

Лабораторная работа №1

Создание простых HTML страниц.

Структура HTML документов, теги форматирования.

Теоретические сведения

1. Теги задания структуры документа

Язык **HTML** (*Hyper Text Markup Language*) - это язык разметки гипертекста. HTML является описательным языком. В состав языка входят развитые средства для создания различных уровней заголовков, шрифтовых стилей, различные списки, таблицы, иллюстраций, аудио- и видеофрагментов и др.

Основной единицей языка HTML является тег. Теговая модель описывает документ как совокупность контейнеров, каждый из которых начинается и заканчивается тегами. Тег состоит из следующих друг за другом элементов:

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

Кроме тегов, элементами HTML являются CER (*Character Entity Reference*), они предназначены для представления специальных символов в документе HTML, которые могут быть неверно обработаны браузером и начинаются с амперсанда &. Также спецсимволы могут задаваться в виде трехзначных кодов символов **&#nnn;** .

Пример 2.3:

Числовой код	Именная замена	Символ	Описание
"	"	"	Кавычка
&	&	&	Амперсанд
<	<	<	Меньше
>	>	>	Больше
 	 		Неразрывный пробел
¡	¡	!	Перевернутый восклицательный знак
¢	¢	¢	Цент
£	£	£	Фунт
¤	¤	¤	Валюта
¥	¥	¥	Йена
¨	¨	¨	Умляют
©	©	©	Копирайт
«	«	«	Левая угловая кавычка
®	®	®	Зарегистрированная торговая марка
±	±	±	Плюс или минус
»	»	»	Правая угловая кавычка

Все теги HTML по их назначению и области действия можно разделить на следующие основные группы:

1. определяющие структуру документа;
2. оформление блоков гипертекста (параграфы, списки, таблицы, картинки);
3. гипертекстовые ссылки и закладки;
4. формы для организации диалога;
5. вызов программ.

1.2 Структура HTML-документа.

HTML-документ — это один большой контейнер, начинающийся с тега `<HTML>` и заканчивающийся тегом `</HTML>`. Также документ включает два вложенных контейнера: заголовка документа (**HEAD**) и тела документа (**BODY**). Таким образом любой документ на языке HTML имеет следующую минимальную структуру предоставления информации:

```
<HTML>
<HEAD>
  <TITLE> НАЗВАНИЕ </TITLE>
</HEAD>
<BODY>
  ТЕКСТ ДОКУМЕНТА
</BODY>
</HTML>
```

1.3 Основные элементы заголовка

К основным элементам HTML-разметки заголовка относятся:

1. **HEAD** (элемент разметки *HEAD*);
2. **TITLE** (заглавие документа);
3. **BASE** (база URL);
4. **META** (метаинформация);
5. **LINK** (общие ссылки);
6. **STYLE** (описатели стилей);
7. **SCRIPT** (скрипты).

TITLE служит для задания имени документу

```
<TITLE>название документа</TITLE>
```

BASE служит для определения базового URL для гипертекстовых ссылок документа, заданных в неполной форме.

Пример 2.4:

```
<HTML>
<HEAD>
  <BASE HREF=http://gsu.by/cgi-bin/>
</HEAD>
<BODY>
  <BASE HREF=http://gsu.by/Index/>
  ...
</BODY>
</HTML>
```

META содержит управляющую информацию, которую браузер использует для правильного отображения и обработки содержания тела документа.

Пример 2.5: Загрузка документа `Page1.html` через 1 сек:

```
<META HTTP-EQUIV="Refresh" CONTENT="1; URL=Page1.html">
```

Обновление страницы каждые 3 сек:

```
<META HTTP-EQUIV="Refresh" CONTENT="3" >
```

Пример 2.6: пример задания кодировки документа на стороне клиента

```
<META HTTP-EQUIV="Content-type" CONTENT="text/html;
CHARSET=windows-1251">
charset=utf-8
```

Пример 2.7

Запрещение кэширования:

```
<META HTTP-EQUIV="Cache-Control" CONTENT="no-cache">
```

