элемент списка-меню. &lt;LI&gt;Второй элемент. &lt;LI&gt;Третий элемент. &lt;/menu&gt; &lt;/body&gt; &lt;/html&gt; </pre>	

## Задания к практической работе № 4. Работа со списками

### Задание 1. Создание нумерованных списков.

1. Создайте новую Web-страницу в текстовом редакторе Notepad ++ (название файла - **нумерованный список.html**)
2. В элементе <TITLE> укажите название странички "Задание 1. Создание нумерованных и нумерованных списков на Web-странице по образцу".
3. Создание нумерованных и нумерованных списков на Web-странице по образцу".
4. Отформатируйте следующий текст согласно указаниям, данным в скобках курсивом (рисунок 8).

#### **Солнце должно быть:**

*(Шрифт размером 4, красного цвета, выравнивание по центру)*

1. Теплым.
2. Круглым.
3. Желтым.

#### **Снег должен быть:**

*(Шрифт размером 10, синего цвета, выравнивание по левому краю)*

- A. Белым.
- B. Холодным.
- C. Пушистым.

*(Шрифт обоих, списков черный, размер 3).*

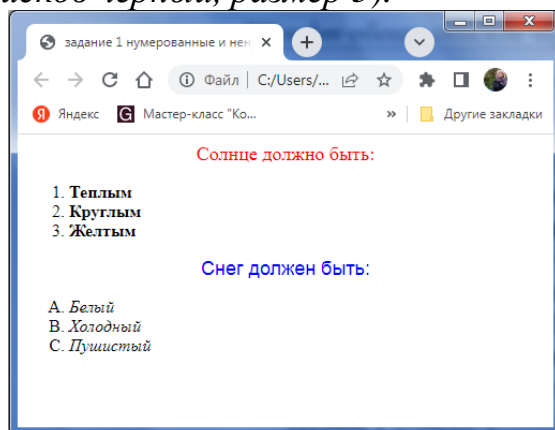


Рисунок 8

**Задание 2.** Создание нумерованных и маркированных списков, списков определений. 1. Создайте новую Web-страницу в текстовом редакторе Notepad ++ (название файла - **маркированный список.html**)

2. В элементе <TITLE> укажите название странички "**Задание 2. Создание нумерованных и нумерованных списков на Web-странице по образцу**".

3. Создайте Ваш HTML-документ, который будет выглядеть следующим образом: **Списки**

#### **Ненумерованный список**

- Пункт 1 списка
- Пункт 2 списка
- Пункт 3 списка

#### **Вложенные списки**

- Пункт 1

1. Пункт 1.1.
2. Пункт 1.2.
- *Пункт 2*
  1. Пункт 2.1.
  2. Пункт 2.2.
- *Пункт 3*
  1. Пункт 3.1.
  2. Пункт 3.2.

### Нумерованный список

- I. Пункт 1
- II. Пункт 2
- III. Пункт 3
- IV. Пункт 4

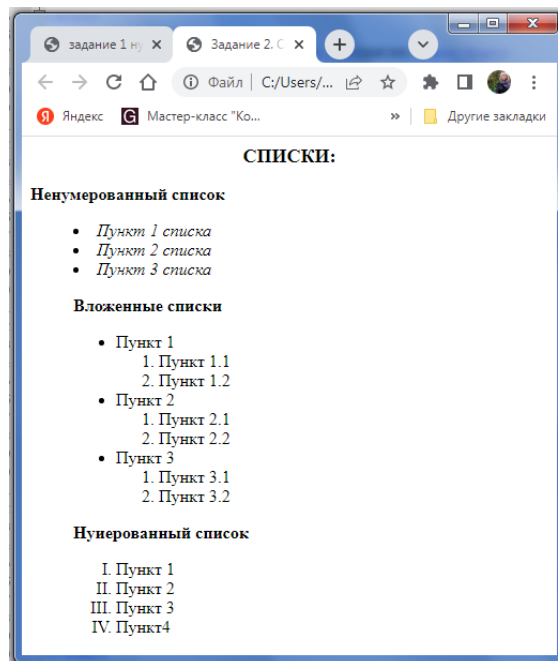


Рисунок 9

### Здание 3. Создание маркированных списков.

1. Создайте новую Web-страницу в текстовом редакторе Notepad ++. (название файла - **актеры список.html**)

В элементе <TITLE> укажите название странички "**Задание 3. Любимые актеры**". Создайте маркированный список, содержащий фамилии ваших любимых актеров.

### Задание 4. Создание маркероанных списков.

1. 1. Создайте новую Web-страницу в текстовом редакторе Notepad ++. (название файла - **радуга.html**)
2. В элементе <TITLE> укажите название странички "**Задание 4. Цвета радуги**".
3. Создайте нумерованный список, перечисляющий цвета радуги (рисунок 10).

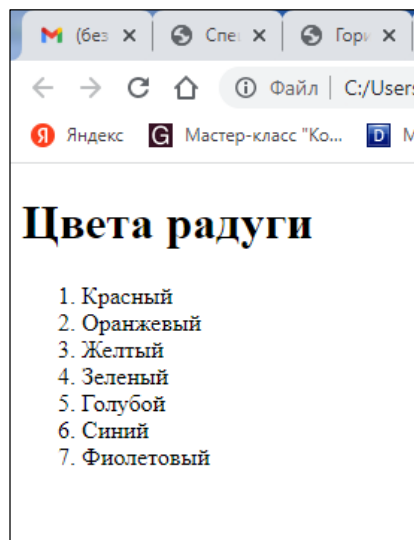


Рисунок 10

**Задание 5.** Создание списка определений.

1. 1. Создайте новую Web-страницу в текстовом редакторе Notepad ++(название файла – **породы собак.html**)
2. В элементе <TITLE> укажите название странички **"Задание 5. Породы собак"**.
3. Создайте список определений, содержащий название и описание породы собак (рисунок 11).

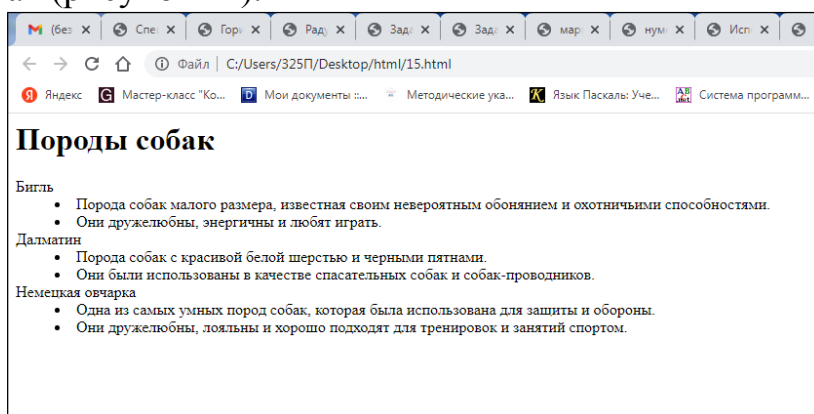


Рисунок 11

**Задание 6.** Создание списка определений.

1. 1. Создайте новую Web-страницу в текстовом редакторе Notepad ++(название файла - **автомобили.html**)
2. В элементе <TITLE> укажите название странички **"Задание 6. Автомобили"**.
3. Создать 3-уровневый список. 1 уровень (верхний) – маркированный список стран; 2 уровень – нумерованный список автомобильных фирм; 3 уровень – маркированный список автомобилей (количество элементов на каждом из уровней – не меньше 2). Создать список определений (любой, количество элементов не меньше 3) (риунок 12).

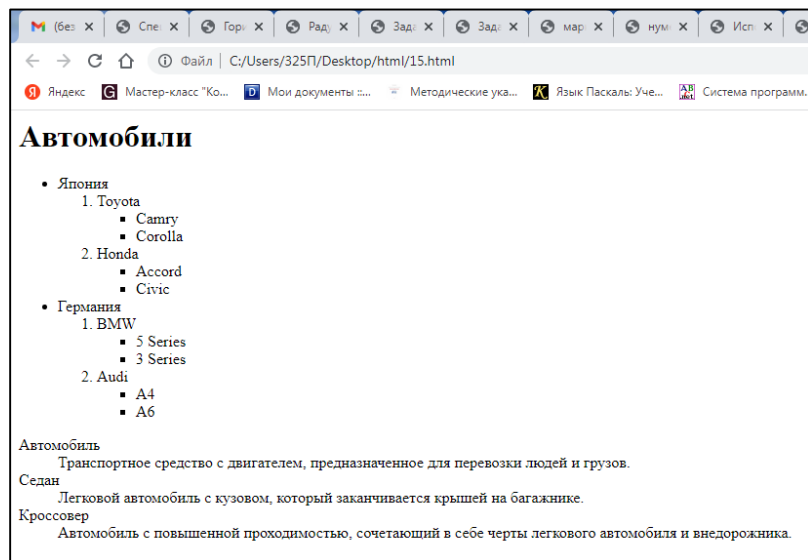


Рисунок 12

**Задание 7.** Создание списка определений.

1. 1. Создайте новую Web-страницу в текстовом редакторе Notepad ++(название файла - **принтеры.html**)
2. В элементе <TITLE> укажите название странички "**Задание 7. Принтеры**".
3. Создать 3-уровневый список. 1 уровень (верхний) – нумерованный список стран; 2 уровень – маркированный список фирм, производящих принтеры; 3 уровень – нумерованный список принтеров (количество элементов на каждом из уровней – не меньше 2) (риунок 13).

