**Лабораторная работа №1.**

**Тема: Разработка макета Web-сайта средстами HTML**

**Основные понятия**

**HTML** *(HyperText Markup Language)* - язык разметки гипертекста - предназначен для создания Web-страниц. Под ***гипертекстом*** в этом случае понимается текст, связанный с другими текстами указателями-ссылками.

HTML представляет собой достаточно простой набор кодов, которые описывают структуру документа. HTML позволяет выделить в тексте отдельные логические части (заголовки, абзацы, списки и т.д.), поместить на Web-страницу подготовленную фотографию или картинку, организовать на странице ссылки для связи с другими документами.

HTML не задает конкретные и точные атрибуты форматирования документа. Конкретный вид документа окончательно определяет только ***программа-броузер*** на компьютере пользователя Интернета.  
HTML также не является языком программирования, но web-страницы могут включать в себя встроенные программы-скрипты на языках ***Javascript*** и ***Visual Basic Script*** и программы-апплеты на языке ***Java***.

Даже, если вы не собираетесь в дальнейшем редактировать "вручную" текст HTML (предполагая использовать графические редакторы), знание языка HTML даст вам возможность как лучше использовать эти средства, так и увеличит ваши шансы сделать HTML-документ более доступным и "читаемым" при просмотре броузерами разных фирм.

Основными компонентами HTML являются:

* **Тег (tag).** Тег HTML это компонент, который командует Web- броузеру выполнить определенную задачу типа создания абзаца или вставки изображения.
* **Атрибут** (или **аргумент**). Атрибут HTML изменяет тег. Например, можно выровнять абзац или изображение внутри тега.
* **Значение**. Значения присваиваются атрибутам и определяют вносимые изменения. Например, если для тега используется атрибут выравнивания, то можно указать значение этого атрибута. Значения могут быть текстовыми, типа *left* или *right*, а также числовыми, как например ширина и высота изображения, где значения определяют размер изображения в пикселях.

Теги представляют собой зарезервированные последовательности символов, начинающиеся с **<** (знака меньше) и заканчивающиеся **>** (знаком больше). Закрытие тега отличается от открытия только наличием символа **'/'**.

Предположим, у нас есть гипотетический атрибут форматирования текста, управляемый кодом **<X>**, и мы хотим применить его к словам "Это мой текст". HTML-последовательность кодов и собственно текста будет выглядеть так:

**<X>Это мой текст</X>**

Теги могут вкладываться друг в друга иерархически, но без пересечений, то есть допустимо вложение вида ***<teg1>*<teg2></teg2>** ***</teg1>***, но не ***<teg1>*<teg2>** ***</teg1>*</teg2>**.

Действие вложенных тегов объединяется. Например, если внутрь тега, создающего жирное начертание шрифта, вложен тег курсива, то в результате получится жирный курсив.

**Первое правило**

Первое правило HTML: закрывайте все, что вы открыли!  
НО! Из этого правила, как и из всех остальных, существуют исключения.

HTML- программа должна начинаться тегом **<HTML>**и заканчиваться тегом **</HTML>**

**<HTML>**

**...... (здесь будут другие теги программы)**

**</HTML>**

HTML- программы состоят из двух основных частей: заголовка и тела. Заголовок ограничивается парой тегов**<HEAD>** и **</HEAD>**, а тело - парой тегов**<BODY>** и **</BODY>**.

В результате HTML- программа выглядит следующим образом:

**<HTML>**

**<HEAD>**

**... (здесь будет заголовок)**

**</HEAD>**

**<BODY>**

**.... (здесь будут другие теги тела программы)**

**</BODY>**

**</HTML>**

Кроме того, каждая HTML- программа имеет заголовок, который помещается в заголовок окна броузера. Заголовок окна броузера создается при помощи двух тегов **<TITLE>** и **</TITLE>** и содержится между тегами **<HEAD>** и**</HEAD>**.

Тогда программа принимает следующий вид:

**<HTML>**

**<HEAD>**

**<TITLE> Основы HTML </TITLE> ;**

**</HEAD>**

**<BODY>**

**.... (здесь будут другие теги тела программы)**

**</BODY>**

**</HTML>**

***Некоторые авторы, пишущие об языке HTML, советуют записывать теги прописными буквами, другие - используют строчные. Редактор HTML - Allaire HomeSite 4.5.1, например, использует по умолчанию нижний регистр для записи тегов. При создании моих страниц использовались оба варианта написания тегов. Как видите, допустимо и то и другое. Современные браузеры допускают запись тегов в любом регистре.***

***Существуют теги и атрибуты "чувствительные" к написанию прописными или строчными буквами. Это регламентируется стандартами языка HTML, определенными***[***Консорциумом W3C***](http://html-exp.narod.ru/help.htm#%D0%BC%D0%B5%D1%82%D0%BA%D0%B0_R2)***.***

***Обращайтесь к***[***первоисточнику***](http://www.w3.org./)***! Хорошее знание технического английского обязательно!***

 При написании HTML-программ возникает необходимость вставки ***комментариев*** - поясняющих текстов, которые невидны при загрузке документа в браузер. Для этой цели служит тег **<!>**. Все, что заключено между символами **<!** и **>** считается комментарием и не отображается в браузере.

