



государственное автономное учреждение  
Калининградской области  
профессиональная образовательная организация  
**«КОЛЛЕДЖ ПРЕДПРИНИМАТЕЛЬСТВА»**

**МЕТОДИЧЕСКИЕ РАЗРАБОТКИ К ВЫПОЛНЕНИЮ  
ЛАБОРАТОРНЫХ РАБОТ ПО ДИСЦИПЛИНЕ  
«ОСНОВЫ ПРОГРАММИРОВАНИЯ»**  
для студентов специальности  
09.02.07 Информационные системы и программирование

Составил:  
преподаватель  
Мамаев П.В.

Калининград  
2023

## ***АННОТАЦИЯ***

Методические разработки лабораторных заданий для студентов специальности 09.02.07 Информационные системы и программирование содержат задания для студентов, предназначенных для проверки уровня сформированности знаний и практических навыков по пройденному материалу учебной дисциплины Основы программирования

Методические разработки, включают лабораторные работы, содержание, а также список литературных и других источников информации, необходимых для выполнения работ.

## Содержание

Введение .....	4
1. Структура документа HTML .....	5
2. Тэги, управляющие текстом .....	7
3. Ссылки .....	11
4. Использование тэга IMG .....	12
5. Списки и свойства таблиц .....	15
6. Colspan, rowspan, сложные таблицы .....	16
7. Фреймы, построение сайта на основе фреймов .....	22
8. Формы .....	28
Список источников .....	31

## Введение

В состав методических рекомендаций входят лабораторные работы к темам, рассматривающих основы языка web-программирования HTML, позволяющие закрепить знания и практические навыки в следующих направлениях :

- Форматирование текста
- Использование мультимедийных ресурсов на сайте
- Работа с таблицами различной сложности
- Вставка и использование форм, фреймов

В результате освоения учебной дисциплины студент должен

**уметь:**

- проектировать и создавать web-документы средствами языка HTML;
- проектировать дизайн сайта;
- осуществлять выбор средств и методов для решения поставленных задач

**знать:**

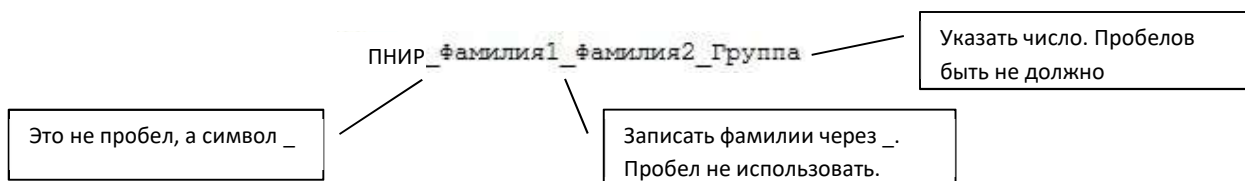
- возможности языка HTML
- основные теги языка HTML и их свойства;
- основные принципы создания сайтов
- основные принципы формирования структуры
- типы таблиц стилей;
- основные свойства языка CSS;
- основные свойства языка JavaScript

## 1. Структура документа HTML.

### Задания:

#### Часть 1.

1. Создать свой каталог на диске D:\, при этом присвоить ему имя в следующем формате:



2. Запустить текстовый редактор Блокнот (Пуск ⇒ Программы ⇒ Стандартные ⇒ Блокнот).

3. Сохранить этот (пока пустой) файл в своём каталоге при этом присвоить ему имя в следующем формате First.html. Для этого выбрать команду Файл ⇒ Сохранить как. В поле Папка – открыть свою папку (см. 1 п.), в поле Имя файла – указать имя будущего файла – First.html, в поле Тип файла – установить – Все файлы.