Запрещение хранения документа после пересылки:

```
<META HTTP-EQUIV="Cache-Control" CONTENT="no-store">
```

Пример 2.8 Описание для evrosport.html

```
<META NAME="description" http-equiv="description"
content=" Канал Eurosport: все о спорте в режиме online, новости
спорта: футбол, хоккей, теннис, баскетбол, бокс; Eurosport;">
```

```
<META NAME="keywords" HTTP-EQUIV="keywords"
CONTENT=" спорт новости спорта канал весь спорт спортивные игры
футбол бокс хоккей баскетбол все спорте eurosport евроспорт ">
```

LINK используется для загрузки внешних описателей стилей.

```
<LINK [REL=тип_отношения] [HREF=URL] [TYPE=тип_содержания]>
```

Пример 2.9 Загрузка стилей из файла style.css

```
<LINK REL=stylesheet href="../css/style.css" TYPE="text/css">
<LINK href="/lecture.css" type=text/css rel=stylesheet>
```

STYLE – предназначен для размещения описателей стилей.

SCRIPT служит для размещения кода JavaScript, VBScript или JScript.

```
<SCRIPT [TYPE=тип_языка_программирования]>
JavaScript/VBScript-код
</SCRIPT>
```

или

```
<SCRIPT [TYPE=тип_языка_программирования]
[ SRC=URL] >
</SCRIPT>
```

1.4 Теги форматирования тела документа

Теги тела документа предназначены для управления отображением информации и задания структуры документа. Описание тела HTML документа начинается с тега **BODY**. Тег определяет видимую часть документа. В этом разделе располагается вся содержательная часть документа (текст статьи, фотографии, формы для заполнения, другие объекты). Тег имеет ряд необязательных атрибутов:

1. **bgscolor=цвет** – Устанавливает цвет фона документа;
2. **background = «имя файла»** – Указывает на url-адрес изображения фона документа (например, файл с расширением *.gif или *.jpeg);

Пример 2.11

```
<BODY BACKGROUND="image.gif">
```

3. **TEXT = цвет** – Устанавливает цвет текста документа;
4. **LINK =цвет** – Цвет гипертекстовых ссылок;

5. VLINK=цвет – Цвет открытых гипертекстовых ссылок;
6. ALINK=цвет – Цвет "активных" гипертекстовых ссылок;
7. LEFTMARGIN=n и TOPMARGIN=n – задают левое и верхнее поле для всей страницы.

<BODY LEFTMARGIN ="40">

Цвет можно задавать как текстом так и в формате RGB.

Таблица 2.2. Цвета			
Название	Код	Название	Код
aqua	#00FFFF	navy	#000080
black	#000000	olive	#808000
blue	#0000FF	purple	#800080
fuchsia	#FF00FF	red	#FF0000
gray	#808080	silver	#C0C0C0
green	#008000	teal	#008080
lime	#00FF00	white	#FFFFFF
maroon	#800000	yellow	#FFFF00

Пример 2.12 цвет текста как синий, фона — белый, пройденные ссылки красные, а новые ссылки зеленые.

<BODY BGCOLOR=#FFFFFF TEXT=#0000FF VLINK=#FF0000 LINK=#00FF00>

Для форматирования документа используются следующие теги.

<H1> ... </H1> — <H6> ... </H6> – Метки вида **<Hi>** (где i — цифра от 1 до 6) описывают заголовки шести различных уровней. Заголовок первого уровня — самый крупный, шестого уровня, естественно — самый мелкий. Тег имеет ряд необязательных атрибутов:
ALIGN=CENTER | RIGHT | LEFT – задает выравнивание заголовка.

Пример:

<H1 ALIGN=CENTER**>Выравнивание заголовка по центру </H1>**

Теги **<P> ... </P>** – описывает абзац. Все, что заключено между **<P>** и **</P>**, воспринимается как один абзац. Тег имеет ряд необязательных атрибутов:

ALIGN=CENTER | RIGHT | LEFT | JUSTIFY – задает выравнивание заголовка.