Еще один тег, который очень важен в HTML-программе, но так же не предназначается для отображения какого-либо объекта в броузере - тег ***<META>***. Этот тег служит специальным целям, а именно - указания языка, на котором написан документ, его кодовой страницы, ключевых слов, используемых поисковыми системами для классификации этого документа и т.п. Теги ***<META>*** обычно вставляются в HTML-программу на заключительном этапе создания Web-страницы -*публикации*.

Для вставки в HTML-программу фрагмента программ, написанных на языке JavaScript или Viual Basic Script *сценариев* используют теги ***<SCRIPT>*** и ***</SCRIPT>***.

**Общая структура HTML-файла**

Суммируя вышесказанное приведем общую структуру HTML-файл :

**<HTML>**

**<HEAD>**

**<Мета-теги>**

**<Функции скриптов>**

**<TITLE>Заголовок документа</TITLE>**

**</HEAD>**

**<BODY>**

**Основная часть документа**

**</BODY>**

**</HTML>**

**Заголовки, абзацы, разрывы строк**

**Заголовки**

Каждый пользователь компьютера, работающий в текстовом редактором Microsoft Word знаком с понятием ***стиля заголовка***. В HTML тоже применяется это понятие для структурирования документа и выделения важности заголовка. Всего существуют 6 стилей заголовка. Каждый из них обозначается в HTML-документе парными тегами ***<Hi>*** и ***</Hi>.*** Здесь ***i*** обозначает важность стиля. ***H1*** обозначает самый важный стиль заголовка, ***H2*** - стиль заголовка второго уровня, а ***H6*** - стиль заголовка самого нижнего уровня.

В подавляющем большинстве случаев для заголовков Web-страниц используют три первых уровня заголовков ***<H1>***, ***<H2>*** и ***<H3>***. Объясняется это тем, что размеры шрифтов оставшихся заголовуов (теги ***<H4>*** - ***<H5>***) меньше размера обычного шрифта Web-страницы.

Вот как в документ можно добавить очень важный заголовок.

**<H1>An important heading</H1>**

А вот результат.

**An important heading**

Посмотрим другие примеры:

**Это заголовок второго уровня**

**Это заголовок третьего уровня**

Это обычный текст

**Это заголовок четвертого уровня**

**Это заголовок пятого уровня**

**Это заголовок шестого уровня**

**Абзацы**

Понятие абзаца в HTML-документе также аналогично понятию абзаца в Microsoft Word. Абзац обозначается в документе парными тегами ***<P>*** и ***</P>*** . Впрочем, применение закрывающего тега не является строго обязательным.  
НО! Специфика тега ***<P>*** заключается в том, что после текста, который находится в его пределах, пустая строка добавляется ***автоматически***.  
Следует помнить и о другой особенности текстовых абзацев: когда текст достигает правой границы окна Web-броузера. переход на новую строку осуществляется автоматически, независимо от расположения тега ***<P>***.  
Для отдельного абзаца можно указать тип, размер и цвет шрифта отличным от стиля остального документа.

Например:

**<P ALIGN="CENTER"><B><I><FONT FACE="Arial" SIZE=6 COLOR="#С07080">My greetings to you!</P>**

А вот и результат -

***My greetings to you!***

**Разрывы строк**

Если в середине строки появилась необходимость ее разорвать - используйте **одиночный**тег переноса строки ***<BR>***.(Это соответствует нажатию клавишной комбинации [Shift]-[Enter] в текстовых процессорах Word). Код <BR> не означает конца логического абзаца, и за строкой с этим кодом дополнительная пустая строка не появится.

Вот и нарушено первое правило!

Кроме тега ***<BR>*** не требует закрытия тег добавления изображения ***<IMG>***.

Использование закрывающего тега абзаца ***</p>*** также не является строго обязательным.

***HTML довольно "демократичен": неправильный тег или неправильное вложение тегов обычно не приводят к "зависаниям" браузера, а только вызывает сообщение об ошибке в строке состояния окна браузера вашего Интернет-читателя. Разумеется ошибки могут привести к неправильному форматированию HTML-документа.***

Примером использования тега ***<BR>*** может служить написание почтового адреса или стихотворения.

