

1. ЛАБОРАТОРНАЯ РАБОТА №1. ИСПОЛЬЗОВАНИЕ *HTML* ДЛЯ РАЗРАБОТКИ СТАТИЧЕСКОГО *WEB* – САЙТА.

1.1.Цель работы

Освоение основ языка разметки гипертекста *HTML*, дескрипторов форматирования текста и технологии создания гиперссылок.

1.2.Теоретические сведения

1.2.1.Основы WWW

World Wide Web – это средство представления информации в Интернет. Более точно WWW можно определить как интерактивную мультимедийную гипертекстовую среду, использующую язык разметки и поддерживающую множество протоколов Интернет[1].

Мультимедийная среда – возможность размещать в Сети не только и не столько текстовую информацию, но и графику, звук, видеоклипы.

Гипертекстовая среда – возможность использования в тексте ссылок на другие порции информации (текст, графику, звуковые и видеоклипы и т.д.), дополнительные источники. Многообразие таких ссылок и порождает Всемирную паутину WWW.

В *Web* эти ссылки называют «Унифицированным указателем ресурсов *URL*». Каждый документ или файл в Интернет имеет собственный *URL*, что позволяет легко сослаться на него из других документов.

Язык разметки *HyperText Markup Language (HTML)* является основой *World Wide Web*. С помощью конструкций языка создаются *Web* страницы (сайты), которые можно просматривать с помощью специальных программ - обозревателей, или как их еще называют - браузеров. Наиболее известными браузерами являются *Microsoft Internet Explorer*, *Mozilla Firefox*, *Opera*, *Google Chrome*.

Протоколы *Internet*. WWW использует собственный протокол связи между отдельными узлами: *HTTP (HyperText Transfer Protocol)*. Любой протокол - это набор правил, которые используются компьютерами для обмена информацией. Кроме *HTTP* можно использовать и другие протоколы, получая доступ и другим приложениям *Internet*: *UseNet* - для доступа к группам новостей, *FTP* - для загрузки файлов; *POP3*, *SMTP* -электронной почте, *Skype* для телефонного общения и др.

WWW - в отличие от телевидения, интерактивная среда. Пользователи сами определяют, какой из узлов посещать, как долго изучать полученную информацию. Более того, на *Web* страницах часто можно ввести диалог с их авторами.

Web использует технологию "клиент - сервер". Это означает, что где — то в сети существует компьютер, на котором работает программное обеспечение *Web* - сервера, управляющее доступом к информации. Различные пользователи со всего мира являются клиентами сервера и получают от него информацию с помощью своих браузеров.

1.2.2. Создание *HTML* – файлов

Основным понятием языка *HTML* является понятие тег - инструкция обозревателю, как отображать текст.

Теговая модель описывает документ как совокупность контейнеров, каждый из которых начинается и заканчивается тегами. Общая структура контейнера:

```
<"имя тега" "список атрибутов">
содержание контейнера
</"имя тега">
```

Большинство тегов спарены т.е. за открывающим тегом следует соответствующий закрывающий тег, а между ними содержится текст

или другие теги. В таких случаях два тега и часть документа, отделенная ими, образуют блок, называемый *HTML* элементом. Некоторые теги являются элементами *HTML* сами по себе, и в рамках спецификации *HTML* для них соответствующий конечный тег необязателен, однако для спецификации *XHTML* закрывать тег обязательно для валидности *WEB* документа.

Независимо от того, как выглядит домашняя страница, и какую информацию необходимо отобразить, существует три тега, которые в соответствии со стандартами *HTML* и *WWW* должны присутствовать в каждой *Web*-странице:

- **<html>** Сообщает браузеру, что документ написан на языке *HTML*.
- **<head>** Отмечает вводную и заголовочную части *HTML*-документа.
- **<body>** Отмечает основной текст и информацию.

Эти теги необходимы *Web*-браузеру для определения различных частей *HTML*-документа, но они не оказывают прямого влияния на внешний вид *Web*-страницы. Они необходимы для того, чтобы последующие нововведения в *HTML* правильно интерпретировали *Web* - страницу, а также для того, чтобы она выглядела одинаково во всех *Web*-браузерах. Например, на *Web*-сервере запущена программа, которая просматривает все *HTML*-документы и составляет их список. Она просматривает только текст, расположенный внутри тегов **<head>**, поскольку здесь находится название документа. Таким образом, если в вашей домашней странице нет тегов **<head>** и **</head>**, она не будет включена в список. Так работает боль-

шинство средств поиска наиболее популярных *Web*-серверов. Они берут информацию из тегов *<head>* (и других).