Пример:

<P ALIGN=justify**> выравнивание абзаца по ширине </P>**

Тег **
** –используется, если необходимо перейти на новую строку, не прерывая абзаца. Очень удобно при публикации стихов

Пример:

<P>

**Однажды в студеную зимнюю пору
**

**Сижу за решеткой в темнице сырой.
**

**Гляжу - поднимается медленно в гору
**

Вскормленный в неволе орел молодой.

</P>

Тег **<PRE>** – позволяет отобразить текст "как есть" (без форматирования), теми же символами и с тем же разбиением на строки.

Тег **...** – задает характеристики шрифта, такие как размер, цвет и гарнитура. Тег имеет ряд атрибутов:

COLOR=ЦВЕТ – Устанавливает цвет текста;

FACE=«ГАРНИТУРА» – Определяет гарнитуру шрифта:

serif –шрифты с засечками (антиквенные), типа **Times**;

sans-serif –рубленные шрифты (шрифты без засечек или гротески), типичный представитель – Arial;

SIZE=X – Задает размер шрифта в условных единицах от 1 до 7. Размер по умолчанию 3.

Пример:

```
<font color="#0000FF" size="6" face="Arial">Мой первый сайт</font>
```

Тег **<DIV>** – предназначен для выделения фрагмента документа с целью изменения вида содержимого. Как правило, вид блока управляется с помощью стилей. Закрывающий тег обязателен. **Синтаксис:**

```
<div>...</div>
```

Атрибуты:

ALIGN="CENTER | LEFT | RIGHT | JUSTIFY" – Задает выравнивание содержимого тега.

TITLE – Добавляет всплывающую подсказку к содержимому.

Пример:

```
<div align="left" title="ПОДСКАЗКА">
```

```
..... ТЕКСТ .....
```

```
</div>
```

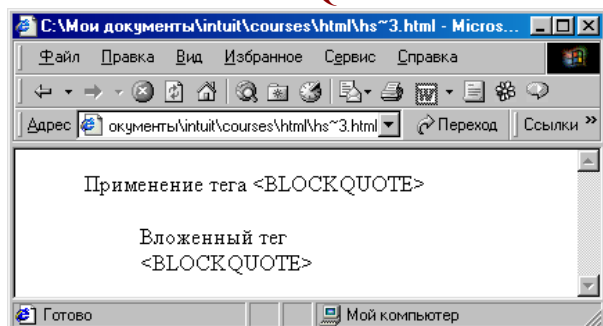
Дополнительные теги управляющие формой отображения, приведены в таблицах 2.3 и 2.4.

Таблица 2.3. Теги, управляющие формой отображения	
Тег	Значение
<I>...</I>	Курсив (Italic)
...	Усиление (Bold)
<TT>...</TT>	Телетайп
<U>...</U>	Подчеркивание
<S>...</S>	Перечеркнутый текст
<BIG>...</BIG>	Увеличенный размер шрифта
<SMALL>...</SMALL>	Уменьшенный размер шрифта
<SUB>...</SUB>	Подстрочные символы
<SUP>...</SUP>	Надстрочные символы

Таблица 2.4. Теги, характеризующие тип информации	
Тег	Значение
...	Типографское усиление
<CITE>...</CITE>	Цитирование
...	Усиление
<CODE>...</CODE>	Отображает примеры кода (например, "коды программ")
<SAMP>...</SAMP>	Последовательность литералов
<KBD>...</KBD>	Пример ввода символов с клавиатуры
<VAR>...</VAR>	Переменная

<code><DFN>...</DFN></code>	Определение
<code><Q>...</Q></code>	Текст, заключенный в двойные кавычки

Тег **<BLOCKQUOTE>** добавляет поля слева и справа от текста. При неоднократном использовании **<BLOCKQUOTE>** текст все больше сжимается к центру.



1.5 Графические объекты. Ссылки

Тег `<hr>` – описывает горизонтальную линию. Тег имеет ряд необязательных атрибутов:

ALIGN=*CENTER | RIGHT | LEFT* – задает выравнивание;

WIDTH=*XX* – ширина линии в процентах/пикселях (точках);

SIZE=*X* – толщина линии в пикселях;

COLOR=*ЦВЕТ* – цвет линии.

