

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ  
СЕВЕРО-КАВКАЗСКАЯ ГОСУДАРСТВЕННАЯ АКАДЕМИЯ

Л.Г. Темирова

# Web-программирование

Практикум для обучающихся 3 курса  
по специальности 01.03.04 Прикладная математика

Черкесск

2019

УДК 681.3  
ББК 32.973-018.2

Рассмотрено на заседании кафедры Математика

Протокол №2 от «21» 09 2018 г.

Рекомендовано к изданию редакционно-издательским советом СевКавГА.

Протокол № 15 от «30» 10 2018 г.

Рецензенты:

д.ф-м.н., профессор А.М. Кочкаров

к.ф-м.н., доцент Е.Х. Бежанова

Т32 **Темирова Л.Г.** Web-программирование: практикум для обучающихся 3 курса специальности 01.03.04 Прикладная математика / Л.Г. Темирова. – Черкесск: БИЦ СКГА, 2019. – 52 с.

Пособие предназначено для изучения и освоения обучающимися специальности 01.03.04 Прикладная математика языка разметки HTML и основ web-программирования как современных технологий разработки web-сайтов в рамках изучения одноименного курса.

Методическое пособие включает в себя лабораторные работы по освоению языка разметки HTML. Даны основные теоретические сведения о тегах, необходимые для выполнения лабораторных и индивидуальных заданий. По каждому индивидуальному заданию написаны коды html в Блокноте.

**УДК 681.3**  
**ББК 32.973-018.2**

© Темирова Л.Г., 2019  
© ФГБОУ ВО СКГА, 2019

## Содержание

Введение .....	4
Лабораторная работа №1 .....	5
Лабораторная работа №2.....	11
Лабораторная работа №3.....	12
Лабораторная работа №4.....	14
Лабораторная работа №5.....	18
Лабораторная работа №6.....	22
Лабораторная работа №7.....	30
Лабораторная работа №8.....	33
Лабораторная работа №9.....	37
Лабораторная работа №10 .....	42
Список литературы .....	51

## **Введение**

В современном мире для любой коммерческой компании является большим плюсом собственное представительство в интернете. Многомиллионная аудитория Всемирной паутины - это широкие возможности для развития бизнеса, продвижения продукции и услуг. Сегодня веб-сайт является мощным маркетинговым инструментом. Это один из наиболее эффективных способов заявить о себе потенциальным покупателям и пользователям. Наличие у фирмы собственного сайта считается признаком надежности, стабильности и профессионализма. Цель методического пособия – дать обучающимся представление об основах Web-технологий, научить их создавать свои собственные сайты на базе изучения тегов html, содержательно обучить технологии CSS.

Данное методическое пособие посвящено изучению основ Web-технологий. В их рамках рассматриваются вопросы создания Web-страниц с помощью HTML. Все лабораторные работы посвящены изучению языка гипертекстовой разметки HTML. После каждой лабораторной работы выполняются индивидуальные задания, также прилагаются вопросы для самоконтроля.

После выполнения всех лабораторных работ, обучающиеся будут уметь: разрабатывать общую структуру сайтов, формировать страницы, из которых состоит эта структура, добавлять интерактивные средства и эффекты мультимедиа и, наконец, размещать готовые сайты в Internet путем загрузки их на Web-сервер.

## Лабораторная работа №1

**Тема: Структура HTML-страницы, основные теги для форматирования текста**

**Цель: Закрепление основных понятий о структуре HTML-страницы, изучение основных тегов**

### Краткие теоретические сведения

Для того, чтобы понять структуру и сценарий Web-документа, необходимо рассмотреть несколько Web-страниц и выявить общие элементы.

Любой Web-документ состоит из тегов. Рассмотрим основные теги, входящие в каждый Web-документ. Прежде всего, это `<HTML></HTML>` Отличительный признак HTML-документа. Одним из принципов языка разметки является многоуровневое вложение элементов. HTML является самым внешним, так как между его стартовым и конечным тегами должна находиться вся Web-страница. Также основным тегом является `<HEAD></HEAD>` Область заголовка Web-страницы. Иными словами, ее первая часть. Так же как и HTML, HEAD служит только для формирования общей структуры документа.

`<BODY></BODY>` Этот элемент включает в себе гипертекст, который определяет собственно Web-страницу. Эта та часть документа, которую разрабатывает автор страницы и которая отображается браузером. Соответственно, конечный тег этого элемента надо искать в конце HTML-файла. Внутри BODY можно использовать все элементы, предназначенные для дизайна Web-страницы. Внутри стартового тега элемента BODY можно расположить ряд атрибутов, обеспечивающих установки для всей страницы целиком. Без этих тегов невозможно создать ни одну Web-страницу.

Структура Web-страницы:

```
<HTML>
  <HEAD>
    <TITLE>Название страницы</TITLE>
  </HEAD>
  <BODY>
    Тело страницы
  </BODY>
</ HTML>
```

Естественно, что конечным тегом `</HTML>` заканчиваются все гипертекстовые документы.

`<HEAD></HEAD>` Область заголовка Web-страницы. Иными словами, ее первая часть.

`<TITLE></TITLE>` Элемент для размещения заголовка Web-страницы. Строка текста, расположенная внутри, отображается не в документе, а в

заголовке окна браузера. Эта особенность часто используется для организации поиска в WWW. Поэтому авторы, создающие Web-страницы, должны позаботиться о том, чтобы строка внутри TITLE, не будучи слишком длинной, достаточно отображала назначение документа.

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

### Заголовки

В примере для заголовка используются теги `<H1></h1>`.

Существует шесть уровней заголовков, которые обозначаются H1...H6. Заголовок уровня H1 самый крупный, а уровень H6 - самый маленький заголовок. Для заголовков можно использовать атрибут **align**, задающий выравнивание влево, по центру или вправо.

`<COMMENT></COMMENT>` Текст комментария.

Текст, помещенный внутри COMMENT, игнорируется браузером. COMMENT может располагаться в любом месте кода Web-страницы. Без конечного тега, здесь по-видимому не обойтись: комментарий должен быть отделен от основного текста.

Существует и другой способ обозначения комментария. Он заключается в использовании восклицательного знака и обрамлении текста комментария двойным тире. Например:

`<!--Строка комментария-->`

Внутри подобной конструкции можно помещать и теги: они не будут восприниматься браузером.

### Теги для форматирования текста

Текст - единственный объект Web-страницы, который не требует специального определения. Иными словами, произвольные символы интерпретируются по умолчанию как текстовые данные. Но для форматирования текста существует большое количество элементов.

`<P></P>` Элемент абзаца (paragraph). Он позволяет использовать только начальный тег, так как следующий элемент P обозначает конец предыдущего и начало следующего абзаца. Вместе с элементом P можно использовать атрибут выравнивания Align.

`<BR>` Элемент, обеспечивающий принудительный переход на новую строку. Он имеет только стартовый (одиночный) тег. В месте его размещения строка заканчивается, а оставшийся текст печатается с новой строки.

`<NOBR></NOBR>` Этот элемент по своему действию является прямой противоположностью предыдущему. Текст, заключенный между его тегами,

будет выведен в одну строку. Длинная строка не уместится на экране, и для ее просмотра придется использовать горизонтальную полосу прокрутки.

**<PRE></PRE>** Элемент для обозначения текста, отформатированного заранее. Подразумевается, что текст будет выведен в том виде, в каком он был подготовлен пользователем.

**<DIV></DIV>** Он позволяет выравнивать содержимое по левому краю, по центру или по правому краю. Для этого стартовый тег должен содержать соответствующий атрибут **Align**.

**<B></B>** Выделение текста полужирным шрифтом. Очень популярный элемент. Использование полужирного шрифта - прием, позаимствованный из текстовых редакторов.

**<BIG></BIG>** Увеличенный размер шрифта

**<SMALL></SMALL>** Уменьшенный размер шрифта

**<I></i>** Выделение текста курсивом

**<STRIKE></STRIKE>** Элемент, создающий перечеркнутое начертание текста. В настоящее время его заменяют более простым: **<S></S>**

**<U></u>** Подчеркнутое начертание текста.

**<STRONG></STRONG>** Элемент, отвечающий за выделение текста. Обычно его применение равносильно использованию элемента для выделения полужирным.

**<SUB></SUB>** Элемент, создающий эффект нижнего индекса.

**<SUP></SUP>** Элемент, создающий эффект верхнего индекса.

**<CITE></CITE>** Предполагается, что этот элемент может быть использован для форматирования цитат и ссылок в обычном понимании этого слова. Текст, расположенный внутри него, выводится по умолчанию курсивом.

**<FONT></FONT>** управляет размером, начертанием и цветом текста. Имеет два атрибута **COLOR, FACE, SIZE**.

Например: **<FONT COLOR= red > текст </ FONT>** - выведет текст красным цветом (можно писать название цвета или его код, который можно найти в Adobe Photoshop – в его палитре инструментов).

Например: **<FONT FACE= Arial > текст</ FONT>** - приведет к выводу текста указанным шрифтом.

**<FONT SIZE = 4 > текст</ FONT>** - изменяет размер шрифта фрагмента текста, можно использовать в любом месте строки. Атрибуты могут перечисляться через пробел в произвольном порядке.

Например: **<FONT SIZE = 4 FACE= Arial COLOR= red> текст </ FONT>**

Цвета обычно кодируются в формате RGB (красный, зеленый, синий). Записывается это как «#RRGGBB», где шестнадцатеричное значение составляющих: RR-красный; GG - зеленый; BB - синий.

00 — обозначает отсутствие яркости, а FF — максимальную яркость.

Шестнадцатеричной называется система счисления, в которой для записи чисел используются 16 символов: 0,1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F. Таким образом, число FF соответствует десятичному числу 256.

### Примеры некоторых цветов, закодированных таким образом:

#000000 - Черный	#00FFFF - Бирюзовый
#FFFFFF - Белый	#0000FF- Синий
#FF0000 - Красный	#FF00FF- Фиолетовый
#FFFF00 -Желтый	#FF800- Оранжевый
	#8000FF- Лиловый

### Теги для создания фона

Один из самых полезных для дизайна - атрибут, определяющий фон страницы. Его появление можно уподобить маленькой революции в WWW, так как одинаковые серые Web-страницы благодаря ему расцвели яркими цветными узорами. Фон страницы задается в начале основной части документа, т.е. атрибутами тэга <BODY>.

**Атрибуты тэга <BODY>** для изменения цвета текста и фона:

**TEXT** – задает цвет текста, например: <BODY TEXT= 7EA3B8> значением атрибута может служить как название цвета, так и его код.

**BGCOLOR**– задает цвет фона, например:

<BODY BGCOLOR = BLACK >

<BGCOLOR= #RRGGBB>

<BODY **BACKGROUND** = \*.gif > - использовать в качестве фона изображения из файла.

**BGPROPERTIES=FIXED** - создание фона – водяного знака (фон, который не перемещается вместе с текстом) <BODY **BGPROPERTIES=FIXED**>

Поскольку фон страницы может изменяться, необходимо иметь возможность подбирать соответствующий цвет текста. Для этого имеется следующий атрибут:

**text= #RRGGBB**

Для задания цвета гиперссылок используется атрибут:

**link= RRGGBB**

Также можно задать цвет для использованных гиперссылок:

**vlink= RRGGBB**

Гипертекст, расположенный внутри элемента BODY, может иметь произвольную структуру. Ее определяют в первую очередь назначение Web-страницы и фантазия разработчика.

### Вставка графических изображений

<**IMG SRC=имя файла.расширение** > вставка картинки из файла

**Атрибуты тэга <IMG>:**

**SRC=имя файла.расширение**

**ALT**=название картинку или поясняющий текст (Используется в качестве текстового описания к данному изображению, появляется при наведении указателя мыши на картинку)

**WIDTH** = ширина (в пикселях или в %)

**HIGHT** = высота (в пикселях или в %)

**BORDER** = число в пикселях определяет рамку вокруг изображения

**HSPACE** = число в пикселях

**VSPACE** = число в пикселях

*LEFT*

**ALIGN** = *RIGHT*

*CENTER*

Используется, чтобы сдвинуть картинку к левому, правому краю, по центру и установить обтекание картинку текстом, а затем продолжить текст в пустой области за объектом.

*LEFT*

**CLEAR** = *RIGHT*

*ALL*

Используется, чтобы очистить поле слева, справа или с обеих сторон графического объекта, а затем продолжить текст в пустой области за объектом.

### **Вставка видео и звука**

**<BGSOUND SRC =имя файла>** - вставка звукового файла

Например: **<BGSOUND SRC=BLIP.WAV>**

**<IMG DYNSRC =имя файла>**- вставка видео файла

*n*

**LOOP** = *INFINITE*

Количество повторений видео или звукового клипа и непрерывное повторение соответственно.

**Индивидуальное задание 1.** Создать структуру HTML-страницы. Внутри созданной страницы использовать выше перечисленные основные теги. Написать ФИО, № группы, название кафедры, используя теги форматирования текста.

#### **Пример выполнения индивидуального задания 1**

**<HTML>**

**<HEAD>**

**<TITLE>** Создание структуры HTML-страницы, основные теги и теги форматирования текста **</TITLE>**

**</HEAD>**

**<BODY BGCOLOR="e5e5e5">**

**<H1><CENTER>**Индивидуальное задание №1 **</CENTER> </H1> <BR>**

**<B>**Название: **<I>**Структура HTML-страницы, основные теги и теги форматирования текста.**</I></B><BR><BR>**

```

<B>Цель работы: <I> Закрепление основных понятий о структуре HTML-страницы,
назначение основных тегов</I></B>
<BR><BR><H2>Выполнил</H2>
<STRONG>Ф.И.О.:</STRONG><U>Иванов Иван Иванович</u><BR>
<STRONG>Группа:</STRONG><U>ПМ-161</U><BR>
<STRONG>Кафедра:</STRONG><U>Математики</U><BR></BODY>
</HTML>

```

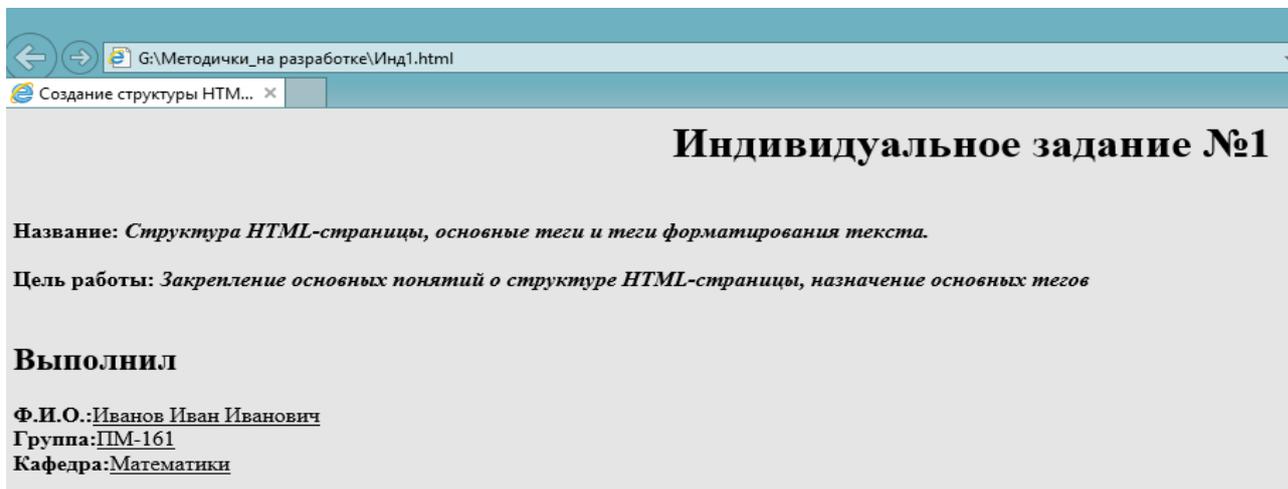


Рисунок 1 – Результат выполнения индивидуального задания 1

## Индивидуальное задание 2. Форматирование текста и вставка графических изображений

1. Открыть папку Пример\_1 и запустить index.htm
2. В главном меню Internet Explorer выполнить команду Вид/В виде HTML.
3. Сопоставить изображение в окне Internet Explorer и текст HTML- документа в блокноте, определить какие теги и атрибуты тегов позволяют отформатировать текст.
4. Создать свою папку и скопировать содержимое папки Пример\_1/ Материалы\_1.
5. Создать в блокноте и сохранить в этой папке HTML-документ, используя файлы из папки Материалы\_1 (Вариант\_1 или Вариант\_2).
6. Тексты перенести из текстовых файлов в HTML-документ копированием. Картинки вставить соответствующим тегом. Применить различное взаимное расположение текста и картинки: с обтеканием и без, выровнять по центру, по левому краю и т.д. Результат должен быть примерно такой, как на рисунке 2.

