

Федеральное государственное бюджетное образовательное  
учреждение высшего образования  
«Брянский государственный аграрный университет»

КАФЕДРА ИНФОРМАЦИОННЫХ СИСТЕМ И ТЕХНОЛОГИЙ

УЛЬЯНОВА Н.Д.

# ОСНОВЫ ЯЗЫКА HTML

Учебно-методическое пособие



Брянская область  
2016

УДК 004.43

Ульянова Н.Д. Основы языка HTML. Учебно-методическое пособие.-  
Брянск: Издательство Брянский ГАУ, 2016.- 64 с.

В пособии кратко раскрывается понятие о языке HTML, излагаются принципы создания веб-страниц, приводятся конкретные примеры создания web-приложений.

Пособие предназначено для студентов направления подготовки 09.03.03 Прикладная информатика профиль Прикладная информатика в экономике, , а также студентов различных направлений высших учебных заведений, имеющих практические навыки работы с Интернет-ресурсами и в области программирования с целью изучения основ языка HTML и выполнения лабораторно-практических занятий.

Рекомендовано к изданию решением методической комиссии экономического факультета от 30.06.2016г., протокол №9.

Рецензент:

к.э.н., доцент кафедры информационных систем и технологий Лысенкова С.Н.

© Брянский ГАУ, 2016

©Н.Д. Ульянова, 2016

## ВВЕДЕНИЕ

Интернет-программирование (Web-программирование) - раздел программирования, ориентированный на разработку web-приложений. На современном этапе развития общества во всех сферах жизнедеятельности человека широкое распространение получили компьютерные сайты. Умея работать с необходимыми в профессиональной деятельности навыками интернет-программирования, человек информационного общества приобретает новое восприятие мира.

В предлагаемом пособии рассматриваются технологии и концепция создания web-страниц, принципы разработки структуры web-страниц, основные элементы языка HTML, средства оформления страниц, правила использования фреймов и форм. Описываются способы представления внешнего вида страниц с использованием каскадных таблиц стилей. Представлены практические задания и последовательность их выполнения при проведении лабораторных работ.

Пособие предназначено для профиля подготовки бакалавров «Прикладная информатика (в экономике)» направления «Прикладная информатика», а также студентов IT-направлений высших учебных заведений, имеющих практические навыки работы с Интернет-ресурсами и в области программирования.

Представленные материалы имеют целью формирование *компетенций и освоение* обучающимися видов профессиональной деятельности в соответствии с ФГОС ВО и ОПОП ВО по направлению подготовки 09.03.03 Прикладная информатика (уровень бакалавриата).

## 1. Общая структура сайта

*Интернет-программирование (Web-программирование)* - раздел программирования, ориентированный на разработку web-приложений.

*Web-приложение* - программа, обеспечивающая функционирование динамического сайта сети Интернет.

Термин *Веб-сайт* происходит от слова website: web - «паутина», «сеть» и site - «место», букв. «место в сети».

*Сайт* - это совокупность web-страниц, объединённых по смыслу и расположенных на одном сервере.

Сделать сайт - значит научиться делать и связывать эти странички.

Страница содержит картинки, текст, аудио и видео файлы, ссылки на другие сайты, а также указание браузеру, как все это отобразить.

*Структура сайта* - основа для выстраивания последовательности и формы отображения имеющихся данных на сайте. При правильной структуре сайта пользователям максимально удобно переходить от одной странички к другой и изучать необходимые для них сведения.

При планировании структуры сайта необходимо продумать следующее: а) структуру каталогов, б)структуру навигации, в)необходимость заставки сайта.

### *Структура каталогов*

С самого начала следует упорядочить содержимое проекта. Все картинки складываются в одну папку, служебные файлы в другую, сами страницы также можно разделить (например, по разделам). Так будет проще ориентироваться. Чтобы определиться какую структуру выбрать для своего сайта, следует знать о разных видах структур сайта и выбирать наиболее подходящую.

### *Виды структуры сайта.*

1. *Линейная* - структура, в которой вся информация представлена в линейном виде. Здесь не может быть разделов, и пользователь не может перескакивать с одной страницы на другую в любой последовательности.

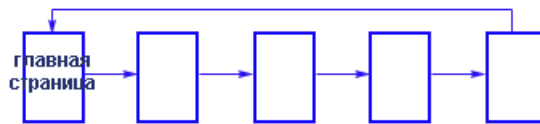


Рис. 1. Схема линейной структуры сайта

Все элементы послушно следуют друг за другом. Пользователь получает возможность лишь плавно переходить по заданной траектории, так, как будто он читает книгу. Данный вид структуры сайта отлично бы подошел для создания презентаций либо онлайн-книг, его очень просто реализовать. Навигация здесь присутствует минимальная, на каждой странице следует обозначить лишь три основные ссылки: на главную страницу, на предшествующую и последующую.

В качестве расширенной формы этого вида структуры сайта представлена линейная структура с ответвлениями.

2. *Линейная с ответвлениями* - структура, которая содержит несколько веток и на главной странице посетитель может выбрать, по какой из них идти дальше, и в зависимости от выбора ему будет представлена та или иная цепочка сведений.

Но самым многофункциональным и развитым можно назвать древовидный вид структуры сайта. Это форма представления данных, выраженная в форме многоуровневой иерархии.

3. *Иерархическая (древовидная)* - структура, в которой вся структура может содержать большое количество разделов и подразделов, внутренних страниц любой вложенности и в произвольной последовательности.

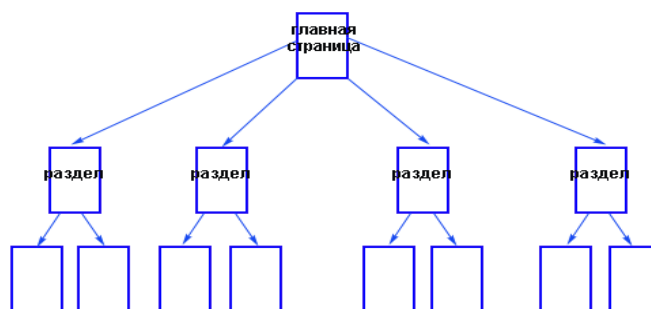


Рис. 2. Схема иерархической структуры сайта

В таком формате можно выполнить совершенно любой веб-ресурс, причем вне зависимости от того, посвящен он одной тематике или сразу нескольким различным областям.

4. *Решетчатая* - структура, которая выполняется практически по тем же стандартам, что и древовидная, но в составе сайта есть особая функция, которая помогает в один клик переместиться с одной части веб-проекта в совершенно противоположную.

5. *Произвольная* - структура, в которой страницы расположены в свободном порядке. Такая структура оправдана только для небольших сайтов.

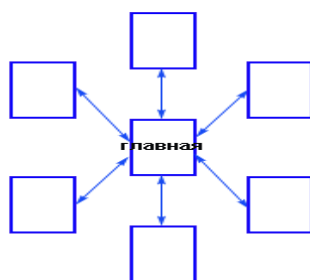


Рис. 3. Схема произвольной структуры сайта

### *Структура навигации*

Это очень важная часть планирования ресурса. Способ доступа к информации на сайте должен быть простым и понятным. Пользователь в любую минуту должен знать, где он находится, куда может пойти, как туда добраться, как вернуться назад. Для решения этих задач используются различные варианты.

#### *Варианты навигации:*

- можно как-то выделить страницу в меню, на которой находится пользователь, а само меню сделать видимым на всех страницах сайта;

- навигационные ключи, т.е. на каждой странице (сверху) указать полный путь до этой страницы по схеме: главная страница - название раздела - название страницы.

Существует *Правило трех кликов*, которое гласит, что до любой страницы сайта пользователь должен добраться не более, чем за три клика (перехода).

Основные системы навигации.

1. *Текстовая система навигации* – оформленные различными способами текстовые ссылки.

Это самый распространенный вид. Текстовая навигация должна присутствовать даже, если используются другие системы навигации.

Различают:

- *Вертикальное меню*, которое располагается справа или слева каждой страницы.

- *Горизонтальное меню*, которое располагается, как правило, сразу под шапкой сайта.

2. *Кнопки* - являются вторыми по частоте использования элементами навигации.

Кнопки могут быть любого размера, формы, цвета и стиля. Используя их, необходимо подключить к каждой обработчики событий (например, на javascript), которые будут описывать действия по нажатию на кнопку или наведения на нее курсора мыши.

3. *Навигационные карты*. Используется изображение и к различным его областям привязываются ссылки.

## **2. Уровни создания сайта**

1. *Конструкторы сайтов* - специальные программы для создания простейшего сайта.

2. *CMS (сокращение от слов content manager system - "система управления контентом")* или *"Движок сайта"* – используется для управления сайтом из панели управления. Имеется возможность добавления новых и редактирования старых страниц сайта.

3. *Язык HTML* - специальный язык разметки гипертекстовых документов для создания web-страниц.

4. *CSS (каскадные таблицы стилей)* - технология описания внешнего вида документа, написанного языком разметки.

5. *JavaScript* – объектно-ориентированный скриптовый язык программиро-

вания. Применяется в основном для того, чтобы сделать страницы интерактивными, а также сделать внешний вид страниц более привлекательным.

6. *PHP* - язык программирования, созданный для разработки динамических web-страниц. Этот язык позволяет не только осуществлять диалог с пользователем, но и запоминать их.

Практически во всех современных браузерах для ОС Windows, Linux достаточно хорошо реализованы web-стандарты совместимости программного обеспечения. Они позволяют разделить структуру документа, его оформление и поведение.

Существует три основных направления применения стандартов:

*HTML* - язык разметки гипертекстовых документов;

*CSS* - каскадные таблицы стилей;

*DOM* - объектная модель документа.

Изучая разработку web-сайтов следует сразу же опираться на «трех китов» - разметку структуры, описание внешнего вида, программирование поведения. За структуру отвечает XHTML (или HTML). Внешний вид описывается на языке CSS. «Оживить» же страницу позволяет JavaScript, обращающийся к ее отдельным элементам в соответствии с DOM.

### 3. Основные понятия и структура HTML-документов



Основной технологией для создания веб-страниц является специальный язык HTML. Данный язык описывает логическую структуру документа, управляет формированием текста и размещением вставных объектов.

*HTML (Hyper Text Mark-up Language)* - специальный язык разметки гипертекстовых документов для создания Web-страниц.

*HTML-документы (документы в формате HTML)* – документы, созданные с помощью языка *HTML*.

*HTML-файл* характеризуется расширением *html*.



*Код HTML-документа* можно создавать при помощи любого текстового редактора или специализированных HTML-редакторов и конвертеров. Для просмотра кода в качестве Web-страницы следует загрузить его в браузер.

Основными конструкциями языка являются теги - инструкции по формированию HTML-файла.

*Теги (tags, метки)* – фрагменты, заключенные между открывающей и закрывающей угловыми скобками.

Большинство тегов *парные* (стартовый тэг и завершающий тэг), то есть на каждую открывающую метку вида <...> есть закрывающая метка </...> с тем же именем, но с добавлением косой черты. Пару, состоящую из открывающего и закрывающего тегов, называют *контейнером*.

HTML-документ состоит из двух частей:

- 1) *заголовок* – содержит информацию, описывающую документ в целом;
- 2) *тело документа* – основное содержание документа.