Пример:

```
<HR SIZE=4 WIDTH=50% color=green align= center >
```

Тег `<MARQUEE>...</MARQUEE>` – позволяет представить текст в виде бегущей строки.

Этот тэг может иметь следующие атрибуты:

BGCOLOR – цвет фона;

HEIGHT – высота бегущей строки;

WIDTH – ширина бегущей строки;

DIRECTION – направление перемещения текста (left|right) (по умолчанию – left).

Пример:

```
<marquee width=80% height=10 color= yellow >
```

Это бегущая строка

```
</marquee>
```

Тег **** предназначен для отображения на веб-странице изображений в графическом формате GIF, JPEG или PNG. Этот тег имеет единственный обязательный параметр **src**, который определяет адрес файла с картинкой. При этом вокруг изображения отображается рамка, которую можно убрать, добавив параметр **border="0"** в тег ****. Рисунки также могут применяться в качестве ссылок. Тег имеет ряд атрибутов:

ALIGN=*"BOTTOM | LEFT | MIDDLE | RIGHT | TOP"* – Определяет как рисунок будет выравниваться по краю и способ обтекания текстом. Наиболее популярные параметры – **left** и **right**, создающие обтекание текста вокруг изображения

BORDER=*"ТОЛЩИНА РАМКИ"* – Толщина рамки вокруг изображения.

HEIGHT=*"ВЫСОТА"* – Высота изображения.

WIDTH=*"ШИРИНА"* – Ширина изображения.

HSPACE=*"ОТСТУП ПО ГОРИЗОНТАЛИ"* – Горизонтальный отступ от изображения до окружающего контента.

VSPACE=*"ОТСТУП ПО ВЕРТИКАЛИ"* – Вертикальный отступ от изображения до окружающего контента.

SRC=*"URL"* – Путь к графическому файлу.

Пример:

Разместить картинку из файла square.gif на странице.

```

```

Разместить с права от текста картинку square.gif с заданными значениями высоты и ширины.

```

```

Разместить картинку sample.gif расположенную в папке **images** с заданными значениями высоты и ширины и рамкой вокруг изображения..

```

```

Самыми распространенными графическими форматами в Web являются PNG, GIF и JPEG. Формат GIF - поддерживает изображения с глубиной цвета больше восьми бит. Используется для четких контурных рисунков. Допускает анимацию. JPEG – для изображений фотографического качества.

Тег **<A>** является одним из важных элементов HTML и предназначен для создания ссылок. В зависимости от присутствия параметров **name** или **href** тег **<A>** устанавливает ссылку или якорь. **Якорь** – закладка внутри страницы, которую можно указать в качестве цели ссылки. При использовании ссылки, которая указывает на якорь, происходит переход к закладке внутри веб-страницы. Допускается переход к якорю и на другой странице.

Для создания ссылки необходимо:

- 1) задать ссылку (объект на который необходимо нажать);
- 2) указать адрес документа, на который следует перейти.

Адрес ссылки может быть абсолютным и относительным. Абсолютные адреса работают везде, независимо от имени сайта или веб-страницы. Относительные ссылки, построены относительно текущего документа или корня сайта.

Синтаксис:

```
<a href="URL">...</a>
```

```
<a name="идентификатор">...</a>
```

Закрывающий тег Обязателен.

Атрибуты:

1. **href** – Задаёт адрес документа, на который следует перейти. Поскольку в качестве адреса ссылки может использоваться документ любого типа, то результат перехода по ссылке зависит от конечного файла. Так, архивы (файлы с расширением zip или rar) будут сохраняться на локальный диск. По умолчанию новый документ загружается в текущее окно браузера,
2. **name** – Устанавливает имя якоря внутри документа.
3. **target** – Имя окна или фрейма, куда браузер будет загружать документ.
4. **title** – Добавляет всплывающую подсказку к тексту ссылки.