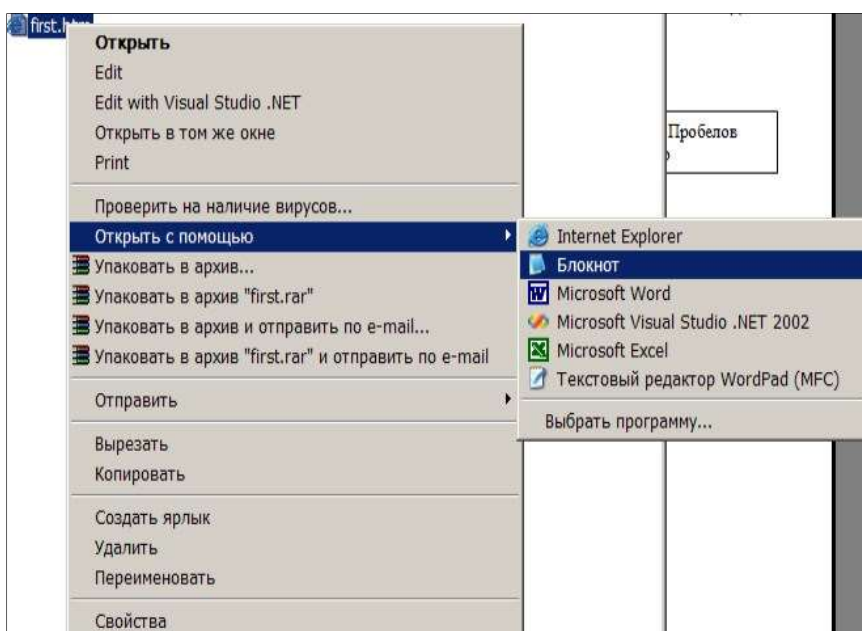


Рис. 1

4. Закрывать созданный файл и открыть его снова следующим способом, щёлкнув на его имени правой клавишей мыши и выбрав пункт (см. рис. 1).

5. В этом файле набрать код для создания простейшей Web-страницы, учитывая синтаксис языка HTML.

- В качестве заголовка Web-документа ввести – Свои фамилии (свою фамилию).

- В содержании документа создать заголовок первого уровня, соответствующий теме лабораторной работы. Установить его выравнивание по центру горизонтали.

- Далее, не используя никакие теги, в виде столбца (в три строки) набрать свою Фамилию Имя Отчество.

- Закончить создание Web-документа, закрыв открытые теги и сохранить его, выбрав команду Файл ⇒ Сохранить. Файл не закрывать.

6. Открыть созданный Web-документ с помощью Internet Explorer (или другим браузером) (см. рис. 1) в автономном режиме.

7. Изменяя размеры окна открытого Web-документа (см. рис.2), ответить на вопросы:

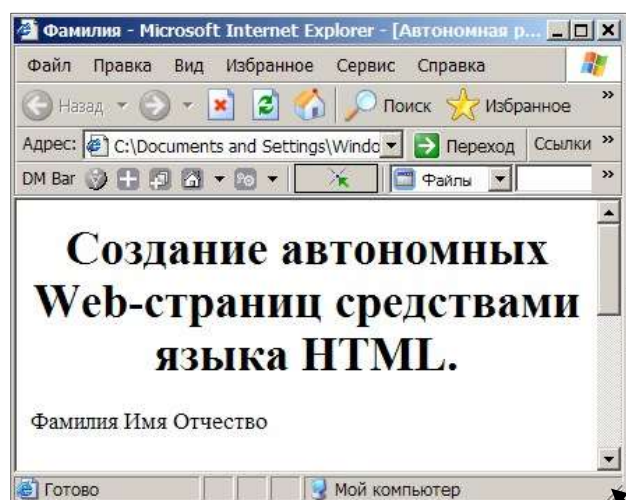


Рис. 2.

- *В каком элементе окна располагается заголовок Web-документа?*
  - *Что происходит содержимым документа при изменении размеров окна?*
  - *Как отображается текст содержимого Web-страницы, который записан в исходном файле в виде столбца и почему?*
  - *Как создать выравнивание заголовка содержимого документа по центру?*
8. Не закрывая окно Браузерс созданным Web-документом, активизировать окно Блокнот с программным кодом Web-страницы. Если его нет, открыть снова.
9. Внести изменения в код, таким образом, чтобы Фамилия Имя Отчество после загрузки отображались в виде столбца. После изменений в исходном файле его сохранить, а Web-страницу обновить. Для этого выбрать кнопку Обновить на панели инструментов Internet Explorer.

## Часть 2.

10. В своём каталоге создать ещё один Web-документ с именем Link.html . Для этого нужно открыть в Блокноте файл First.html, созданный в первой части работы, и сохранить его с новым именем, установив Тип файла – Все файлы. Сам файл First.html не изменять и не удалять.
11. В документе Link.html удалить текст – Фамилия Имя Отчество и вместо него добавить заголовок второго уровня " Форматирование абзацев. ".
12. Скопировать с интернета два небольших фрагмента текста (произвольных)
13. Обозначить вставленные тексты, как первый и второй абзацы.
14. Сохранить вторую ( Link.html ) Web-страницу.
15. Открыть в Блокноте файл First.html, созданный в первой части работы и в конце его содержимого создать гиперссылку на текст "Переход на страницу – Форматирование абзацев". При выборе этой ссылки должен загрузиться Web-документ Link.html.
16. Файлы Link.html и First.html не удалять и сохранить на диске для следующих работ.
17. Ответить на следующие вопросы и подготовиться к защите.