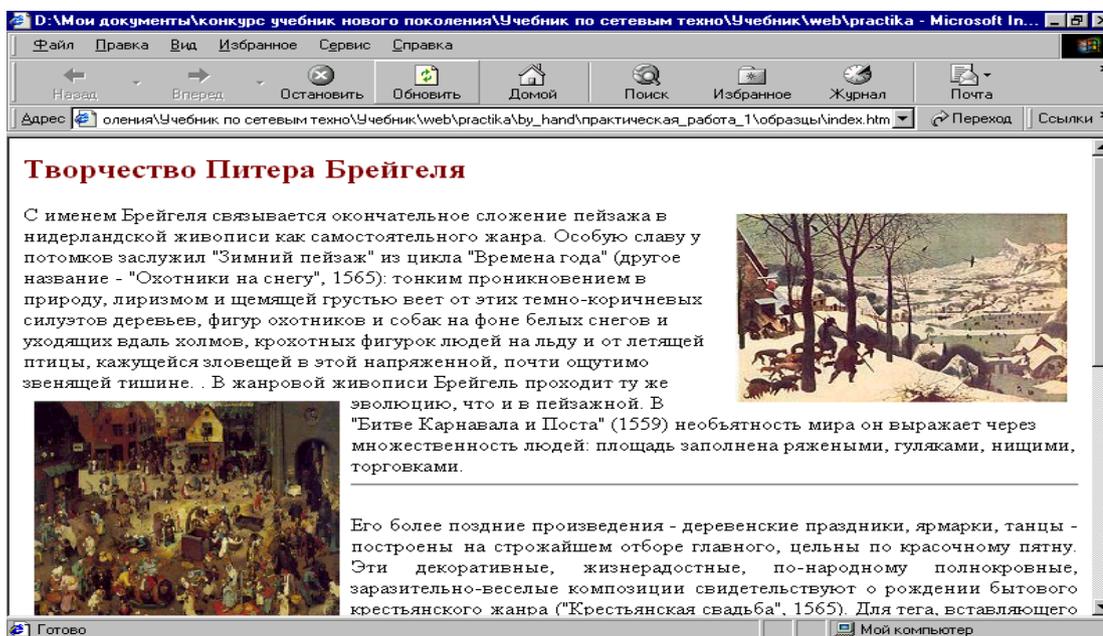


Рисунок 2 – Результат выполнения индивидуального задания 2

## Лабораторная работа №2

**Тема: Теги для вставки звука и видео**

**Цель: Изучение и закрепление знаний по тегам для вставки звука и видео**

### Вставка видео и звука

**<BGSOUND SRC =*имя файла*>** - вставка звукового файла

Например: **<BGSOUND SRC="BLIP.WAV">**

**<IMG DYNSRC =*имя файла*>**- вставка видео файла

**Атрибут тэгов <BGSOUND SRC> и <IMG DYN SRC>**

*n*

**LOOP = INFINITE**

Количество повторений видео или звукового клипа и непрерывное повторение соответственно.

### Индивидуальное задание 1. Вставка звука и видео

Действие:

1. Открыть папку Пример\_2 и запустить index.htm
2. В главном меню Internet Explorer дать команду Вид/В виде HTML.
3. Сопоставить изображение в окне Internet Explorer и текст HTML- документа в блокноте, определить какие теги и атрибуты тегов позволяют вставить звук и видео.
4. Создать свою папку и скопировать содержимое папки Пример\_2
5. Создать в блокноте и сохранить в своей папке HTML-документ, используя звуковые и видео файлы из папки Материалы\_2.

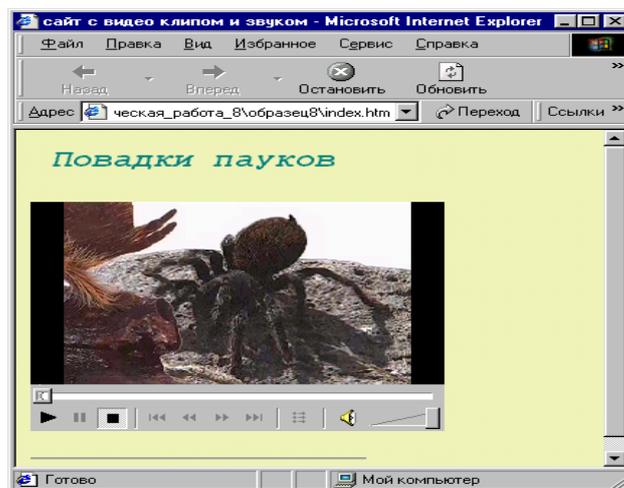


Рисунок 3 – Результат выполнения индивидуального задания 1

## Контрольные вопросы

1. Назовите теги для вставки видео и звука
2. Назовите атрибуты у выше названных тегов

## Лабораторная работа №3

### Тема: Списки

**Цель: Изучение и закрепление знаний по тегам для формирования нумерованных и маркированных списков**

Списки (list) добавлены в HTML, несомненно, под влиянием успеха текстовых редакторов. Список отличается от обычного текста, прежде всего тем, что пользователю не надо думать о нумерации его пунктов: эту задачу берет на себя программа. Если список добавляется новыми пунктами или укорачивается, нумерации корректируется автоматически. В случае нумерованных списков программа ставит перед каждым пунктом маркеры: кружочки, прямоугольники, ромбы или другие изображения. В результате список принимает удобочитаемый, «фирменный вид».

Теги для создания списков можно условно разделить на две группы: одни определяют общий вид списка и позволяют использовать атрибуты, а другие задают его внутреннюю структуру. Существует несколько разновидностей списков: нумерованные (маркированные) и нумерованные.

#### **Теги для нумерованных списков:**

**<UL>** (Unordered List) – начало списка

**</UL>** – конец списка

**<LI>** (List Item) – элемент списка

**Атрибут тэга <UL> для нумерованных списков:**

*DISC*

**TYPE=** *CIRCLE*

*SQUARE*

создаёт маркеры в виде окружностей, пустых квадратов, закрашенных квадратов соответственно.

#### **Теги для нумерованных списков:**

**<OL>** (Ordered List) – начало списка

**</OL>** – конец списка

**<LI>** (List Item) – элемент списка

**TYPE** – есть атрибут тэга **<OL>** для нумерованных списков: нумерация арабскими цифрами, русскими буквами, римскими цифрами

#### Значение атрибута **TYPE**

Атрибут	Последовательность нумерации
type="1"	1, 2, 3, 4, ...
type="i"	i, ii, iii, iv, ...

type="I"	I, II, III, IV, ...
type="a"	a, b, c, d, ...
type="A"	A, B, C, D, ...

**SKIP**=количество пропущенных номеров

Например: <OL SKIP=4> Такая запись означает, что будет пропущено четыре следующих номера.

**START**= номер, с которого будет начата нумерация списка

Например: <OL TYPE=I START=3> Такая запись означает, что нумерация начнется с римской цифры III.

**VALUE**= номер, который будет присвоен данному пункту списка

Например:

<LI VALUE = 13>

<LI VALUE = 25>

Такая запись означает, что после номера 13 в списке будет стоять номер 25.

### Индивидуальное задание 1

Действие:

1. Открыть папку Пример\_3.
2. Запустить index.htm изучить теги для формирования списков
3. В меню Internet Explorer выполнить команду Вид/В виде HTML.
4. Сопоставить изображение в окне Internet Explorer и текст HTML- документа в блокноте, определить какие теги и атрибуты тегов позволяют создать маркированный, нумерованный и многоуровневый списки.
5. Создать свою папку. Создать в блокноте и сохранить в этой папке HTML-документ, используя список из текстового файла папки Материалы\_3.
6. Результат должен быть примерно такой, как на представленном ниже рисунке 4.

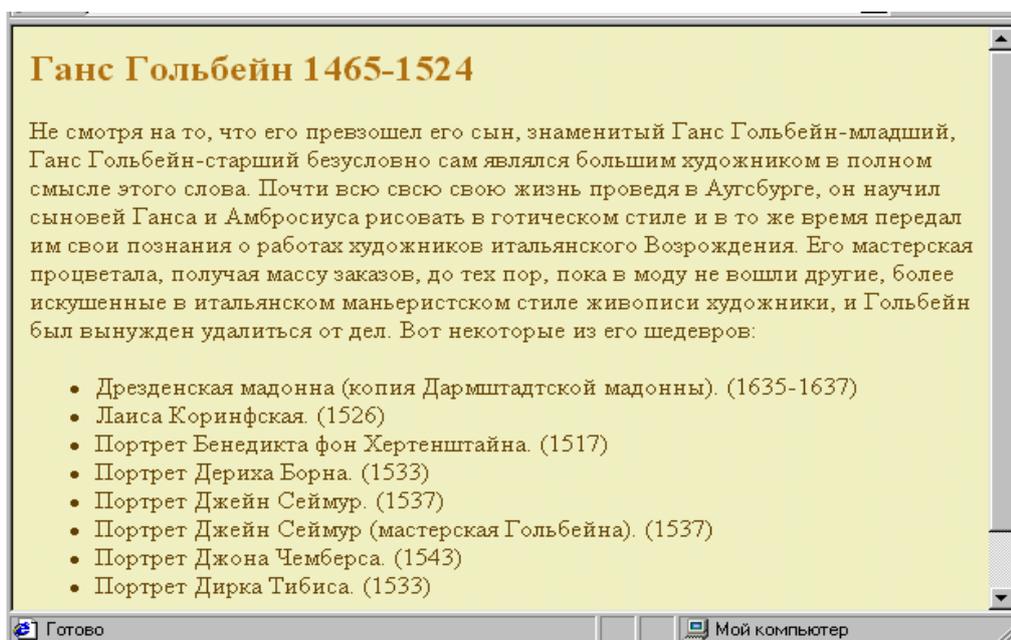


Рисунок 4 – Результат выполнения индивидуального задания 1

### Контрольные вопросы

1. Для чего были добавлены списки в HTML?

2. Какие бывают списки?
3. Теги для определения списка?
4. Атрибуты списка их назначение?
5. Атрибут TYPE для списка?

#### Лабораторная работа №4

**Тема: Таблицы**

**Цель: Изучение и закрепление знаний по тегам для формирования простых и сложных таблиц и научиться создавать HTML-страницы с помощью таблиц**

#### Краткие теоретические сведения

Таблицы в HTML формируются нетрадиционным способом - построчно. Сначала с помощью элемента TR необходимо создать ряд таблицы, в который затем элементом TD помещаются ячейки.

**TABLE** - элемент для создания таблицы. Обязательно должен иметь начальный и конечный тэги.

*Ячейки* внутри таблицы создаются с помощью элементов TR, TD, TH и CAPTION. Параметры: **align** - определяет способ горизонтального выравнивания таблицы, возможные значения: left, center, right. Значение по умолчанию - left; **valign** - определяет способ вертикального выравнивания таблицы.

Возможные значения: top, bottom, middle; **border** - определяет ширину внешней рамки таблицы (в пикселях). При BORDER="0" или при отсутствии этого параметра рамка отображаться не будет; **cellpadding** - определяет расстояние (в пикселях) между рамкой каждой ячейки таблицы и содержащимся в ней материалом; **cellspacing** - определяет расстояние (в пикселях) между границами соседних ячеек; **width** - определяет ширину таблицы.

Ширина задается либо в пикселях, либо в процентном отношении к ширине окна браузера. По умолчанию этот параметр определяется автоматически в зависимости от объема содержащегося в таблице материала; **height** - определяет высоту таблицы. Высота задается либо в пикселях, либо в процентном отношении к высоте окна браузера. По умолчанию этот параметр определяется автоматически в зависимости от объема содержащегося в таблице материала; **bgcolor** - определяет цвет фона ячеек таблицы. Задается либо RGB-значением в шестнадцатеричной системе, либо одним из 16 базовых цветов; **background** - позволяет заполнить фон таблицы рисунком. В качестве значения необходимо указать URL рисунка.

**CAPTION** - задает заголовок таблицы. Содержание заголовка должно состоять только из текста. Использование блочных элементов в этом случае недопустимо. Атрибуты: **align** - определяет способ вертикального выравнивания заголовка таблицы. Возможные значения: **top** - помещает

заголовок над таблицей (значение по умолчанию); **bottom** - помещает заголовок под таблицей.

**TR** - создает новый ряд (строку) ячеек таблицы. Ячейки в ряду создаются с помощью элементов TD и TH. Атрибуты: **align** - определяет способ горизонтального выравнивания содержимого всех ячеек данного ряда. Возможные значения: **left, center, right**; **valign** - определяет способ вертикального выравнивания содержимого всех ячеек данного ряда. Возможные значения: **top, bottom, middle**; **bgcolor** - определяет цвет фона для всех ячеек данного ряда. Задается либо RGB-значением в шестнадцатеричной системе, либо одним из 16 базовых цветов.

**TD/TH** - элемент TD создает ячейку с данными в текущей строке. Элемент TH также создает ячейку, но определяет ее как ячейку-заголовок. Такое разграничение позволяет браузерам оформлять содержимое ячейки-заголовка и ячеек с данными разными шрифтами. В качестве содержимого ячейки можно использовать другие таблицы.