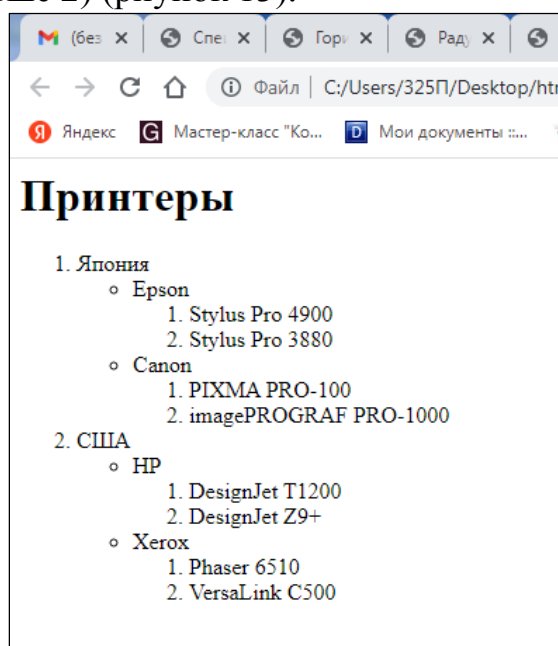


Рисунок 13

**Задание 8.** Создание списка определений.

1. 1. Создайте новую Web-страницу в текстовом редакторе Notepad ++(название файла - **мониторы.html**)
2. В элементе <TITLE> укажите название странички "**Задание 8. Мониторы**".

3. Создать 3-уровневый список. 1 уровень (верхний) – маркированный список стран; 2 уровень – нумерованный список фирм, производящих мониторы; 3 уровень – нумерованный список мониторов (количество элементов на каждом из уровней – не меньше 2).

4. Создать список определений (любой, количество элементов не меньше 3) (рисунок 14)

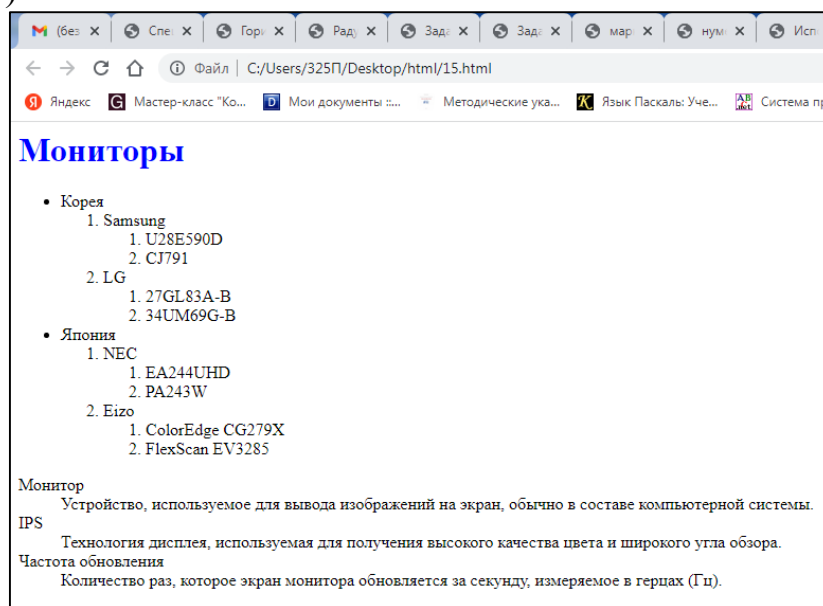


Рисунок 14

папка	<b>Первые шаги _ФИО</b>	
файл	1. <b>Задание _1.html</b> 2. <b>Задание _2.htm.</b>	<b>Выполняют все</b>
	<b>актеры список.html</b> <b>породы собак.html</b> <b>принтеры.html</b>	<b>Выполняют четные номера ПК</b>
	<b>радуга.html</b> <b>автомобили.html</b> <b>мониторы.html</b>	<b>Выполняют нечетные номера ПК</b>

### Контрольные вопросы

1. Перечислите основные типы списков в HTML-документах.
2. Охарактеризуйте маркированный (нумерованный) список.
3. Дайте характеристику нумерованных списков.
4. Дайте характеристику вложенным спискам.
5. Списки определений в HTML-документах.



## **Практическая работа № 5. Вставка графических изображений**

**Цель:** научиться размещать графические изображения на Web-странице, и изменять их размер и размещать горизонтальные линии на странице. Использовать карты изображения.

### **Роль графики**

Как интересный журнал или проспект теряет в своей привлекательности без цветных иллюстраций, так любой HTML-документ кажется сухим и невзрачным без использования графики. Значение графических изображений в аспекте создания электронных документов нельзя переоценить – реклама и коммерческие предложения компаний и юридических лиц

становятся более выразительными и яркими, иллюстрации и схемы способны превратить скучный перечень услуг или расценок в интересный информативный материал, любой художественный рассказ или произведение будет восприниматься легче и естественнее при наличии картинок или фотографий.

Однако всегда и во всем следует помнить о чувстве меры. HTML-документ, перенасыщенный иллюстративным материалом, будет неоправданно отвлекать внимание пользователя от истинного содержания страницы – информации. К тому же чрезмерное увлечение графикой влечет за собой увеличение времени загрузки электронного документа, что может сказаться на общей оценке посетителем вашего интернет-ресурса.

Также необходимо четко и внимательно отбирать графический материал для последующего размещения в HTML-документах, следить за его соответствием тематике вашего интернет-ресурса в целом и содержанию отдельной страницы в частности.

Следует помнить о том, что графика призвана привлечь внимание посетителя, заострить его интерес на конкретных моментах или формулировках, но ни в коем случае не должна отвлекать от основного содержания Web-сайта. Разумеется, данное утверждение имеет ряд оговорок и исключений из правил (например, в отношении компьютерных галерей и прочих ресурсов, где ставка сделана именно на графические изображения). И, тем не менее, в большинстве случаев необходимо строго следить за количеством графики на ваших HTML документах.

### **Характеристика графических стандартов**

Любая графическая информация может храниться в двух основных форматах – векторном и растровом.

Форматом, избранным для демонстрации Web-графики в Интернете, стал растровый формат, поскольку относительно малый размер мониторов пользователей не позволяет выводить на экран изображения больших размеров.

На сегодняшний день для создания графических изображений, предназначенных для размещения в электронных документах, используются три основных стандарта: GIF, JPEG и PNG.

### **Вставка графических изображений**

Чтобы вставить изображение, нужно только иметь это самое изображение в формате GIF (файл с расширением \*.gif) или JPEG (файл с расширением \*.jpg или \*.jpeg) и одну строчку в HTML-тексте.

**<IMG>** – элемент для создания ссылки на графический файл (image). Он не содержит конечного тэга – вся необходимая информация задается при помощи атрибутов. Этот элемент является универсальным: с его помощью можно использовать изображения в гиперссылках,

вставлять картинки в таблицы, просто размещать рисунки на Web-странице, решать задачи дизайна и т. д.

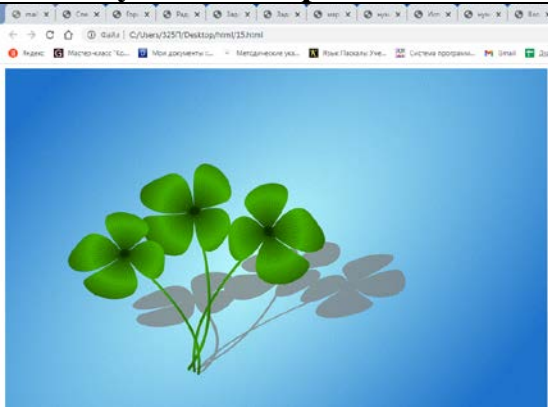
Необходимым атрибутом является

**SRC** – указатель на графический файл: SRC = "Ссылка на файл".

Допустим, нам нужно включить в документ изображение, записанное в файл picture.gif, находящееся в одном каталоге с HTML-документом. Тогда строчка будет вот такая:

**<IMG SRC="picture.gif">**

Очень полезным атрибутом является **ALT**. Он позволяет выводить текст в тех местах, где должны располагаться рисунки. Страница может загружаться достаточно долго, и пока графические файлы на «подходе», пользователь должен видеть, какие изображения он сможет получить.

Пример 18	Образец полученной страницы
<pre>&lt;HTML&gt; &lt;HEAD&gt; &lt;TITLE&gt;Вставка изображения&lt;/title&gt; &lt;/head&gt; &lt;BODY&gt; &lt;IMG SRC="картинка.gif" ALT="Картинка"&gt; &lt;/body&gt; &lt;/html&gt;</pre>	

Встретив такую метку, браузер покажет на экране текст “Картинка” и начнет загружать на его место картинку из файла picture.gif. Атрибут ALT может оказаться необходимым для старых браузеров, которые не поддерживают изображений, а также на случай, если у браузера отключена автоматическая загрузка изображений (при медленном подключении к Интернет это делается для экономии времени).

Файл, содержащий изображение, может находиться в другом каталоге или даже на другом сервере. В этом случае необходимо указать его полное имя (путь к файлу):

Можно использовать атрибуты выравнивания:

align = "bottom" — по нижнему краю;

align = "left" — влево;

align = "middle" — по центру;

align = "right" — вправо;

align = "top" — по верхнему краю.

Высоту и ширину области, в которой демонстрируется рисунок, задают при помощи атрибутов **height - высота** и **width - ширина**. В том случае, когда задается один из этих атрибутов, рисунок масштабируется таким образом, чтобы его высота или ширина соответствовали заданной. Второй размер устанавливается автоматически, в соответствующей пропорции.

Таким образом, применение только одного из атрибутов изменяет оба размера рисунка. Если задать явным образом оба атрибута, то рисунок будет масштабироваться по двум осям в соответствии с заданным размером. Масштабирование, как правило, ухудшает качество изображения.

Например, реальный размер рисунка 76x121 пикселей. Напишем два варианта отображения данного рисунка.