#### Обязательные теги (метки)

Тег	Описание
<HTML> ... </HTML>	Открытие и закрытие HTML-документа
<HEAD> ... </HEAD>	Начало и конец заголовка документа
<TITLE> ... </TITLE>	Начало и конец названия документа
<BODY> ... </BODY>	Начало и конец тела HTML-документа, между которыми располагаются текст и теги документа

Все документы HTML имеют одну и ту же структуру, определяемую фиксированным набором тегов структуры.

*Общая структура HTML-документа:*

```
<HTML>
  <HEAD>
    <TITLE>
      Заголовок документа
    </TITLE>
  </HEAD>
  <BODY>
    Текст документа
  </BODY>
</HTML>
```

Положение структурных тегов в документе нетрудно определить, даже если они опущены. Поэтому стандарт языка требует только наличия тега <TITLE>. Тем не менее, при создании документа HTML опускать структурные теги не рекомендуется.

Многие метки, помимо имени могут содержать атрибуты, то есть элементы, дающие дополнительную информацию о том, как браузер должен обработать текущую метку.

*Атрибуты тега* – относящиеся к тегу уточнения.

Спецификация атрибута включает:

- \* имя атрибута;
- \* знак равенства =;
- \* значение атрибута, которое задается строкой символов.

В HTML-документ можно включать комментарии, позволяющие скрыть часть текста от браузера. Все, что заключено между последовательностями символов <!-- и -->, при просмотре страницы остается невидимым. Комментарии не могут быть вложенными друг в друга.

#### **4. Элементы языка HTML**

*Элемент языка HTML* – часть документа между открывающим и закрывающим тегом.

Весь документ HTML можно рассматривать как элемент HTML. Для непарных тегов элемент совпадает с тегом, который его определяет.

Большинство элементов языка HTML описывает части содержания документа и помещается внутри тега BODY. Данные элементы делятся на:

- 1) *текстовые элементы* – описывают свойства отдельных фраз и еще более мелких частей текста;
- 2) *блочные элементы* – относятся к частям текста уровня абзаца (параграфы, списки, формы, таблицы, картинки и т.п.).

В большинстве документов основными функциональными элементами являются заголовки и абзацы.

Теги тела документа предназначены для управления отображением информации в программе интерфейса пользователя. Они описывают гипертекстовую структуру базы данных при помощи встроенных в текст контекстных гипертекстовых ссылок.

Тело документа состоит из иерархических контейнеров и заставок; заголовков; блоков; адресов; текста, разбитого на области действия стилей (подчеркивание, выделение, курсив); математических описаний, графики и гипертекстовых ссылок.

*Метки заголовков* – описывают относительные размеры заголовков различных уровней.

*Формат:* <H1> ... </H1>, где I – цифра от 1 до 6.

Заголовок первого уровня – самый крупный, шестого – самый мелкий.

*Разбиение текста на строки и абзацы*

*Метки абзаца* – описывают начало и конец абзаца.

*Формат:* <P> ... </P>.

Закрывающий тег необязательный – подразумевается, что он стоит перед тегом, который задает начало очередного абзаца документа.

Язык HTML не содержит средств для создания абзацного отступа (красной строки), поэтому при отображении на экране компьютера абзацы разделяются пустой строкой или используются специальные метки.

Любая последовательность, состоящая только из пробелов и символов конца строки, при отображении документа рассматривается как одиночный пробел. При этом не осуществляется перехода на следующую строку – для этой цели используется текстовый элемент, задаваемый специальным тегом.

*Метка конца строки* <BR> - используется, когда нужно не организовать новый абзац, а просто перейти на новую строку.

В качестве ограничителя абзацев также используется горизонтальная линия.

*Метка* <HR> - описывает горизонтальную линию.

Данная метка может включать атрибуты:

*SIZE* – определяет толщину линии в пикселях;

*WIDTH* – определяет размер линии в процентах от ширины экрана;

*COLOR* – определяет цвет линии (работает для IE).

*Пример:* <HR SIZE=4 WIDTH=50% COLOR=RED>.

Метки <H1> и <P> могут содержать дополнительный атрибут *ALIGN*, принимающий одно из значений: *LEFT*, *CENTER*, *RIGHT*, *JUSTIFY* (выравнивание по левому краю, по центру, по правому краю, по ширине окна браузера соответственно).

*Пример:* <P ALIGN= center > \*абзац с выравниванием по центру.

*Правила вложения элементов*

1. Элементы не должны пересекаться.
2. Блочные элементы могут содержать вложенные блочные и текстовые элементы.
3. Текстовые элементы могут содержать вложенные текстовые элементы.
4. Текстовые элементы не могут содержать вложенные блочные элементы.

## 5. Форматирование текста

Браузеры позволяют объединять на одной странице несколько видов шрифтов, вне зависимости от того, какой из них задан по умолчанию в браузере пользователя.

*Управление параметрами шрифта*

*Формат:* <FONT> ... </FONT>.

Тег позволяет изменять цвет, размер и тип шрифта текста, находящегося между начальным и конечным тегами. Вне тегов <FONT> и </FONT> используется шрифт, указанный в элементе BASEFONT.

Тег <FONT> должен содержать один из атрибутов:

- 1) *COLOR* – задает цвет текста;
- 2) *FACE* – определяет используемый шрифт;
- 3) *SIZE* – определяет размер шрифта. Возможные значения:  
– целое число от 1 до 7;

– относительный размер с указанием знака.

*Пример:*

```
<FONT SIZE=3 FACE="Courier New" COLOR="Magenta">Моноширинный  
фиолетовый текст 3 размера  
</FONT>
```

*Физический стиль шрифтового выделения* – прямое указание на модификацию текущего шрифта.

#### Теги форматирования текста

Тег	Описание
<B> ... </B>	Полужирный шрифт
<I> ... </I>	Курсив
<U> ... </U>	Подчеркнутый шрифт
<S> ... </S>	Вычеркнутый шрифт
<TT> ... </TT>	Текст с фиксированной шириной символа (имитация пишущей машинки)
<BIG> ... </BIG>	Крупный шрифт
<SMALL> ... </SMALL>	Мелкий шрифт
<SUP> ... </SUP>	Надстрочные символы
<SUB> ... </SUB>	Подстрочные символы

*Цветовая гамма* документа определяется атрибутами, размещенными внутри метки *BODY*:

- атрибут *BGCOLOR* определяет цвет фона документа;
- атрибут *TEXT* определяет цвет текстового наполнения документа.

Цвет можно задавать двумя способами:

- 1) *название цвета* – red, green, ... (всего 16 цветов);
- 2) *код из системы RGB* - в виде последовательности из трех пар шестнадцатеричных символов.

*Пример:* *BGCOLOR*=#808080 – фон серого цвета

*Элементы анимации текста*

Бегущая строка – теги <MARQUEE> ... </MARQUEE>.

Регулируется время «пробегания» строки, цвет фона, ширина бегущей строки, цвет шрифта, вид его написания, наклон, размер.

## Лабораторная работа 1. HTML. Общая структура web-документа.

### Упражнение 1. Создание простейшей web-страницы

#### Порядок работы:

1. Запустить текстовый редактор Блокнот.

2. Ввести текст документа:

```
<HTML>
    <HEAD>
        <TITLE>Моя страничка</TITLE>
    </HEAD>
    <BODY>
        Тело <!-- Это комментарий --> документа
    </BODY>
</HTML>
```

3. Сохранить документ Пример1Фамилия.html, где текст «Фамилия» заменить собственной фамилией.

4. Запустить браузер. Открыть файл Пример1Фамилия.html.

### Упражнение 2. Изучение приемов форматирования абзацев

#### Порядок работы:

1. В коде документа изменить цвет текста на красный и цвет фона – на желтый, добавив в тег <BODY> запись TEXT=red BGCOLOR= yellow.

2. Удалить весь текст, находящийся между тегам <BODY> и </BODY>.

3. Ввести текст документа:

```
<H1 ALIGN=CENTER>Пример 2</H1>
<H2 ALIGN=RIGHT>Иосиф Бродский </H2>
<P >
Откуда к нам пришла зима, <BR>
не знаешь ты, никто не знает.
<P ALIGN=CENTER>
Умолкло все. Она сама <BR>
```

холодных губ не разжимает.

```
<HR SIZE=2 WIDTH=50% COLOR=RED>
```

4. Сохранить документ под именем Пример2Фамилия.html
5. Запустить программу браузер и открыть файл.

### Упражнение 3. Шрифтовые выделения

#### Порядок работы:

1. Удалить весь текст, находящийся между тегами <BODY> и </BODY>.
2. Ввести текст документа:

```
<B>Полужирный шрифт</B> <BR>
```

```
<I>Курсив</I><BR>
```

```
<U>Подчеркнутый шрифт</U><BR>
```

```
<TT>Шрифт пишущей машинки</TT><BR>
```

```
<BIG>Крупный шрифт</BIG><BR>
```

```
<SMALL>Мелкий шрифт</SMALL><BR>
```

```
x<sup>342</sup> так пишутся показатели степени<BR>
```

```
a<sub>i</sub>, а так – индексы<BR>
```

```
<P ALIGN=JUSTIFY><B>Выравнивание по ширине окна браузера</B>
```

Разнообразие внешнего вида HTML-страниц сравнимо только с разнообразием обложек журналов, газет и рекламы: здесь разнообразные шрифты, иллюстрации, анимационные фигуры и другие элементы.

```
</P>
```

3. Сохранить документ под именем Пример3Фамилия.html
4. Запустить браузер и открыть файл.

#### Задание

1. Создайте документ с именем index.html, в заголовке которого поместите собственные имя и фамилию и установите цвет фона по желанию.
2. Создайте название «Домашнее задание №1», используя заголовок второго

уровня.

3. Разместите в документе три пословицы следующим образом: первая выравнивается по левому краю, пишется полужирным шрифтом; вторая - по центру курсивным шрифтом; третья - по правому краю подчеркнутым шрифтом.

4. Пословицы оформите шрифтами разного цвета.

5. Разделите пословицы горизонтальными линейками с различными значениями атрибутов.

## 6. Списки

Абзацы языка HTML можно организовывать в списки, выводить их на экран в отформатированном виде. HTML поддерживает несколько видов списков.

*1. Упорядоченный (нумерованный) список*

Формат: `<OL> ... </OL>`.

*2. Неупорядоченный (маркированный) список*

Формат: `<UL> ... </UL>`.

Эти списки оформляются одинаково. Каждый новый элемент списка следует начинать с метки `<LI>`.

Вид нумерации и вид маркера задаются атрибутом `TYPE`.

*Виды нумерации:*

1 – арабские цифры;

a – строчные буквы;

A – заглавные буквы;

i – малые римские цифры;

I – большие римские цифры.

*Пример:* `<OL TYPE=i>`

Атрибут `START` определяет начальное значение нумерованного списка.

*Три вида разделителей элементов маркированного списка друг от друга:*

1) `disk` - элементы отделяются заштрихованным диском;

2) `circle` - элементы отделяются незаштрихованным кругом;

3) `square` - элементы отделяются незаштрихованным квадратом.



### 3. Список определений (словарь специальных терминов)

Формат: <DL> ... </DL>.