Атрибуты: **align** - определяет способ горизонтального выравнивания содержимого ячейки. Возможные значения: **left, center, right**.

Для TD по умолчанию выполняется выравнивание по левому краю, а для TH - центрирование; **valign** - определяет способ вертикального выравнивания содержимого ячейки. Возможные значения: **top, bottom, middle**. По умолчанию происходит выравнивание по центру, если значение этого параметра не было задано ранее в элементе TR; **width** - определяет ширину ячейки. Ширина задается в пикселях или в процентном отношении к ширине таблицы; **height** - определяет высоту ячейки. Высота задается в пикселях или в процентном отношении к ширине таблицы; **colspan** - определяет количество столбцов, на которые простирается данная ячейка. По умолчанию имеет значение 1; **rowspan** - определяет количество рядов, на которые простирается данная ячейка. По умолчанию имеет значение 1; **nowrap** - блокирует автоматический перенос слов в пределах текущей ячейки; **bgcolor** - определяет цвет фона ячейки. Задается либо RGB-значением в шестнадцатеричной системе; **background** - заполняет ячейку фоновым рисунком, при этом указать URL рисунка.

**Пример:**

```
<html>
<body>
<table border="1" cellpadding="0" cellspacing="0" width="100">
<tr>
<td colspan="2">А</td><td>В</td>
<td rowspan="2">Б</td>
</tr><tr>
<td rowspan="2">В</td><td colspan="2">Г</td></tr><tr>
<td>Д</td><td>Е</td><td>Ж</td></tr><tr>
<td colspan="2">З</td><td colspan="2">И</td>
</tr>
</table>
</body>
</html>
```

А	В	Б	
В	Г		
	Д	Е	Ж
З		И	

Результат работы вышеприведенного примера.

### Теги для формирования таблиц

**<TABLE></TABLE>** – начало и конец таблицы

**<CAPTION></CAPTION>** – начало заголовка и конец заголовка таблицы

заголовок располагается прямо по центру относительно ширины таблицы

**<TH></TH>** (Table Header) – начало заголовков столбцов или строк таблицы и конец заголовков столбцов или строк таблицы

**<TR></TR>** (Table Row) – начало строки таблицы и конец строки таблицы

**<TD></TD>** - начало ячейки таблицы и конец ячейки таблицы

**Атрибуты тэга <TABLE>**

- **WIDTH**=ширина таблицы в пикселях или %

- **BORDER**=ширина границы таблицы в пикселях или %

- **CELLSPACING**= ширина промежутков между ячейками в пикселях или %

- **CELLPADDING**= ширина промежутков между содержимым ячейки и её границами в пикселях или %

**Например:**

```
<TABLE BORDER=3 CELLSPACING=2 CELLPADDING=2 WIDTH="80%">
```

```
<CAPTION> ... заголовок таблицы ... </CAPTION>
```

```
<TR>
```

```
  <TD> первая ячейка таблицы </TD>
```

```
  <TD> вторая ячейка таблицы </TD>
```

```
</TR> ...
```

```
...
```

```
</TABLE>
```

**Атрибуты тэга <TH>, <TR>, <TD>**

Таблица может быть простой или сложной, когда несколько строк или столбцов объединяются

**ROWSPAN** = количество объединяемых строк

**COLSPAN** = количество объединяемых столбцов

**BGCOLOR** = цвет фона заголовка, строки, ячейки

**Пример сложной таблицы**

```
<HTML>
```

```
<BODY background=Puzzle.jpg><br>
```

```
<br><br>
```

```
<TABLE border=5 cellspacing=3>
```

```
<caption><b><font size=4 color=E88CB7>Заголовок таблицы</caption></b>
```

```
  <tr><th ROWSPAN=4 bgcolor=silver><font color=FBDC2F>Заголовок 1
```

```

<tr><TD>Ячейка 1
  <TD>Ячейка 2
  <TD>Ячейка 3
<tr><TD>Ячейка 4
  <TD>Ячейка 5
  <TD>Ячейка 6
</TABLE>
</BODY>
</HTML>

```

### Индивидуальное задание 1 (Таблицы простые и сложные)

Действие:

1. Открыть папку Пример\_4.
2. Запустить index.htm
3. В меню Internet Explorer выполнить команду Вид/В виде HTML.
4. Сопоставить изображение в окне Internet Explorer и текст HTML- документа в блокноте, определить какие теги и атрибуты тегов позволяют создать таблицу.
5. Создать свою папку.
6. Создать в блокноте и сохранить в этой папке HTML-документ, используя табличные данные из файла, который находится в папке Материалы 4
7. Текст перенести из текстового файла в HTML-документ копированием.
8. Цвета подобрать и определить их коды в редакторе Photoshop.
9. Результат должен быть примерно такой, как на рисунке 5

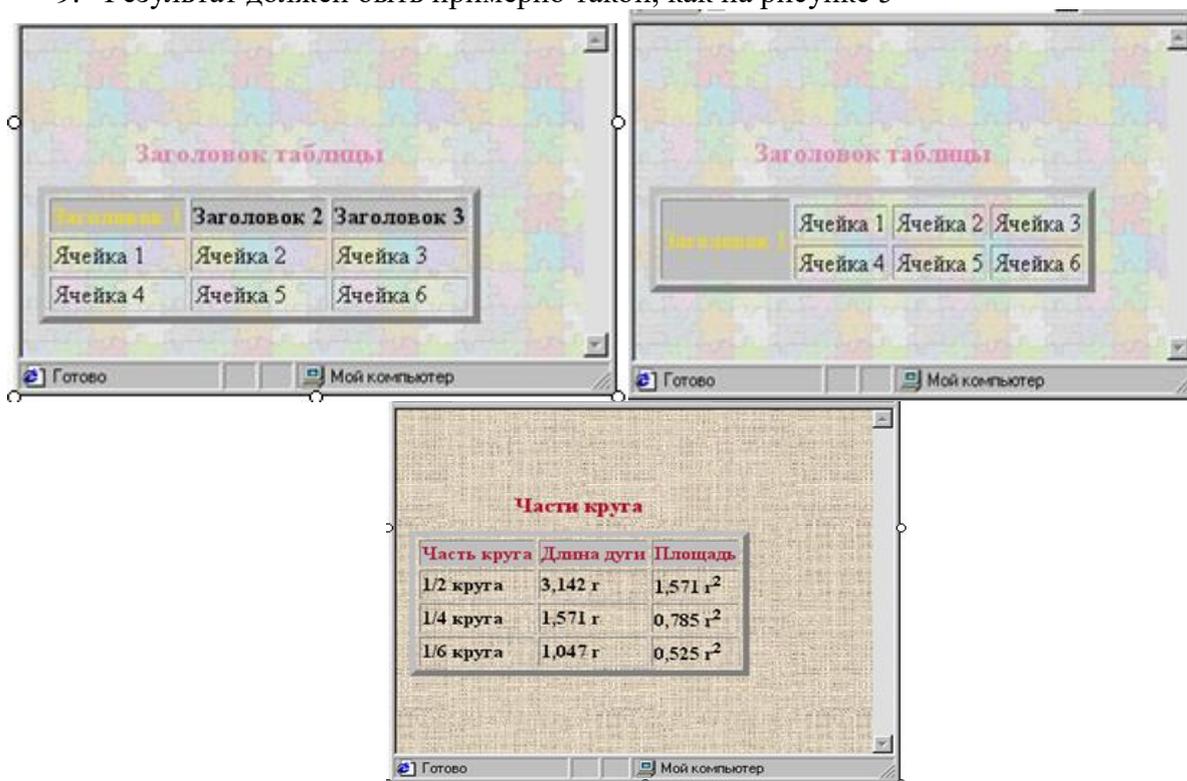


Рисунок 5 – Результаты выполнения индивидуального задания 1

### Контрольные вопросы

1. Теги для формирования таблиц

2. Все атрибуты тега таблицы
3. Что такое сложная таблица
4. Что такое простая таблица
5. Как объединить строки и столбцы

## Лабораторная работа №5

**Тема: Гиперссылки**

**Цель: Изучение основных тегов для организации ссылок в пределах одного документа и ссылки на другие документы**

### Гиперссылки

Тело документа может содержать ссылки на другие документы, текст и другую форматированную информацию.

**Для организации перехода внутри одного HTML-файла следует ввести необходимый текст в двух местах:**

- записать метку в том месте HTML- файла, куда необходимо перейти.
- записать переход на эту метку в том месте, откуда будет выполнен переход.

Гипертекстовая ссылка состоит из двух частей: указателя и адреса (URL-адреса).

### Ссылки на метки

Можно организовать ссылку на другую часть этого же документа, если предварительно расставить в начале таких частей - **метки**:

`<A HREF= #имя метки >активный текст или изображение </A>`

Создает ссылку на метку в текущем HTML-документе:

`<A NAME= имя метки >начало текстового фрагмента </A>`

Можно также создать ссылку на метку в другом документе, если в нем имеется метка:

`<A HREF= имя файла#имя метки >активный текст или изображение </A>`

`<A NAME= "МЕТКА" ></A>`, где МЕТКА – это любое, состоящее из букв английского алфавита и цифр. Для перехода на эту метку запишите: `<A HREF= МЕТКА> Перейти на метку </A>`.

*Например:* Требуется организовать переход с текущего места документа в начало HTML – файла. Переход осуществляется левой кнопки мыши на текст «Вернуться в начало». Для выполнения этого перехода задайте:

- в начале HTML – файла записать: `<A NAME= "A0" ></A>`

- в необходимом для перехода месте HTML– файла записать:

`<A HREF= HTML0 >Вернуться в начало раздела </A>`.

Для перехода из любого места HTML–файла h1.htm в начало HTML – файла h2.htm, в файле h1.htm следует записать: `<A HREF= «h2.htm»> Переход к файлу h2.htm </A>`

Можно перейти на заданную метку МЕТКА другого документа h2.htm с помощью следующей команды: `<A HREF= «h2.htm#МЕТКА»> Переход в начало файла h2.htm </A>`

Для записи гипертекстовой ссылки в HTML – файл используйте следующую команду: `<A HREF= «URL»> Переход по гипертекстовой ссылке </A>`

При нажатии левой кнопкой мыши на текст перехода по гипертекстовой ссылке Вы перейдете на ресурс, заданный в URL. Для возврата используйте клавишу «Back» браузера.

### **Гипертекстовая ссылка с помощью изображения**

Допустим, что Вы выводите на экран изображение IMAGE- файл, который называется, например, picture.gif. Вы хотите организовать страницу так, чтобы при нажатии левой кнопки мыши на изображение перейти на другую домашнюю страницу, например, на [HTTP://WWW.HOME.COM](http://www.home.com), то следует записать следующую команду:

```
<A HREF= «HTTP://WWW.HOME.COM»><IMG SRC=HOME.gif>
```

### **Создание ссылки на почтовый адрес**

Для организации ссылки в домашней страничке на почтовый адрес следует записать в конце первой домашней страницы HTML – файла:

```
<A HREF= mailto:ataev_ruslan@hotmail.com>Webmaster</A>
```

Формат тэга, создающего ссылку на другой документ (HTML-документ или файл с изображением, звуком, видео):

```
<A HREF= имя файла. расширение> активный текст или изображение </A>
```

Если файл находится в другой папке, то нужно указывать путь к нему.

Например:

```
<A HREF= "web/sprav.pps"> активный текст или изображение </A>
```

Если создается ссылка на ресурс Интернета, то нужно указать его URL:

```
<A HREF= URL- адрес> активный текст или изображение </A>
```

Например:

```
<A HREF=http:// upm.ipschool.spb.ru> активный текст или изображение </A>
```

### **Бегущая строка**

<MARQUEE> – начало бегущей строки

</MARQUEE> – конец бегущей строки

Фрагмент текста, картинки или любые другие объекты между этими тэгами будут двигаться так, как определяют атрибуты.

**Атрибуты тэга <MARQUEE>**

**WIDTH** = *ширина области бегущей строки в пикселях или в % от высоты экрана*

**BGCOLOR** = *название цвета или его код*

Определяет цвет фона бегущей строки.

**DIRECTION** = *LEFT*

*RIGHT*

Определяет направление бегущей строки

*n*

**LOOP** = *INFINITE*

Определяет количество "прокруток" бегущей строки: при первом значении атрибута – n, при втором значении атрибута – бесконечно (по умолчанию)

*TOP*

**ALIGN** = *MIDDLE*  
*BOTTOM*

Позволяет разместить текст в верхней, средней или нижней части области бегущей строки.

*SCROLL*  
**BEHAVIOR** = *SLIDE*  
*ALTERNATE*

Определяет режим вывода ("поведения") бегущей строки: прокрутка циклическая, выход и остановка и прыжки соответственно.

**SCROLLAMOUNT** = *скорость перемещения текста или графики в строке [1...3000]*

**HIGHT**=*высота бегущей строки (в пикселях или в % от высоты экрана )*

**HSPACE**= *ширина в пикселях левого и правого полей между областью бегущей строки и окружающим её текстом или графикой*

**VSPACE**= *размер отступа в пикселях сверху и снизу от бегущей строки до текста или графики.*

**Индивидуальное задание 1.** Создание ссылки с текста, с картинки, на другой HTML- документ, на URL-адрес. Формирование бегущей строки

Действие:

1. Открыть папку Пример\_5 и запустить index.htm, изучить код html
2. Открыть этот HTML-документ в блокноте.
3. Создать свою папку и скопировать содержимое папки Материалы \_5 (выбрать один из вариантов).
4. Создать и сохранить в этой папке HTML-документ, используя выбранные файлы, так чтобы он содержал бегущую строку и ссылки на один из сайтов, адрес которого взять из текстового документа сайт.doc
5. Создать ещё один HTML-документ, используя оставшиеся файлы папки Материалы \_5 (из того же варианта).
6. Организовать ссылки с одного HTML-документа на другой.

**Индивидуальное задание 2. Ссылки на метки**

Действие:

1. Открыть папку Пример\_6.
2. Создать HTML-документ, используя файлы из папки Материалы\_6.
3. Запустить index.htm
4. В главном меню Internet Explorer дать команду Вид/В виде HTML.
5. Сопоставить изображение в окне Internet Explorer и текст HTML- документа
6. В блокноте, определить какие теги и атрибуты тегов позволяют создавать ссылки на метки в документе.
7. Создать в папке Мои документы свою папку.
8. Создать в блокноте и сохранить в этой папке HTML-документ, используя файл из папки Материалы\_6 (расставить метки, создать ссылки на метки и предусмотреть ссылку возврата к началу документа).
9. Результат должен быть примерно такой, как на рисунке 6.