### **Контрольные вопросы:**

#### *Вопросы первой части.*

1. Вопросы из текста задания.
2. Какие теги должны всегда присутствовать в программном коде Web-документа?
3. Как создать заголовок содержимого документа?
4. Как определяется размер заголовка и его выравнивание?
5. За счёт чего производится перевод строки в отображаемом документе?
6. Что нужно сделать, чтобы сохранить текстовый файл, как Web-документ?

#### *Вопросы второй части.*

7. Как создаются такие элементы документа, как абзацы и как они отображаются?
8. Что нужно сделать, чтобы создать гиперссылку?
9. Что называется внешней и внутренней ссылками? В каких случаях они используются?

## 2. Тэги, управляющие текстом

**Лабораторная работа №1.** Создание простейшего HTML-документа. Форматирование шрифта и абзаца

**Цель работы:**

*Научиться создавать простейший гипертекстовый документ средствами текстового редактора Блокнот.*

*Научиться использовать теги форматирования шрифта и абзаца.*

**Задание на выполнение**

1. Создать файл с гипертекстовым документом:

- Запустить редактор Блокнот, ввести текст:

**Приветствую Вас на моей первой web-страничке!**

- Сохранить файл в созданной папке. При сохранении, в окне диалога **Сохранить как...** в строке **Тип файла:** выбрать вариант **Все файлы (\*.\*)**, а в строке **Имя файла** задать имя с расширением **.htm**, например **1\_name.htm** (где **name** – ваше имя)

- Закрыть документ, найти его пиктограмму в окне **Мой компьютер** или в окне программы **Проводник**.

- Открыть файл. Проанализировать, *с помощью какого приложения отображается файл* и как выглядит введенная фраза.

2. Ввести теги, определяющие структуру html-документа:

- С помощью контекстного меню открыть файл с помощью редактора Блокнот. Ввести приведенные ниже теги, в разделе заголовка документа (между тегами **<TITLE>** **</TITLE>**) указать свою фамилию.

```
<HTML>
```

```
<HEAD> <TITLE> Фамилия </TITLE>
```

```
</HEAD>
```

```
<BODY>
```

```
Приветствую Вас на моей первой web-страничке!
```

```
</BODY>
```

```
</HTML>
```

- **Сохранить** документ под тем же именем, обновить его отображение в браузере (выполнить **Вид/Обновить** или нажать кнопку **Обновить** на панели инструментов). Проанализировать произошедшие изменения в отображении документа.

3. Отредактировать документ:

- Вызвать меню браузера **Вид/Просмотр HTML-кода** и добавить после текста **«Приветствую Вас на моей первой web-страничке!»** текст подписи:

```
Студент группы NNN Фамилия Имя
```

Сохранить документ (но не закрывать) и обновить его просмотр в браузере.

- Используя одиночный тег **<BR>**, отредактировать документ так, чтобы подпись начиналась с новой строки, а **Фамилия Имя** – в следующей строке. Просмотреть в браузере новый вариант.

**Внимание!** *После каждого изменения документ нужно сохранять, а просмотр в браузере начинать с обновления загрузки документа с помощью кнопки «Обновить» на панели инструментов.*

4. Оформить фрагменты текста с помощью стилей **Заголовков**:

- Первую строку документа оформить стилем **Заголовок 1-го уровня** с помощью парного тега **<H1> ...</H1>**. Вторую строку оформить как **Заголовок 6-го уровня**, а третью как **Заголовок 4-го уровня**.

- Просмотреть документ в браузере, изменяя настройку отображения шрифтов (меню Вид / Размер шрифта / Самый крупный, Средний, Мелкий и Самый мелкий).

- Поменять стиль оформления первой строки на **Заголовок 2 уровня**, второй строки - на **Заголовок 5 уровня**, последней строки - на **Заголовок 3-го уровня**.

5. Выполнить форматирование шрифта:

- После строки **Фамилия Имя** добавить еще одну строку текста

**Нас утро встречает прохладой**

- Оформить фразу по приведенному ниже образцу.

**Нас *утро* встречает *прохладой***

В слове УТРО все буквы должны иметь **разные цвета**. В слове ПРОХЛАДОЙ оформить буквы ПРО – **красным** цветом, ОЙ – **синим**.