Данный список содержит элементы двух видов:

- 1) определяемый термин - парный тег <DT>;
- 2) определение определения – парный тег <DD>.

Метка <DL COMPACT> - компактный список определений.

Разрешается вложение списков друг в друга. Число уровней вложенности не ограничено. Вложенные списки могут быть комбинированными.

#### *Комбинирование списков*

Отступ для каждого вида списка устанавливает браузер, однако, если это необходимо, для его увеличения можно добавить "пустой" список.

Можно создать собственные маркеры для использования в нумерованных списках. Вместо тэга <LI> используется тэг <IMG>, который указывает на графический файл используемого маркера.

## **Лабораторная работа 2. HTML. Использование списков**

### **Упражнение 1. Создание маркированного списка**

#### **Порядок работы:**

1. Запустить текстовый редактор Блокнот и открыть файл Пример1Фамилия.
2. Ввести текст тела документа:

```
<FONT COLOR=red SIZE=7>
<UL>
    <LI> Программирование</LI>
    <LI> Алгоритмизация
    <LI> Проектирование
</UL>
</FONT>
```

3. Сохранить документ Список1Фамилия.html
4. Запустить браузер. Открыть созданный файл.
5. Задать специальную нумерацию, поместив в тег <UL> запись

TYPE=SQUARE, во второй тэг <LI> внести атрибут TYPE= DISK, в третий - TYPE= CIRCLE. Просмотреть изменения в маркировке элементов списка.

## Упражнение 2. Создание списка с нестандартными маркерами

### Порядок работы:

1. Поместить файл star.ico из папки Server/Учебные материалы/Интернет-программирование в папку с сохраненным документом Список1Фамилия.html.
2. Для файла Список1Фамилия в тело документа внести изменения:

```
<UL>
    <IMG SRC=star.ico ALIGN=top>
    <NBR>Программирование<BR>
    <IMG SRC=star.ico ALIGN=top>
    <NBR>Алгоритмизация<BR>
    <IMG SRC=star.ico ALIGN=top>
    <NBR>Проектирование<BR>
</UL>
```

3. Сохранить документ и обновить файл. Просмотреть изменения в маркировке элементов списка.

## Упражнение 3. Создание нумерованного списка

### Порядок работы:

1. Для файла Список1Фамилия в тело документа добавить текст:

```
<OL>
    <LI>Долгосрочные кредиты</LI>
    <LI>Краткосрочные кредиты</LI>
    <LI>Кредиторская задолженность</LI>
</OL>
```

2. Сохранить документ и в браузере открыть созданный файл.
3. Задать специальную нумерацию, поместив в тег <OL> запись START= 15

TYPE= I.

4. Сохранить документ и обновить файл.

#### **Упражнение 4. Создание списка определений**

##### **Порядок работы:**

1. Запустить текстовый редактор Блокнот и открыть файл Пример1Фамилия.

2. Ввести текст тела документа:

```
<DL>
```

```
<DT>Отдел маркетинга
```

```
<DD>Данный отдел занимается продвижением товаров и  
услуг
```

```
<DT>Финансовый отдел
```

```
<DD>Данный отдел занимается финансовыми операциями
```

```
</DL>
```

3. Сохранить документ Список2Фамилия.html и в браузере открыть созданный файл.

#### **Упражнение 5. Создание вложенного списка**

##### **Порядок работы:**

1. Запустить текстовый редактор Блокнот и открыть файл Пример1Фамилия.

2. Ввести текст тела документа:

```
<OL>
```

```
<LI>Актив</LI>
```

```
<UL>
```

```
<LI>Внеоборотные активы</LI>
```

```
<LI>Оборотные активы</LI>
```

```
<LI>Убытки</LI>
```

```
</UL>
```

```
<LI>Пассив</LI>
```

```
<UL>
```

```
<LI>Капитал и резервы</LI>
```

```
<LI>Долгосрочные пассивы</LI>  
<LI>Краткосрочные пассивы</LI>  
</UL>
```

```
</OL>
```

3. Сохранить документ Список3Фамилия.html и в браузере открыть созданный файл.

### Задание

1. Создайте файл Фамилия.html и включите в него списки (маркированные, нумерованные, определений) до четырех уровней вложенности.
2. Используйте различные маркеры для маркированных и стартовые значения для нумерованных списков.

## 7. Таблицы

Таблицы необходимы не только для представления определенного вида информации в табличном виде, они также являются удобным инструментом структурирования всего документа в целом или его части. Многие web-дизайнеры используют таблицы для точного размещения элементов документа.

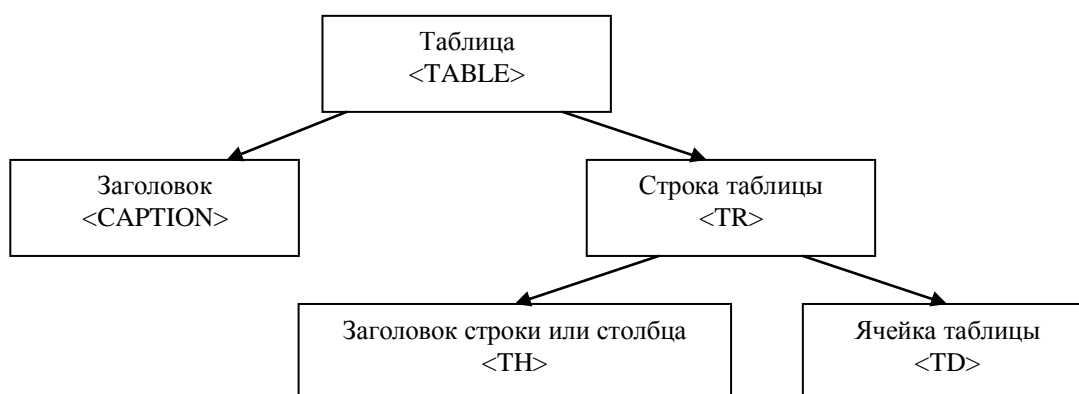


Рис. 4. Основные элементы таблицы (перечисленные теги являются парными).

Тэг `<TABLE>` - контейнер, в котором размещается содержимое таблицы.

Таблица строится по строкам:

- для обозначения строки используется контейнер `TR`,
- для обозначения заголовков столбцов (строк) – контейнер `TH`,

- для данных в ячейках – контейнер TD.

Таблицы всегда должны быть прямоугольными (другие формы не допускаются). Заголовки выделяются полужирным шрифтом и центрируются в своих ячейках. Данные имеют обычный шрифт и выравниваются по левой стороне ячейки.

#### Атрибуты элементов, используемых при создании таблицы

Атрибут	Элемент	Назначение
ALIGN	Таблица, строка, ячейка	Выравнивание таблицы или данных по горизонтали LEFT, CENTER, RIGHT
	Заголовок	Размещение заголовка над (TOP) или под таблицей (BOTTOM)
VALIGN	Строка, ячейка	Выравнивание по вертикали (TOP, MIDDLE, BOTTOM)
WIDTH	Таблица, ячейка	Ширина в пикселах (или в процентах от ширины страницы для таблицы)
HEIGHT	Ячейка	Высота
COLSPAN	Ячейка	Протяженность в несколько столбцов
ROWSPAN	Ячейка	Протяженность в несколько строк
CELLSPACING	Таблица	Зазор между ячейками
CELLPADDING	Таблица	Зазор между содержимым ячейки и ее границей в пикселах
BORDER	Таблица	Отображение границ ячеек и внешней рамки таблицы в пикселах
BGCOLOR	Таблица, ячейка	Цвет фона
BODERCOLOR	Таблица	Цвет рамки ячейки

*Смежные ячейки* таблицы могут объединяться с целью размещения большего количества данных

Для соединения двух смежных ячеек в одном столбце используется атрибут ROWSPAN тэга <TH> или <TD>, например: <TD ROWSPAN=2>.

Для объединения двух смежных ячеек в одной строке нужно использовать атрибут COLSPAN тех же тэгов, например: <TD COLSPAN=2>.

Допускается изменение не только цвета фона и шрифтов всей таблицы и ее отдельных ячеек, но и вставка вместо текста графических изображений, гиперссылок и других элементов.

Существуют альтернативные способы представления табличных данных: с использованием списка; с использованием изображения или как предварительно отформатированный текст.

## Лабораторная работа 3. HTML. Работа с таблицами

### Упражнение 1. Создание простейшей таблицы

#### Порядок работы:

1. Запустить текстовый редактор Блокнот и открыть файл Пример1Фамилия.
2. Ввести текст заголовка: Простейшая таблица
3. Ввести текст тела документа:

```
<TABLE>
```

```
<CAPTION>Сведения о фильмах</CAPTION>
```

```
<TR><TH>Название фильма</TH><TH>Страна</TH><TH>Год про-  
изводства</TH></TR>
```

```
<TR><TD>Бриллиантовая рука</TD><TD>СССР</TD><TD>1971  
</TD></TR>
```

```
<TR><TD>Титаник</TD><TD>США</TD><TD>1998</TD></TR>
```

```
<TR><TD>Вор</TD><TD>Россия</TD><TD>1998</TD></TR>
```

```
</TABLE>
```

4. Сохранить документ Таблица1Фамилия.html
5. Запустить браузер. Открыть созданный файл.
6. Добавить оформление таблицы и выравнивание ее по центру в теге TABLE:

```
<TABLE BORDER=3 ALIGN=center>
```

7. Просмотреть изменения таблицы в браузере.

### Упражнение 2. Объединение ячеек таблицы

#### Порядок работы:

1. Запустить текстовый редактор Блокнот и открыть файл Пример1Фамилия.
2. Ввести текст заголовка: Пример таблицы с объединенными ячейками
3. Ввести текст тела документа:

```
<TABLE BORDER=1 ALIGN=center CELLSPACING=10 CELLSPAD-  
DING=5 BGCOLOR="green">
```

```
<TR><TH>Названия</TH><TH>Параметры</TH><TH COLSPAN=3>В  
см/футах</TH></TR>
```

```

<TR>
  <TH ROWSPAN=2>Возраст</TH><TH>Рост</TH>
  <TD>175/5.8</TD>
  <TD>180/6</TD>
  <TD>185/6.5</TD>
</TR>
<TR>
  <TH>Вес</TH><TD>68кг</TD><TD>79 кг</TD><TD>90 кг</TD>
</TR>
<TR>
  <TH ROWSPAN=2>Интересы</TH>
  <TD>устойчивые</TD>
  <TD>вялые</TD>
  <TD>яркие</TD>
  <TD>средние</TD>
</TR>
</TABLE>

```

4. Сохранить документ Таблица2Фамилия.html

5. Запустить браузер. Открыть созданный файл.

### Упражнение 3. Использование цвета в таблице

#### Порядок работы:

1. Запустить текстовый редактор Блокнот и открыть файл Пример1Фамилия.