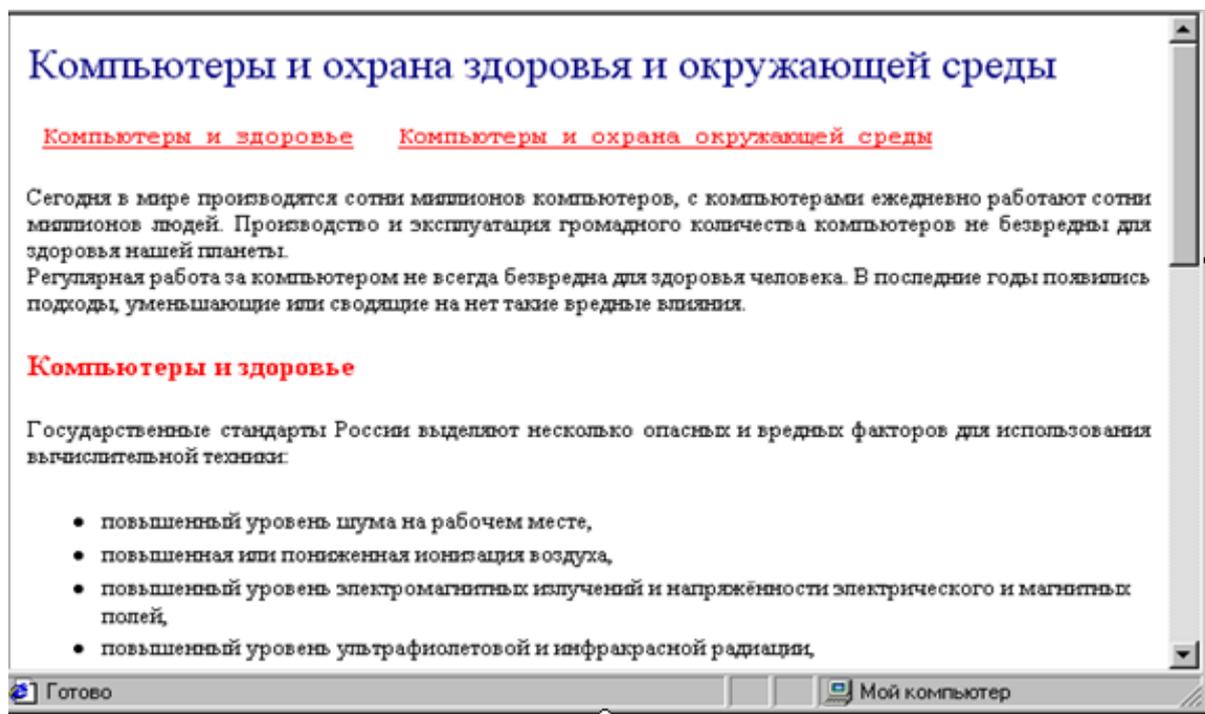


Рисунок 6 – Результаты выполнения индивидуального задания 2

### Контрольные вопросы

1. Что такое гипертекст
2. Как осуществляется ссылка в пределах одного документа
3. Как организовать ссылку на другой документ
4. Как организовать ссылку на почтовый ящик
5. Что такое бегущая строка
6. Назовите тег бегущей строки и его атрибуты

## Лабораторная работа № 6

**Тема: Изучение фреймовой структуры HTML-страницы**

**Цель: Научиться создавать HTML-страницы с помощью фреймов**

### Краткие теоретические сведения

#### Фреймы

**Кадры (frames)** – это независимые части, на которые можно разбить окно браузера и в каждую часть можно загружать отдельную страницу. Эти страницы могут быть связаны между собой ссылками.

Количество частей (кадров) и их размеры, выбор горизонтального или вертикального деления, т.е. **кадровая (фреймовая) структура** задается тэгом:

**<FRAMESET>... </FRAMESET>**- Определяет расположение фреймов на странице.

**Внимание!** Использование тэга **<FRAMESET>** исключает использование тэга **<BODY>**! Следует запомнить, что не все браузеры отображают фреймы корректно!

Определяя структуру, указывают части сверху вниз и слева направо с помощью атрибутов.

**Атрибуты тэга <FRAMESET>**

**ROWS** = *ширина 1 части, ширина 2 части,...* (в пикселях или %)- определяет количество и размеры горизонтальных фреймов.

*Например:* **<FRAMESET ROWS =100,300,100>**;

или в процентах **<FRAMESET ROWS =25%,35%,40%>**

**COLS** = *ширина 1 части, ширина 2 части,...* (в пикселях или %)- определяет количество и размеры вертикальных фреймов.

**Атрибуты тэга <FRAME>**

**<FRAME SRC=имя файла.расширение>** - позволяет задать, какой html-документ, текстовый или графический, будет загружаться в отдельный кадр.

**SRC** - его значение есть имя документа, который будет в этот кадр загружаться.

**NAME** = *имя кадра*- задает имя кадра. Это имя потом используется в качестве значения атрибута **target** в тэге **<A HREF>** для того, чтобы документ, на который указывает ссылка, загружался в нужном кадре.

*Например:*

**<FRAME SRC = my.htm NAME = right>**

Если при этом в другом кадре загружен документ, содержащий тэг:

**<A HREF = 1.jpg target= right >загрузить рисунок </A>** тогда по щелчку на словах "загрузить рисунок" рисунок 1.jpg загрузится в кадр, в котором до этого был загружен документ my.htm.

Имена кадров могут начинаться с цифры или буквы. Для удобства можно указывать в качестве имени кадра слово, указывающее расположение кадра, например: **top, bottom, right, left.**

Еще **атрибуты** тэга <FRAME>:

**scrolling** - определяет наличие полей прокрутки фрейма, возможные значения: yes - отображать, no - не отображать, auto - отображать, если нужно.

**Noresize** - не позволяет мышкой изменять размеры (границы) между фреймами (кадрами)

**frameborder** - определяет наличие рамок у фрейма. Возможные значения: yes/no - отображать/ не отображать.

**NOFRAMES** - определяет, что будет отображать браузер, если он не поддерживает фреймы.

**Пример:** это текст файла, задающего фреймы (frame1.html, frame2.html, frame3.html)

```
<FRAMESET FRAMEBORDER="0" FRAMESPACING="0" BORDER="0"
COLS="250,*">
<FRAME SRC="frame1.html" NAME="page">
<FRAMESET ROWS="150,*">
<FRAME SRC="frame2.html" NAME="menu1" MARGINWIDTH="0">
<FRAME SRC="frame3.html" NAME="menu2" MARGINWIDTH="0">
</FRAMESET>
<NOFRAMES>Ваш браузер не поддерживает фреймы</NOFRAMES>
</FRAMESET>
<BODY></BODY>
```

Фреймовую структуру также можно создать с помощью тэгов для создания таблиц и тэга <IFRAME> ...</IFRAME>

*Например:*

```
<td>
<iframe width=600 height=330 align=center scrolling=none frameborder=0
name="frame" src=1.htm></iframe>
</td>
```

На рисунке 7 представлено окно, в котором фреймовая структура организована именно таким способом.

Соответствующий HTML-код:

```
<HTML>
<BODY link=brown alink=gold vlink=brown>
<table cellpadding=0 cellspacing=0 width=600 height=360>
<tr>
<td colspan=2 width=750 height=20><center><b>Костюмы разных эпох
</b></center>
<tr>
<td width=350 height=10><a href=egypt.htm target="frame"
<font size=5>Древний Египет</font></a>
```

```

<td rowspan=7 >
<iframe width=400 height=330 align=center scrolling=none
frameborder=0 name="frame" src=egypt.htm></iframe>
<tr>
<td width=350 height=10><a href=viking.htm target="frame"
<font size=5>Викинги</font></a>
<tr>
<td width=350 height=10><a href=sredn.htm target="frame"
<font size=5>Средние века</font></a>
<tr>
<td width=350 height=10><a href=XX.htm target="frame"
<font size=5>XX век</font></a>
<tr>
<tr><td width=350 height=20>
<tr><td width=350 height=20>
<tr><td width=350 height=20>
</table>
</BODY>
</HTML>

```

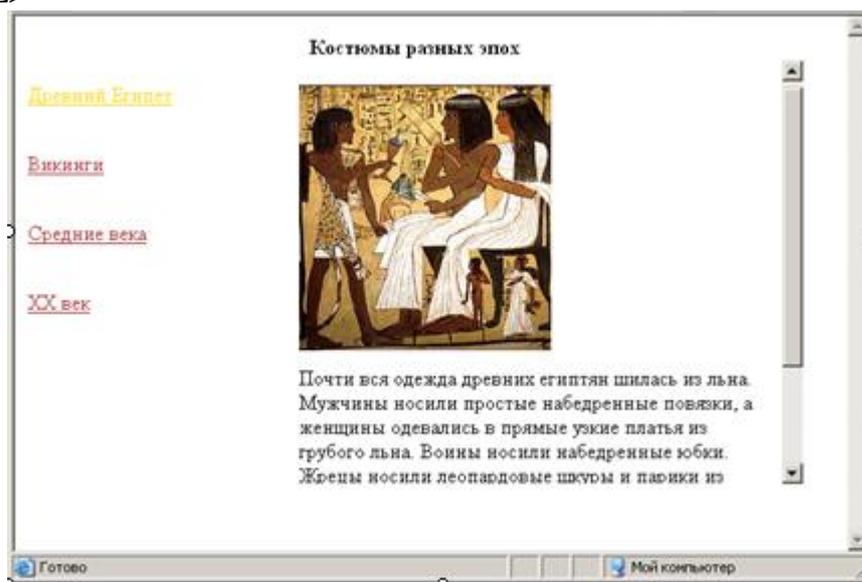


Рисунок 7 – Фреймовая структура, организованная с помощью таблиц и IFRAME

**Индивидуальное задание 1.** Создать страницу ссылок на ресурсы сети, обязательно с использованием фреймов. Разбить всю страницу на 4 фрейма: 2 по горизонтали и 2 по вертикали. В левом верхнем фрейме создать меню, состоящее из следующих ссылок:

- Поисковые системы;
- Каталоги;
- Литература.

В правом верхнем фрейме разместить какой-нибудь логотип, картинку. В левом нижнем фрейме выводить ссылки из левого верхнего фрейма.

- Для поисковой системы – ссылки [www.yandex.ru](http://www.yandex.ru), [www.google.ru](http://www.google.ru), [www.aport.ru](http://www.aport.ru), [www.rambler.ru](http://www.rambler.ru), [www.filesearsh.ru](http://www.filesearsh.ru) оформить в виде нумерованного списка;

- Для каталогов – ссылки [www.list.ru](http://www.list.ru), [www.catalog.ru](http://www.catalog.ru), [www.rambler.ru](http://www.rambler.ru), [www.filesearsh.ru](http://www.filesearsh.ru) оформить в виде нумерованного списка;

- Для литературы – все ссылки оформить как списки с определениями, то есть разложить по рубрикам библиотека ([www.lib.ru](http://www.lib.ru), [www.aldebaran.ru](http://www.aldebaran.ru)), развлечение – игры, музыка, клипы и тд. ([www.games.ru](http://www.games.ru), [www.partypoker.com.ru](http://www.partypoker.com.ru), [www.mp3real.ru](http://www.mp3real.ru), [www.zaycev.net](http://www.zaycev.net), [www.delit.net](http://www.delit.net), [www.mp3.uz](http://www.mp3.uz)), техническая литература ([www.citforum.ru](http://www.citforum.ru), [www.intuit.ru](http://www.intuit.ru), [www.kurs-lab.ru](http://www.kurs-lab.ru)).

Все поисковые системы выводить с их логотипом.

В правом нижнем фрейме отображать непосредственно содержание ссылок из левого нижнего фрейма, то есть загружать сами сайты.

**Пример выполнения 1-го индивидуального задания: изучение возможностей языка HTML – фреймы и списки.**

HTML код:

### **Index.htm**

```
<HTML>
<HEAD>
<META NAME="Content-Language" CONTENT="ru">
<TITLE>Практическое задание №2 - Использование фреймов,
списков</title>
</head>
<FRAMESET cols="25%,*" FRAMEBORDER="2" и BORDER="5">
<FRAMESET rows="20%,*" FRAMEBORDER="3" и BORDER="5">
<frame name="toc" src="Frame1.html" SCROLLING="Auto" NORESIZE>
<frame name="menu2" src="Frame4.html" SCROLLING="Auto" NORESIZE>
</frameset>
<FRAMESET rows="20%,*" FRAMEBORDER="3" и BORDER="5">
<frame name="logo" src="Frame3.html" SCROLLING="Auto" NORESIZE>
<frame name="main" src="Frame2.html" SCROLLING="Yes" NORESIZE>
</frameset></frameset><noframes>
<p>Ваш браузер не поддерживает фреймы, пожалуйста, обновите его.
</p></noframes>
</html>
```

### **Frame1.htm**

```
<HTML>
<HEAD>
<TITLE>Frame1</title>
```

```
</head>
<BODY bgcolor="c0e0e0">
<UL>
<LI><A HREF="Find.html" target="menu2">Поисковые системы</a>
<LI><A HREF="Catalogs.html" target="menu2">Каталоги</a>
<LI><A HREF="Literature.html" target="menu2">Литература</a>
</ul>
</body>
</html>
```

### **Frame2.htm**

```
<HTML>
<HEAD>
<TITLE>Frame2</title>
</head>
<BODY bgcolor="d0e0e0">
<H2><CENTER>Добро пожаловать на сайт</center></h2>
</body>
</html>
```

### **Frame3.htm**

```
<HTML>
<HEAD>
<TITLE>Frame3</title>
</head>
<BODY bgcolor="d0e0e0">
<IMG SRC="Mts2.gif" width="138" height="86" align="right">
</body>
</html>
```

### **Frame4.htm**

```
<HTML>
<HEAD>
<TITLE>Frame4</title>
</head>
<BODY bgcolor="c0e0e0">
</body>
</html>
```

### **Find.htm**

```
<HTML>
<HEAD>
<TITLE>Find</title>
</head>
<BODY bgcolor="d0e0e0">
```

```

<IMG SRC="logo_yandex.gif" width="50">
<A HREF="http://www.yandex.ru" target="main">www.yandex.ru</a><BR>
<IMG SRC="logo Rambler.gif" width="50">
<A HREF="http://www.rambler.ru" target="main">www.rambler.ru</a><BR>
<IMG SRC="logo_aporu.gif" width="50">
<A HREF="http://www.aport.ru" target="main">www.aport.ru</a><BR>
<IMG SRC="logo_google.gif" width="25"><IMG SRC="logo_google2.gif"
width="10">
<IMG SRC="logo_google3.gif" width="15">
<A HREF="http://www.google.ru" target="main">www.google.ru</a><BR>
<IMG SRC="logo_filesearch.gif" width="50">
<A HREF="http://www.filesearch.ru" target="main">www.filesearch.ru</a><BR>
</body>
</html>