Схематично разметку любой страницы можно представить следующим образом:

```
<!Спецификация DOCTYPE>
<html>
<head (Заголовок)>
[<Надпись заголовка>]
[<Стили>]
[<Скрипты>]
</head>
<body (Тело страницы)>
[<Разметка страницы>]
</body>
</html>
```

Спецификация документа должна быть указана, поскольку в противном случае браузер не будет иметь возможности корректно интерпретировать разметку. В различных браузерах по умолчанию отображение различно!

В соответствии с терминологией *SGML*, *html*-страница является документом, а элемент *html* – корневым элементом.

1.2.3. Основные элементы разметки

1.2.3.1. Элемент *html*

Элемент *html* является обязательным для любой страницы. Определяет границы *html*-документа.

1.2.3.2. Элемент *head*

Не является обязательным. Представляет собой контейнер, в котором размещаются элементы заголовка документа.

Внутри элемента *head* могут быть определены следующие элементы:

Таблица 1.1 — допустимые элементы внутри *head*.

Тэг	Описание
<i><title></i>	Определяет заглавие документа. Обычно браузеры отображают этот заголовок в заголовке окна.
<i><base /></i>	Определяет базовый адрес или адрес по умолчанию для всех ссылок на странице. Ссылки на странице, содержащие относительный путь будут дополняться базовым адресом. При перемещении такого документа все ссылки будут сохранены.

Тэг	Описание
<code><link /></code>	Определяет связи между документом и внешними ресурсами. Например подключение стилей отображения.
<code><meta /></code>	Определяет метаданные о документе. Среди них могут быть как дополнительные ключевые слова, используемые для поиска, так и указание кодировки страницы.
<code><script></code>	Определяет скрипты, выполняемые на стороне браузера.
<code><style></code>	Определяет информацию о стилях отображения в документе. В отличие от <code><link></code> эти стили будут внедрены в документ.

Следует отметить, что для страниц на русском языке обязательно необходимо указывать кодировку текста. Примеры:

```
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<meta http-equiv="Content-Type" content="text/html; charset=windows-1251" />
<meta http-equiv="Content-Type" content="text/html; charset=koi8-r" />
```

В настоящее время целесообразно использовать только *UNICODE*-кодировки, например *utf-8*.

1.2.3.3. Элемент *body*

Представляет собой контейнер, в котором размещаются элементы, описывающие тело (или содержимое) документа.

Исторически этот элемент имел дополнительные атрибуты, позволявшие выбрать цвет фона, шрифта, фоновый рисунок, однако эти атрибуты устарели и их использование запрещено.

Элемент содержит ряд стандартных атрибутов, таких как класс, стиль, заголовок элемента, язык.

1.2.4. Элементы разметки текста

Рассмотрим еще несколько часто используемых элементов, которые применяются для разметки текста.

Таблица 1.2 — Некоторые элементы разметки текста

Тэг	Описание
<code><p></code>	Параграф. Позволяет указать область текста, относящуюся к одному параграфу. Параграфы при отображении обычно отделяются увеличенным отступом. В рамках параграфа игнорируются переводы строк в исходном тексте. Формат переноса может быть задан стилями, но не имеет отношения к исходной разметке <i>Пример:</i> <code><p>Некоторый текст.</p></code> <i>Будет выведено:</i> Некоторый текст.