Браузеры показывают текст в своем окне, автоматически осуществляя перенос слов. Поэтому, если вы считаете необходимым запретить разрыв блока текста с пробелами между словами - воспользуйтесь специализированным символом ***&nbsp;*** - символом неразрывного пространства (non-breaking space). Например,

**Освежающий и бодрящий напиток Coca&nbsp;Cola,**

**приобрел широкую популярность в нашей стране.**

А вот и результат:

Освежающий и бодрящий напиток Coca Cola, приобрел широкую популярность в нашей стране.

*Cовет*

Не следует использовать цепочку символов ***&nbsp;*** для выравнивания текста в окне браузера. Для этих целей рекомендуется использовать *таблицы стилей*.

**Предварительно отформатированный текст**

Достоинством браузеров является их способность самостоятельно распределять текст в окне браузера. Но иногда вы не нуждаетесь в этой услуге, хотите самостоятельно определить представление вашего текста в окне браузера. Например, вы хотите представить код программы в наиболее удобочитаемом виде. Такую возможность вам предоставляет тег ***<pre>***. Посмотрим пример:

**<pre>**

**void Node::Remove()**

**{**

**if (prev)**

**prev->next = next;**

**else if (parent)**

**parent->SetContent(null);**

**if (next)**

**next->prev = prev;**

**parent = null;**

**}**

**</pre>**

Это выполнится так

**void Node::Remove()**

**{**

**if (prev)**

**prev->next = next;**

**else if (parent)**

**parent->SetContent(null);**

**if (next)**

**next->prev = prev;**

**parent = null;**

**}**

Это сделано при помощи [*таблицы стилей*](http://html-exp.narod.ru/css.htm).

Обратите внимание на то, что в случае использования тега **<pre>**, текст отображается браузером точно в таком виде, как он был создан в HTML-документе. Сохраняются все пробелы, табуляции и переводы строк. Исключением является только новая строка, следующая сразу же за тегом **<pre>**. Таким образом, эти два примера кода HTML на экране дисплея будут показаны одинаково:

**<pre>предварительно отформатированный текст</pre>**

**<pre>**

**предварительно отформатированный текст**

**</pre>**

А именно:

**предварительно отформатированный текст**

**предварительно отформатированный текст**

У этого тега существует необязательный атрибут, указывающий желаемый размер строки в символах, а именно:

**<PRE width="N">**

**Предмет истории есть жизнь народов и человечества.**

**Непосредственно уловить и обнять словом - описать жизнь не только**

**человечества, но одного народа, представляется невозможным.**

Обычно, для отображения предварительно отформатированного текса используются моноширинные шрифты, все символы которого имеют одинаковую ширину. Для того, чтобы браузер "не забыл" об этом следует применить *таблицы стилей*. Например, это можно сделать так:

Результат такого определения стиля для тега <pre> вы можете видеть на этой странице.

*Совет*

Если вы устанавливаете цвет текста, желательно также установить цвет фона. Это поможет вам избежать ситуации, когда буквы будут практически неразличимы на близком к ним по цвету фоне.

Совет № 2. Вместо того, чтобы устанавливать цвет фона для элемента <pre>, установите цвет фона для элемента <body>. Например, это можно сделать так:

**<style type="text/css">**

**body { color: black; background: white; }**

**pre { color: green; font-family: monospace; }**

**</style>**

**Выделенный текст**

В тех случаях, когда необходимо обратить особое внимание пользователя на тот или иной фрагмент текста, следует использовать стандартные средства форматирования.

Стандартные средства форматирования представлены специальными тегами, которые ***обязательно*** являются ***парными*** (т.е. имеются открывавающий и закрывающий теги).

Чтобы увеличить размер шрифта на один пункт, используйте тег <big>.

Чтобы выделить текст **полужирным шрифтом**, воспользуйтесь **тегом <b>** или **тегом <strong>**.

Чтобы выделить текст *курсивом*, воспользуйтесь *тегом <i>* или ***тегом <em>***.

***Вы заметили, что в данном случае выделение текста тегами <i> и <em> различаются?***

***Объяснение простое - в HTML-коде этой страницы используется внедренная таблица стилей, в которой для тега <em> указан шрифт: полужирный курсив.***

Шрифт пишущей машинки можно имитировать с помощью тега <tt>.

С помощью тега <small> размер шрифта можно уменьшить на один пункт.

Существуют и другие теги, которые также предназначены для выделения текста.