```

### Catalogs.htm

```

<HTML>
<HEAD>
<TITLE>Catalogs</title>
</head>
<BODY bgcolor="d0e0e0">
<OL type="1">
<LI><A HREF="http://www.yandex.ru" target="main">www.yandex.ru</a>
<LI><A HREF="http://www.rambler.ru" target="main">www.rambler.ru</a>
<LI><A HREF="http://www.aport.ru" target="main">www.aport.ru</a>
<LI><A HREF="http://www.google.ru" target="main">www.google.ru</a>
<LI><A HREF="http://www.filesearch.ru"
target="main">www.filesearch.ru</a>
</ol>
</body>
</html>

```

### Literature.htm

```

<HTML>
<HEAD>
<TITLE>Literature</title>
</head>
<BODY bgcolor="d0e0e0">
<DL>
<DT><B><BIG>Библиотеки</big></b>
<DD><A HREF="http://www.lib.ru" target="main">www.lib.ru</a>
<DD><A HREF="http://www.aldebaran.ru"
target="main">www.aldebaran.ru</a>
</dt>
<DT><B><BIG>Развлечение - игры, музыка, клипы и тд.</big></b>

```

```

<DD><A HREF="http://www.games.ru" target="main">www.games.ru</a>
<DD><A HREF="http://www.partypoker.com.ru"
target="main">www.partypoker.com.ru</a>
<DD><A HREF="http://www.mp3real.ru" target="main">www.mp3real.ru</a>
<DD><A HREF="http://www.zaycev.net" target="main">www.zaycev.net</a>
<DD><A HREF="http://www.delit.net" target="main">www.delit.net</a>
<DD><A HREF="http://www.mp3.uz" target="main">www.mp3.uz</a>
</dt>
<DT><B><BIG>Техническая литература</big></b>
<DD><A HREF="http://www.citforum.ru" target="main">www.citforum.ru</a>
<DD><A HREF="http://www.intuit.ru" target="main">www.intuit.ru</a>
<DD><A HREF="http://www.kurs-lab.ru" target="main">www.kurs-lab.ru</a>
</dt>
</dl>
</body>
</html>

```

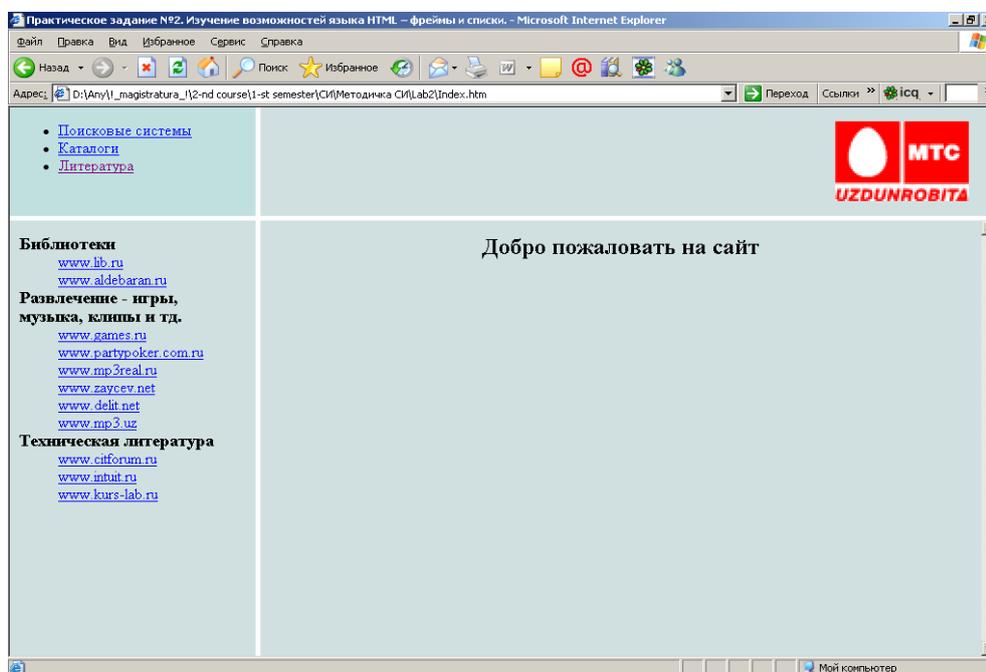


Рисунок 8 – Результат выполнения 2-го индивидуального задания

## Индивидуальное задание 2. Фреймы, ссылки в заданный фрейм

Действие:

1. Открыть папку Пример\_7.
2. Создать HTML-документ, используя файлы из папки Материалы\_7.
3. Запустить index.htm
4. В главном меню Internet Explorer выполнить команду Вид/В виде HTML.
5. Сопоставить изображение в окне Internet Explorer и текст HTML- документа в блокноте, определить какие теги и атрибуты тегов позволяют создавать фреймы и ссылки на картинку, выводимую в определённом фрейме.
6. Создать в папке Мои документы свою папку.
7. Скопировать в эту папку содержимое папки Материалы\_7.

8. Создать в блокноте HTML-документы, используя файлы из папки Материалы\_7 (разбить страницу на фреймы, сделать ссылки из левого фрейма на вновь созданные HTML-документы).
9. Каждый HTML-документ, на который делается ссылка, должен содержать текстовый фрагмент и картинку (для размещения текста и картинки использовать невидимую таблицу и форматирование текста по ширине).
10. Результат должен быть примерно такой, как на рисунке 9.

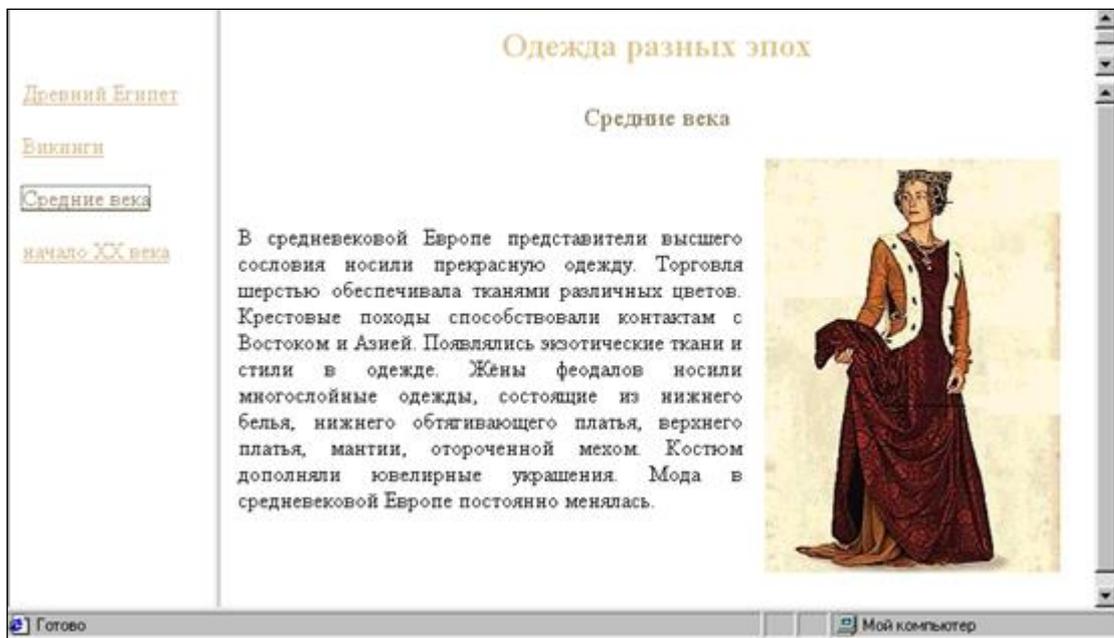


Рисунок 9 – Результат выполнения индивидуального задания 2

### Контрольные вопросы

1. Что такое фрейм
2. Теги для организации фрейма
3. Объясните назначение каждого атрибута
4. Что такое index.html
5. В чем разница: FRAME, FRAMESET, IFRAME

## Лабораторная работа № 7

**Тема: Карты изображений в HTML-документах**

**Цель: Научиться создавать HTML-страницы с использованием карты изображений, организовать ссылки на различные элементы рисунка**

### Краткие теоретические сведения

Можно организовать ссылки с картинки на другие документы, но можно также организовать ссылки с разных частей картинки, если предварительно выделить эти части (области) и определить их координаты. Такую картинку называют **изображение-карта** (map).

**<MAP>... </MAP>** Этот тэг позволяет определить карту. Между открывающим и закрывающим тэгами **<MAP>** определяются области карты при помощи тэгов:

**<AREA>**

Для каждой области карты должен быть создан свой элемент **<AREA>**, который должен включать атрибут, определяющий ссылку **HREF=адрес ссылки**

**Атрибуты тэга <MAP>**

Задаётся имя изображения-карты при помощи атрибута **NAME**.

**NAME=имя**

**Атрибуты тэга <AREA>**

**ALT=название области изображения-карты**

Это атрибут для задания текста, заменяющего изображение-карту, не является обязательным.

**TARGET=имя фрейма**

Определяет имя фрейма, в котором будет отображаться документ, на который делается ссылка с данной области.

Атрибут, определяющий форму области на карте

*rect*

**SHAPE = poly**

*circle*

*point*

**COORDS =x,y,x1,y1**

Описывает координаты прямоугольной области *rect* (значениями атрибута будут координаты левого верхнего угла прямоугольника и правого нижнего, т.е. координаты противоположных его вершин).

**COORDS =x,y,x1,y1 ... xn,yn**

Описывает координаты многоугольной области *poly* (значениями атрибута будут пары координат всех вершин многоугольника).

**COORDS =x,y,r**

Описывает координаты области, имеющей вид окружности *circle* (значениями атрибута будут координаты центра окружности и значение радиуса).

**COORDS =x,y**

Описывает координаты области, имеющей вид точки (значениями атрибута будут её координаты).

Для того, чтобы записать координаты выделенной области в качестве значения атрибута COORDS, нужно их определить:

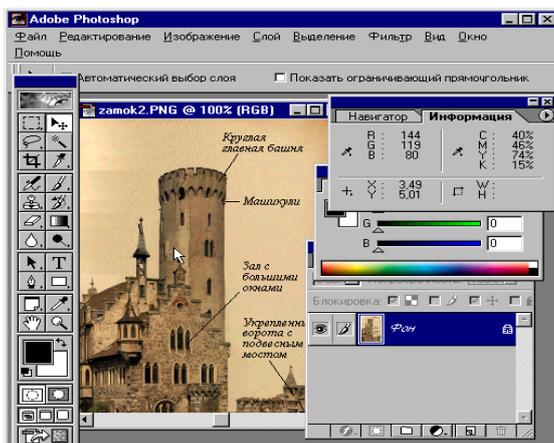
-загрузить картинку (которая станет изображением-картой) в графический редактор;

-выделить с помощью соответствующего инструмента область (прямоугольную, окружность, или область произвольной формы);

-в окне **Информация** нужно щёлкнуть в левом нижнем углу и выбрать Пиксели в качестве единицы измерения;

-выделив, например, прямоугольную область на рисунке и подведя указатель мыши к левому верхнему углу, в окне Информация рядом с обозначениями X и Y можно увидеть и записать координаты этой вершины.

-переместить указатель мыши в правый нижний угол и записать координаты противоположной вершины прямоугольника. Эти пары координат и будут значениями атрибута COORDS.



### Атрибуты тэга <IMG> для изображения-карты

**USEMAP = #имя**

Определяет имя карты. Это имя должно совпадать с именем, которое указано в качестве значения атрибута NAME тэга <MAP>.

*Например:*

```
<HTML>
```

```
<MAP NAME="karta">
```

```
<AREA alt=область1 SHAPE=rect COORDS="77,75,128,130" HREF=1.txt  
TARGET="OKNO">
```

```
<AREA SHAPE=poly COORDS="113,206,164,134,218,209" HREF=2.txt  
TARGET=OKNO>
```

```
<AREA SHAPE=circle COORDS="57,209,50" HREF=3.txt TARGET=OKNO>
```

```
</MAP>
```

```
<A HREF="karta.map">
```

```
<IMG SRC="1.jpg" USEMAP="#karta" alt="области">
```

```
</A>
```

```
</HTML>
```

## Индивидуальное задание 1. Изображение-карта

Действие:

1. Открыть папку Пример\_7 и запустить index.htm
2. В главном меню Internet Explorer выполнить команду Вид/В виде HTML.
3. Сопоставить изображение в окне Internet Explorer и текст HTML- документа в блокноте, определить какие теги и атрибуты тегов позволяют разделить страницу на кадры (фреймы).
4. Щёлкнуть на правом фрейме правой кнопкой мыши и выбрать из контекстного меню команду Просмотр в виде HTML.
5. Определить какие теги и атрибуты тегов позволяют создать изображение-карту.
6. Создать в папке Мои документы свою папку.
7. Создать в блокноте HTML-документы, используя файлы из папки Материалы\_7 (1-ый HTML-документ должен разделить страницу на кадры, во 2-ой HTML-документ вставить картинку с изображением-картой; создать также HTML-документы, которые будут загружаться в левом фрейме при щелчке на соответствующих областях изображения -карты.)
8. Для определения координат областей на изображении-карте воспользоваться графическим редактором Photoshop (фильтром MapTag).
9. Результат должен быть примерно такой, как на рисунке

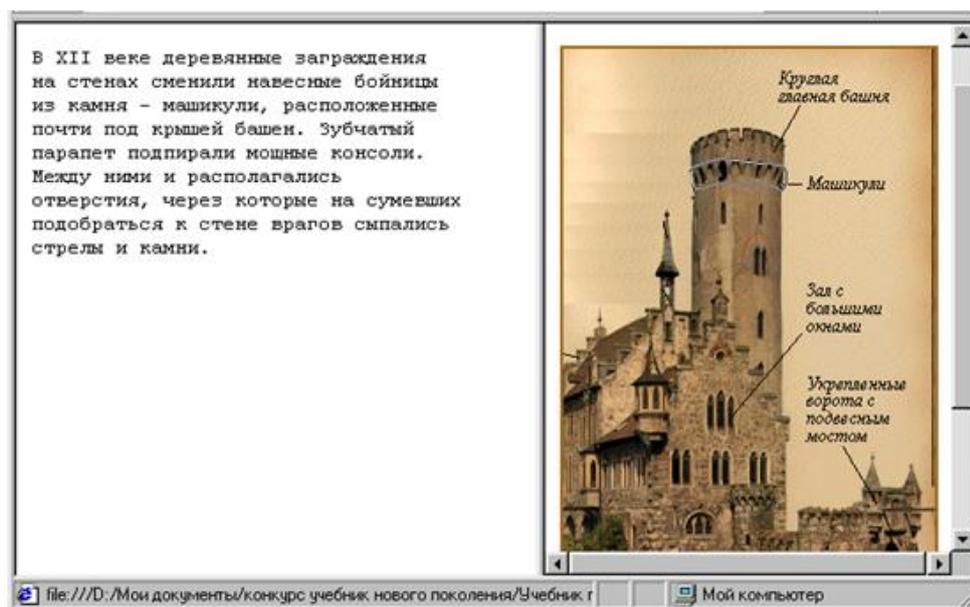


Рисунок 10 – Результат выполнения индивидуального задания

### Контрольные вопросы:

1. Перечислить все теги для изображения–карты и действия их атрибутов
2. Что означает TARGET="OKNO"
3. Что определяет тег **USEMAP** = #имя
4. Объясните назначение каждого атрибута

## Лабораторная работа № 8

**Тема: Изучение тегов для организации форм**

**Цель: Научиться моделировать формы для веб-страниц**

### Краткие теоретические сведения

#### ФОРМЫ

**Форма** – это набор таких знакомых нам по диалоговым окнам элементов, как поля ввода, поля выбора, переключатели. Форма – это то, что позволяет создавать интерактивные страницы, т.е. организовывать диалог с пользователем, создавать заполняемые анкеты, опросники и т.д.