2. Ввести текст заголовка: Пример таблицы с цветовым оформлением

3. Ввести текст тела документа:

```

<TABLE BORDER=8 WIDTH=90% BGCOLOR="white"
CELLPADDING=10 ALIGN=center>

```

```

<TR>

```

```

  <TH WIDTH=20%>Простой заголовок</TH>

```

```

  <TH WIDTH=80% COLSPAN=2>Заголовок на 2 столбца</TH>

```

```

</TR>
<TR BGCOLOR="silver">
    <TH WIDTH=20%>Заголовок на 1 строку</TH>
    <TD WIDTH=40%>Ячейка 1</TD>
    <TD WIDTH=40% BGCOLOR="teal">Ячейка 2</TD>
</TR>
<TR>
    <TH WIDTH=20% ROWSPAN=3>Заголовок на 3 строки</TH>
    <TD ALIGN=center>Ячейка 3</TD>
    <TD ALIGN=right>Ячейка 4</TD>
</TR>
<TR>
    <TD><FONT COLOR="red">Ячейка 5</FONT></TD>
    <TD ROWSPAN=2 ALIGN=center VALIGN=bottom>Ячейка 6</TD>
</TR>
<TR>
    <TD>Ячейка 7</TD>
</TR>
</TABLE>

```

4. Сохранить документ [Таблица3Фамилия.html](#)

5. Просмотреть созданную таблицу.

6. Для создания эффекта выпуклой рамки таблицы изменить значение атрибута **BORDER** до 20. Просмотреть результат.

#### **Упражнение 4. Использование изображений в таблицах.**

##### **Порядок работы:**

1. Разместить изображение вместо фона таблицы упражнения 3 - в тег `<TABLE>` добавить атрибут `background="auto.jpg"`. Просмотреть результат и удалить атрибут.
2. Разместить изображение вместо текста отдельной ячейки таблицы – в тег `<TD>` вместо текста Ячейка 7 добавить тег ``.



## Задание

Создайте файл tableФамилия.html, в котором разместите собственную таблицу, включающую не менее 3 столбцов и 4 строк.

## 8. Гипертекстовые ссылки

*Гипертекст* – форма организации семантической информации, разделенной на фрагменты, для каждого из которых указаны переходы к родственным фрагментам с указанием типа взаимосвязи. Гипертекстовая ссылка обеспечивает переход к соответствующему смысловому фрагменту. Она является фрагментом текста документа и потому задается текстовым элементом языка HTML.

*Создание гиперссылки* - метки: `<A> ... </A>`.

Ссылка состоит из двух частей:

- первая часть (указатель)– это то, что пользователь видит на Web-странице;
- вторая часть (URL-адрес) - адресная часть ссылки, дающая инструкцию браузеру.

*URL-адрес* - это стандартизированный способ записи адреса ресурса в сети Интернет.

Указатели бывают двух типов – текстовые и графические.

*Текстовые указатели* представляют собой слово или слова, подчеркнутые прямой линией выделенные цветом.

*Графические указатели* не подчеркиваются и не выделяются цветом, но вокруг них можно сделать рамку. Они часто имеют вид точки, звездочки или другого маркера.

Обязательным является атрибут задания файла ссылки *HREF*, т.е.

`<A HREF="URL или относительный адрес файла">`.

Гиперссылки бывают двух видов.

1. *Внешняя* – ссылка на отдельный Web-документ. Используется *абсолют-*

*ный адрес URL*, который указывает путь к конкретному документу (сайту).

*Абсолютный адрес* - URL-адрес, полностью описывающий расположение ресурса в глобальной сети (определяет компьютер, каталог и файл).

## 2. Внутренняя.

А) Ссылка на отдельные страницы - в качестве значения атрибута используется *относительный адрес URL*, который указывает путь к конкретной странице документа.

*Относительный адрес* - URL-адрес файла, расположенный на том же компьютере, что и документ, в котором находится указатель этой ссылки.

Это означает, что если браузер загрузил страницу, находящуюся по адресу `http://www.mysite.com/page`, то относительный адрес `/picture` подразумевает адрес `http://www.mysite.com/page/picture`, т. е. подкаталог, расположенный на той же машине.

Б) Ссылка на определенное место внутри страницы - *система якорей*. Месту присваивают имя – для этого в метку гиперссылки включают атрибут *NAME*. Значение этого атрибута – *имя якоря* - произвольная последовательность латинских букв или цифр без пробелов. Для ссылки на якорь его имя указывается после символа #, а сама метка заключается в кавычки.

Пример: `<A NAME=NACHALO> Глава 1 </A>`

`<A HREF="#NACHALO">к Главе 1 </A>`

Тэг `<BASE>` располагается в начале документа и содержит URL-адрес, относительно которого в документе построена вся адресация.

Пример: `<base href="http://www.server.com">`

Задание цветов ссылок:

- атрибут *LINK* определяет цвет гиперссылки;
- атрибут *VLINK* определяет цвет ссылки на документ, который уже был просмотрен ранее;
- атрибут *ALINK* определяет цвет ссылки, когда на нее указывает курсор.

Пример: `<BODY LINK=RED, VLINK=PURPLE, ALINK=GREEN>`

## Лабораторная работа 4. HTML. Использование гипертекстовых ссылок

### Упражнение 1. Использование внутренних гиперссылок

#### Порядок работы:

1. Запустить текстовый редактор Блокнот и открыть файл Пример1Фамилия.
2. Ввести текст заголовка: Внутренние ссылки
3. Ввести текст тела документа:

```
<H2>Давайте поуплетно разберем стихотворение</H2>
```

```
<a href=#1>Первое четверостишие</a><br>
```

```
<a href=#2>Второе четверостишие</a><br>
```

```
<a href=#3>Третье четверостишие</a><br>
```

```
<a href=#4>Четвертое четверостишие</a><br>
```

```
<H3><a name=1>Первое четверостишие</a></H3>
```

```
<pre>
```

```
Покорно мне воображенье  
В изображенье серых глаз.  
В моем тверском уединенье  
Я горько вспоминаю вас.
```

```
</pre>
```

```
<H3><a name=2>Второе четверостишие</a></H3>
```

```
<pre>
```

```
Прекрасных рук счастливый пленник,  
На левом берегу реки,  
Мой знаменитый современник,  
Случилось, как хотели вы.
```

```
</pre>
```

```
<H3><a name=3>Третье четверостишие</a></H3>
```

```
<pre>
```

```
Вы, приказавший мне: довольно,  
Поди, убей свою любовь!  
И вот я таю, я безвольна,
```

Но все сильнее скучает кровь.

```
</pre>
```

```
<H3><a name=4>Четвертое четверостишие</a></H3>
```

```
<pre>
```

И если я умру, то кто же

Мои стихи напишет вам!

Кто стать звенящими поможет

Еще не сказанным словам?

```
</pre>
```

4. Сохранить документ [Ссылка1Фамилия.html](#)

5. Запустить браузер. Открыть созданный файл и уменьшить размер окна по горизонтали, оставив видимыми только ссылки. Проверить работу гиперссылок.

## **Упражнение 2. Использование внешних гиперссылок. Создание страницы ссылок на серверы музеев России и Мира**

1. Запустить текстовый редактор [Блокнот](#) и открыть файл [Пример1Фамилия](#).

2. Ввести текст заголовка: [Внешние ссылки](#)

3. В тело документа ввести текст со ссылкой на сайт музея Лувр:

```
<H2>Музеи России и мира</H2>
```

```
<A HREF=http://www.louvre.fr>Парижский музей Лувр</A>
```

4. Сохранить страницу [Museums.html](#). Просмотреть ее в браузере и проверить работу гиперссылки.

5. В сети Интернет найти 3 ссылки на сайты музеев России и Мира с помощью поисковых серверов. Сохранить в папке [images](#) логотипы найденных сайтов.

*Примечание:* при сохранении логотипов изменить при необходимости их имена на более запоминающиеся, например [logo.jpg](#) на [louvre.jpg](#) при сохранении логотипа Лувра.

6. Добавить ссылки найденных сайтов на созданную страницу.

7. Разместить около каждой ссылки соответствующий логотип и создать ссылку на сайт. Например, ввести внутри тега <A>:

```
<IMG SRC="images/louvre.jpg">
```

8. Сохранить файл и просмотреть его в браузере.

### **Задание**

Создайте файл СсылкиФамилия.html, в котором следует поместить собственный текст с внутренними ссылками.

## **9. Работа с графическими изображениями**

Графические иллюстрации являются неотъемлемой частью Web-страниц. Изображения могут сделать текст документа более содержательным, помогают лучше передать суть и содержание документа.

Два основных формата для работы с графическими элементами:

1) *файл формата GIF* (расширение .gif) – изображение содержит 256 цветов, заданных индексной палитрой;

2) *файлы формата JPEG* (расширение .jpeg или .jpg) - предназначены для хранения фотографических изображений, использующих 24-разрядный цвет.

При выборе формата изображения в первую очередь обращают внимание на объем получаемого файла и во вторую – на качество изображения. При загрузке большая часть времени отводится на загрузку иллюстраций, поэтому рекомендуется создать файлы в двух вариантах форматов и выбрать файл с наименьшим объемом.

*Встраивание изображений*

*Формат метки:* <IMG SRC="имя встраиваемого файла">.

*Пример:* <IMG SRC="book.gif">.

Хранение файлов графики желательно в одном определенном каталоге Web-сайта (не корневом). В этом случае можно использовать относительную адресацию в комбинации с тэгом <BASE>, а не указывать полный URL-адрес.

Атрибут SRC определяет не только какое изображение, но и где хранится это изображение.

*Основные варианты расположения:*

•SRC="book.gif" означает, что браузер будет искать изображение с именем

book.gif в той же самой папке (или каталоге), где непосредственно расположен html документ.

- SRC ="images/ book.gif" означает, что изображение располагается в папке нижнего уровня, по сравнению с папкой HTML- документа, который запросил его. Это можно продолжить по уровням вниз настолько это необходимо.

- SRC ="../book.gif" означает, что изображение находится в папке верхнего уровня, по сравнению с папкой HTML-документа, который запросил его.

При отображении рисунка браузер по умолчанию использует его реальные размеры. Если рисунок необходимо отмасштабировать, а также определить взаимодействие с текстом, то применяют специальные атрибуты.

#### Атрибуты, используемые при размещении изображения

Атрибут	Назначение
WIDTH	Ширина рисунка (в пикселах или % от первоначальных параметров)
HEIGHT	Высота рисунка (в пикселах или % от первоначальных параметров)
ALIGN	По горизонтали: LEFT – выравнивает изображение по левому полю, текст «обтекает» изображение справа, RIGHT - выравнивает изображение по правому полю, текст «обтекает» изображение слева, CENTER – центрирует изображение По вертикали: TOP – выравнивает верхний край изображения по верхнему краю текущей строки, MIDDLE – выравнивает центр изображения по базовой линии текущей строки, БОТТОМ - выравнивает нижний край изображения по базовой линии текущей строки
HSPACE	Расстояние между текстом и изображением по горизонтали (в пикселах)
VSPACE	Расстояние между текстом и изображением по вертикали (в пикселах)

Возможны случаи, когда отсутствуют средства просмотра изображений или в браузере отключена автоматическая загрузка изображений. Считается правилом хорошего тона включать в место показа картинки альтернативный текст, с описанием названия и объема загружаемого графического файла.

*Атрибут ALT= "текст"* – загружает альтернативный текст, который отображается вместо изображения, если оно по каким-либо причинам не может быть выведено.