<IMG SRC = "picture.gif" width="76" height="121">

<IMG SRC = "picture.gif" width="176">

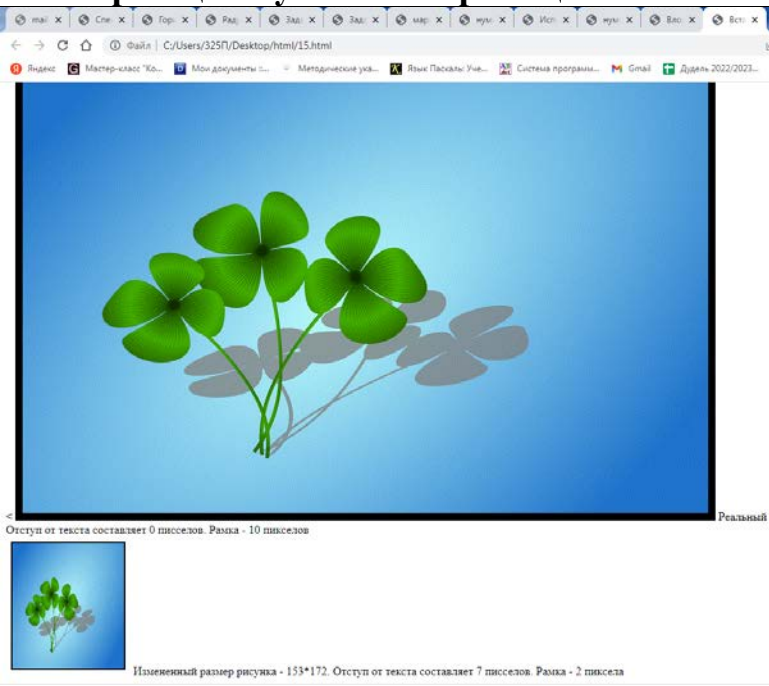
Обратите внимание, что во втором случае изменен размер рисунка (мы изменили ширину, высота будет пропорционально изменена автоматически) При этом происходит потеря качества изображения, поэтому желательно задавать эти атрибуты в соответствии с реальными размерами рисунка.

Атрибут **border** – задает рамку вокруг объекта. border="2" ширина рамки задается в пикселях. При наличии тегов **hspace** и **vspace** вокруг картинки образуется отступ в соответствующее количество пикселей от текста.

hspace = Число пикселей	Пустое пространство справа и слева от рисунка
vspace = Число пикселей	Пустое пространство выше и ниже рисунка

Полностью тэг **IMG** может выглядеть следующим образом:

<IMG src="путь к файлу рисунка" width="ширина картинки" height="высота картинки" hspace="число" vspace=" число" border=" число" align="left..." alt="подпись к рисунку">

Пример. 19	Образец полученной страницы
<pre> &lt;HTML&gt; &lt;HEAD&gt; &lt;TITLE&gt;Вставка изображения&lt;/title&gt; &lt;/head&gt; &lt;BODY&gt; &lt;IMG SRC="картинка.gif" "10"&gt; Реальный      размер рисунка      - 172*153. Отступ от текста составляет 0 пикселей. Рамка - 10 пикселей &lt;BR&gt; &lt;IMG SRC="picture.gif" width="153" height="172" hspace="7" </pre>	

<pre> vspace="7" border = "2"&gt; Измененный размер рисунка - 153*172. Отступ от текста составляет 7 пикселей. Рамка - 2 пиксела &lt;/body&gt; &lt;/html&gt; </pre>	
---------------------------------------------------------------------------------------------------------------------------------------------------------------------	--

## Задания к практической работе № 6. Вставка графических изображений

### Задание 1. Размещение графики на Web-странице

- Создайте новую Web-страницу в редакторе Notepad ++. Файл назовите «Графика 1.html»

- В элементе <TITLE> укажите название странички «Задание 1. Размещение графики на Web-странице».

- Если в тэге <BODY> не указывать цвет странички, то по умолчанию фон будет белым. Сделайте цвет фона странички оранжевым.

- Разместите на страничке любой рисунок. Для размещения изображения вам требуется указать путь к файлу в элементе <IMG>. Чтобы упростить описание пути к этому графическому файлу, скопируйте этот файл в ту же папку, в которой будет сохранена Web-страничка.

Если файл находится не в одной папке с Web-страничкой, то необходимо указать относительный путь к этому файлу.

- Сделайте подпись к рисунку с помощью атрибута **alt**.
- Создайте рамку вокруг рисунка шириной **5 пиксела**.
- Над рисунком поместите заголовок самого большого размера и выровняйте его по центру страницы.

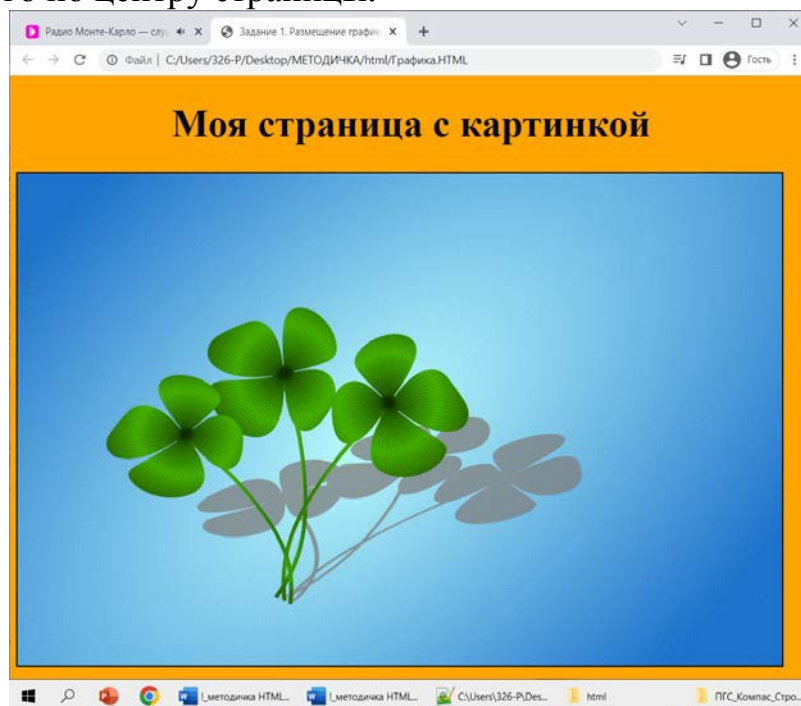


Рисунок 15

### Задание 2. Изменение размеров изображения на Web-странице.

- Создайте новую Web-страницу в редакторе Notepad ++. Файл назовите «Графика 2.html»

- В элементе <TITLE> укажите название странички «Задание 2. Изменение размеров изображения на Web-странице».

- Пусть фон странички будет зеленого цвета.
- Определите размер выбранного вами изображения.

- Вставьте это изображение на страничку, задав его размеры: **ширина 100, высота 200 пикселей**.
- Просмотрите в браузере полученную страничку. Обратите внимание на следующие вещи:
  - а. В связи с тем, что вы непропорционально изменили размеры изображения, произошло искажение рисунка;
  - б. Так как вы увеличили размер изображения, произошла потеря качества рисунка (обратите внимание на края изображения);
  - с. Для размещения изображений на Web-страничке, лучше использовать прозрачный фон рисунка.

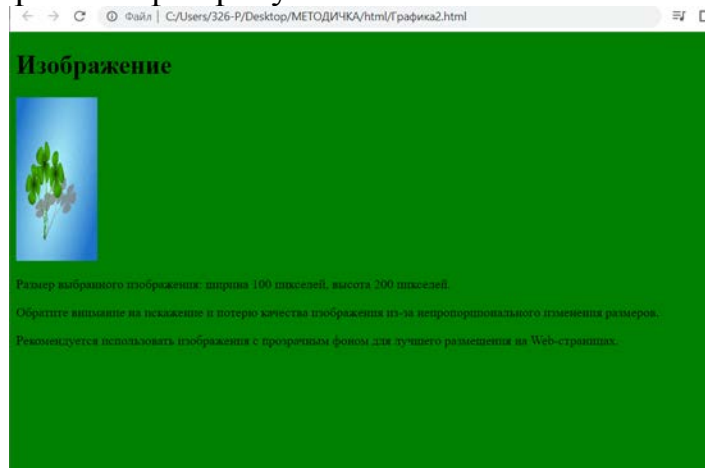


Рисунок 16

### Задание 3. Использование рисунка в качестве гиперссылки

• Выполнить задание вы можете, если успешно справились с *Заданием 3*. • На страничках `page1.htm`, `page2.htm` и `page3.htm`, вместо текстовых гиперссылок используйте рисунки.

Для этого:

• Вставьте на первую страничку одно изображение и создайте гиперссылку для перехода на вторую страницу:

```
<A href="page2.htm"> <IMG src="*.gif"> </a>
```

• самостоятельно вставьте другое изображение для перехода с первой странички на третью;

• задайте бордюр обоим изображениям, равным 10;

• Откройте в браузере первую страничку и, переходя по ссылкам со странички на страничку, убедитесь, что переходы работают верно.

• Самостоятельно вставьте изображения на страницы `page2.htm` и `page3.htm` и задайте соответствующие гиперссылки.

• Откройте в браузере первую страничку и, переходя по ссылкам со странички на страничку, убедитесь, что кнопки на всех страницах работают верно.