**<FORM>...</FORM>** Этот тэг используется для создания заполняемых форм. Между **<FORM>** и **</FORM>** можно использовать тэги для создания полей нескольких типов: текстовые поля в одну или несколько строк, группы радиокнопок - их еще называют переключателями, окошки, в которые можно устанавливать флажки, меню, командные кнопки (см. рисунок 11).

Форма заявки на участие в конференции

Имя	<input type="text"/>
Отчество	<input type="text"/>
Фамилия	<input type="text"/>
Контактный телефон	<input type="text"/>
Пол	<input type="radio"/> М <input type="radio"/> Ж
Секции, в которых примите участие	<input type="checkbox"/> 1 <input type="checkbox"/> 2 <input type="checkbox"/> 3 <input type="checkbox"/> 4
Город	<input type="text" value="Москва"/>
Тема доклада и аннотация	<input type="text" value="Москва"/>
Тема:	<input type="text"/>
Вы можете отправить доклад	<input type="text"/> <input type="button" value="Обзор..."/>
<input type="button" value="Послать"/> <input type="button" value="Сброс"/>	

Рисунок 11 – Пример формы, содержащей разные поля

#### Атрибуты тэга <FORM>

**ACTION=адрес**- обязательный атрибут, он определяет URL, по которому будет отправлено содержимое формы. Это может быть либо адрес электронной почты, либо путь к скрипту сервера, обслуживающему данную форму. Например, **ACTION =mailto:имя@сервер.домен** либо для запуска на сервере посредством протокола HTTP специальной программы, обслуживающей данную форму. Например: **ACTION = "http://www.prim.com/cgi- bin/reg.pl"**

**METHOD**- определяет, какой из HTTP методов будет использоваться для пересылки серверу (адрес которого задан предыдущим атрибутом) содержимого текущей формы. Это может быть либо GET, либо POST (по умолчанию используется GET).

**ENCTYPE** - этот атрибут определяет механизм, который следует использовать при кодировании содержания данной формы. По умолчанию используется application/x-www-form-urlencoded.

**TYPE=password** - в данном случае вводимые символы отображаются на экране в виде условных значков, таких как \*. Делается это, чтобы скрыть текст от глаз при вводе паролей с клавиатуры.

**TYPE=checkbox** - этот тип используется для ввода в заполняемую форму простых значений булевого типа ("да"/"нет"). *Например:*

```
<input type=checkbox name=section value=yes> 1
<input type=checkbox name=section value=yes> 2
<input type=checkbox name=section value=yes> 3
<input type=checkbox name=section value=yes> 4
```

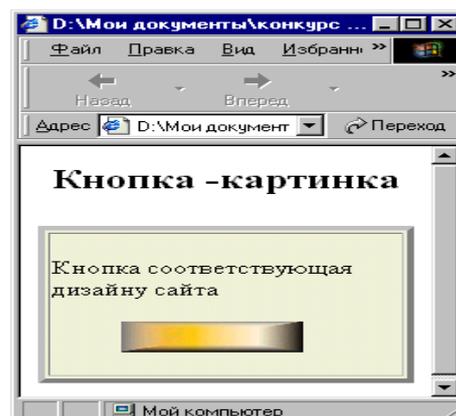
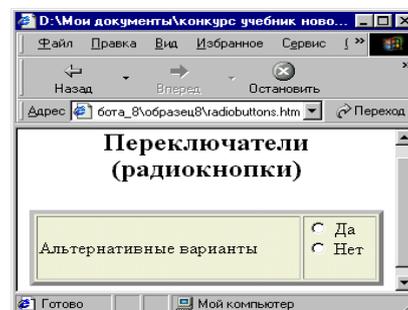
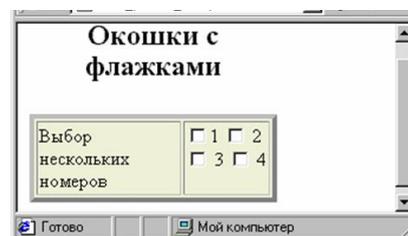
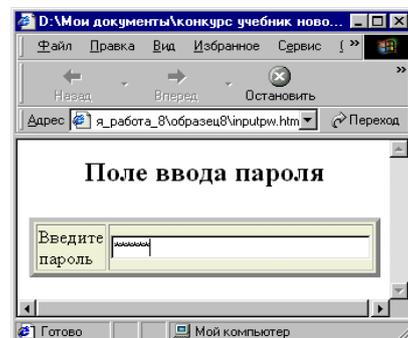
**TYPE=radio** - используется для ввода в форму параметра, являющегося результатом однозначного выбора из набора альтернативных вариантов. Этим вариантам при разметке ставится в соответствие группа "радиокнопок", в каждую из которых должен быть записан один и тот же атрибут NAME. В радиокнопках обязательно следует указывать также и атрибут VALUE. При заполнении формы среди группы радиокнопок только активированная генерирует пару "название/значение" в соответствующем поле данных. В каждой группе радиокнопок одна должна быть изначально активирована посредством атрибута checked.

*Например, альтернативные варианты:*

```
<INPUT type="radio" name=answer value=Yes> Да<br>
<INPUT type="radio" name=answer value=No> Нет
```

**TYPE=image** - этот атрибут создает графический вариант для кнопок, инициирующих передачу данных.

URL для соответствующего изображения задается атрибутом src. Выравнивание картинки осуществляется согласно значению атрибута align. В этом отношении графические изображения кнопок подобны элементам IMG,

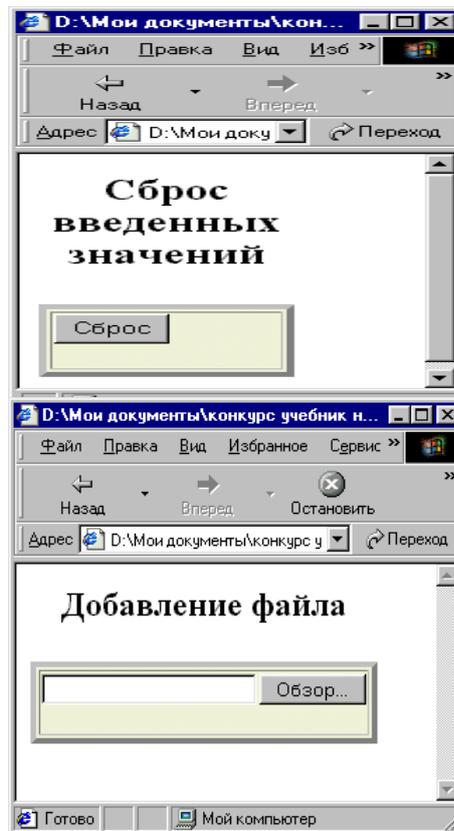


их можно точно так же выравнивать по правому, верхнему, нижнему краю, либо ставить их по центру.

Например, `<INPUT align="middle" name=Ok type=image src=gradient.jpg >`

**TYPE=reset** - этот атрибут позволяет создать кнопку, на которой можно щелкнуть, чтобы вернуть все поля формы в исходное состояние (как было в момент загрузки документа). Информация о кнопке перезагрузки никогда не включаются в набор сведений, пересылаемых на сервер после заполнения формы. Например, `<input type=reset>`

**TYPE=file** - дает пользователям возможность дополнить содержимое текущей формы файлом. При разметке этого элемента обычно создается поле для ввода текста, к которому прилагается кнопка. Щелчок на этой кнопке приводит к раскрытию нового окошка, где можно просмотреть имеющиеся файлы, и выбрать один из них. Имя файла можно также ввести непосредственно в исходном текстовом поле. Точно так же, как и в случае `type=text`, можно использовать здесь атрибут `size`, чтобы выбрать ширину данного поля формы (единицей измерения здесь служит средняя ширина символов). Можно также установить верхний предел для длины вводимого имени файла посредством атрибута `maxlength`.



Например, `<input type=file name=photo size=40>`

### Создание в формах меню

#### **<SELECT>...</SELECT>**

Тэг `<SELECT` создает в заполняемой форме меню типа "выбор одного пункта из многих", либо "несколько пунктов из многих". Между открывающим и закрывающим тэгами `SELECT` должен быть один или несколько элементов `OPTION`, описывающих отдельные пункты меню. Меню типа "один из многих" обычно реализуется как выпадающее меню, в то время как меню типа "несколько из многих" обычно предстает в виде списка с окошками, в которые можно устанавливать флажки против каждого пункта.

#### **Атрибуты тэга <SELECT>**

**NAME=имя** - Сообщает название для данного качества, которое затем будет использоваться во время передачи данных на сервер, чтобы указать, какие пункты в меню выбраны. Каждому пункту меню соответствует пара значений "название/величина".

**SIZE=количество** - В меню типа "несколько из многих" устанавливает количество одновременно видимых пунктов.

**MULTIPLE**- Этот атрибут указывает, что в данном меню пользователи могут сразу выбрать несколько пунктов. По умолчанию - только один пункт.

Атрибуты тэга **<OPTION>**

**SELECTED**- Если для тэга **<OPTION>** указан атрибут **selected**, то соответствующий этому элементу пункт меню уже при загрузке документа изначально помечается как выбранный. Однако если в меню типа "один из нескольких" изначально таким образом помечено более одного пункта, то это будет ошибкой.

**VALUE=значение** - Задает значение, которое соответствует данному пункту меню. В последнем случае это значение будет объединено с названием

*Например:*

```
<SELECT NAME=town>  
<OPTION VALUE=a> Москва  
<OPTION VALUE=b> Санкт-Петербург  
<OPTION VALUE=c> Архангельск  
<OPTION VALUE=d> Астрахань  
</SELECT>
```

### Создание в формах текстовых полей

**<TEXTAREA>...</TEXTAREA>** Задает поля для ввода нескольких строк текста. Обычно это поле содержит текст инициализации, который при загрузке документа изначально будет записываться в данное поле.

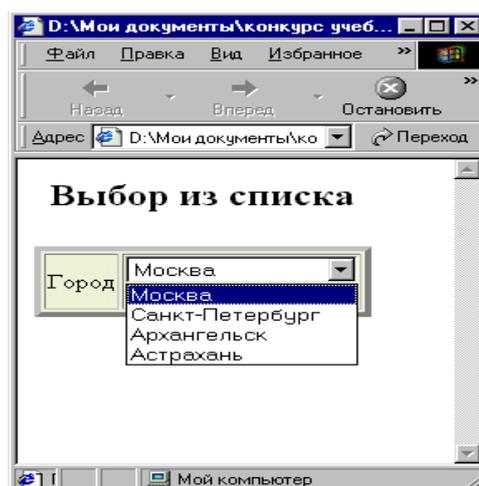
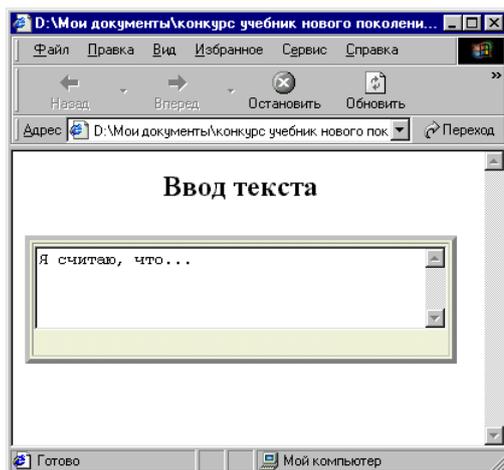
Атрибуты тэга **<TEXTAREA>**

**NAME=имя** - Определяет название, которое будет использовано для идентификации данного поля **textarea** при предоставлении заполненной формы на сервер.

**ROWS=количество**- Задает количество строк текста, видимых на экране.

**COLS=количество**- Определяет ширину создаваемого текстового поля (единицей измерения служит средняя ширина символов).

*Например:* **<TEXTAREA NAME=address ROWS=4 COLS=40>**Я считаю, что...  
**</TEXTAREA>**



В таблице 1 собраны и представлены все теги для создания форм

**Таблица 1 –Теги создания форм**

<code>&lt;form&gt;&lt;/form&gt;</code>	Создает формы
<code>&lt;select multiple name="NAME" size=?&gt;&lt;/select&gt;</code>	Создает скролируемое меню. Size устанавливает кол-во пунктов меню, которое будет показано на экране, остальные будут доступны при использовании прокрутки.
<code>&lt;option&gt;</code>	Указывает каждый отдельный элемент меню
<code>&lt;select name="NAME"&gt;&lt;/select&gt;</code>	Создает ниспадающее меню
<code>&lt;option&gt;</code>	Указывает каждый отдельный элемент меню
<code>&lt;textarea name="NAME" cols=40 rows=8&gt;&lt;/textarea&gt;</code>	Создает окно для ввода текста. Columns указывает ширину окна; rows указывает его высоту.
<code>&lt;input type="checkbox" name="NAME"&gt;</code>	Создает checkbox. За тегом следует текст.
<code>&lt;input type="radio" name="NAME" value="x"&gt;</code>	Создает radio кнопку. За тегом следует текст.
<code>&lt;input type="text" name="foo" size=20&gt;</code>	Создает строку для ввода текста. Параметром Size указывается длина в символах.
<code>&lt;input type="submit" value="NAME"&gt;</code>	Создает кнопку "Принять"
<code>&lt;input type="image" border=0 name="NAME" src="name.gif"&gt;</code>	Создает кнопку "Принять" - для этого используется изображение
<code>&lt;input type="reset"&gt;</code>	Создает кнопку "Отмена"

### Контрольные вопросы:

1. Что такое Форма

2. Перечислите все теги для создания формы
3. Перечислите название атрибутов формы и их действия
4. Как работает тег SELECT
5. Как работает тег TEXTAREA

## Лабораторная работа № 9

**Тема: Каскадные таблицы стилей CSS. Шрифты и цветовая гамма.**

**Цель: Научить применять свойства шрифта и использовать цветовую гамму с помощью CSS.**

### Краткие теоретические сведения

#### Применение CSS

Каскадные таблицы стилей или CSS (Cascading Style Sheets) были революцией для WWW. Если до этого Web-дизайнер не знал, как будет выглядеть его творение в разных программах Web-обозревателей, то теперь он может контролировать все: от начертания шрифта до положения картинки на странице.