*Атрибут BORDER* тэга <img> - помещение изображения в рамку, ширина которой задается в пикселах.

*Пример:* <IMG SRC ="comp.jpg" BORDER =5>

Встраивание изображения в ячейку таблицы.

*Пример:* <TD>< IMG SRC="flower.jpg"></TD>

Существует возможность использования изображения в качестве фона веб-документа. При отображении страницы, содержащей фоновый рисунок, рабочая область экрана заполняется этим рисунком последовательно (как паркетом), начиная от верхнего левого угла окна.

*Атрибут BACKGROUND* в теге BODY - использование изображения в качестве фонового рисунка.

*Создание графического указателя*

Изображение документа может использоваться как *изображение-ссылка* (на экране изображение отличается рамкой и при наведении на него указателя, он изменяется на указательный палец).

*Формат:* <IMG SRC="URL или относительный адрес файла">

*Пример.* <A HREF="http:\\yandex.ru">  
<IMG SRC="D:\КИТ\Yandex.gif">  
</A>

Щелкая по уменьшенному изображению, загружается указанный ресурс из сети Интернет.

## **Лабораторная работа 5. HTML. Использование изображений**

### **Упражнение 1. Создание изображения и использование его на Web-странице**

#### **Порядок работы:**

1. Открыть программу Paint.
2. Задать размеры нового рисунка, например 200x200 точек (меню Рисунок пункт Атрибуты).
3. Залить рисунок произвольным фоновым цветом.

4. Инструментом Кисть нанести произвольный рисунок на заданный фон.
5. Сохранить рисунок под именем pic.gif (в формате GIF).
6. Открыть документ Пример1Фамилия.htm в программе Блокнот.
7. После тега <BODY> ввести текст:

Поскольку время копирования изображения определяет время<BR> копирования всей страницы, желательно представлять изображение<BR> в том формате, который обеспечивает минимальный размер файла.
8. Установить текстовый курсор в начало текста.
9. Ввести тег <IMG SRC="pic.gif">.
10. Сохранить документ под именем Картинка1Фамилия.htm.
11. Запустить обозреватель. Просмотреть созданный файл.

## **Упражнение 2. Способы размещения изображения на Web-странице**

### **Порядок работы:**

1. Открыть документ Пример1Фамилия.htm в программе Блокнот и ввести следующий текст:

```
<BODY>
```

```
<P>
```

```
<IMG SRC="book.jpg">
```

Этот текст следует сразу же за изображением.

```
<P>
```

Этот текст прерывается <IMG SRC="book.jpg"> изображением.

```
<P>
```

В этом случае изображение появляется после текста.

```
<IMG SRC="book.jpg">
```

```
</BODY>
```

2. Сохранить документ под именем Картинка2Фамилия.htm.
3. Запустить обозреватель и просмотреть созданный файл.



### Упражнение 3. Выравнивание текста по краю изображения

#### Порядок работы:

1. Открыть документ Пример1Фамилия.htm в программе Блокнот и ввести следующий текст:

```
<BODY>
```

```
<P>
```

```
<IMG SRC=" book.jpg" ALIGN=TOP>
```

Этот текст выровнен с верхней частью изображения.

```
</P>
```

```
<P>
```

```
<IMG SRC=" book.jpg" ALIGN=MIDDLE>
```

Этот текст выровнен с серединой изображения.

```
</P>
```

```
<P>
```

```
<IMG SRC=" book.jpg" ALIGN=BOTTOM>
```

Этот текст выровнен с нижней частью изображения.

```
</P>
```

```
</BODY>
```

2. Сохранить документ под именем Картинка3Фамилия.htm.

3. Запустить обозреватель и просмотреть созданный файл.

### Упражнение 4. Позиционирование изображения на Web-странице

#### Порядок работы:

1. Открыть документ Пример1Фамилия.htm в программе Блокнот и ввести следующий текст:

```
<P>
```

```
<IMG SRC=" book.jpg" ALIGN=LEFT>
```

```
<FONT SIZE=7> Книга - один из видов печатной продукции: непериодическое издание, состоящее из сброшюрованных
```

бумажных листов (страниц) или тетрадей, на которых нанесена типографским или рукописным способом текстовая и графическая (иллюстрации) информация, имеющая объём более сорока восьми страниц и, обычно, в твердом переплете.

```
<IMG SRC="book.jpg" ALIGN=RIGHT>
```

```
<FONT SIZE=7>
```

Электронная книга - версия книги в электронном (цифровом) виде. Данный термин применяется как для произведений, представленных в цифровой форме, так и в отношении устройств, используемых для их прочтения. Электронными текстами называются и учебные электронные пособия, в состав которых, помимо текста, включаются растровые изображения, примеры данных, упражнения, специально написанные комментарии и ответы на возможные вопросы.

```
</P>
```

2. Сохранить документ под именем Картинка4Фамилия.htm.
3. Запустить обозреватель и просмотреть созданный файл.
4. Изменить размеры второго изображения и уменьшить размер шрифта второго абзаца, чтобы информация размещалась на страницу:

```
<IMG SRC="book.jpg" WIDTH=140 HEIGHT=100>
```

5. Просмотреть изменения.
6. Поместить первое изображение в рамку, добавив в тег IMG атрибут BORDER=5.
7. Просмотреть изменения, затем изменить толщину границы до 10.
8. Ввести альтернативное описание первого изображения, добавив в тег IMG атрибут: ALT="Книга". При просмотре в браузере (например, Internet Explorer) при наведении указателя мыши на изображение отображается текст этого атрибута.

## Задание.

Создать документ ИзображенияФамилия.html. Вставить 4 рисунка, используя все атрибуты размещения и позиционирования, но присваивая им различные значения. Добавить текст, разместив его различными способами.

## 10. Организация фреймов

Язык HTML позволяет в рамках одной web-страницы отобразить несколько документов. Для этого страница должна быть разбита на несколько областей – фреймов. В результате разбивки пользователь, просматривающий страницу, может изучать только одну ее часть, независимо от остального содержимого. Фактически браузер, распознающий фреймы, загружает разные страницы в разные секции (фреймы) экрана.

*Фрейм (frame, кадр)* – раздел (область) окна браузера, в котором показывается свой HTML-документ.

Каждый фрейм может иметь свои полосы прокрутки, ссылки, графические изображения и т. д. Фреймы могут функционировать независимо или влиять друг на друга, используя ссылки, указывающие на другие фреймы.

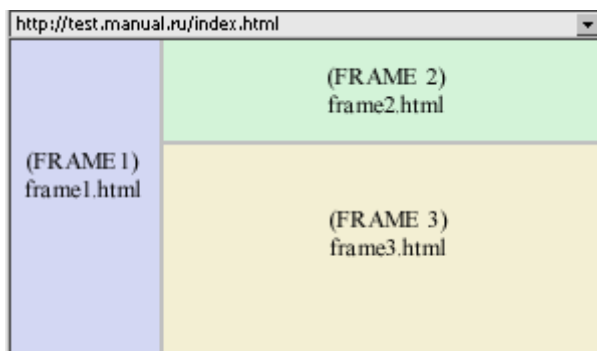


Рис. 5. Пример окна браузера с фреймами

*Документ группы фреймов* - Web-страница, которая разделена на фреймы.

Документы группы фреймов содержат стандартный заголовок, задаваемый тегом HEAD, но в отличие от стандартных HTML-документов, они не содержат тега BODY. Вместо него используется контейнер <FRAMESET>, который применяется для определения строк и столбцов отдельных фреймов, каждый из ко-

торых обозначается тегом <FRAME>.

Элементы для создания фреймов и работы с ними.

*Тег определения фреймовой (оконной) структуры документа* (размеры и расположение фреймов на странице).

*Формат метки:* <FRAMESET> ...</FRAMESET>.

Атрибуты тега определяют способ разбиения окна:

*ROWS* - окно разбивается горизонтальными линиями;

*COLS* - окно разбивается вертикальными линиями;

*Оба атрибута* - создается сетка фреймов.

Значение любого из этих атрибутов – перечисленные через запятую размеры отдельных фреймов (в пикселах, процентах или относительных единицах от ширины окна). Последнее значение может быть определено с помощью символа «\*» - выделяется оставшееся пространство.

*Пример:* <FRAMESET ROWS="25%, 50%, 25%"> - задаст три кадра, размером по 25%, 50% и 25% от высоты окна браузера.

В абсолютных единицах стоит указывать размеры кадра лишь для размещения небольших изображений, в остальных же случаях лучше пользоваться относительными величинами.

*Пример:* <FRAMESET COLS="\*,2\*,3\*">

Можно определить любое число рядов и столбцов; необходимым условием является указание хотя бы одного из атрибутов ROWS или COLS.

Кадр не может быть единственным.

*Атрибут BORDER* - используется только с тегом <FRAMESET> и устанавливает ширину всех рамок для всех кадров контейнера FRAMESET.

*Тег <FRAME>* - определение фрейма и его свойств внутри FRAMESET-структуры. Число тегов <FRAME> обязательно должно быть равно числу кадров, определенных в теге <FRAMESET>. Тег содержит:

- 1) *обязательный атрибут SRC*, с помощью которого указывается абсолютный или относительный адрес документа, который загружается во фрейм;
- 2) *атрибут NAME*, позволяющий задать имя созданной области в виде по-

следовательности латинских букв и цифр, использованной как значение этого атрибута

*Пример:* <FRAME SRC="text.htm" NAME="left">

Созданное имя может использоваться для ссылки к фрейму. Обычно ссылка задаётся из другого фрейма, располагающегося на той же самой странице.

*Пример:* <FRAME SRC="sample.htm" NAME="frame\_1">

Следует ввести в тег A атрибут TARGET, значение которого совпадает с ранее определенным именем области. При переходе по данной гиперссылке новый документ загрузится в указанный фрейм.

*Пример:* <A HREF="other.htm" TARGET="frame\_1">Кликните здесь для загрузки документа other.htm во фрейм с именем frame\_1</A>

3) *атрибут* MARGINWIDTH устанавливает величину левых и правых полей;

4) *атрибут* MARGINHEIGHT устанавливает величину верхних и нижних полей фрейма.

Значения этих атрибутов всегда должны указываться в абсолютных значениях (пикселах). Минимально допустимое значение равно единице.

*Пример:* <FRAME MARGINHEIGHT="5" MARGINWIDTH="7">

5) *атрибут* SCROLLING управление отображением полос прокрутки

Параметр SCROLLING может принимать три значения:

- значение YES вызывает появление полос прокрутки вне зависимости от необходимости этого,

- значение NO - запрещает их появление,

- значение AUTO действует также, как и в случае отсутствия параметра SCROLLING.

6) *атрибут* NORESIZE - запрещение возможности "перекраивания" страницы.

Этот атрибут не имеет значений. Указав его в одном кадре, тем самым запрещается изменять размеры и всех смежных кадров.

*Строковый (плавающий) фрейм*

Тэг <IFRAME> - вставка фрейма в любое место обычного HTML документа. Ширина строкового фрейма задается с помощью атрибута WIDTH, а высоту с помощью атрибута HEIGHT.