Тэг	Описание

	Перевод строки в любом месте текста, включая параграфы. Параграф при этом не разрывается. <i>Пример:</i> <p>Некоторый
 текст.</p> <i>Будет выведено:</i> Некоторый текст.
<pre>	Устаревший, но широко применяемый элемент, позволяющий обеспечить вывод текста, включая все переводы строк. <pre>Некоторый текст.</pre> <i>Будет выведено:</i> Некоторый текст.
	<i>Unordered list.</i> Позволяет разметить список, элементы которого в зависимости от значения атрибута type (устаревшего) будут отмечены как круг, квадрат или окружность. В разметке каждый элемент списка указывается тегами <i>Пример:</i> <ul type="disk"> Кофе Чай Молоко <i>Будет выведено:</i> Кофе Чай Молоко
	<i>Ordered list.</i> Позволяет разметить список, элементы которого будут отмечены порядковым номером. В разметке каждый элемент списка указывается тегами Имеются атрибуты (устаревшие) start, позволяющий указать стартовый номер, а также type, позволяющий указать тип нумерации: 1(арабские числа), A, a (буквы латинского алфавита), I, i (римские числа). <i>Пример:</i> Кофе Чай Молоко <i>Будет выведено:</i> <input type="checkbox"/> Кофе <input type="checkbox"/> Чай <input type="checkbox"/> Молоко
<h1>...<h6>	Устаревшие теги, предназначенные для того, чтобы выделить заголовки соответствующего уровня. Рекомендуется использовать стили.

1.2.5. Создание таблиц в HTML

настоящее время таблицы в HTML используются, в основном, для форматирования и оформления страниц, хотя и первоначальное их назначение как метода представления информации не утратило своего значения. Таблицы дают широчайшие возможности для оформления интернет-страниц. Рассмотрим подробнее их использование[2].

Элемент <table> служит контейнером для элементов, определяющих содержимое таблицы. Любая таблица состоит из строк и ячеек, которые

задаются с помощью тегов `<tr>` и `<td>`. Таблица начинается с тэга `<table>` и заканчивается тэгом `</table>`. Общая структура таблицы:

```
<table>
...
</table>
```

Тэг `<table>` может включать несколько атрибутов:

- **`align="left / center / right"`** – устанавливает расположение таблицы по отношению к полям документа;
- **`background="URL"`** – Определяет изображение, которое будет использоваться в качестве фонового рисунка таблицы;
- **`bgcolor="цвет"`** – Устанавливает цвет фона таблицы;
- **`width="значение"`** – Задаёт ширину таблицы. Её можно задать в пикселях (например, `width=400`) или в процентах от ширины страницы (например, `width=80%`). Если общая ширина содержимого превышает указанную ширину таблицы, то браузер будет пытаться «втиснуться» в заданные размеры за счёт форматирования текста. В случае, когда это невозможно, например, в таблице находятся изображения, параметр `width` будет проигнорирован, и новая ширина таблицы будет вычислена на основе её содержимого;
- **`border="толщина"`** – устанавливает ширину внешней рамки таблицы и ячеек в пикселях (например, `BORDER=4`). Если атрибут не установлен, таблица показывается без рамки;
- **`bordercolor="цвет"`** – Устанавливает цвет рамки таблицы;
- Для задание контейнера для строк таблицы служит тег `<tr>`.

Общая структура:

```
<table>
<tr>...</tr>
<tr>...</tr>
...
</table>
```

Атрибуты тега:

- **`bgcolor="цвет"`** – Устанавливает цвет фона строки таблицы;
- **`bordercolor="цвет"`** – Устанавливает цвет рамки вокруг строки. Рамка показывается, когда установлен параметр `border` с ненулевым значением у тега `<table>`.
- **`align="left / center / right / justify"`** – Задаёт выравнивание содержимого ячеек строки по горизонтали. Выравнивание осуществляется для всех ячеек в пределах одной строки.
- **`valign="top / middle / bottom / baseline"`** – Устанавливает вертикальное выравнивание содержимого ячеек в строке. По умолчанию содержимое ячейки располагается по её вертикали в центре *middle*.

- Допустимые значения:
- *top* – Выравнивание содержимого ячеек по верхнему краю строки;
- *middle* – Выравнивание посередине;
- *bottom* – Выравнивание по нижнему краю.
- **Пример.** Задать таблицу с синим фоном. Для первой строки установить золотой цвет.

```
<table bgcolor="blue" width="30%"> <tr bgcolor="#ffcc00">
<td> ячейка 1 </td>
</tr>
<tr>
<td> ячейка 2 </td>
</tr>
</table>
```