Предположим, нужно изменить цвет текста в HTML-документе с черного на синий. Для этого помещаете текст в пару тегов <FONT> и </FONT> следующего вида: <P><FONT color=blue >Это синий текст</FONT></P>.

А теперь представим, что вы внесли определение внешнего вида текста в другое место документа: P.bluetext {color: blue }

Эта строка обозначает, что мы определили для текста, находящегося внутри тега <P> и помеченного стилевым классом bluetext, синий цвет шрифта. Такая конструкция HTML **называется определением стиля или просто стилем.**

В результате в HTML-тексте у нас останутся только теги логического форматирования текста:

```
<P class="bluetext">Это синий текст</P>
```

Здесь мы поместили нужный текст с помощью атрибута **class**, присвоив ему значение bluetext. Атрибут **class** задает имя стилевого класса для тега, и его поддерживают все теги.

Можно переназначить цвета текста для всех тегов <P>. В этом случае не задаем имя стилевого класса: P {color: blue}.

Или можно задать форматирование для стилевого класса, не привязанного ни к какому тегу: **.bluetext {color: blue }**

И теперь можно присваивать стилевой класс тексту, заключенному в любые теги:

```
<H1 class="bluetext">Это синий цвет</H1>
```

```
<CENTER class="bluetext">Это синий цвет</CENTER>
```

```
Это <B class="bluetext">жирный синий</B> текст
```

Можно дать специальное форматирование тегу только в том случае, если он заключен внутрь другого тега:

```
H7 B {color: blue}
```

И теперь:

```
<H7><B>Этот</B> текст будет синим</H7>
```

```
<P>А <B>этот</B> - не будет!</P>
```

Более того, можно встроить определение стиля прямо в тег:

```
<P style="color: blue">Это синий текст</P>
```

Это достигается при помощи атрибута **style**, который также поддерживают все теги HTML.

И еще один способ привязать стиль к какому-либо тегу - использовать атрибут **id**, задающий уникальное имя элемента HTML.

```
#header of document { font-size: 20pt }
```

Здесь задан размер шрифта 20 пунктов.

```
<H1 id="header of document">Это заголовок документа</H1>
```

Можно задавать несколько атрибутов в определении стиля. В этом случае они разделяются точкой с запятой:

```
P {color: blue; font-size: 9pt; text-align: center }
```

Определение стилей, вынесенные в заголовок HTML-документа, составляют таблицу стилей. Таблица стилей заключается в теги `<STYLE>` и `</STYLE>`:

```
<Style [type="text/css"]>
```

```
...
```

```
</STYLE>
```

Тег `<STYLE>` может содержать необязательный атрибут **type**, содержащий обязательное значение **text/css**.

Таблицу стилей можно вынести в отдельный файл и использовать сразу в нескольких документах. В этом случае в заголовке HTML-документа необходимо разместить тег `<LINK>`, указывающий на эту таблицу стилей:

```
<LINK rel="stylesheet" href="{Адрес файла таблицы стилей}">
```

### Свойства шрифта

**font** - задает параметры шрифта элемента страницы.

Заменяет атрибуты **font-family**, **font-height**, **font-size**, **font-style**, **font-variant** и **font-weight**. Значения этих атрибутов могут располагаться в любом порядке.

```
font: {font-family} [{font-height}] [{font-size}] [{font-style}] [{font-variant}]  
[font-weight];
```

Значение по умолчанию - `normal "Times New Roman"`.

Альтернативный формат:

```
font: caption|icon|menu|message-box|small-caption|status-bar;
```

В этом случае доступны шесть предопределенных значений, задающие один из стандартных шрифтов, используемых в элементах интерфейса Windows:

**font: caption;** - шрифт заголовка кнопок, текстовых меток и т.п.;

**font: icon;** - шрифт подписей под пиктограммами;

**font: menu;** - шрифт пунктов меню;

**font: message-box;** - шрифт содержимого стандартных окон-предупреждений;

**font: small-caption;** - мелкий шрифт заголовков;

**font: status-bar;** - шрифт содержимого строки состояния.

Поддерживается IE начиная с 4.0

font-family - указывает имя или семейство шрифта.

font-family: {Имя шрифта}|serif|san-serif|fantasy|monospace;

В качестве значения этого атрибута задается либо непосредственно имя нужного шрифта, либо одно из пяти predetermined значений, задающих имя шрифтового семейства. Можно задавать одновременно несколько шрифтов, разделив их имена запятыми; в этом случае Web-обозреватель сможет выбрать из них тот, который установлен на компьютере клиента. Если имя шрифта содержит пробелы, его следует взять в кавычки.

font-family: "Times New Roman",sans-serif;

Поддерживается IE.

font-weight - задает "жирность" шрифта.

font-weight: normal|bold|bolder|lighter|100..900;

"Жирность" может быть задана следующими способами. font-weight: normal; - обычный;

font-weight: lighter; - светлее;

font-weight: bold; - жирный;

font-weight: bolder; - жирнее;

font-weight: от 100 до 900 - любое значение, кратное 100

Значение по умолчанию normal.

Поддерживается IE начиная.

font-size - задает размер шрифта.

Возможно задание либо абсолютного размера шрифта в одной из поддерживаемых CSS единиц измерения, либо как процент от размера шрифта родителя. Также доступны девять определенных значений.

font-size: 200% - относительная величина (проценты)

font-size: 150px - размер в пикселях

font-size: 300pt - размер в пунктах

font-size: {xx-small,small,medium,large,x-large,xx-large} - задают один из семи размеров шрифтов, поддерживаемых HTML

font-style - задает начертание шрифта.

font-style: normal|italic|oblique;

font-style: normal; - задает обычный вид шрифта (используется по умолчанию);

font-style: italic - курсивное начертание;

## Цветовая гамма

color - Определяет цвет элемента.

color: {Цвет};

Поддерживается IE.

background - задает все свойства фона элемента страницы в один прием. Заменяет собой атрибуты background-attachment, background-color, background-image, background-position и background-repeat.

background: [{background-color}] [{background-image}] [{background-repeat}] [{background-attachment}] [{background-position}];

Значения этих свойств могут располагаться в любом порядке.

Поддерживается IE.

background-color - задает фоновый цвет Web-страницы или ее элемента.

background-color: {Цвет}|transparent;

scrollbar-track-color - задает цвет рабочей части полосы прокрутки, т.е. той ее части, по которой перемещается бегунок.

scrollbar-track-color: {Цвет};

**Индивидуальное задание 1.** Создать HTML-страницу с использованием CSS.

Определить для тега <P> синий цвет шрифта, семейство шрифта, размер шрифта (на своё усмотрение).

Определить универсальный стиль, в котором переопределить цвет текста, размер шрифта, семейство шрифта (на своё усмотрение).

Переопределите тег <body>: цвет фона, размер шрифта, цвет шрифта, семейство шрифта (на своё усмотрение).

**Пример выполнения практического задания: Каскадные таблицы стилей CSS. Использование свойств шрифта и цветовой гаммы.**

HTML код:

**Index.htm**

```
<HTML>
```

```
<HEAD>
```

```
<LINK rel="stylesheet" type="text/css" href="main.css">
```

```
<TITLE>Практическое задание.CSS. Использование свойств шрифта и  
цветовой гаммы.</title></head>
```

```
<BODY>
```

```
<TABLE BORDER="1" WIDTH="95%" HEIGHT="80%"  
ALIGN="CENTER" VALIGN="CENTER">
```

```
<CAPTION><H1>Практическое задание</h1></caption>
```

```
<TR>
```

```
<TD colspan="2">
```

```
Переопределённый тег &ltBODY&gt;<BR>
```

```
BODY {COLOR: black; background-color: c5c5c5; font-family: sans-  
serif; font-size: 200%;}
```

```
</td>
```

```
</tr>
```

```
<TR>
```

```
<TD WIDTH="50%">
```

После переопределения тега &ltP&gt в P {COLOR: blue; font-family: Courier; font-size: 20px;} текст выглядит следующим образом:

```
<P>Переопределённый текст</p>
```

```

</td>
<TD WIDTH="50% ">
А это универсальный стиль .universal {COLOR: purple; font-size: 30px; font-
family: Comic;}
<H1 class="universal">Вот так выглядит текст переопределённый
универсальным стилем</h1>
</td>
</tr>
</table><BR>
</body></html>
main.css
P {COLOR: blue; font-family: Courier; font-size: 20px;}
.universal {COLOR: purple; font-size: 30px; font-family: Comic;}
BODY {COLOR: black; background-color: c5c5c5; font-family: sans-serif;
font-size: 200%;}

```

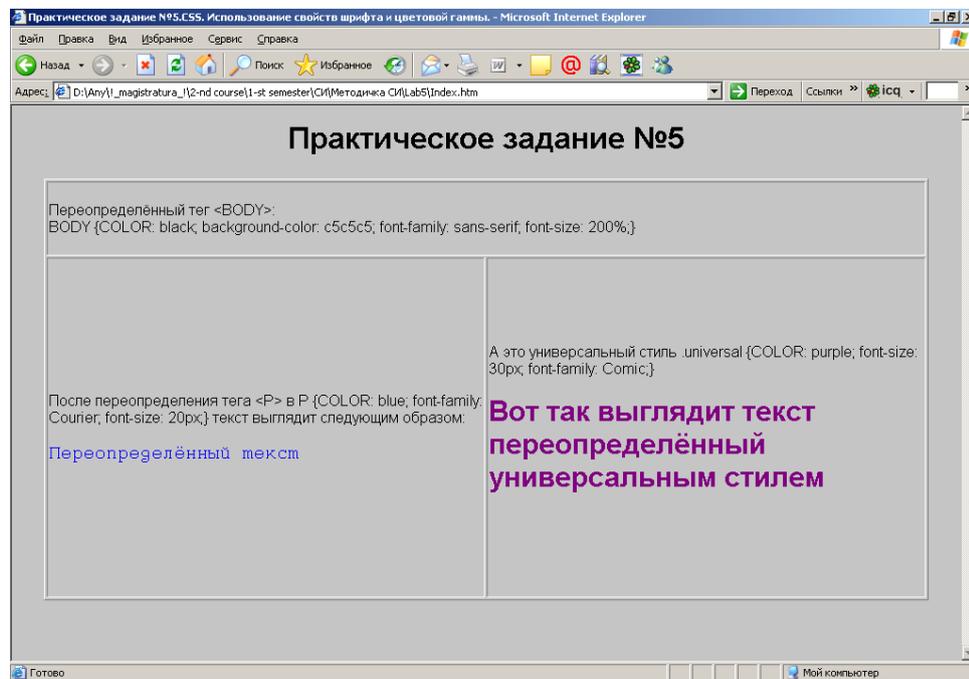


Рисунок 5 – Результат выполнения индивидуального задания

## Лабораторная работа №10

**Тема: Каскадные таблицы стилей CSS. Использование свойств текста и псевдостилей гиперссылок**

**Цель: Научить применять свойства текста и использовать псевдостили гипертекста**

### Краткие теоретические сведения

#### Свойства текста

`text-decoration` - задает специальное оформление текста: подчеркнутый, зачеркнутый и т.п.

`text-decoration: none|underline|overline|line-through|blink;`

`text-decoration: none;` - отменяет любое специальное оформление (значение по умолчанию для большинства тегов);

`text-decoration: underline;` - подчеркивает текст (значение по умолчанию для тегов `<A>`, `<INS>` и `<U>`);

`text-decoration: overline;` - черта сверху текста;

`text-decoration: line-through;` - зачеркивает текст (значение по умолчанию для тегов `<DEL>`, `<S>` и `<STRIKE>`);

`text-decoration: blink;` - мигание текста (отсутствует в IE)

`text-decoration: uppercase` - все буквы становятся заглавными;

`text-decoration: lowercase` - все буквы становятся маленькими (прописными);

`text-decoration: capitalize` - каждое слово в строке начинается с большой буквы.

Поддерживается IE

`text-underline-position` - задает местонахождение линии подчеркивания: выше или ниже текста. Имеет смысл, если атрибут `text-decoration` равен `underline` или `overline`.

`text-underline-position: below|above;`

`text-underline-position: below;` - помещает линию подчеркивания под текстом (значение по умолчанию);

`text-underline-position: above;` - над текстом ("надчеркивание").

Поддерживается IE.

`text-align` - определяет горизонтальное выравнивание текста.

`text-align: left|right|center|justify;`

`text-align: left;` - выравнивание текста по левому краю (значение по умолчанию);

`text-align: right;` - выравнивание по правому краю;

`text-align: center;` - выравнивание по центру;

`text-align: justify;` - выравнивание по обоим краям (по ширине).

Поддерживается IE

`text-align-last` - задает горизонтальное выравнивание последней строки абзаца.

text-align-last: auto|inherit|left|right|center|justify;

text-align: auto; - выравнивает последнюю строку абзаца так же, как и остальные строки (основываясь на значении атрибута text-align) (значение по умолчанию);

text-align: inherit; - так же, как выровнен текст родителя;

text-align: left; - выравнивание текста по левому краю;

text-align: right; - выравнивание по правому краю;

text-align: center; - выравнивание по центру;

text-align: justify; - выравнивание по обоим краям (по ширине).

Поддерживается IE начиная с 5.5

text-indent - устанавливает отступ красной строки.

text-indent: {Отступ}|{Отступ}%;

Отступ может быть задан как абсолютной величиной, так и процентом от ширины родителя. Значение по умолчанию 0.

Поддерживается IE

text-height - интервал между строками текста.

text-height: {Интервал}|{Интервал}%;

Интервал может быть задан как абсолютной величиной, так и процентом от ширины родителя.

Поддерживается IE.

text-transform - задает преобразование регистра символов текста.

text-transform: none|capitalize|uppercase|lowercase;

text-transform: none; - отключает любые преобразования регистра символов (значение по умолчанию);

text-transform: capitalize; - преобразует первую букву каждого слова текста в верхний регистр;

text-transform: uppercase; - преобразует все символы текста в верхний регистр;

text-transform: lowercase; - в нижний регистр.

Поддерживается IE.

clear - задает поведение текста при "обтекании" им некоторых элементов страницы, таких как изображения.

clear: none|left|right|all;

Атрибут задается для текста, а не для элемента страницы, который он будет "обтекать".

clear: none; - разрешает тексту "обтекать" элемент страницы (значение по умолчанию);

clear: left; - запрещает тексту "обтекать" элемент страницы с левой стороны;

clear: right; - с правой стороны;

clear: all; - с обеих сторон.

Поддерживается IE и NN начиная с 4.0

word-spacing - определяет дополнительное расстояние между словами в тексте.

word-spacing: normal|{Величина};

Значение этого атрибута может быть задано либо абсолютной величиной в одной из поддерживаемых CSS единиц измерения, либо предопределенным значением `normal`, задающим стандартную величину расстояния между символами.