Пример:

```
<IFRAME SRC="fr_a.html" WIDTH="400" HEIGHT="150"></IFRAME >
<IFRAME SRC="fr_b.html" WIDTH="700" HEIGHT="250"></IFRAME >
```

## Лабораторная работа 6. HTML. Фреймы.

### Упражнение 1. Создание Web-документа с использованием фреймов

#### Порядок работы:

1. Создать файл frames.html в папке Фреймы1 и записать в нем следующие теги:

```
<HTML>
<FRAMESET ROWS="*,*">
<FRAMESET COLS="*,*">
  <FRAME SRC="frame1.html" NAME="fr1">
  <FRAME SRC="frame2.html" NAME="fr2">
</FRAMESET>
  <FRAME SRC="frame3.html" NAME="fr3">
</FRAMESET>
</HTML>
```

2. Создать файлы frame1.html, frame2.html, frame3.html и заполнить их следующим образом:

#### **frame1.html:**

```
<HTML><BODY BGCOLOR=white>
  <H1>Frame1</H1>
  <A HREF=frame2.html TARGER="fr3">Ссылка на 2
  кадр</A>
</BODY></HTML>
```

#### **frame2.html:**

```
<HTML><BODY BGCOLOR=red TEXT=yellow>
```

```
<H1>Frame2</H1>
<A HREF=frame3.html TARGER=_top>Frame3 во все
окно</A>
</BODY></HTML>
```

### **frame3.html:**

```
<HTML><BODY BGCOLOR=purple>
<H1>Frame3</H1>
<A HREF=frame1.html TARGER=_self>Frame1 в это
окно</A>
</BODY></HTML>
```

3. При помощи браузера загрузить файл frames.html и посмотреть на результат.

## **Упражнение 2. Создание фреймовой структуры с навигацией**

### **Порядок работы:**

1. Создать шесть файлов в папке Фреймы2: Index.html, kontakt.html, navigation.html, produkte.html, start.html, impressum.html.

### **Index.html**

```
<HTML>
<HEAD>
<TITLE>Фреймы</TITLE>
</HEAD>
<FRAMESET cols="300,*" frameborder="0">
<frame src="navigation.html" name="Navi" scroll-
ing="no"/>
<frame src="start.html" name="Содержание" scroll-
ing="no"/>
</FRAMESET>
</HTML>
```

### **start.html**

```
<HTML>
```

```
<HEAD>
    <TITLE>Старт</TITLE>
</HEAD>
<BODY>
    <H1>Добро пожаловать!</H1>
    <p>Здесь вы найдете информацию о наших
    услугах</p>
</BODY>
</HTML>
```

### **navigation.html**

```
<HTML>
    <HEAD>
        <TITLE>Навигация</TITLE>
    </HEAD>
    <BODY>
<h1> Навигация</h1>
<p><a href="produkte.html" tar-
get="Содержание">Продукция</a></p>
<p><a href="kontakt.html" tar-
get="Содержание">Контакт</a></p>
<p><a href="impressum.html" tar-
get="Содержание">Выходные данные</a></p>
    </BODY>
</HTML>
```

### **produkte.html**

```
<HTML>
    <HEAD>
        <TITLE>Продукция</TITLE>
    </HEAD>
    <BODY>
```



```
<H1>Продукция</H1>
<p>Мы вам предлагаем:</p>
<p>Подушки из силикона</p>
<p>Удобные матрасы</p>
</BODY>
</HTML>
```

### **kontakt.html**

```
<HTML>
  <HEAD>
    <TITLE>Контакт</TITLE>
  </HEAD>
  <BODY  >
    <H1>Контакт</H1>
    <p>Вы можете написать нам письмо по адресу maga-
zin@mail.ru</p>
  </BODY>
</HTML>
```

### **impressum.html**

```
<HTML>
  <HEAD>
    <TITLE>Выходные данные</TITLE>
  </HEAD>
  <BODY>
    <H1>Выходные данные</H1>
    <p>Разработчик сайта - студент Иванов С.Д.</p>
  </BODY>
</HTML>
```

3. При помощи браузера загрузить файл Index.html и проверить работу гиперссылок.

### Упражнение 3. Строковые фреймы

#### Порядок работы:

1. Создать файл stroka.html следующего содержания и сохранить его в папке

#### Фреймы2

```
<html>
  <body>
    <p> Это строковый фрейм: </p>
    <iframe src="produkte.html" width='400'
height='150'></iframe>
    <p> Еще один строковый фрейм:</p>
    <iframe src="kontakt.html" width='700'
height='250'></iframe>
  </body>
</html>
```

3. При помощи браузера загрузить файл и посмотреть на результат.

## 11. Работа с формами

Организовать интерактивный диалоговый режим в web-документе позволяют формы. С помощью языка HTML можно создавать простые формы, предполагающие ответы типа "да" и "нет", разрабатывать сложные формы для заказов или для того, чтобы получить от своих читателей какие-либо комментарии и пожелания.

*Форма* – набор полей, позволяющих передавать информацию программам-обработчикам. В область поля пользователь может ввести некоторую информацию, либо выбрать какую-то опцию.

После того как пользователь отправит информацию, она обрабатывается программой (скриптом), размещённой на сервере. *Скрипт* – это короткая программа, специально созданная для обработки каждой формы.

Внешне формы представляют собой набор примитивных стандартных элементов – кнопки, флажки, переключатели, ниспадающие меню.

Формат метки: *<FORM> ...</FORM>*.

Метка включает три атрибута:

- 1) *обязательный атрибут ACTION* – определяет адрес URL, который примет и обработает данные формы (если этот атрибут не определен, данные отправляются по адресу страницы, на которой помещена форма);
- 2) *атрибут METHOD* – определяет способ передачи данных из формы обработчику;

Допустимые значения:

- *POST* – данные отправляются отдельно от URL (по электронной почте);
- *GET* – данные передаются вместе с URL;
- 3) *атрибут ENCTYPE=* – определяет, каким образом данные из формы будут закодированы для передачи обработчику

В форме могут использоваться следующие теги:

- 1) *непарный тег <INPUT>* – создание различных элементов управления (в том числе текстовые поля и командные кнопки);
- 2) *парный тег <SELECT>* - создание списков (обычных и раскрывающихся) – отдельные пункты задаются с помощью парного тега *<OPTION>*, который допустим только внутри данного элемента;
- 3) *парный тег <TEXTAREA>* - создание поля для ввода нескольких строк текста.

Каждый из этих тегов должен иметь атрибуты: *NAME* - определяет имя поля формы, *VALUE* - значение поля. Имя задается латинскими буквами (не более 20 символов).

Для текстового поля и текстовой области атрибут *VALUE* приобретает значение, соответствующее содержимому этого поля, заданного пользователем; для списков атрибут *VALUE* соответствует содержимому выбранного элемента *OPTION*.

Тег *<INPUT>* должен содержать обязательный атрибут *TYPE*, определяющий конкретный тип элемента управления.

## Основные типы элементов

Тип	Назначение
<i>TEXT</i>	определяет окно для ввода строки текста; может содержать атрибуты <i>SIZE</i> =[число] (ширина окна ввода в символах) и <i>MAXLENGTH</i> =[число] (максимально допустимая длина вводимой строки в символах)
<i>PASSWORD</i>	Определяет окно для ввода пароля *аналогичен типу <i>TEXT</i> , только показываются звездочки
<i>RADIO</i>	Определяет переключатель; может содержать атрибут <i>CHECKED</i> (показывает, что кнопка помечена); в группе переключателей с одинаковыми именами может быть только один помеченный переключатель
<i>CHECKBOX</i>	Создается флажок, который можно пометить; может содержать атрибут <i>CHECKED</i> (показывает, что флажок установлен); в группе может быть несколько флажков
<i>SUBMIT</i>	Создается кнопка отправки формы
<i>IMAGE</i>	Создается графическая кнопка отправки
<i>RESET</i>	Создается кнопка очистки формы

### Примеры:

- 1) `<INPUT TYPE=TEXT SIZE=20 NAME=user VALUE=«Сергей»>`
- 2) `<INPUT TYPE=PASSWORD NAME=pw SIZE=15 MAXLENGTH=8>`
- 3) `<INPUT TYPE=RADIO NAME=modem VALUE="33600 бит/с" CHECKED>`  
33600 бит/с  
`<INPUT TYPE=RADIO NAME=modem VALUE="55400 бит/с">` 55400 бит/с
- 4) `<INPUT TYPE=CHECKBOX NAME=comp VALUE="PC">` Персональные компьютеры  
`<INPUT TYPE=CHECKBOX NAME=comp VALUE="WS" CHECKED>` Рабочие станции
- 5) `<INPUT TYPE=RESET VALUE=«Очистить поля формы»>`

Пользователь может использовать формы с различными целями. С помощью описанных выше элементов создавать заполняемые анкеты, опросники и различные поля для ввода текста пользователем с возможностью последующей отправки заполненной формы на ваш сервер. Простейшим примером является размещение формы, куда читатели, посетившие сайт, смогут записать свои отзывы.

## Лабораторная работа. HTML. Web-формы

### Упражнение 1. Простейшая форма с полем для ввода многострочной текстовой информации

#### Порядок работы:

1. Создать Web-документ с формой:

```
<HTML><BODY>  
    <FORM>  
        <TEXTAREA> Текст в форме для информации пользователя  
    </TEXTAREA>  
    </FORM>  
</BODY></HTML>
```

2. Просмотреть в браузере
3. Увеличить ширину формы, чтобы информация размещалась в одной строке (атрибут COLS для тега TEXTAREA)

### Упражнение 2. HTML-документ, содержащий форму со списком

#### Порядок работы:

1. Создать Web-документ с формой:

```
<HTML>  
<BODY>  
    Выберите время года:  
    <FORM>  
        <SELECT NAME=year>  
            <OPTION SELECTED VALUE="winter"> зима  
            <OPTION VALUE="spring"> весна  
            <OPTION VALUE="summer"> лето  
            <OPTION VALUE="autumn"> осень  
        </SELECT>  
    </FORM>  
</BODY> </HTML>
```

2. Просмотреть результат в браузере, попробовать выбрать конкретное время года.

### **Упражнение 3. Использование в форме элементов с вводом значений.**

#### **Порядок работы:**

1. Создать документ с возможностью ввода информации:

```
<HTML>
<BODY>
  <FORM>
    Введите номер телефона:
    <INPUT TYPE="TEXT" NAME="phone"
      SIZE="15" MAXLENGTH="12">
  </FORM>
</BODY>
</HTML>
```

2. Заменить ввод телефонного номера на ввод пароля из не более 8 символов: ввести текст «Введите пароль», заменить тип поля на PASSWORD, имя на SECRET, установить ограничение на количество символов.

3. Открыть документ в браузере и попробовать ввести слово «информатика».

### **Упражнение 4. Использование в форме элементов с флажками.**

#### **Порядок работы:**

1. Создать документ с возможностью выбора из списка:

```
<HTML>
<FORM>
<P><B> С какими операционными системами вы знакомы? </B>
</P>
  <P><INPUT TYPE="CHECKBOX" NAME="OPTION1" VALUE="A1"
    CHECKED> Windows 95/98<BR>
  <INPUT TYPE="CHECKBOX" NAME="OPTION2" VALUE="A2">
```

Windows 2000<BR>

```
<INPUT TYPE="CHECKBOX" NAME="OPTION3" VALUE="A3">
```

System X<BR>

```
<INPUT TYPE="CHECKBOX" NAME="OPTION4" VALUE="A4">
```

Linux<BR>

```
<INPUT TYPE="CHECKBOX" NAME="OPTION5" VALUE="A5">
```

X3-DOS</P>

```
<P><INPUT TYPE="SUBMIT" VALUE="Выбрать"></P>
```

```
</FORM>
```

```
</HTML>
```

2. Просмотреть в браузере.