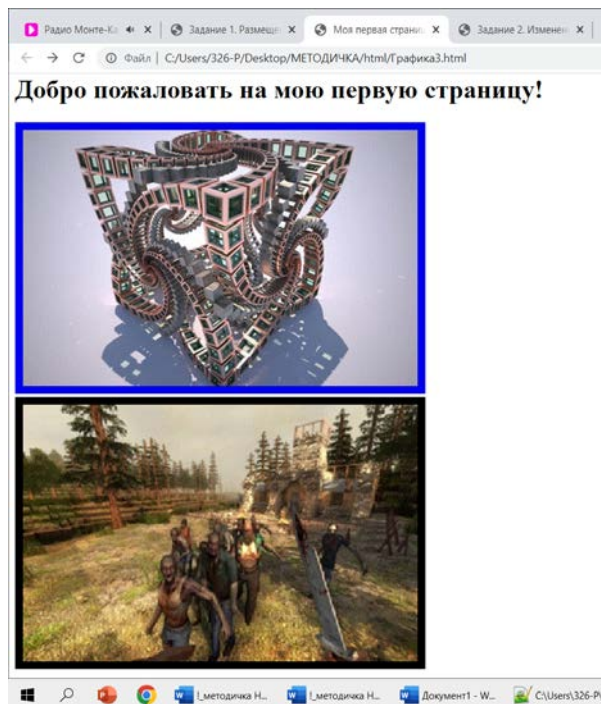


Рисунок 17

### Результат выполнения практической № 5

папка	Первые шаги _ФИО
файл	1. Графика 1.htm 2. Графика 2.htm 3. Графика 3.htm

### Контрольные вопросы

1. Роль графики. Характеристика графических стандартов
2. Вставка графических изображений в HTML-документ.
3. Использование карт изображений



## Практическая работа № 7. Таблицы HTML-документов

**Цель работы:** познакомиться с основными приемами создания таблиц на языке гипертекстов, рассмотреть основные теги и атрибуты, научиться создавать, форматировать, заполнять таблицы, проектировать структуру таблицы.

Одним из наиболее мощных и гибких средств представления информационных данных в HTML по праву являются таблицы. В повседневной жизни, решая те или иные задачи, мы часто сталкиваемся с таблицами. Однако в HTML таблицы не ограничиваются удобным средством структурирования информации. Сегодня таблица становится основой большинства электронных документов, структура которых может включать самые разнообразные элементы HTML. Таблицы, изначально избранные в качестве визуального способа представления данных, сейчас имеют гораздо более важную функцию – управление структурой HTML-документа в целом. Удобство размещения данных в таблице неоспоримо, а преимущества перед другими средствами представления информации (например, списки) позволяют считать таблицы основополагающим структурным элементом любого HTML документа.

Таблицы являются очень удобным средством форматирования данных на Web странице. Основное удобство заключается в том, что браузер берет на себя заботу о прорисовке рамки таблицы. Размер рамки может быть автоматически согласован с размером окна просмотра в браузере и, разумеется, с размером находящихся в ячейках таблицы строк текста и рисунков. Кроме этого, таблицы позволяют решать чисто дизайнерские задачи: выравнивать части страницы друг относительно друга, размещать рядом рисунки и текст, управлять цветовым оформлением и т. д. Контур таблицы показывать не обязательно, поэтому тег `<TABLE>` применим для размещения текстовых и графических компонент в разных местах страницы. При создании таблиц используется принцип вложения: внутри основного элемента таблицы (TABLE) создается ряд элементов, определяющих строки (TR), а внутри этих элементов размещаются элементы для описания каждой ячейки в строке (TD, TH).

### Основные методы создания таблиц

`<TABLE> </table>`

Внешний элемент таблицы. Он позволяет задавать общие свойства таблицы и отделяет ее структуру от остальной части Web-страницы. Если начало и конец таблицы не определены, то никакие табличные команды не будут работать.

Рассмотрим параметры этого элемента.

#### Параметр align

Таблицу можно выравнивать по горизонтали при помощи атрибута align. Данный параметр устанавливает тип выравнивания всей таблицы относительно ширины HTML документа.

align="left" – влево;

align="center" – по центру;

align="right" – вправо.



### Параметры width и height

Параметры width и height отвечают соответственно за ширину и высоту таблицы. Значение этих параметров может быть указано в процентах или пикселях: **Пример.**

<TABLE WIDTH="500" HEIGHT="50%">

**Пример:** width =250 задает таблицу шириной 250 пикселей; width = 50% задает таблицу на половину ширины страницы при любом размере изображения на экране.

В случае указания ширины или высоты в пикселях таблица будет иметь фиксированные размеры. Такая таблица при изменении размеров окна браузера останется неизменной. При использовании в качестве единицы измерения процентов ширина/высота таблицы будет варьироваться в зависимости от размеров окна браузера. В этом случае размер ячеек таблицы будет уменьшаться/увеличиваться пропорционально заявленному значению в процентах. Содержимое таблицы будет перемещаться в окне браузера соответственно изменению его размеров.

Для управления видом рамки используются два атрибута. Дело в том, что браузер создает изображение рамки, имитируя ее трехмерность (выпуклость). На рамке можно различить фронтальную и боковую наклонную грани. Кроме этого, имитируется различие в их освещенности.

### Параметр border

Параметр **border** задает ширину рамки. Если рамка не задана, то получим таблицу без рамок, тот же результат дает <TABLE border=0>. Также как и ширину можно задать в процентах и пикселях.

border = Ширина в пикселях

### Параметр bordercolor

Параметр **bordercolor** предназначен для определения цвета рамки вокруг таблицы (поддерживается только Internet Explorer). Значение указывается в шестнадцатеричном формате или в виде наименования (например, green).

### Параметр cellpadding

Параметр cellpadding задает расстояние между соседними ячейками таблицы по вертикали и горизонтали. Расстояние между соседними ячейками (как по вертикали, так и по горизонтали) всегда будет одинаково, т. к. HTML не позволяет назначать разные значения в пределах данного параметра.

cellpadding = Ширина в пикселях

Если значение этого атрибута равно нулю, рамка получается тонкой, заостренной. Если данный параметр не задан, по умолчанию задается величина, равная 2 пикселям. Атрибут полезен для размещения текста и графики именно там, где нужно.

### Параметр cellspacing

Для всех ячеек таблицы можно задать размер пустого пространства между содержимым таблицы и ее границами. Для создания и регулирования отступа между рамкой ячейки и ее содержимым (по вертикали и горизонтали) используется параметр **cellpadding**. cellpadding = Число пикселей

Задание этого атрибута делает ячейки больше. Между рамкой таблиц и данными всегда сохраняется определенное расстояние. В некоторых случаях это позволяет улучшить восприятие таблицы, сделать текст в ячейках легко читаемым.

При значении параметра `cellpadding`, равном нулю, содержимое ячейки будет вплотную прижато к рамке таблицы, что ухудшит восприятие информации (это утверждение еще более актуально для таблиц с прозрачной рамкой, т. е. значением `border="0"`, – в этом случае текст соседних ячеек сольется друг с другом).

### Параметр **unit**

Параметр **unit** тега `<TABLE>` определяет единицы измерения, используемые при указании размеров всей таблицы и ее отдельных столбцов. Может принимать три значения: • **unit = en** - это значение задает единицу измерения, равную EN-пробелу, т.е. ширине буквы n (зависит от размера шрифта);

• **unit = relative** используется для задания относительной ширины столбцов в процентах от общей ширины таблицы. (Числа при этом воспринимаются как ширина в процентах); • **unit = pixels**- это значение применяется, когда нужно точно задать ширину столбца на экране. Например, тег `<TABLE unit=pixels width=340>` сформирует таблицу шириной 340 пикселей. Так как разные мониторы имеют разное разрешение, то при задании ширины таблиц и столбцов в пикселях ячейки на экранах с более высоким разрешением могут оказаться меньше. Например, таблица шириной 640 пикселей на мониторе с разрешением 640\*768 займет весь экран, а на мониторе с разрешением 1024\*768 немногим более половины ширины экрана.

### Параметр **colspan**

Параметр **colspan** = перечисляет все столбцы и для каждого задает выравнивание и размер. Для столбца существует 5 способов выравнивания: **L** - по левому краю; **C** - по центру; **R** - по правому краю; **J** - и по левому и по правому; **D** - по десятичной точке или запятой. **Пример:**

```
<TABLE UNIT= PIXELS COLSPAN="L10 C15 R20 J25 D 30">
```

Первый столбец имеет ширину 10 пикселей и его содержимое выравнивается по левому краю; второй столбец имеет ширину 15 пикселей и его содержимое выравнивается по центру; третий столбец имеет ширину 20 пикселей и его содержимое выравнивается по правому краю; четвертый столбец имеет ширину 25 пикселей и его содержимое выравнивается по левому краю и по правому краю; пятый столбец имеет ширину 30 пикселей и его содержимое выравнивается по десятичным запятым.

### Параметр **bgcolor**

Для всей таблицы может быть задан цвет фона:

```
bgcolor="Цвет"
```

или

```
bgcolor="#RRGGBB".
```

В этом случае атрибут `bgcolor` помещается внутри элемента `<TABLE>`.

**Пример.**

```
<TABLE bgcolor="red" border="2" cellspacing="2" cellpadding="2"> <TR>
<TD>Ячейка 1</td>
<TD>Ячейка 2</td>
</tr>
</table>
```

Параметры тега <TABLE> перечислены в таблице 3

Таблица 3

Параметр	Функция
border	Создание рамки вокруг таблицы
bordercolor	Определение цвета рамки вокруг таблицы
cellspacing	Указание расстояния между соседними ячейками
cellpadding	Создание отступа от границы ячейки до ее содержимого
align	Указание типа выравнивания таблицы по ширине окна браузера
width	Определение ширины таблицы
height	Определение высоты таблицы
unit	Определяет единицы измерения, используемые при указании размеров всей таблицы и ее отдельных столбцов
colspan	Перечисляет все столбцы и для каждого задает выравнивание и размер

### <CAPTION> </caption>

Элемент для задания заголовка таблицы. Заголовок выводится на экране вне рамки таблицы. Положением заголовка можно управлять:

Align = "top" – заголовок над таблицей;

align = "bottom" – заголовок под таблицей.