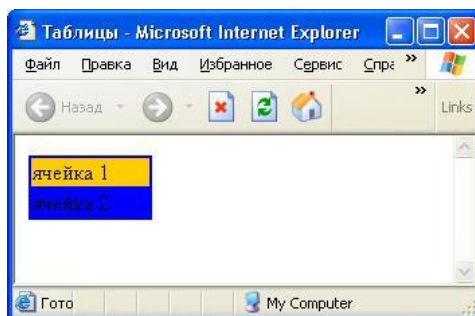


Рис. 1.1 — таблица с синим фоном

Пример. Задать таблицу с черным цветом сетки. Для первой строки установить красный цвет сетки

```
<table bordercolor="black" border="1" width="100%">
<tr bordercolor="red"> <td> ячейка 1 </td> </tr>
<tr>
<td> ячейка 2 </td> </tr>
</table>
```

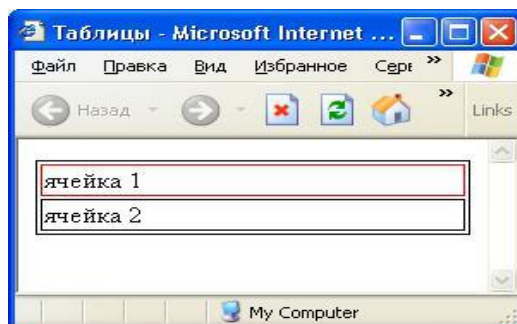


Рис. 1.2 — пример таблица

Для задание ячеек таблицы предназначен тег **<td>**. Контейнер должен размещаться внутри тега **<tr>**.

```
<table>
<tr> <td>ячейка 1, 1-строки </td> .... <td> ячейка n, 1-строки
</td> </tr>
<tr> <td>ячейка 1, 2-строки </td> .... <td> ячейка n, 2-строки
</td> </tr>
...
</table>
```

Атрибуты тега:

- **align="left / center / right / justify"** – Задаёт выравнивание содержимого ячейки по горизонтали.
- **valign="top / middle / bottom / baseline"** – Устанавливает вертикальное выравнивание содержимого ячейки.
- **background="URL"** – Определяет изображение, которое будет использоваться в качестве фонового рисунка таблицы.
- **bgcolor="цвет"** – Устанавливает цвет фона ячейки.
- **bordercolor="цвет"** – Устанавливает цвет рамки вокруг ячейки.
- **colspan="число"** – Устанавливает число ячеек, которые должны быть объединены по горизонтали.
- **rowspan="число"** – Устанавливает число ячеек, которые должны быть объединены по вертикали.
- **height="значение"** – Браузер сам устанавливает высоту таблицы и ее ячеек исходя из их содержимого. Однако при использовании параметра **height** высота ячеек будет изменена. Здесь возможны два варианта. Если значение **height** меньше, чем содержимое ячейки, то этот параметр будет проигнорирован. В случае, когда установлена высота ячейки, превышающая ее содержимое, добавляется пустое пространство по вертикали
- **width="значение"** – Задаёт ширину ячейки. Если общая ширина содержимого превышает указанную ширину ячейки, то браузер

будет пытаться «втиснуться» в заданные размеры за счет форматирования текста. В случае, когда это невозможно, например, в ячейке находится изображения, параметр **width** будет проигнорирован, и новая ширина ячейки будет вычислена на основе ее содержимого.

```
<table width="75%" border="2" cellpadding="0" cols="3" bgcolor="gold" align="center" bordercolor="black" >
<tr align="center">
<td rowspan="2">Папа</td>
<td colspan="2">День</td>
</tr>
<tr align="center">
<td colspan="2">Понедельник</td>
```



```

<td>Вторник</td>
</tr>
<tr>
<td rowspan="2" bgcolor="red">8:00-9:30</td>
<td>Пение</td>
<td>Рисование</td>
</tr>
<tr>
<td>Информатика</td>
<td>Математика</td>
</tr>
<tr>
<td rowspan="2" bgcolor="red">9:40-11:10</td>
<td>Философия</td>
<td>История</td>
</tr>
<tr>
<td>Физика</td>
<td>Химия</td>
</tr>
</table>

```

Пара	День	
	Понедельник	Вторник
8:00-9:30	Пение	Рисование
	Информатика	Математика
9:40-11:10	Философия	История
	Физика	Химия

Рис. 1.3 — Таблица расписания

1.2.6. Вставка изображений

Для вставки изображений применяется элемент ``. Элемент `img` имеет обязательный атрибут `src`, который должен содержать `URL` графического файла. Любой современный браузер поддерживает отображение форматов `png`, `jpg`, `gif`. Рекомендуется использование формата `png` по причине отсутствия лицензионных ограничений.