## **Упражнение 5. Создание двух форм с выбором одного переключателя**

### **Порядок работы:**

1. Создать документ с возможностью выбора из списка:

```
<HTML>
```

```
<BODY>
```

Форма 1.

```
<FORM>
```

```
<P><B> Укажите Ваш пол:</B></P>
```

```
<INPUT TYPE="radio" NAME="choice" VALUE="men"
```

```
CHECKED> Мужской
```

```
<INPUT TYPE="radio" NAME="choice" VALUE="women"> Жен-  
ский
```

```
</FORM>
```

```
<HR>
```

Форма 2.

```
<FORM>
```

```
<P><B>Определите свой тип мышления?</B></P>
```

```
<P><INPUT TYPE="radio" NAME="dzen" VALUE="optim">
```

```

    ОПТИМИСТ</P>
        <P><INPUT TYPE="radio" NAME="dzen" VAL-
    UE="pessim"> пессимист</P>
        <P><INPUT TYPE="radio" NAME="dzen" VALUE="real"
    CHECKED> реалист</P>
        <P><INPUT TYPE="submit" VALUE="Выбрать"></P>
    </FORM>
</BODY>
</HTML>

```

2. Просмотреть в браузере.

## Упражнение 6. Создание формы с кнопками

### Порядок работы:

1. Создать документ:

```

<html>
<body>
    <form>
        <p><input type="button" value=" Нажми меня " ></p>
    </form>
</body>
</html>

```

2. Просмотреть в браузере.

3. Добавить в форму кнопку типа SUBMIT для отправки данных на сервер:

```

<p><input value="Введите текст"></p>
<p><input type="submit"></p>

```

4. Просмотреть текст на кнопке, который автоматически добавит браузер (он различается в зависимости от браузера).

5. Добавить кнопку типа RESET для очистки введенной в поле формы информации:

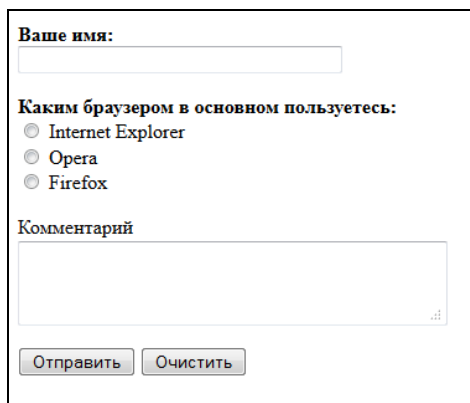
```

<input type="reset" value="Очистить"></p>

```



6. Просмотреть результат.



The image shows a web form with the following elements:

- A text input field labeled "Ваше имя:".
- A section titled "Каким браузером в основном пользуетесь:" containing three radio button options: "Internet Explorer", "Opera", and "Firefox".
- A text area labeled "Комментарий" with a small "x" icon in the bottom right corner.
- Two buttons at the bottom: "Отправить" (Submit) and "Очистить" (Clear).

Рис. 6. Форма с кнопками

**Задание.** Создать форму в виде вопросника, содержащую поле для ввода текстовой информации, список с возможностью выбора, 2 кнопки (подобно примеру на рис. 6).

## 12. Каскадные таблицы стилей

### *Понятие и возможности таблиц стилей*

Для описания внешнего вида документа используется язык CSS (каскадные таблицы стилей), о котором впервые заговорили в 1995 году. Главная цель CSS - отделить структуру документа от его оформления и позволить автору или пользователю самому решать, как должен выглядеть тот или иной элемент содержания HTML-страницы. Другим достоинством таблиц стилей является возможность обеспечить единый стиль оформления некоторого набора HTML-документов.

*Стиль* - правило, сообщающее браузеру, как интерпретировать отдельные теги HTML.

*Каскадные таблицы (листы) стилей CSS (Cascading Style Sheets) или язык иерархических стиливых спецификаций* - набор правил отображения, применяемых в документе, к которому присоединяется соответствующая таблица стилей. Данная технология используется для форматирования документов HTML.

*Возможности CSS:*

а) определение начертания и размера шрифта, цвет фона и переднего плана, фоновое изображение, размеры границ и другие характеристики;

б) управление отображением стандартных и новых элементов с наложением ограничений;

в) сочетание установок браузера, используемых по умолчанию, с непосредственными инструкциями автора документа и пользователя.

*Описание стиля* включает селектор тега, указывающий имя тега, на который распространяется правило стиля, и фигурные скобки, содержащие одну или более пар объявлений вида "свойство\_стиля: значение", разделенных точкой с запятой.

```
селектор_тега {  
    свойство1: значение1;  
    свойство2: значение2;  
    свойство3: значение3;  
}
```

*Пример:* font-size: 12pt; - установка размера шрифта.

В HTML включены десятки команд-свойств (font-size, font-style, color, margin-right и т. д.). Каждое свойство может принимать несколько значений, одно из которых приписывается ему по умолчанию. Все свойства, используемые в таблицах стилей, условно можно разделить на несколько групп:

- *управляющие видом шрифта* (гарнитура, кегль, цвет, наклон, жирность, ...);
- *управляющие форматированием абзаца* (отступ красной строки, выравнивание, интерлиньяж, расстояние между словами, ...);
- *управляющие свойствами блока* (отступы слева-сверху-справа-снизу, местоположение блока в документе, видимость блока, ...);
- *другие* (цвет ссылок, изменение внешнего вида курсора, ...).

Первая группа. *Теги-селекторы* – имя тега записывается в качестве селектора:

```
p {  
    font-size: 16pt;  
    font-weight: bold;
```

```
}
```

Вторая группа. *Множество тегов* – форматируется список тег-селекторов, которые отделены запятыми:

```
p, li {  
    font-size: 16pt;  
    font-weight: bold;  
}
```

Третья группа. *Селекторы каскадных типов, вложенных один в другой* – вложенные теги перечисляются через пробел:

```
ol ol li {  
    font-size: 12pt;  
    font-style: italic;  
}
```

#### *Подключение стиля*

Существует три способа связи CSS и HTML (подключения стиля):

- 1) внутренние стили;
- 2) глобальные стили;
- 3) связанные стили.

*Внутренние стили (Inline-описание или описание, встроенное в тег)* – использование дополнительного атрибута *style* для определения нужные стилевых параметров в любом теге. Определение стиля действует в пределах лишь одного тега. Этот способ не слишком отличается, к примеру, от прямого описания внешнего вида при помощи тега `<font>`.

#### *Пример:*

```
<p style="color:red; text-align:center;">Этот параграф переопределен стилем</P>
```

*Глобальные стили - описание стиля в заголовке HTML-страницы.* Команды размещаются между тэгами *html* и *body*. В парном теге *style* указывается принадлежность CSS и имя тега, для которого создается стиль, с фигурными скобками (селектор). Затем в фигурные скобки вписываются все CSS-команды для данного селектора.

*Пример:*

```
<head>
  <style type="text/css">
    h1 {color: blue;}
  </style>
</head>
<body>
<h1> Синий заголовок </h1>
</body>
```

Тэг `<style>` имеет единственный атрибут `type`, определяющий тип mime (Multipurpose Internet Mail Extension, стандарт электронной почты Internet). Определяется он как "text/css" для того, чтобы браузеры, не поддерживающие таблицы стилей, могли игнорировать тэг `<style>`.

*Связанные стили* - вынесение описания стилей во внешний файл (внешнюю таблицу стилей). Для этого все описание стилей помещается в отдельный текстовый файл с расширением `*.css`.

В раздел *head* документа включается строка с тегом `<link>`:

```
<link rel="STYLESHEET" type="text/css" href="list.css">
```

Первые два параметра этого тега являются зарезервированными именами, требующимися для того, чтобы сообщить браузеру, что на этой страничке будет использоваться CSS. Третий параметр – `href = <URL>` – указывает на файл, который содержит описания стилей. Этот параметр должен содержать либо относительный путь к файлу – в случае, если он находится на том же сервере, что и документ, из которого к нему обращаются – или полный адрес *URL* (`<http://...>`) в случае, если файл стилей находится на другом сервере.

*Пример файла со стилями:*

```
H1 {
  color: #000080;
  font-size: 200%;
  font-family: Arial, Verdana, sans-serif;
```

```
text-align: center;
}
P {
padding-left: 20px;
}
```

### *Наследование и каскадность*

Применение CSS к документам HTML основано на принципах наследования и каскадирования.

*Наследование* - заключается в том, что свойства форматирования любого элемента наследуются от элемента, в котором он размещается. CSS свойства всегда имеют некоторые значения, даже если автор их не определил.

Этим свойством можно пользоваться для минимизации описания стилей. Но следует помнить, что в различных браузерах могут быть различные значения по умолчанию и отображение страницы при этом будет различаться.

*Каскадность стилей* - возможность переопределять стили с учетом расставленных приоритетов таблиц стилей (элемент получает все больше параметров на наследстве, образуя тем самым каскады).

#### *Приоритеты таблиц стилей:*

- 1) самый низкий приоритет - стили браузера по умолчанию;
- 2) далее приоритет - связанные стили;
- 3) чуть больший приоритет - внедренные стили;
- 4) наивысший приоритет - inline-стили;
- 5) самым высоким приоритетом обладают стили, объявленные автором страницы или пользователем, с помощью сопроводительного свойства стиля *!important*.

#### *Пример:*

```
h1 {color: maroon !important;}
```

## Лабораторная работа 8. HTML. Стили. Способы применения

### Упражнение 1. Применение внутренней таблицы стилей

#### Порядок работы:

```
<HTML>
<HEAD>
<TITLE>Первый пример использования стилей</TITLE>
</HEAD>
<BODY>
    <p style="color: red; margin-left: 20px">
        Это текст со стилем.
    </p>
    <p> Это текст без стиля
</BODY>
</HTML>
```

### Упражнение 2. Применение глобальных таблиц стилей

#### Порядок работы:

```
<HTML>
<HEAD>
<TITLE>Второй пример использования стилей</TITLE>
    <STYLE>
        h1 {text-align: center}
        p {background-color: grey; font-size:14pt}
        body {background-color:#b0c4de}
    </STYLE>
</HEAD>
<BODY>
    <h1> Все о стилях</h1>
    <p> Использовать таблицы стилей очень удобно
</BODY> </HTML>
```

### Упражнение 3. Применение оформления со списком тег-селекторов

**Задание.** Установить белый цвет фона документа (свойство background-color), свойства управления шрифтом:

- цвет (свойство color) - серый;
- семейство шрифта (гарнитура) (свойство font-family) - Times;
- размер шрифта (свойство font-size) - 14 пунктов.

Добавить свойства, связанные с форматированием абзаца:

- дополнительный отступ красной строки (свойство text-indent) - 10pt;
- выравнивание (свойство text-align) - по ширине документа;
- отступ от левого края (свойство padding-left) - 20pt.