### <TR> </tr>

Элемент, создающий строку таблицы (сокращение от Table Row – строка таблицы). Он не имеет конечного тега. Строка заканчивается там, где начинается следующая, то есть где стоит следующий элемент <TR>. Если в таблице два набора тегов <TR></tr>, то в ней будут две строки. Если N наборов <TR></tr>, то в таблице N строк.

### <TH> </th>

Элемент ячейки, которая является заголовком столбца или строки таблицы (Table Header, заголовок таблицы). Этот элемент должен располагаться внутри элемента <TR>. Ячейка-заголовок отличается от обычной тем, что браузер выводит текст внутри нее выделенным (как правило, полужирным) шрифтом. Для элемента ячейки предусмотрено много атрибутов.

### <TD> </td>

Этот элемент определяет обычную ячейку таблицы. Число тегов <TD></td> в строке определяет число ячеек. Строка с пятью наборами тегов <TD></td> содержит пять ячеек. Число ячеек в строке не должно быть постоянным. В одной строке может быть 5 ячеек, в другой 3.

Элементы – **<TR>**, **<TH>** и **<TD>** – не имеют конечных тегов. Функцию конечного тега выполняет следующий элемент, который определяет структуру таблицы. Однако все же рекомендуется их указывать для предотвращения ошибок, которые могут возникнуть при создании сложных вложенных таблиц.

### Цветные таблицы

#### Параметры bgcolor и background

Параметры **bgcolor** и **background** используются для работы с фоном ячеек или целых рядов таблицы. **bgcolor** позволяет задать определенный цвет ячеек, а **background** – указать путь к графическому изображению, которое будет служить фоном.

Параметр bgcolor используется для тегов **<TR>**, **<TD>** и **<TH>**. Параметр background применим только к тегам ячеек **<TD>** и **<TH>**.

#### Пример.

```
<TR>
<TD bgcolor = "red" >
</tr>
</td>
```

Параметр **bordercolor** задает цвет границы каждой ячейки таблицы, например: **<TD bordercolor=yellow> </td>**

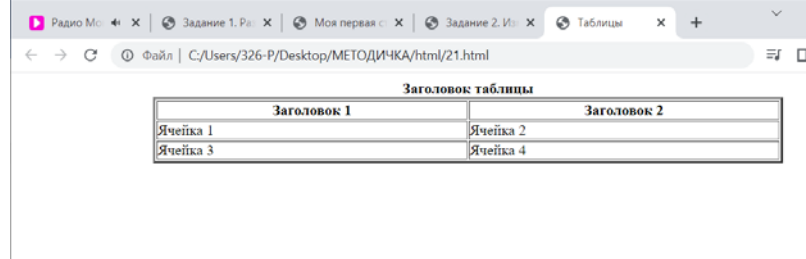
Здесь цвет не в кавычках.

Атрибуты **bordercolordark** = и **bordercolorlight** = задают два цвета для окраски краев таблицы (создает “трехмерное” изображение),

#### Пример.

```
<TR>
<TD bordercolorlight = yellow bordercolordark = blue>
```

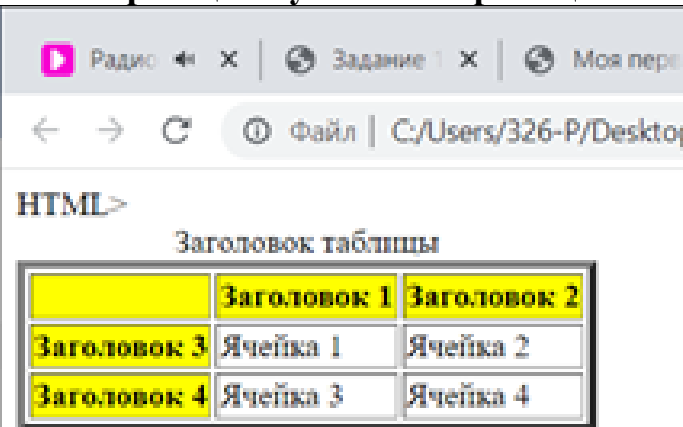
Теперь, зная, какие элементы используются для построения таблицы, можно создать простейшую таблицу:

Пример. 21	Образец полученной страницы
<pre>&lt;HTML&gt; &lt;HEAD&gt; &lt;TITLE&gt;Таблицы&lt;/title&gt; &lt;/head&gt; &lt;BODY&gt; &lt;BASEFONT SIZE=3&gt; &lt;TABLE CELLPADDING=0 CELLSPACING=3 BORDER=3 ALIGN=CENTER WIDTH=70%&gt; &lt;CAPTION VALIGN="top"&gt;&lt;B&gt;Заголовок ок таблицы&lt;/B&gt;&lt;/caption&gt; &lt;TR&gt; &lt;TH&gt;Заголовок 1&lt;/th&gt; &lt;TH&gt;Заголовок 2&lt;/th&gt;&lt;/tr&gt;</pre>	

<pre> &lt;TR&gt; &lt;TD&gt;Ячейка 1&lt;/td&gt;       &lt;TD&gt;Ячейка       2&lt;/TD&gt;&lt;/tr&gt; &lt;TR&gt; &lt;TD&gt;Ячейка 3&lt;/td&gt;       &lt;TD&gt;Ячейка 4&lt;/td&gt;&lt;/tr&gt; &lt;/table&gt; &lt;/body&gt; &lt;/html&gt; </pre>	
-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--

Первая строка таблицы содержит только ячейки-заголовки. Текст, расположенный после элементов `<TD>`, представляет собой содержимое ячейки. Таблица может форматироваться автоматически (если не заданы атрибуты), с учетом объема данных в ячейках.

Последний пример можно несколько усложнить. При необходимости можно создать заголовки и для столбцов, и для строк:

Пример. 22	Образец полученной страницы
<pre> &lt;HTML&gt; &lt;HEAD&gt; &lt;TITLE&gt;Таблицы&lt;/title&gt; &lt;/head&gt; &lt;BODY&gt; &lt;TABLE border=4 cellspacing=3&gt; &lt;CAPTION&gt; Заголовок таблицы &lt;/caption&gt; &lt;TR&gt;&lt;TH bgcolor="yellow"&gt; &lt;TH bgcolor="yellow"&gt;Заголовок 1 &lt;TH bgcolor="yellow"&gt; Заголовок 2 &lt;TR&gt;&lt;TH      bgcolor      ="yellow"&gt; Заголовок 3 &lt;TD&gt; Ячейка 1 &lt;TD&gt; Ячейка 2 &lt;TR&gt;&lt;TH      bgcolor="yellow"&gt; Заголовок 4 &lt;TD&gt; Ячейка 3 &lt;TD&gt; Ячейка 4 &lt;/table&gt; &lt;/body&gt; &lt;/html&gt; </pre>	

Обратите внимание: несмотря на то, что левая верхняя ячейка не используется, для нее задан цвет фона так же, как и для других ячеек-заголовков. Это необходимо сделать для того, чтобы рамка таблицы в этом месте была правильно прорисована.

### Основные параметры тегов `<TR>`, `<TH>` и `<TD>`

#### Параметр `align`

Если в случае использования этого параметра в теге <TABLE> вся таблица выравнивается определенным образом, то в данном случае назначается тип выравнивания для конкретной ячейки или ряда таблицы. Если необходимо, чтобы каждая ячейка содержала разные типы выравнивания, придется указывать соответствующее значение параметра align в каждом теге <TD> или <TH>. Если требуется задать один тип выравнивания для всего ряда (включающего все ячейки), значение параметра align прописывается в теге <TR>.

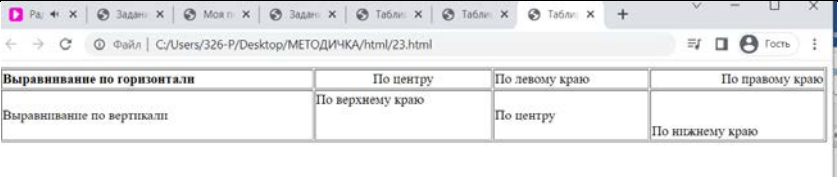
Параметр **align** определяет выравнивание текста и графики по горизонтали несколькими способами:

- а) align = left выравнивает по левому краю с учетом отступа, заданного атрибутом cellpadding; б) align = center располагает содержимое по центру;
- в) align = right выравнивает по правому краю с учетом отступа, заданного атрибутом cellpadding .

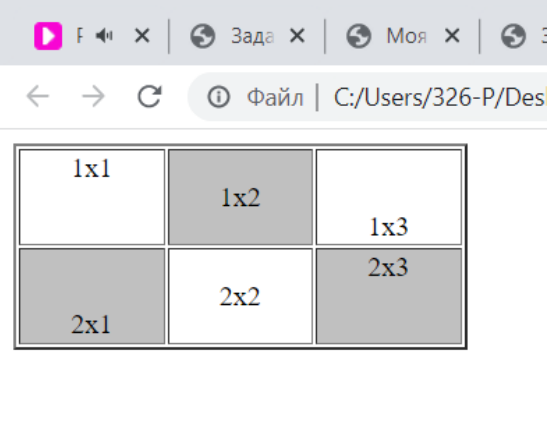
### Параметр valign

Параметр **valign** осуществляет выравнивание текста и графики по вертикали несколькими способами.

- а) valign = top - выравнивает содержимое ячейки по ее верхней границе; б) valign = middle - центрирует содержимое ячейки по ее вертикали;
- в) valign = bottom - выравнивает содержимое ячейки по ее нижней границе.