Кроме того, в HTML включена поддержка математических символов и научных обозначений. Для построения простейших равенств и выражений вам могут пригодиться два тега ***<sub>*** (нижний индекс) и ***<sup>*** (верхний индекс). Например:

A2+B2=C2

CO2=углекислый газ

В HTML-коде это записано так:

**<p align="center">A<sup>2</sup>+B<sup>2</sup>**

**=C<sup>2</sup></p>**

**<p align="center">CO<sub>2</sub>=углекислый газ</p>**

# Добавление изображений

Изображения делают страницу привлекательной. Теги изображений представляет собой исключение из правил - для них них не требуется закрывающего тега, зато у них есть ряд важных атрибутов. Теги изображения состоят из тега **<img>**и источника изображения (адрес графического файла), указанного в атрибуте **src**. В качестве адреса графического файла указывается либо URL-адрес, либо имя файла (возможно с указанием пути). Например,

**<IMG SRC="images/"Tourgojk\_lake.jpg>**

Рекомендуется включать в тег следующие атрибуты для каждого изображения:

* **width="x".** Определяет ширину изображения
* **heigth="x".** Определяет высоту изображения
* **border="x".** Значение *x* в данном случае определяет толщину рамки, которую браузер будет рисовать вокруг изображения.

*Совет*

Для этого атрибута нужно использовать значение,не равное нулю.

* **alt="x".**Атрибут *alt* позволяет вводить описание изображения. Это очень полезно для людей, отключающих изображение.

## Увеличение скорости загрузки изображения

Для увеличения скорости передачи графического изображения в теге <IMG> можно использовать атрибут *lowsrc*, который принимает в качестве аргумента адрес графического файла. Для использования этого атрибута необходимо создать два файла: один с высоким разрешением (формат JPG), другой - с низким (формат GIF). Тогда тег

**<img src = "images/big\_flower.jpg" align="middle"**

**lowsrc="images/small\_flower.gif">**

предпишет браузеру сначала загрузить файл small\_flower.gif, а затем по мере приема заменить его файлом big\_flower.jpg.

Другой способ ускорения загрузки заключается в задании размеров изображения с помощью атрибутов *width* (ширина) и *heigth* (высота), измеряемые в пикселах. Если указать эти атрибуты, то броузер сначала выделит место под графику, подготовит макет документа, отобразит текст, и только потом загрузит графику. При этом браузер сжимает или растягивает изображение, встраивая его в рамки указанного размера.

## Обтекание текстом изображения

При создании HTML-документа, вы можете выбрать как используемое вами изображение будет расположено относительно текста, а именно:

1. в одну линию с текстом;
2. слева от текста;
3. справа от текста.

Для управления расположением изображения относительно текста используется аттрибут *align*. Этот атрибут может принимать значения: *left* - изображение выравнивается по левому полю, *right* - изображение выравнивается по правому полю, *middle* - посередине.

**Задание №1.** Создайте web-сайт (3-4 связанные между собой html-страницы), согласно требованиям представленным в таблице 1. Наполните страницу небольшим контентом, согласно тематике.

Таблица 1 – Варианты для выполнения задания №1.

|  |  |
| --- | --- |
| **Вариант** | **Тематика страницы** |
| 1 | Спорт |
| 2 | Автомобили |
| 3 | Электроника |
| 4 | Программирование |
| 5 | Развлечения |
| 6 | Наука |
| 7 | Недвижимость |
| 8 | Книги |
| 9 | Животные |
| 10 | Игры |

**Задание №2.**

1) Создайте документ содержащий HTML-словарь тегов форматирования приведенных ниже. При создании словаря используйте список определений.

<abbr><acronym><address>**<article><aside>**

<b><bdi><bdo><big><blockquote><br>

<cite><code>**<command>**

**<datalist>**<del>**<details>**<dfn><div>

<em>

**<figcaption><figure><footer>**

<h1><h2><h3><h4><h5><h6>**<header>**

<i><ins>

<kbd>**<keygen>**

**<main><mark>**<menu>**<meter>**

**<nav>**<nobr>

**<output>**

<pre>**<progress>**<q>

<s><samp><small>**<source>**<span><strike><strong><sub>**<summary>**<sup>

**<time><track>**<tt>

<u>

<var><video>

**<wbr>**

Жирным выделены теги поддержка которых есть только в HTML5. Создайте 5 страниц с подробным описанием тегов (синткаксис, поддержка браузеров, их возможные атрибуты, пример использования) согласно вашему варианту.

|  |  |
| --- | --- |
| **Вариант** | **Теги** |
| 1 | abbr code **mark time** video |
| 2 | acronym dfn **keygen progress** small |
| 3 | address blockquote **main summary** var |
| 4 | big **command header** ins samp |
| 5 | **datalist footer** nobr strike var |
| 6 | code **figcaption meter** pre sup |
| 7 | **figure** kbd nobr small **wbr** |
| 8 | cite **header** menustrong var |
| 9 | big del **time track** var |
| 10 | div i **output progress** q |

2) В верхней части документа создайте внутренние ссылки (в качестве ссылки выступает заглавная буква английского алфавита), которые ведут в соответствующие букве части словаря. Добавьте возможность вернуться в начало документа.