Установить темно-красный цвет (свойство color) заголовков с первого по третий уровень.

#### Порядок работы:

```
<HTML>
<HEAD>
<TITLE>Третий пример использования группы селекто-
ров</TITLE>
  <STYLE TYPE="text/css">
    BODY {background-color: white}
    P    {
      /* шрифт */
        color: green;
        font-family: Times, serif;
        font-size: 14pt;
      /* абзац */
        text-indent: 10pt;
        text-align: justify;
        padding-left: 20pt;
    }
    H1, H2, H3 {color: red; text-align: center}
  </STYLE>
</HEAD>
<BODY>
<h1>Каскадные таблицы стилей</h1>
<h2>Возможности CSS</h2>
```

<p>Главная цель CSS – отделить структуру документа от его оформления и позволить автору или пользователю самому решать, как должен выглядеть тот или иной элемент содержания HTML-страницы. Другим достоинством таблиц (иногда называемых листами) стилей является возможность обеспечить единый стиль оформления некоторого набора HTML-документов

</p> </BODY>

</HTML>

#### **Упражнение 4. Применение стиля с помощью связанных таблиц стилей**

##### **Порядок работы:**

Создать внешний файл main.css:

```
body {background:black; font-size:24pt; color:red; font-family:Arial Black}
h1 {color:white;text-align: center}
```

Создать файл Index.html:

```
<html>
<head>
    <TITLE> Четвертый пример использования связанных
таблиц стилей </TITLE>
    <LINK rel="stylesheet" type="text/css"
href="main.css">
</head>
    <body>
    <h1>Заголовок</h1>
        Содержание документа
    </body>
</html>
```

##### **Задание**

Создайте HTML-документ, в котором:

- заголовку первого уровня задайте стиль оформления, определяющий цвет текста, выравнивание, шрифт, размер шрифта;
- заголовку второго уровня задайте выравнивание, начертание, размер шрифта;



- установите цвет фона документа.

### 13. Классы стилей

Селекторы специально ограничены одним или несколькими тегами. Если необходимо обозначить стилями индивидуальные элементы HTML-страницы или задать разные стили для одного тега, то следует подразделить их на классы.

*Назначение класса* - возможность присваивать стили не всем одинаковым элементам страницы, а избирательно.

*Синтаксис для классов:*

*Тег.* *Имя класса* { *свойство1: значение; свойство2: значение; ...* }

Чтобы указать в коде HTML, что тег используется с определенным стилем, к тегу добавляется параметр *CLASS* = "*имя класса*" присваивающийся любому элементу страницы.

*Пример:*

```
.bc { font-weight: bold; text-align: center }
```

...

```
<p class = "bc"> Текст параграфа</p>
```

```
<td class = "bc">текст</td>
```

Таким образом, можно присвоить описанный класс любым текстовым элементам страниц. При написании названия классов необходимо соблюдать *регистр символов*, согласно названию класса в описании стиля!

*Универсальный класс* - может быть применен к любому тегу, имя класса начинается с точки.

*Пример:*

```
.small { font-size: 9pt; color: #eeeeee; text-align: center; }
```

...

```
<p class=small>Накладываем стиль на этот текст</p>
```

*Теговый класс* - может быть применен к конкретному тегу, имя класса начинается с указания тега, после точки записывается имя класса.

*Пример:*

```
p.small { font-size: 9pt; }
```

...

```
<p class=small>Этот текст будет выведен стилем small</p>
```

```
<td class=small>А этот останется неизменным</td>
```

*Псевдокласс* - может быть применен к части текста или к определенному его состоянию.

*Пример:*

```
a {text-decoration: none; }
```

```
a:hover {text-decoration: underline; }
```

*Пример (определение буквицы в начале абзаца):*

```
p:first-letter { font-size: 200%; font-weight: bold; }
```

*Тег* `<span>` используется как контейнер стиля для применения стилевых свойств к группе элементов уровня строки или к одному элементу.

*Пример:*

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

```
.bold { font-weight: bold }
```

```
.ital { font-style: italic }
```

```
.sm_cap { font-variant: small-caps }
```

```
</style>
```

...

```
<p> Одна из ошибок начинающих создателей сайтов — избыточное использование на странице <span class="bold">жирного</span>, <span class="ital">курсивного</span> шрифтов, написание текста только <span class="sm_cap">заглавными</span> буквами. </p>
```

*Тег* `<div>` используется для формирования структуры документа в виде блоков текста.

*Пример:*

```
<div style="color: green; font-weight: bold"><h1>Это заголовок</h1></div>
```

*Комментарии* - используются при создании таблицы стилей.

*Пример:*

*/\* Этот текст является комментарием \*/*

## Лабораторная работа 9. HTML. Стили. Классы.

### Упражнение 1. Использование классов с тегами

#### Порядок работы:

1. Создать HTML-документ:

```
<html>
  <head>
    <title>Классы</title>
    <style type="text/css">
      P { /* Обычный параграф */
        text-align: justify;
      }
      P.cite { /* Параграф с классом cite */
        text-align: justify;
        color: navy;
        font-style: italic; /* Курсивное начертание */
      }
    </style>
  </head>
  <body>
    <p> При работе на вычислительной технике необходимо
    сесть так, чтобы руки с предплечьями образовывали прямой
    угол, глаза поставить на расстояние 30-40 см от рабочей
    поверхности монитора. Набирать на клавиатуре следует по-
    душечками пальцев короткими отрывистыми ударами.</p>
    <p class="cite"> Предельно допустимая длина ногтей для
    женщин составляет 12-15 мм, для мужчин 3-5 мм. При выходе
    длины ногтей за нормы регламентируемые ГОСТом, во избежа-
    нии риска поцарапать поверхность дорогостоящей техники,
    оператор допускается к работе на компьютере только в вер-
    хонках.</p>
  </body>
</html>
```

2. Разметить текст «Инструкция» как головок второго уровня по центру страницы используя свойства стиля.

## Упражнение 2. Использование классов для части текста

### Порядок работы:

1. Создать HTML-документ:

```
<html>
  <head>
    <title>Классы</title>
    <style type="text/css">
      .cite {
        color: navy; /* Синий цвет текста */
        font-style: italic; /* Курсивное начертание */
      }
    </style>
  </head>
  <body>
    <p> Следует тщательно позаботиться о своем рабочем ме-
сте. Освещение в помещении отрегулировать таким обра-
зом, чтобы источник света находился сбоку или сзади опера-
тора. Во избежании медицинских осложнений <b
class="cite"> стул рекомендуется выбирать с мягким сиде-
нием </b>.</p>
  </body>
</html>
```

## Упражнение 3. Использование классов со стилем первой буквы предложе- ния

### Порядок работы:

1. Создать HTML-документ:

```
<html>
  <head>
    <title>Классы</title>
    <style type="text/css">
      .inicial {
        color: red;
        font-size: 200%; /* Размер текста */
      }
    </style>
```

```
</head>
<body>
  <p><span class="inicial">Н</span>абирать на клавиатуре
следует подушечками пальцев короткими отрывистыми удара-
ми.</p>
</body> </html>
```

#### **Упражнение 4. Применение стиля с псевдоклассами (вертикальное меню)**

##### **Порядок работы:**

##### 1. Создать HTML-документ:

```
<html>
<head>
<TITLE> Четвертый пример использования классов</TITLE>
  <style type="text/css">
    ul
    {
      list-style-type:none; /*отмена маркера в списке */
      margin:0; /*отступ от границы */
      padding:0; /* отступ от текста */
    }
    a:link,a:visited /*переопределение стиля для ссылки и
посещенной ссылки */
    {
      display:block;
      font-weight:bold; /*жирность шрифта */
      color:#FFFFFF;
      background-color:#98bf21;
      width:120px;
      text-align:center;
      padding:4px;
      text-decoration:none; /*ссылки без оформления*/
      text-transform:uppercase; /*прописные символы*/
    }
    a:hover,a:active/*переопределение стиля при наведении
и активации ссылки*/
    {
      background-color:#7A991A; text-
      decoration:underline; /*подчеркивание*/
    }
  </style>
```

```
</head>
<body>
  <ul>
    <li><a href="#home">Главная</a></li>
    <li><a href="#news">Новости</a></li>
    <li><a href="#contact">Контакты</a></li>
    <li><a href="#about">О нас</a></li>
  </ul>
</body>
</html>
```

### **Задания**

Создать HTML-документ, в котором размещается информация о достопримечательностях населенного пункта. Оформить таблицу стилей с применением классов в виде отдельного файла. В файле стилей следует использовать элементы оформления из задания лабораторной работы № 8.

### **ЛИТЕРАТУРА**

1. Кисленко Н.П. HTML. Самое необходимое. СПб: БХВ-Петербург, 2012 – 352 с.
2. Мержевич Влад Самоучитель HTML [Электронный ресурс] - Режим доступа: <http://htmlbook.ru/samhtml>
3. Мержевич Влад Самоучитель CSS [Электронный ресурс] - Режим доступа: <http://htmlbook.ru/samcss/znacheniya-stilevykh-svoystv>
4. HTML. Материал из Википедии. [Электронный ресурс] - Режим доступа: <http://ru.wikipedia.org/wiki/HTML>
5. HTML-книга. Самоучитель HTML. Справочник HTML. Самоучитель CSS. HTML-справочник. [Электронный ресурс] - Режим доступа: <http://html.manual.ru>
6. Справочник CSS [Электронный ресурс] - Режим доступа: <http://htmlbook.ru>

## СОДЕРЖАНИЕ

Введение .....	3
1. Общая структура сайта .....	4
2. Уровни создания сайта.....	7
3. Основные понятия и структура HTML-документов.....	8
4. Элементы языка HTML.....	10
5. Форматирование текста.....	12
Лабораторная работа 1. HTML. Общая структура web-документа .....	14
6. Списки.....	16
Лабораторная работа 2. HTML. Использование списков.....	17
7. Таблицы.....	20
Лабораторная работа 3. HTML. Работа с таблицами.....	22
8. Гипертекстовые ссылки.....	25
Лабораторная работа 4. HTML. Использование гипертекстовых ссылок ..	27
9. Работа с графическими изображениями.....	29
Лабораторная работа 5. HTML. Использование изображений.....	31
10. Организация фреймов.....	35
Лабораторная работа 6. HTML. Фреймы.....	38
11. Работа с формами.....	42
Лабораторная работа 7. HTML. Web-формы.....	45
12. Каскадные таблицы стилей.....	49
Лабораторная работа 8. HTML. Стили. Способы применения.....	54
13. Классы стилей.....	57
Лабораторная работа 9. HTML. Стили. Классы.....	59
Литература .....	62

Учебное издание  
Ульянова Наталья Дмитриевна

Основы языка HTML  
Учебно-методическое пособие

Компьютерный набор произвела Ульянова Н.Д.

Редактор Лебедева Е.М.

---

**Подписано к печати** Формат 60x84. 1/16. Бумага печатная

Усл.п.л. 4,0. Тираж 50 экз. **Изд.№**

---

Издательство Брянского государственного аграрного университета  
243365, Брянская обл., Выгоничский район, п. Кокино, БГАУ