Другой обязательный атрибут — `alt`. Его использование необходимо для того, чтобы в браузерах, которые не могут отобразить изображение, вместо изображения был выведен текст.

Обратите внимание на то, что значение атрибута `alt` также используется поисковыми системами для того, чтобы ассоциировать изображение, указанное в атрибуте `src` с текстом.

Необязательные атрибуты *width* и *height* указывают ширину и высоту изображения в пикселах или процентах. Их использование необходимо для корректного отображения страницы до того, как браузер загрузил изображение. Помните о том, что каналы связи не идеальны, а скорости порядка 10-100 килобит до сих пор встречаются в условиях плохого покрытия у беспроводных модемов, включая технологии *3G* и *LTE*! Если не учитывать скорость загрузки страниц, пользователи, просматривающие такие страницы, будут видеть постоянно меняющуюся разметку.

Пример:

```

```

1.2.7. Вставка ссылок

Гиперссылки на *Web* - страницы - одно из основных свойств WWW. С помощью таких ссылок и родилась Всемирная паутина.

Для создания гипертекстовой ссылки используется специальный парный тег

```
<a href= "Значение ссылки"> </a>.
```

Рассмотрим подробнее структуру гиперссылки на примере:

```
<a href="http://www.sevsu.ru " title="NEWS SevGU"> Новости СевГУ </a>.
```

Здесь *http://www.sevsu.ru* - представляет *URL* - адрес файла в *Internet*.

«Новости СевГУ» - текст, который видит пользователь на экране обозревателя. Этот текст обычно выделен синим цветом, и при наведении на него мыши появляется характерный указатель в виде указательного пальца.

Если документ, на который ссылается данная ссылка, находится в той же самой директории, что и исходный документ, то в качестве *URL* - адрес файла можно просто указать имя файла, например.

```
<a href="Second_Page.html"> Следующая страница </a>.
```

Разумеется, ссылаться можно не только на *HTML*-страницы, но и на любые файлы, будь то картинки, будь то фильмы, будь то музыка, будь то архивы, будь то ещё всё, что угодно.

Последний атрибут - это *"title"*. Этот атрибут задаёт текст, который будет виден при наведении мышки на ссылку.

Что касается того, что внутри тега, то в большинстве случаев - это обычный текст. Очень часто делают картинку внутри тега *<a>*, тогда эта ссылка будет в виде картинки.

Тег ** означает конец гиперссылки.

1.3. Ход выполнения работы

В качестве примера, разработаем простой сайт, состоящий из 2 страниц. Главный *html* файл будет иметь имя *index.html*. В него запишем следующий код :

```
<!doctype html>
<html lang="ru">
  <head>
    <meta charset="utf-8" />
    <title> Главное меню </title>
  </head>
  <body background="images/фон.jpg">
    
    
      <h1 align="center"> Главное меню </h1>
      <h2 align="center"><i> Небольшое вступление </i></h2>
      <p align="left"> <font size="5"> Привет, меня зовут Вася и
этот сайт обо мне. <br>
      Обучаюсь в <u>Севастопольском Государственном университе-
те,</u> <br>

      в институте радиоэлектроники и информационной безопасно-
сти, <br>
      на кафедре <font color="blue"><u> "Физика живых систем"
</u></font><br><br>
      Учусь в основном на <font
color="red"><b>"Отлично"</b></font></p><br>
      Про мои увлечения, также перейдя по этой ссылке: <a
href="hobbi.html" > Увлечения и хобби</a></font> <br>
</body>
</html>
```

Для корректного отображения текста на странице необходимо в секции **<head>** прописать обязательно кодировку страницы.

```
<meta charset="utf-8" />
```

Что бы установить фон страницы, в теги **<body>** нужно указать атрибут **background**, в котором нужно указать название изображения, которое будет выступать фоном.

```
<body background="images/фон.jpg">
```

Тег **** как было описано в пункте 1.2.6. позволяет вставить изображение на страницу сайта. Для этого в атрибуте **src** пропишем ссылку на изображение:

```

```

Атрибут *align* позволяет выровнять изображение по левому краю страницы.