Значение по умолчанию `normal`.

Поддерживается IE .

`word-wrap` - устанавливает, будет ли строка, выходящая за границы элемента страницы и не содержащая пробелов, переноситься по словам.

`word-wrap: normal|break-word;`

`word-wrap: normal;` - запрещает переносить строки по словам (значение по умолчанию);

`word-wrap: break-word;` - разрешает.

Поддерживается IE начиная с 5.5

`word-mode` - задает направление строк текста: горизонтальное или вертикальное.

`word-mode: lr-tb|tb-rl;`

`word-mode: lr-tb;` - задает обычное горизонтальное расположение строк текста; текст пишется слева направо и сверху вниз (значение по умолчанию);

`word-mode: tb-rl;` - поворачивает текст на 90° по часовой стрелке; при этом он будет писаться сверху вниз и справа налево.

Поддерживается IE начиная с 5.5

`letter-spacing` - определяет расстояние между символами в тексте.

`letter-spacing: normal|{Величина};`

Значение этого атрибута может быть задано либо абсолютной величиной в одной из поддерживаемых CSS единиц измерения, либо предопределенным значением `normal`, задающим стандартную величину расстояния между символами.

Значение по умолчанию `normal`.

Поддерживается IE начиная с 4.0

`line-height` - задает вертикальное расстояние между строками текста.

`line-height: normal|{Y}|{Y}%;`

Высота может быть задана как абсолютной величиной, так и процентом от высоты родителя. Предопределенное значение `normal` задает стандартное расстояние.

Значение по умолчанию `normal`.

Поддерживается IE

`direction` - задает порядок чтения текста: слева направо или справа налево.

`direction: ltr|rtl|inherit;`

`direction: ltr;` - задает порядок чтения слева направо (значение по умолчанию);

`direction: rtl;` - справа налево;

`direction: inherit;` - заставляет наследовать порядок чтения у родителя.

Для документов, составленных на европейских языках, порядок чтения всегда слева направо (`ltr`).

Поддерживается IE начиная с 5.0

unicode-bidi - задает поведение встроенных элементов при изменении направления письма с помощью атрибута direction.

unicode-bidi: normal|embed|bidi-override;

unicode-bidi: normal; - меняет направление письма и у родителя (используется по умолчанию);

unicode-bidi: embed; - меняет направление письма только у встроенного элемента;

unicode-bidi: bidi-override; - аналогично embed за тем исключением, что направление письма меняется согласно значению атрибута direction, независимо от локальных установок Web-обозревателя.

Поддерживается IE начиная с 5.0

accelerator - позволяет указать, содержит ли текст элемента страницы клавишу-ускоритель.

accelerator: true|false;

Клавиша-ускоритель - это особая клавиша на клавиатуре, при нажатии которой вместе с клавишей <Alt> происходит переход к данному элементу страницы.

accelerator: true; - указывает, что текст содержит клавишу-ускоритель;

accelerator: false; - не содержит.

Значения по умолчанию не имеет.

*Пример использования:*

```
<LABEL for="txtName"><U style="accelerator: true">И</U>мя</LABEL>
```

```
<INPUT type="text" id="txtName" accesskey="B" value="Имя пользователя">
```

В этом случае символ "И" в слове "Имя" будет подчеркнут. Если в региональных настройках операционной системы Windows 2000 была выбрана опция Скрыть индикаторы клавиш-ускорителей до нажатия Alt, этот символ не будет подчеркнут, пока пользователь не нажмет клавишу <Alt> на клавиатуре.

Поддерживается IE начиная с 5.0

Псевдостили гиперссылок

Псевдостили применяются к гиперссылкам <A>.

active - применяется к активным гиперссылкам, т.е. гиперссылкам, на которых находится фокус ввода пользователя.

{Задание стиля гиперссылки}:active {Определение стиля};

Аналогичен атрибуту alink тега <BODY>.

*Пример:*

```
A:active {color: lime;}
```

активная гиперссылка будет ярко-зеленой.

По умолчанию в IE активные гиперссылки выделяются красным цветом.

Поддерживается IE начиная с 4.0

hover - применяется к гиперссылкам, когда пользователь помещает над ними курсор мыши.

{Задание стиля гиперссылки}:hover {Определение стиля};

Пример:

```
A:hover {color: lime; text-decoration: none;}
```

Гиперссылка, при наведении на нее мыши, будет ярко-зеленой и неподчеркнутой.

По умолчанию в IE гиперссылки подчеркиваются, когда пользователь помещает над ними курсор мыши.

Поддерживается IE начиная с 4.0

link - применяется к не посещенным еще пользователем гиперссылкам.

```
{Задание стиля гиперссылки}:link {Определение стиля};
```

Аналогичен атрибуту link тега <BODY>.

*Пример:*

```
A:link {color: black;}
```

По умолчанию в IE непосещенные гиперссылки отображаются синим цветом.

Поддерживается IE начиная с 3.02

visited - применяется к уже посещенным пользователем гиперссылкам.

```
{Задание стиля гиперссылки}:visited {Определение стиля};
```

Аналогичен атрибуту vlink тега <BODY>.

*Пример:*

```
A:link {color: indigo;}
```

По умолчанию в IE посещенные гиперссылки отображаются бордовым цветом. Поддерживается IE начиная с 3.02

### **Индивидуальное задание 1. Создать HTML-страницу с использованием CSS.**

Определить для всех ссылок на странице цвет, размер, семейство шрифта. Изменения свойств ссылки должны происходить в следующих случаях: в момент нахождения курсора над ней, посещенной ссылки, активной ссылки и в обычном состоянии. Определить один общий класс для изменения текста на все заглавные буквы, а другой – на прописные.

**Пример выполнения индивидуального задания: Каскадные таблицы стилей CSS. Использование свойств текста и псевдостилей гиперссылок.**

HTML код:

```
Index.htm
```

```
<HTML>
```

```
<HEAD>
```

```
<TITLE> Индивидуальное задание. CSS. Использование свойств текста и псевдостилей гиперссылок.</title>
```

```
</head>
```

```
<FRAMESET rows="15%, *">
```

```
<frame name="head" src="head.htm" SCROLLING="No" NORESIZE>
```

```

<FRAMESET cols="25%, *">
<FRAMESET rows="60%, *">
<frame name="menu1" src="menu1.htm" SCROLLING="Auto" NORESIZE>
<frame name="menu2" src="menu2.htm" SCROLLING="Auto" NORESIZE>
</frameset>
<frame name="main" src="main.htm" SCROLLING="Auto" NORESIZE>
</frameset>
</frameset>
</html>

```

### **head.htm**

```

<HTML>
<HEAD>
<LINK rel="stylesheet" type="text/css" href="main.css">
<TITLE> Индивидуальное задание.CSS. Использование свойств текста и
псевдостилей гиперссылок.</title>

```

```

</head>
<BODY>
<H1 class="txtdecor"> Индивидуальное задание №6</h1>
</body>
</html>

```

### **menu1.htm**

```

<HTML>
<HEAD>
<LINK rel="stylesheet" type="text/css" href="main.css">
<TITLE> Индивидуальное задание.CSS. Использование свойств текста и
псевдостилей гиперссылок.</title>

```

```

</head>
<BODY>
<UL>
<LI><A HREF="main.htm" target="main">Главная
<LI><A HREF="subject.htm" target="main">Предмет
<LI><A HREF="teacher.htm" target="main">Преподаватель
<LI><A HREF="theme.htm" target="main">Тема
<LI><A HREF="task.htm" target="main">Задание</a>
<LI><A HREF="fac.htm" target="main">Факультет</a>
<LI><A HREF="course.htm" target="main">Курс</a>
<LI><A HREF="group.htm" target="main">Группа</a>
<LI><A HREF="lastname.htm" target="main">Фамилия</a>
<LI><A HREF="firstname.htm" target="main">Имя</a>
</ul>
</body>
</html>

```

### **menu2.htm**

```

<HTML>
<HEAD>
<LINK rel="stylesheet" type="text/css" href="main.css">
<TITLE> Индивидуальное задание.CSS. Использование свойств текста и
псевдостилей гиперссылок.</title>
</head>
<BODY>
<UL>
<LI><A HREF="nonetxttrans.htm" target="main">Обычные буквы
<LI><A HREF="uppertxttrans.htm" target="main">Заглавные буквы
<LI><A HREF="lowertxttrans.htm" target="main">Прописные буквы
</ul>
</body>
</html>

```

### **main.htm**

```

<HTML>
<HEAD>
<LINK rel="stylesheet" type="text/css" href="main.css">
<TITLE> Индивидуальное задание.CSS. Использование свойств текста и
псевдостилей гиперссылок.</title>
</head>
<BODY>
<IMG SRC="img1.jpg" width="600" height="450">
</body>
</html>

```

### **main.css**

```

BODY {background-color: e5e5e5}
a {text-decoration: none; color: Purple;}
a:link      {color: Purple;   font-size: 12px; font-family: Arial, Helvetica,
sans-serif;}
a:visited   {color: Navy;    font-size: 12px; font-family: Arial, Helvetica,
sans-serif;}
a:active    {color: Red; font-size: 14px; font-family: Arial, Helvetica, sans-
serif; text-decoration: blink;}
a:hover     {color: Lime;    font-size: 14px; font-family: Arial, Helvetica,
sans-serif; text-decoration: underline;}
.txtdecor {text-decoration: underline; text-align: center; color: red}
.txttrans1 {text-transform: uppercase;}
.txttrans2 {text-transform: lowercase;}

```

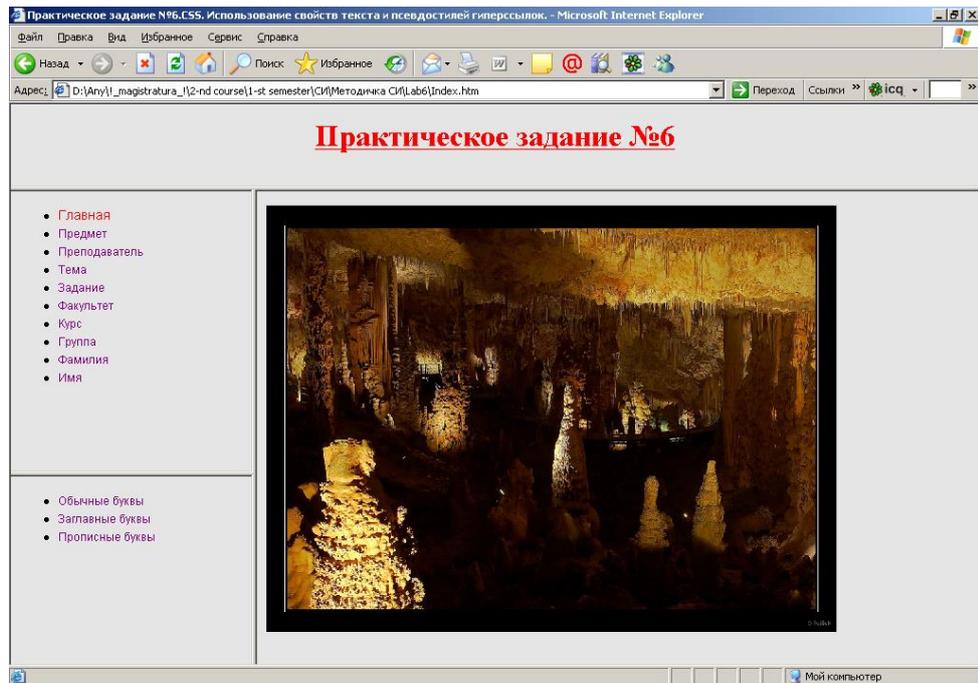


Рисунок 6 – Результат выполнения индивидуального практического задания

### Контрольные вопросы

1. Свойства текста и его атрибуты.
2. Псевдостили гиперссылок и его атрибуты.
3. Какой параметр задаёт регистр букв, и объясните назначение каждого атрибута?

## Список литературы

1. Баранов, Р.Д. Практические аспекты разработки веб-ресурсов [Электронный ресурс]: учебное пособие/ Р.Д. Баранов, С.А. Иноземцева, А.А. Рябова. — Электрон. текстовые данные. — Саратов: Вузовское образование, 2018. — 121 с. — 978-5-4487-0263-1. — Режим доступа: <http://www.iprbookshop.ru/75692.html>
2. Лучанинов, Д.В. Основы разработки web-сайтов образовательного назначения [Электронный ресурс]: учебное пособие/ Д.В. Лучанинов. — Электрон. текстовые данные. — Саратов: Ай Пи Эр Медиа, 2018. — 105 с. — 978-5-4486-0174-3. — Режим доступа: <http://www.iprbookshop.ru/70775.html>
3. Торопова, О.А. Основы web-программирования. Технологии HTML, DHTML [Электронный ресурс]: учебное пособие/ О.А. Торопова, И.Ф. Сытник. — Электрон. текстовые данные. — Саратов: Саратовский государственный технический университет имени Ю.А. Гагарина, ЭБС АСВ, 2014. — 106 с. — 978-5-7433-2606-8. — Режим доступа: <http://www.iprbookshop.ru/76493.html>
4. Буренин, С.Н. Web-программирование и базы данных [Электронный ресурс]: учебный практикум/ С.Н. Буренин. — Электрон. текстовые данные. — М.: Московский гуманитарный университет, 2014. — 120 с. — 978-5-906768-17-9. — Режим доступа: <http://www.iprbookshop.ru/39683.html>
5. Зудилова, Т.В. Web-программирование HTML [Электронный ресурс]/ Т.В. Зудилова, М.Л. Буркова. — Электрон. текстовые данные. — СПб.: Университет ИТМО, 2012. — 70 с. — 2227-8397. — Режим доступа: <http://www.iprbookshop.ru/65748.html>
6. Зудилова, Т.В. Web-программирование JavaScript [Электронный ресурс]/ Т.В. Зудилова, М.Л. Буркова. — Электрон. текстовые данные. — СПб.: Университет ИТМО, 2012. — 68 с. — 2227-8397. — Режим доступа: <http://www.iprbookshop.ru/65749.html>
7. Мелькин, Н.В. Искусство продвижения сайта. Полный курс SEO [Электронный ресурс]: от идеи до первых клиентов/ Н.В. Мелькин, К.С. Горяев. — Электрон. текстовые данные. — М.: Инфра-Инженерия, 2017. — 268 с. — 978-5-9729-0139-5. — Режим доступа: <http://www.iprbookshop.ru/68990.html>
8. Шапошников, И. Web-сайт своими руками [Текст]: учеб. пособие/ Шапошников И.- СПб.: БХВ-Петербург, 2000.- 224 с.
9. Кожемякин, А. А. «HTML и CSS в примерах. Создание Web-страниц»/ А. А. Кожемякин.- Изд-во: Альтекс-А, 2014.- 416 с.
10. Шмитт, К. Создание WEB-страниц средствами CSS/ К. Шмитт. — Издательство: КУДИЦ-Образ, 2003. — 368 с.