Пример. 23	Образец полученной страницы
<pre> &lt;HTML&gt; &lt;HEAD&gt; &lt;TITLE&gt;Таблицы&lt;/title&gt; &lt;/head&gt; &lt;BODY&gt; &lt;TABLE width="100%" border="1" align="center"&gt; &lt;TR&gt; &lt;TD&gt;&lt;B&gt;Выравнивание по горизонтали&lt;/b&gt;&lt;/td&gt; &lt;TD align="center"&gt; По центру &lt;/td&gt; &lt;TD align="left"&gt; По левому краю &lt;/td&gt; &lt;TD align="right"&gt; По правому краю &lt;/td&gt; &lt;/tr&gt; &lt;TR&gt; &lt;TD &lt;B&gt;Выравнивание по вертикали &lt;/b&gt;&lt;/td&gt; &lt;TD valign="top"&gt; По верхнему краю &lt;BR&gt;&lt;BR&gt;&lt;/td&gt; </pre>	

<pre> &lt;TD valign="middle"&gt; По центру &lt;/td&gt; &lt;TD valign="baseline"&gt; &lt;BR&gt;&lt;BR&gt;По нижнему краю &lt;/td&gt; &lt;/tr&gt; &lt;/table&gt; &lt;/body&gt; &lt;/html&gt; </pre>	
---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--

Пример. 24	Образец полученной страницы
<pre> &lt;HTML&gt; &lt;HEAD&gt; &lt;TITLE&gt;Таблицы&lt;/title&gt; &lt;/head&gt; &lt;BODY&gt; &lt;TABLE border=2&gt; &lt;TR&gt; &lt;TD height="55" width="80" valign="top"&gt; &lt;center&gt;1x1&lt;/center&gt; &lt;/td&gt; &lt;TD width="80" bgcolor="silver"&gt; &lt;center&gt;1x2&lt;/center&gt; &lt;/td&gt; &lt;TD width="80" valign="bottom"&gt; &lt;center&gt;1x3&lt;/center&gt; &lt;/td&gt; &lt;/tr&gt; &lt;TR&gt; &lt;TD height="55" width="80" bgcolor="silver" valign="bottom"&gt; &lt;center&gt;2x1&lt;/center&gt; &lt;/td&gt; &lt;TD width="80" &gt; &lt;center&gt;2x2&lt;/center&gt; &lt;/td&gt; &lt;TD width="80" bgcolor="silver" valign="top"&gt; &lt;center&gt;2x3&lt;/center&gt; &lt;/td&gt; &lt;/tr&gt; &lt;/table&gt; &lt;/body&gt; &lt;/html&gt; </pre>	

### Параметры width и height

Параметры width и height отвечают соответственно за ширину и высоту таблицы. Значение этих параметров может быть указано в процентах или пикселях: width = Ширина в пикселях (процентах)

height =Высота в пикселях (процентах)

## Параметр nowrap

Обычно любой текст, не помещающийся в одну строку ячейки, переходит на следующую строку. Параметр **nowrap** запрещает принудительный перенос строки в ячейке или табличном ряде. При использовании атрибута **nowrap** (он не имеет параметров) с тегами <TH> и <TR> длина ячейки расширяется настолько, чтобы заключенный в ней текст поместился в одну строку. В этом случае в ячейке будет создана одна строка, а таблица может уйти за край окна. Например, если название фирмы ее главный дизайнер хочет записать в одну строку.

Не рекомендуется использовать данный параметр во всех ячейках, т. к. это может сильно понизить уровень масштабируемости таблицы

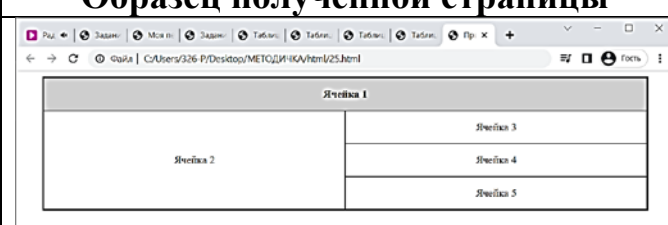
## Параметр rowspan и colspan

Иногда при построении таблиц возникает необходимость в объединении нескольких соседних ячеек в одну. Атрибуты **rowspan** и **colspan** позволяют объединять ячейки в таблице.

Параметр **rowspan**, используемый в тегах <TH> и <TR> задает число строк, на которые растягивается ячейка. Если указано `rowspan = n`, то  $n$  строк должно находиться ниже. НЕЛЬЗЯ поместить этот атрибут в последней строке таблицы.

При задании атрибута `rowspan = n` и условию, что  $n > 1$ , соответствующая ячейка займет не одну, а  $n$  строк и, соответственно, будет иметь размер в  $n$  раз больший, чем обычная ячейка данного столбца.

Параметр **colspan** (**Column Span**, соединение столбцов) используется для того, чтобы сделать какую-нибудь ячейку шире, чем верхняя или нижняя и растянуть ячейку над любым количеством обычных ячеек. При помощи атрибута **colspan** можно создавать ячейки, рас положенные сразу в нескольких столбцах. Например, атрибут `colspan = 2` растянет ячейку на ширину 2 столбцов.

Пример. 25	Образец полученной страницы
<pre>&lt;HTML&gt; &lt;HEAD&gt; &lt;TITLE&gt;Пример объединения ячеек с помощью COLSPAN и ROWSPAN&lt;/title&gt; &lt;/head&gt; &lt;BODY bgcolor="#FFFFFF" text="black" leftmargin="40" rightmargin="40" marginwidth="40"&gt; &lt;TABLE align="center" border="2" cellspacing="0" cellpadding="5" width="100%" height="200"&gt; &lt;TR align="center" bgcolor="#CECECE"&gt; &lt;TH colspan="2"&gt;Ячейка 1&lt;/th&gt; &lt;/tr&gt;</pre>	



<pre> &lt;TR&gt; &lt;TD align="center" rowspan="3"&gt;Ячейка   2&lt;/td&gt; &lt;TD align="center"&gt;Ячейка 3&lt;/td&gt; &lt;/tr&gt; &lt;TR&gt; &lt;TD align="center"&gt;Ячейка 4&lt;/td&gt; &lt;/tr&gt; &lt;TR&gt; &lt;TD align="center"&gt;Ячейка 5&lt;/td&gt; &lt;/tr&gt; &lt;/table&gt; &lt;/body&gt; &lt;/html&gt; </pre>	
--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--

Пустые ячейки используются для того, чтобы у ячейки были границы, но не было содержимого. Если нужна рамка, можно воспользоваться пустой строкой **<BR>** или `&nbsp; non-breaking space` - неразрывающий пробел. Можно задать ширину ячеек в пикселях или относительных единицах, но не вводить значений.

Перечень параметров тегов **<TR>**, **<TD>** и **<TH>** приведен в таблице 4  
Таблица 4- Перечень параметров тегов

Параметр	Функция	Применение
align	Выравнивание содержимого ячейки или ряда по горизонтали	<b>&lt;TR&gt;</b> , <b>&lt;TD&gt;</b> , <b>&lt;TH&gt;</b>
valign	Выравнивание содержимого ячейки или ряда по вертикали	<b>&lt;TR&gt;</b> , <b>&lt;TD&gt;</b> , <b>&lt;TH&gt;</b>
width	Определение ширины ячейки или ряда	<b>&lt;TR&gt;</b> , <b>&lt;TD&gt;</b> , <b>&lt;TH&gt;</b>
height	Определение высоты ячейки или ряда	<b>&lt;TR&gt;</b> , <b>&lt;TD&gt;</b> , <b>&lt;TH&gt;</b>
b bgcolor	Указание цвета для фона ячейки или ряда	<b>&lt;TR&gt;</b> , <b>&lt;TD&gt;</b> , <b>&lt;TH&gt;</b>
background	Указание рисунка для фона ячейки	<b>&lt;TD&gt;</b> , <b>&lt;TH&gt;</b>
nowrap	Запрет принудительного переноса строки в ячейке или ряду	<b>&lt;TR&gt;</b> , <b>&lt;TD&gt;</b> , <b>&lt;TH&gt;</b>
colspan	Объединение соседних ячеек по горизонтали	<b>&lt;TD&gt;</b> , <b>&lt;TH&gt;</b>
rowspan	Объединение соседних ячеек по вертикали	<b>&lt;TD&gt;</b> , <b>&lt;TH&gt;</b>

## Группировка данных

При построении таблиц мы можем легко задать единый тип выравнивания для отдельной ячейки и даже целого табличного ряда. Однако гораздо, чаще может возникнуть необходимость в едином выравнивании информационных данных конкретного столбца. Столбец в таблице – это последовательность ячеек, располагающихся в разных рядах.

Стандартными средствами HTML нам пришлось бы задавать один и тот же тип выравнивания для отдельно взятой ячейки, формирующей нужный столбец:

```
<TABLE>
<TR>
<TD align="right">Ячейка 1 с выравниванием вправо</td> <TD
align="center">Ячейка 1 с выравниванием по центру</td> </tr>
<TR>
<TD align="right">Ячейка 2 с выравниванием вправо</td> <TD
align="center">Ячейка 2 с выравниванием по центру</td> </tr>
</table>
```

При работе с браузером Internet Explorer, труд разработчика электронного документа может быть значительно облегчен за счет таких тегов, как `<col>` и `<colgroup>`. Теги `<col>` и `<colgroup>` предназначены для определения свойств отображения табличных данных, сгруппированных по конкретному признаку. Они оба могут иметь параметр `span`, задающий количество соседних столбцов, и параметр `align`, устанавливающий единый тип выравнивания для выбранного столбца (возможные значения: по левому краю, по правому краю, по центру; формат записи аналогичен тегам `<TD>` и `<TH>`). Помимо этого тег `<colgroup>` может содержать дополнительный параметр вертикального выравнивания данных – `valign` (возможные значения: по верхнему краю, по нижнему краю, по середине; формат записи аналогичен тегам `<TD>` и `<TH>`). Разница между тегами `<col>` и `<colgroup>` заключается в условном объединении (группировке) взаимосвязанных данных отдельно взятого столбца таблицы.