- Оформить строку с подписью (Студент группы NNN **Фамилия Имя**) **курсивом**, размер шрифта задать относительным изменением. Использовать теги <FONT SIZE=«+2»> и <I>

- Просмотреть полученный документ в браузере.

6. Выполнить форматирование абзацев:

- Создать новый документ **2\_name.htm**, сохранить его в той же рабочей папке.

- Ввести текст (использовать копирование текста из документа **1\_name.htm**):

<HTML>

<HEAD> <TITLE> **Фамилия** </TITLE>

</HEAD>

<BODY>

**Приветствую Вас на моей второй web-страничке! <BR> Монолог Гамлета**

</BODY>

</HTML>

- Выровнять текст **по центру**.

- Ввести текст:

Быть иль не быть - вот в чем вопрос. Что благороднее: сносить удары неистовой судьбы - иль против моря невзгод вооружиться, в бой вступить. И все покончить разом...

- Оформить выравнивание абзаца **по ширине**.

- Ограничить абзац горизонтальными разделительными линиями сверху и снизу, используя тег <HR>.

- Скопировать монолог и разбить его на абзацы. Выровнять **по центру**.

Быть иль не быть - вот в чем вопрос.

Что благороднее: сносить удары

Неистойвой судьбы - иль против моря

Невзгод вооружиться, в бой вступить

И все покончить разом...

- Сохранить документ.

- Просмотреть документ в окне браузера, изменяя размер окна.

*Таблица основных тегов HTML-документа. Теги форматирования шрифта и абзаца*

Назначение	Вид тегов	Примечание
<b>Общая структура документа HTML</b>		
Тип документа	<HTML></HTML>	Начало и конец документа
Имя документа	<HEAD></HEAD>	Не отображается браузером
Заголовок	<TITLE></TITLE>	Содержимое строки заголовка



		окна браузера
Тело документа	<BODY></BODY>	Содержимое WEB-страницы
<b>Структура содержания документа</b>		
Внутренние заголовки различного уровня	<H№> текст </H№>	Где № – номер уровня заголовка (от 1 до 6). Например, <H1>...</H1> - заголовок 1-го уровня.
Заголовок с выравниванием	<H№ ALIGN="LEFT  CENTER  RIGHT"> текст </H№>	LEFT - по левому краю, CENTER - по центру, RIGHT - по правому краю.
<i>Форматирование абзацев</i>		
Создание абзаца (параграфа)	<P> текст </P>	Абзацы отделяются двойным межстрочным интервалом
Перевод строки внутри абзаца	 	Одиночный тег
Выравнивание абзаца	<P ALIGN="LEFT">текст </P> <P ALIGN="CENTER">текст </P> <P ALIGN="RIGHT"> текст</P> <P ALIGN="JUSTIFY"> текст </P>	LEFT - по левому краю CENTER - по центру RIGHT - по правому краю JUSTIFY – по ширине
Разделительная горизонтальная линия между абзацами	<HR SIZE=«?»>	Одиночный тег. «?» - толщина линии в пикселях. Толщину линии можно не указывать.
<i>Форматирование шрифта</i>		
Жирный	<B> текст </B>	<B> <b>Жирный</b> </B>
Курсив	<I> текст </I>	<I> <i>Курсив</i> </I>
Подчеркнутый	<U> текст </U>	<U> <u>Подчеркнутый</u> </U>
Перечеркнутый	<S> текст </S>	<S> <del>Перечеркнутый</del> </S>
Увеличенный размер	<BIG> текст </BIG >	
Уменьшенный размер	<SMALL> текст </SMALL>	
Верхний индекс	<SUP> текст </SUP>	<SUP> Верхний индекс </SUP>
Нижний индекс	<SUB> текст </SUB>	<SUB> Нижний индекс </SUB>
Размер шрифта	<FONT SIZE=«?»> текст </FONT>	? - значения от 1 до 7 или относительное изменение (например, +2)
Базовый размер шрифта	<BASEFONT SIZE=«?»>	Одиночный тег ? – размер от 1 до 7; по умолчанию равен 3 и задается для всего документа в целом
Гарнитура шрифта	<FONT FASE=«название1, название2, ...»> текст </FONT>	Текст оформляется первым, установленным на компьютере шрифтом из списка названий
Цвет шрифта	<FONT COLOR="\$\$\$\$\$\$"> текст </FONT>	Цвет задается либо ключевым словом, либо шестнадцатеричным кодом с

		символом # RED –красный, #FF0000 – шестнадцатеричный код – красного цвета
--	--	--