Пример 1. Использование тега <A>

Задание ссылки с использованием изображения

```
<p><a href="images/xxx.jpg">Посмотрите на мою фотографию!</a></p>
```

Задание ссылки с использованием текста

```
<p><a href="tip.html">Как сделать такое же фото?</a></p>
```

Описание атрибутов тега <A>

Параметр **HREF** – Задаёт адрес документа, на который следует перейти. Поскольку в качестве адреса ссылки может использоваться документ любого типа, то результат перехода по ссылке зависит от конечного файла. Так, архивы (файлы с расширением zip или rar) будут сохранять-

ся на локальный диск. По умолчанию новый документ загружается в текущее окно браузера, однако это свойство можно изменить с помощью параметра **target**.

Пример 2. Использование параметра href

```
<p><a href="knob.html">Относительная ссылка</a></p>
```

```
<p><a href="http://www.gsu.ru/html/example/knob.html">Абсолютная ссылка</a></p>
```

Параметр **NAME** - Параметр **name** используется для определения **якоря внутри** страницы. Синтаксис:

```
<a name="закладка">...</a>
```

Между тегами `` и `` текст писать не обязательно, так как требуется лишь указать местоположение перехода по ссылке.

Вначале следует задать в соответствующем месте закладку и установить ее имя параметром **name** тега `<A>`. Имя ссылки на закладку начинается символом **#**, после чего идет название закладки. Можно также делать ссылку на закладку, находящуюся в другой веб-странице и даже другом сайте. Для этого в адресе ссылки надлежит указать ее адрес и в конце добавить символ решетки **#** и имя закладки.

Пример 3. Создание якоря с помощью параметра name

```
<p><a name="top"></a></p>
```

```
<p> Текст
```

```
...
```

```
</p>
```

```
<p><a href="#top">Наверх</a></p>
```

TARGET – управляет открытием страницы. В XHTML применение этого параметра запрещено. Синтаксис

```
<a target="имя окна">...</a>
```

В качестве аргумента используется имя окна или фрейма, заданное параметром **name**. Если установлено несуществующее имя, то будет открыто новое окно. В качестве зарезервированных имен используются следующие:

1. **_blank** – Загружает страницу в новое окно браузера.
2. **_self** – Загружает страницу в текущее окно. (значение по умолчанию)
3. **_parent** – Загружает страницу во фрейм-родитель, если фреймов нет, то этот параметр работает как **_self**.
4. **_top** – Отменяет все фреймы и загружает страницу в полном окне браузера, если фреймов нет, то этот параметр работает как **_self**.
- 5.

Пример 4. Открытие ссылки в новом окне

```
<p>
```

```
<a href="new.html" target="_blank">Открыть в новом окне</a>
</p>
```

TITLE - Добавляет поясняющий текст к ссылке в виде всплывающей подсказки, которая отображается, при подводе курсора мыши. Синтаксис:

```
<a title="текст">...</a>
```

Пример 5. Создание всплывающей подсказки


```
<p><a href="zoo.html" title="Рисунки различных животных и не только  
...">Рисунки</a></p>
```

Задание активных изображений

Активное изображение (image maps) — это изображение чувствительные к щелчкам мыши, с активными областями, которые ссылаются на другие страницы или ресурсы.

Процесс создания активного изображения состоит из двух этапов. Сначала необходимо определить на *картинке* области, которые нужно сделать активными, а потом соотнести их со ссылками на другие URL. Границы активных областей задаются координатами углов прямоугольника и многоугольника или центра и радиуса круга. Приведем пример *активных изображений*.

```
<IMG SRC="image.gif" ALT="Изображения" USEMAP="#imap">
```

```
<MAP NAME="imap">
```

```
<AREA SHAPE="rect" COORDS="0,0,100,100"
```

```
HREF="http://www.GSU.BY/index.html">
```

```
<AREA SHAPE="rect" COORDS="100,0,200,100"
```

```
HREF="http://www.GSU.BY/help.html">
```

```
<AREA SHAPE="default" nohref>
```

```
SHAPE= rect | poly | circle
```

Создание активного изображения можно использовать вспомогательными программами, например **MapEdit**.

Задание 1.