Пример. 26	Образец полученной страницы								
<pre>&lt;HTML&gt; &lt;HEAD&gt; &lt;TITLE&gt;Группировка данных таблицы&lt;/title&gt; &lt;/head&gt; &lt;BODY BGCOLOR="#FFFFFF " TEXT="black" LEFTMARGIN="40" RIGHTMARGIN="40" MARGINWIDTH="40 "&gt; &lt;TABLE align="center" border="2" cellspacing="0"</pre>	<table><tr><td>Ячейка 1</td><td>Ячейка 2</td><td>Ячейка 3</td><td>Ячейка 4</td></tr><tr><td>Ячейка 5</td><td>Ячейка 6</td><td>Ячейка 7</td><td>Ячейка 8</td></tr></table>	Ячейка 1	Ячейка 2	Ячейка 3	Ячейка 4	Ячейка 5	Ячейка 6	Ячейка 7	Ячейка 8
Ячейка 1	Ячейка 2	Ячейка 3	Ячейка 4						
Ячейка 5	Ячейка 6	Ячейка 7	Ячейка 8						

<pre> cellpadding="5" width="100%" height="200"&gt; &lt;COLGROUP align="center" span="2"&gt; &lt;COLGROUP align="right" span="2"&gt; &lt;TR&gt; &lt;TD&gt;Ячейка 1&lt;/td&gt; &lt;TD&gt;Ячейка 2&lt;/td&gt; &lt;TD&gt;Ячейка 3&lt;/td&gt; &lt;TD&gt;Ячейка 4&lt;/td&gt; &lt;/tr&gt; &lt;TR&gt; &lt;TD&gt;Ячейка 5&lt;/td&gt; &lt;TD&gt;Ячейка 6&lt;/td&gt; &lt;TD&gt;Ячейка 7&lt;/td&gt; &lt;TD&gt;Ячейка 8&lt;/td&gt; &lt;/tr&gt; &lt;/table&gt; &lt;/body&gt; &lt;/html&gt; </pre>	
--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--

### **Колонтитулы таблицы**

Браузер Internet Explorer позволяет использовать дополнительные теги структурирования табличных данных, а именно – <thead>, <tbody> и <tfoot>. Эти теги предназначены для создания колонтитулов таблицы различных уровней (соответственно для верхнего, основного (содержательного) и нижнего уровней таблицы).

Теги верхнего и нижнего колонтитулов <thead> и <tfoot> могут быть использованы в структуре таблицы лишь единожды, причем для них необязательно наличие закрывающих тегов.

Тег основного колонтитула <tbody> может встречаться неоднократно в пределах одной таблицы, однако требует своего закрывающего тега. Верхний и нижний колонтитулы функционально очерчивают логические заголовки соответствующего уровня и применимы, в основном, в больших таблицах, не помещающихся в пределах одной страницы электронного документа.

Основные колонтитулы выполняют функцию, аналогичную тегам группировки <soy> и <colgroup>, расставляя логические метки по ходу изложения основной содержательной части таблицы.

### **Прорисовка структуры таблицы**

Еще одной замечательной возможностью нестандартного представления таблиц, работающей только в браузере Internet Explorer, является прорисовка внутренней структуры таблицы, а именно – рамок и линеек. Другими словами, возможно оформить таблицу таким образом, что от рамки останутся только верхняя и нижняя границы, а между ячейками останется только вертикальная линейка.

За изменение свойств рамки отвечает параметр **frame**, а с помощью параметра **rules** варьируется внешний вид линеек таблицы (оба параметра применимы внутри тега <TABLE>).

Возможные значения параметра frame приведены в таблице 5

Таблица 5 - Значения параметра frame

<b>значение</b>	<b>Функция</b>
border	Рамка с четырех сторон
above	Рамка только сверху
below	Рамка только снизу
hsides	Верхняя и нижняя части рамки
vsides	Левая и правая части рамки
lhs	Только левая часть рамки
rhs	Только правая часть рамки
void	<b>Нет рамок</b>

Возможные значения параметра rules приведены в таблице 6.

Таблица 6 - Значения параметра rules

<b>Значение</b>	<b>Функция</b>
all	Отображение линейки целиком
groups	Часть линейки, разделяющая сгруппированные данные
cols	Часть линейки, разделяющая столбцы
rows	Часть линейки, разделяющая строки
none	Отсутствие линейки

### Пример. 27

```
<HTML> <HEAD>
<TITLE>Нестандартные возможности прорисовки
структуры таблицы</title>
</head>
<BODY BGCOLOR="#FFFFFF" TEXT="black">
<TABLE ALIGN="center" BORDER="1" CELSPACING="0"
CELLPADDING="4" WIDTH="100%" FRAME="hsides"
RULES="rows"> <CAPTION><B>Таблица 1</B></caption>
<TR>
<TD>Ячейка 1</td>
<TD>Ячейка 2</td>
<TD>Ячейка 3</td>
<TD>Ячейка 4</td>
</tr>
<TR>
<TD>Ячейка 5</td>
<TD>Ячейка 6</td>
<TD>Ячейка 7</td>
<TD>Ячейка 8</td>
</tr>
</table>
<TABLE ALIGN="center" BORDER="1" CELSPACING="0"
CELLPADDING="4" WIDTH="100%" FRAME="box" RULES="groups">
<CAPTION><B>Таблица 2</B></caption>
<TR>
<TD>Ячейка 1</td>
<TD>Ячейка 2</td>
<TD>Ячейка 3</td>
```

### Образец полученной страницы

Таблица 1			
Ячейка 1	Ячейка 2	Ячейка 3	Ячейка 4
Ячейка 5	Ячейка 6	Ячейка 7	Ячейка 8
Таблица 2			
Ячейка 1	Ячейка 2	Ячейка 3	Ячейка 4
Ячейка 5	Ячейка 6	Ячейка 7	Ячейка 8
Таблица 3			
Ячейка 1	Ячейка 2	Ячейка 3	Ячейка 4
Ячейка 5	Ячейка 6	Ячейка 7	Ячейка 8

```

        <TD>Ячейка 4</td>
    </tr>
    <TR>
        <TD>Ячейка 5</td>
        <TD>Ячейка 6</td>
        <TD>Ячейка 7</td>
        <TD>Ячейка 8</td>
    </tr>
</table>
<TABLE    ALIGN="center"    BORDER="1"    CELLSPACING="0"
CELLPADDING="4" WIDTH="100%" FRAME="above" RULES="all">
<CAPTION><B>Таблица 3</B></caption>
<TR>
    <TD>Ячейка 1</td>
    <TD>Ячейка 2</td>
    <TD>Ячейка 3</td>
    <TD>Ячейка 4</td>
</tr>
<TR>
    <TD>Ячейка 5</td>
    <TD>Ячейка 6</td>
    <TD>Ячейка 7</td>
    <TD>Ячейка 8</td>
</tr>
</table>
</body>
</html>

```

## **Вложенные таблицы**

Одной из замечательных особенностей HTML-таблиц по праву считается поддержка многоуровневой вложенности. Другими словами, одна таблица может включать другую, та, в свою очередь, еще одну и т. д.

### **Преимущества вложенных таблиц**

Особенность вложенных таблиц, в отличие от других способов представления данных в электронном документе, позволяет более точно размещать отдельные элементы страницы относительно друг друга и относительно границ самого документа, отображаемого браузером.

Например, два разнородных блока текста и нумерованный список, размещенные внутри тега <BODY>, невозможно разместить на одном уровне, а тем более на одном уровне со смещением в какую-либо сторону. Использование таблиц с легкостью решает эту проблему, позволяя располагать различные элементы и их комбинации в разных местах документа посредством видимых и невидимых ячеек рядов таблицы.

Вот почему в последнее время преобладающее большинство HTML-документов создается на основе таблиц, где в качестве несущей основы берется таблица с невидимыми краями, содержащая вложенные таблицы с разным оформлением, отличающимися значениями параметров.

Подводя итог сказанному, можно выделить следующие преимущества вложенных таблиц:

- гибкая масштабируемость структуры электронного документа в целом;
- широкие возможности позиционирования отдельных элементов страницы;
- многоуровневое представление разнородных информационных данных;
- расширенные оформительские возможности;
- поддержка популярными браузерами.

Правила построения вложенных таблиц ничем не отличаются от создания таблиц одного уровня – используются те же теги и параметры, задаются те же свойства и значения. Единственное, о чем нельзя забывать в ходе создания сложных вложенных таблиц, это:

- каждая таблица последующего уровня размещается внутри тега-контейнера <TD> или <TH> таблицы предыдущего уровня;
- вложенная таблица не может быть создана за пределами вышеназванных тегов ячейки таблицы;
- таблица одного уровня может содержать любое количество вложенных таблиц другого уровня, идущих друг за другом в пределах тега ячейки таблицы верхнего уровня;
- количество тегов таблиц всех уровней должно соответствовать количеству закрывающих тегов этих же таблиц.