**Таблица основных цветов**

Цвет	Color's name	Шестнадцатеричный код цвета		
		Red	Green	Blue
Черный	black	00	00	00
Темно-синий	navy	00	00	80
Голубой	blue	00	00	FF
Зеленый	green	00	80	00
Темно-зеленый	teal	00	80	80
Салатный	lime	00	FF	00
Бледно-голубой	aqua	00	FF	FF
Вишневый	maroon	80	00	00
Фиолетовый	purple	80	00	80
Оливковый	olive	80	80	00
Серый	gray	80	80	80
Светло-серый	silver	C0	C0	C0
Красный	red	FF	00	00
Лиловый	fuchsia	FF	00	FF
Желтый	yellow	FF	FF	00
Белый	white	FF	FF	FF

### 3. Ссылки.

1. **Основные сведения.** Существует три типа ссылок: *внутристраничные* – они задают переходы в пределах одной страницы; *внутрисистемные* – ссылки между страницами в пределах одного и того же сервера; и *межсистемные* – ссылки на страницы, расположенные на удаленных узлах Web. Для определения ссылок предназначен специальный тег, который называется Anchor (якорь).

#### Элемент А.

*Описание:* определяет ссылку или якорь.

*Начальный тег:* **необходим**

*Конечный тег:* **необходим**

*Определения атрибутов:*

**name** = “строка” – именуется текущий якорь, который может стать якорем назначения для другой гиперссылки. Значением этого атрибута должно быть уникальное имя якоря. Областью видимости этого имени является текущий документ.

**href** = “строка” – определяет размещение ресурса Web, определяя таким образом ссылку с текущего элемента на якорь назначения, определённый этим атрибутом.

*Примеры:*

<code>&lt;a name="info"&gt;Информация&lt;/a&gt;</code>	Создается метка с именем info.
<code>&lt;a href="http://www.google.ru"&gt;ЧГПУ&lt;/a&gt;</code>	Создается ссылка на поисковую систему Google.

Цвет ссылки можно задать при помощи атрибутов тэга **BODY**:

**link** = “color” – устанавливает цвет непосещённых гиперссылок.

**vlink** = “color” – устанавливает цвет посещённых гиперссылок.

**alink** = “color” – устанавливает цвет гиперссылок при выборе пользователем.

*Пример:*

<code>&lt;body link="#ff0000" vlink="#00ff00"&gt;</code>	Цвет непосещённых гиперссылок – красный, посещённых – синий.
--	--

Для указания ссылки на электронный ящик в значение атрибута href должно быть “mailto:имя\_электронного\_ящика”.

*Пример:*

<code>&lt;a href="свой_е-мэйл"&gt;Напишите мне&lt;/a&gt;</code>	Создается ссылка «Напишите мне», щелкнув по которой, можно написать и отправить письмо.
---	---

#### 1. Внутристраничные ссылки:

1. Создать имя (метку) для точки назначения, в которую должен осуществляться переход. Метка создается с помощью тега якоря, используя его атрибут NAME (например, `<A NAME="info">Дополнительная информация </A>`). Фраза "Дополнительная информация" при этом никак не будут выделены в тексте документа.

2. Для создания гиперссылки на эту точку документа используется тег `<A>` с атрибутом HREF=, при этом к имени якоря присоединяется знак #: `<A HREF="#info">Просмотр дополнительной информации </A>`. Такой фрагмент HTML-текста приведет к появлению в документе выделенного фрагмента (в нашем случае

фразы Просмотр дополнительной информации), при нажатии на который произойдет переход к строке Дополнительна информация.

## 2. Внутрисистемные ссылки:

1. В файле, в который мы хотим перейти необходимо создать метку аналогично п. 1.1.1.

2. Создать гиперссылку аналогично на созданную вами страницу

3. **Задание:** В файл *index.html* добавить абзац «Кратко о моих увлечениях и хобби можно посмотреть *здесь*» после абзаца со словами

4. В начале файла *obotne.html* фразу «Кратко о себе» заключить в теги `<A NAME="info">` и `</A>`.

5. В документе *main.html* слово «здесь» оформить гиперссылкой на документ *obotne.html* на якорь «*info*»

6. Изменить цвет непосещенных гиперссылок на красный, а посещенных на черный.

7. Опробовать действие гиперссылки.

8. В файл *obotne.html* внизу страницы поместить абзац, состоящий из одного слова «Назад», выровненного по центру.

9. Организовать обратный переход.