На рабочем диске создать папку «My site». В простом текстовом редакторе (например, NotePad++), создать **HTML** документ следующей структуры:

1. стартовая страница (**index.html**), должна содержать информацию следующего характера – Название лабораторной работы, цель, ФИО, группа. На странице разместить ссылку на главную страницу сайта (**main.html**), переход на которую также должен выполняться автоматически через 5 сек;

2. главная страница сайта (**main.html**), представляет собой оглавление следующего списка тем:

а) биографические сведения о авторе страницы (**biography.html**), фото обязательно;

б) увлечения, хобби (**hobby.html**). Текст не менее 2-3 страницы, изображения, списки и т.д.

в) мои любимые фотографии (изображения) (**fotos.html**). Показ изображений организовать в виде слайд – шоу с интервалом в 2 – 3 сек. Каждая страница должна содержать подписи к рисункам и ссылку на следующую страницу. При необходимости изменить масштаб изображения.

г) свободная тема, по выбору студента (**my.html**).

д) карта сайта, реализующая переходы на страницы **main.html**, **biography.html**, **hobby.html**, **fotos.html**, **my.html**). Реализовать в виде *image maps* – карты изображения (рис. 1).

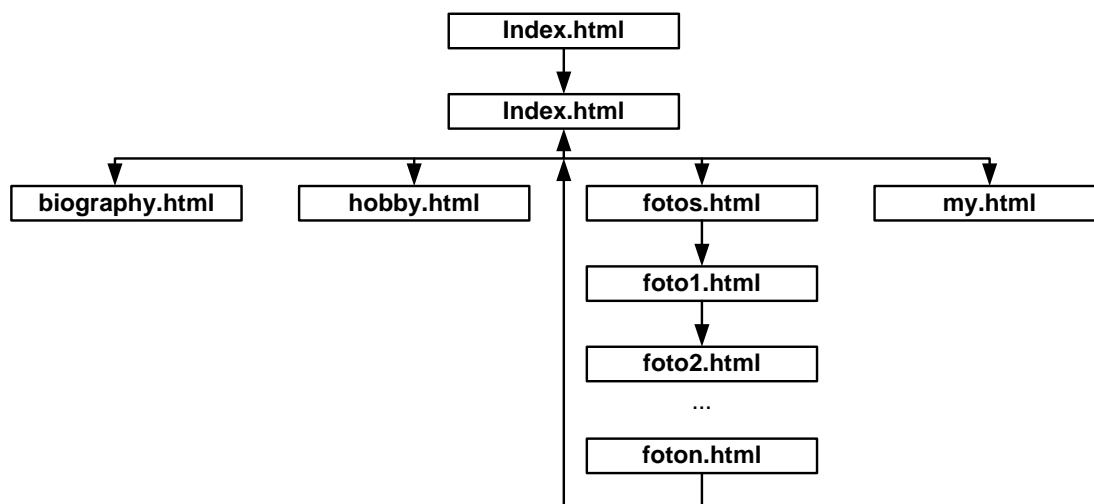


Рисунок 1. – структура сайта

Все страницы должны быть презентабельно оформлены, содержать осмысленную информацию. При оформлении должны использоваться различные элементы оформления и структурирования данных. Ссылки между страницами должны соответствовать рис. 1. Полученный сайт продемонстрировать преподавателю и только после этого оформлять отчет. В отчете привести скриншоты страниц **main.html**, **biography.html** и **html** текст всех страниц.

Вопросы к защите

1. Глобальные компьютерные сети: основные понятия, принципы функционирования;
2. Адресация в Интернете;
3. Архитектура web-приложений. Преимущества и недостатки Web;
4. Современные Web – технологии;
5. Язык HTML. Дать понятие тегу и его атрибутам;
6. структура HTML документа. Теги задания структуры;
7. структура HTML документа. Основные элементы заголовка;
8. Теги форматирования документа. Заголовки, абзацы, разрыв строки;
9. Теги форматирования документа. Горизонтальные линии, форматирование шрифта.
10. Тег FONT;
11. Задание бегущей строки;
12. Размещение изображений;