## Задания к практической работе № 7. Таблицы HTML-документов

### Задание 1. Создание таблицы по образцу.

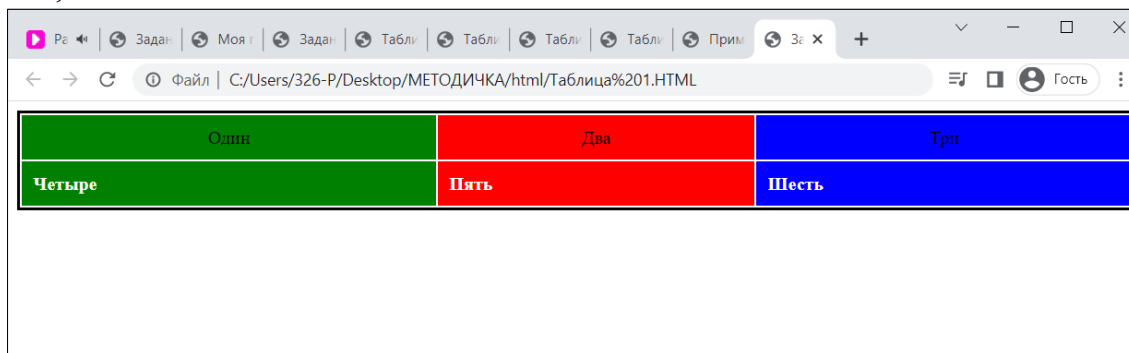
#### •Таблицы HTML-документов

Создайте файл под названием «Таблица 1.HTML» В элементе <TITLE> укажите название странички "Задание 1. Создание таблицы по образцу"

Сделайте фон первого столбца зеленого цвета, второго – красного, третьего – синего.

- Задайте ширину бордюра таблицы, равным 3 пикселям.
- Ширину таблицы сделайте равной ширине экрана.
- Шрифт, которым написан текст внутри таблицы, сделайте белым
- Создайте таблицу, как показано на рисунке
- Текст в первой строке выровняйте по центру по горизонтали и по вертикали.
- Текст во второй строке выделите жирным шрифтом.

Для создания таблицы необходимо использовать теги <table>, <tr>, <td> и <th>.



Один	Два	Три
Четыре	Пять	Шесть

Рисунок 18

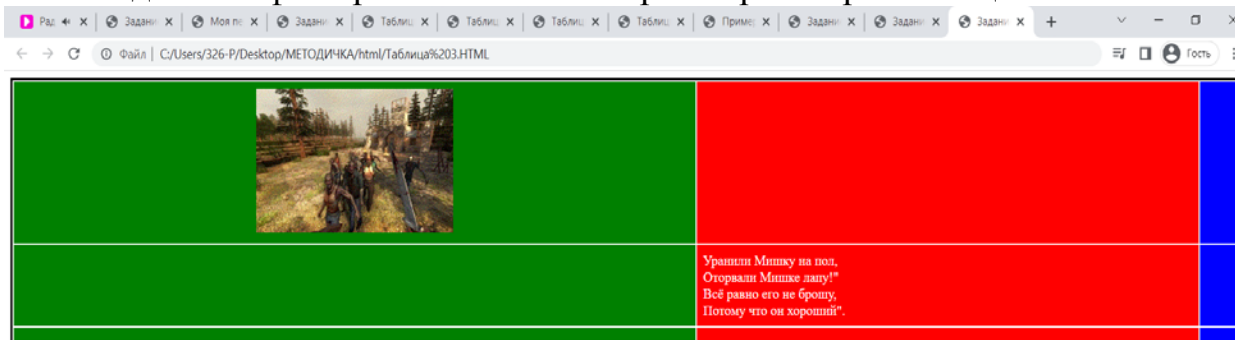
### Задание 2. Создание таблицы по образцу.

Создайте файл под названием «Таблица 2.HTML»

• Для выполнения этого задания воспользуйтесь результатами выполнения Задания 1.

• **Таблицы HTML-документов** В элементе <TITLE> укажите название странички "Задание 4. Заполнение таблицы".

- В первую ячейку первой строки вставьте рисунок **image2.gif**.
- Во второй ячейке второй строки напишите четверостишие (Уронили Мишке на пол...).
- Сделайте фон третьей ячейки второй строки красного цвета.




		
	Уронили Мишке на пол, Огорвали Мишке лапу! Всё равно его не брошу, Потому что он хороший!	

Рисунок 19



### Задание 3. Самостоятельное проектирование и создание таблицы

#### • Таблицы HTML-документов

Создайте файл под названием «Таблица 3.HTML»

Составьте таблицу для расписания ваших занятий. Пусть эта таблица состоит из восьми столбцов и девяти строк. Ширина таблицы 100%.

• В первом столбце укажите время начала и конца ваших занятий. Выравнивание данных в ячейках должно быть по левому краю.

• В остальных столбцах укажите дисциплины по дням недели.

• Выравнивание дней недели по центру ячейки и жирным шрифтом.

• Выравнивание названий дисциплин в ячейках должно быть по левому краю. • У всех столбцов фон сделайте разным цветом.

• Перед таблицей поместите заголовок "РАСПИСАНИЕ ЗАНЯТИЙ", выделив его тэгами <H1> и </h1>. Цвет заголовка красный.

• Между заголовком и таблицей поместите рисунок

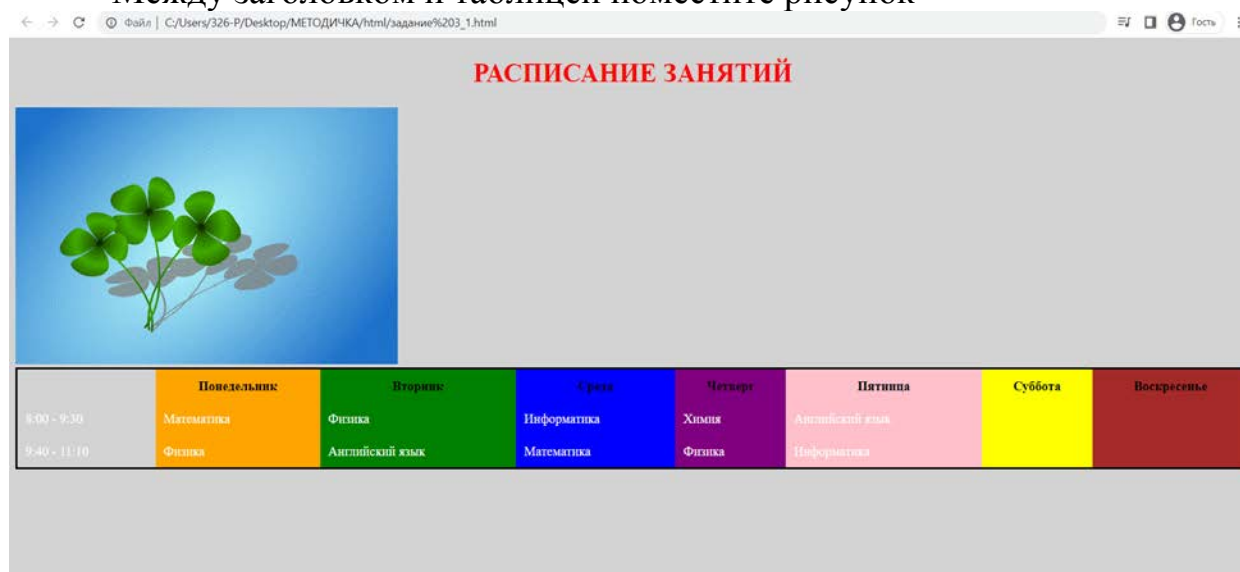


Рисунок 20

### Задание 4.

#### Таблицы HTML-документов

Создайте файл под названием «Таблица 4.HTML»

Составьте справочную таблицу “Цветовая гамма HTML-документа” размером 5\*3. Заполните ячейки таблицы записями кодов различных оттенков красного (строка 1), зеленого (строка 2) и синего (строка 3). Фон ячеек закрасьте теми же цветами, коды которых записаны в них.

#FF0000	#FF3333	#FF6666	#FF9999	#FFCCCC
#00FF00	#33FF33	#66FF66	#99FF99	#CCFFCC
#0000FF	#3333FF	#6666FF	#9999FF	#CCCCFF

Рисунок 21

## Задание 5.

### Таблицы HTML-документов

Создайте файл под названием «Таблица 5.HTML»

Составьте радугу, закрасив ячейки таблицы нужным цветом.



Рисунок 22

**Задание 6.** Создайте файл под названием «**Таблица 6.HTML**»

**Таблицы HTML-документов** Создайте таблицу с колонками для следующих данных: предмет, оценка, дата экзамена/зачета, преподаватель

В первой строке таблицы разместите заголовки колонок

В следующих строках разместите данные, соответствующие каждому предмету и его оценке, а также дату и ФИО преподавателя, который проводил экзамен или зачет.

← → ↻ ⓘ Файл | C:/Users/326-P/Desktop/МЕТОДИЧКА/html/задание%206\_1.html

Предмет	Оценка	Дата экзамена/зачета	Преподаватель
Математика	4	15.05.2023	Башкирцева Галина Алексеевна
Физика	5	20.06.2023	Петров Петр Петрович
Иностранный язык	5	30.06.2023	Дудель Елена Викторовна
История	4	10.05.2023	Иванов Виталий Александрович
Информатика	3	05.06.2023	Александрова Елена Михайловна

Рисунок 23

### Результат выполнения практической № 6

папка	<b>Первые шаги</b> <b>ФИО</b>	
файл	1. <b>Таблица 2.HTML</b> 2. <b>Таблица 6.HTML</b>	Выполняют все
	<b>Таблица 1.HTML</b> <b>Таблица 4HTML</b>	Четные            номера компьютеров
	<b>Таблица 3.HTML</b> <b>Таблица 5.HTML</b>	Нечетные        номера компьютеров

### Контрольные вопросы

1. Основные методы создания таблиц.
2. Параметры, которые используются при создании таблиц в HTML-документах.
3. Группировка данных
4. Колонтитулы таблицы
5. Прорисовка структуры таблицы
6. Вложенные таблицы

### **Использованные источники**

1. <https://code.mu/ru>
2. <https://gospodaretsva.com/>
3. Серветник О.Л Методические рекомендации по выполнению практических работ.