3. **Межсистемные ссылки.** Используя эти ссылки можно установить связь с любой страницей на любом узле Web.

1. В этом случае необходимо создать только гиперссылку, например, ссылка на сервер фирмы Microsoft выглядит следующим образом: `<A HREF="http://www.microsoft.com"> Переход на сервер компании Microsoft </A>`. При нажатии на выделенный фрагмент произойдет переход на сервер фирмы Microsoft.

2. **Задание:** в документ *main.html* добавить абзац «Здесь вы можете посмотреть сайт Камышловского техникума промышленности и транспорта».

3. Оформить абзац по своему усмотрению.

4. Оформить абзац как гиперссылку на адрес `http://kamtechprom.ru/`

5. Адрес e-mail внизу страницы сделать ссылкой. Теперь пользователь сможет отправлять электронные письма по этому адресу.

## 4.Использование тэга IMG

**Тема:** Использование тэга IMG и форматирование текста в Web-документах средствами языка HTML.

**Цель:** Ознакомиться, каким образом создаются и вставляются графические файлы в Web-страницы. Изучить приёмы форматирования текста средствами языка HTML.

### Задания:

1.Открыть свой каталог на диске D:\, если это нет, то создать

2.В своём каталоге открыть с помощью программы Блокнот Web-документ с именем *first.html*, созданный в прошлой работе

3.В этом файле набрать код для создания между заголовка третьего уровня "Использование графических изображений. ", установить его выравнивание по центру горизонтали. Сохранить данный файл с тем же именем, не закрывая.

4.Запустить отредактированный Web-документ с помощью Internet Explorer и убедиться в правильности создания заголовка.

5. В своём каталоге создать графический 24 разрядный файл с именем Bitmap.bmp, в который внести изображения некоторых кнопок (см. рис.2) с панели инструментов программы Internet Explorer.

6. Для этого выполнить следующее:

- Скопировать в Буфер обмена содержимое открытого окна Web-документа first.html в программе Internet Explorer.
- Запустить графический редактор Paint и вставить в него содержимое буфера обмена, используя Shift+Insert.
- Отменить выделение вставленной области клавишей Esc и, используя курсор мыши, выделить нужную область – кнопки на панели инструментов (см. рис.2). Скопировать выделенный фрагмент в буфер обмена, используя Ctrl+Insert.
- Используя клавишу Esc отменить выделение только что скопированного фрагмента и вставить его, используя Shift+Insert. Данный фрагмент должен наложиться на предыдущее изображение в верхнем левом углу (см. рис.1).
- Используя информацию рис.1 определить размеры (ширину и высоту)