Для перехода на другую страницу необходимо создать гиперссылку:

```
<a href="hobbi.html" > Увлечения и хобби</a>
```

В атрибуте *href* тега *<a>* указывается, на какую страницу будет осуществляться переход. Так как *html* файлы *index.html* и *hobbi.html* находятся в одной папке, можно указывать только имя файла, на который будет переход. Между открывающим тегом *<a>* и закрывающим тегом ** указывается название гиперссылки.

Создадим второй файл под именем *hobbi.html* и запишем следующий код:

```
<!doctype html>
<html lang="ru">
  <head>
    <title> Увлечения и хобби </title>
  </head>
  <body background="images/фон.jpg">
    <h2 align="center"> О моих увлечениях и хобби </h2>
    <p align="center"> <a
href="index.html">Главное меню</a>
```

```
<table align="center" border="1" width="40%">
  <td align="center" bgcolor="yellow"
colspan="3"><b>Увлечения</b></td>
  <tr >
    <td> <b>Увлечение</b> </td>
    <td> <b>Любимое</b> </td>
    <td> <b>Картинка</b></td>
  </tr>
  <tr >
    <td> <b><u> Музыка </u></b></td>
    <td> Би-2, Imagine Dragons </td>
    <td>   </td>
  </tr>
  <tr>
    <td> <b><u>Кино</b></u> </td>
    <td> Комедии, детективы, приключения, мюзиклы </td>
    <td>   </td>
  </tr>
  <tr>
    <td> <b><u>Книги</b></u> </td>
    <td> Энциклопедии </td>
    <td> 
 </td>
```

```

</tr>
<tr>
  <td> <b><u>Наука</b></u> </td>
  <td> Биология (альгология), физика, химия </td>
  <td>   </td>
</tr>
<tr>
  <td> <b><u>Игры</b></u> </td>
  <td> World of Warcraft, Dark souls, Spore </td>
  <td>   </td>
</tr>
<td align="center" bgcolor="yellow" col-
span="3"><b>Хобби</b></td>
<tr>
  <td> <b><u>Азартные игры</b></u> </td>
  <td> Игры в карты, фокусы </td>
  <td>  </td>
</tr>
<tr>
  <td> <b><u>PaperCraft</b></u> </td>
  <td> Бумажные изделия в виде голов дракона, черепа
</td>
  <td>  </td>
</tr>
</table>
</body>
</html>

```

В данной странице расположена краткая характеристика увлечений студента. Увлечения размещены в таблице с помощью парного тега **<table>**.

Для возвращения на предыдущую страницу нужно обязательно прописать гиперссылку на эту страницу.

a href="index.html">Главное меню

1.4. Порядок выполнения

1.4.1. Создать личный Web - сайт с информацией о себе, включающий минимум, две страницы. На первой странице с названием *index.html* поместите общую информацию о себе, фотографию. На второй более частные подробности, например, информацию о своих планах на следующую неделю, предыдущие места учебы, работы, Ваших увлечениях и т.п. и оформить в виде таблицы.

1.4.2. Подготовить отчет по работе.

1.5. Содержание отчета

1.5.1. Сформулировать цель работы.

1.5.2. Привести постановку задачи.

1.5.3. Привести краткие теоретические сведения о технологии *HTML*, и описать основные элементы разметки.

1.5.4. Привести тексты программ.

1.5.5. Составить таблицу, содержащую теги использованные для создания *HTML* – документа (см. таблица 1.3).

Таблица 1.3 — имя и назначение тегов

Имя тега	Назначения тега

1.5.6. Привести скриншоты выполнения программы.

1.5.7. Выводы.

1.6. Контрольные вопросы

1.6.1. Что такое технология *HTML*?

1.6.2. Что такое тег?

1.6.3. Привести структуру *HTML* документа. Описать назначение тегов *<html>*, *<head>*, *<meta>*, *<body>*.

1.6.4. Что такое атрибут тега? Формат записи атрибутов.

1.6.5. Опишите теги для форматирования текста в *HTML*

1.6.6. Как представляются гиперссылки в *HTML* документе?

1.6.7. Как включаются графические объекты в *HTML* документы?