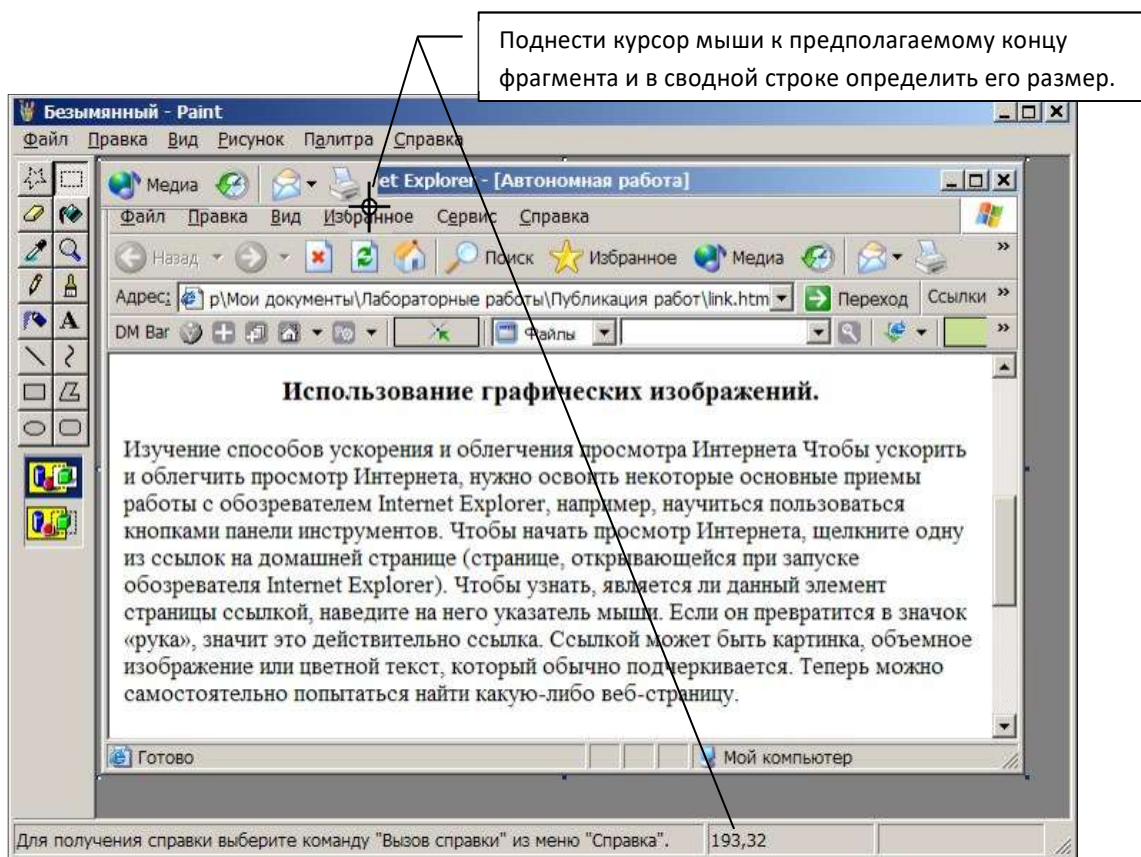


Рис. 1

вставленного фрагмента.

- В этом же редакторе Paint выбрать команду Рисунок ⇨ Атрибуты и в появившемся диалоговом окне ввести ширину и высоту данного изображения. После подтверждения установок в рабочем окне редактора Paint должен остаться только скопированный фрагмент.
- Сохранить данный файл, выполнив при этом следующее:
  - i. открыв при этом свой каталог,
  - ii. указав имя файла – Bitmap,
  - iii. установив тип файла – 24 разрядный рисунок (\*.bmp).
- Проверить наличие сохранённого файла в своём каталоге. Файл должен сохраниться под именем Bitmap.bmp.

7. Открыть с помощью программы Блокнот Web-документ с именем first.html и в этом файле добавить код для вставки во второй абзац изображения из файла Vitmar.bmp, учитывая синтаксис языка HTML. Вставленное изображение должно разместиться на Web-странице в строгом соответствии с образцом (см. рис.2). Расстояние от текста до изображения должно быть 20 пикселей.

8. Внести в программный код Web-документа дополнения для установки разных цветов шрифта на заголовки (см. рис.2).

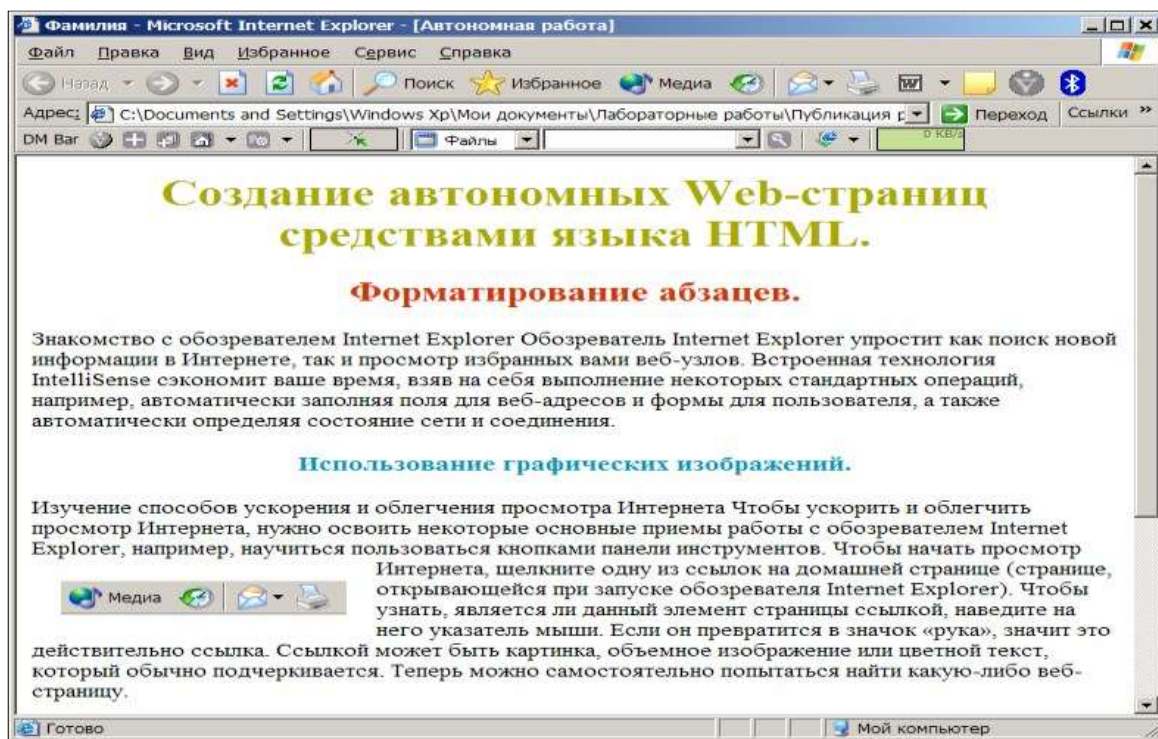


Рис. 2

9. Создать копию файла first.html и присвоить ей имя Linkcopy.htm. Провести следующие эксперименты в файле Linkcopy.htm, поочередно изменяя исходный код, сохраняя и обновляя отображение документа в Internet Explorer:

- i. При вставке графического изображения в Web-документ зарезервировать следующую область вывода 100x20. *В каком виде отображалось изображение, сохранялись ли при этом его пропорции?*
- ii. При вставке графического изображения в Web-документ зарезервировать следующую область вывода 300x220. *В каком виде отображалось изображение, сохранялись ли при этом его пропорции?*
- iii. В своём каталоге создать ещё одну папку и переместить в неё на время файл Vitmar.bmp. *Что появилось после обновления отображения документа?*
- iv. При вставке графического изображения добавить атрибут, который выводит альтернативный текст вместо изображения. *Проверить как он выводится? Вернуть графический файл обратно.*
- v. Установить в качестве фонового узора графический файл. *Проверить как он выводится?*
- vi. На различные фрагменты текста первого абзаца файла Linkcopy.htm установить следующие типы шрифта: Courier New, ItalicT.
- vii. На различные фрагменты текста второго абзаца установить следующие атрибуты шрифта: полужирный, подчёркнутый, курсив, отображение цитат, выделение. *Проверить как выводится текст с данными атрибутами?*

10. Файлы first.html не удалять и сохранить для следующих работ.

11. Ответить на следующие вопросы и подготовиться к защите.

### Контрольные вопросы:

1. Какими комбинациями клавиш можно скопировать в буфер обмена содержимое активного окна и содержимое всего рабочего стола?
2. Какие специальные возможности могут иметь файлы типа GIF?
3. Какие основные отличия существуют в следующих форматах графических файлов: GIF, JPEG и BMP?
4. Что означает выражение 24 разрядный графический файл?
5. Как узнать размер фрагмента изображения в пикселях?
6. Как создать графический файл нужного размера?
7. Что нужно сделать, чтобы вставить содержимое графического файла в Web-документ?
8. Каким образом устанавливаются поля изображения?
9. Что нужно сделать, чтобы установить необходимое обтекание изображения текстом?
10. Как установить цвет и тип шрифта?
11. Что нужно сделать, чтобы установить нужное начертание и размер шрифта? Как выводится текст с данными атрибутами?

### 5. списки и свойства таблиц

#### Лабораторная работа №2.

**Тема:** Использование списков и таблиц в Web-документах средствами языка HTML.

**Цель:** Ознакомиться, каким образом создаются списки на Web-странице. Изучить приёмы создания и форматирования таблиц средствами языка HTML.

#### Задания:

18. Открыть свой каталог на диске D:\, если это нет, то создать (см. лабораторную работу №1).
19. В своём каталоге открыть с помощью программы Блокнот файл с именем Link.htm созданный в предыдущей работе и сохранить его с новым именем Table.htm, указав нужный тип файла. При этом предыдущие файлы Link.htm и First.htm не удалять и не изменять. Если файла Link.htm, по каким то причинам нет, то создать файл Table.htm, как пустую Web-страницу, в окне заголовка которой должна быть Ваша фамилия и добавить два заголовка второго уровня (см. рис. 1 и 2).
20. В файле Table.htm удалить всё из раздела – содержимое, кроме заголовков второго и третьего уровня.
21. Изменить название заголовков (см. рис. 1 и 2). и установить оба заголовка на второй уровень.
22. Внести программные строки для создания списка, формат которого должен соответствовать образцу (см. рис. 1). Первые три уровня списка должны быть нумерованные, а четвёртый – маркированный.
23. Запустить отредактированный Web-документ с помощью Internet Explorer и убедиться в правильности создания списка.
24. После второго заголовка добавить программные строки для создания таблицы, формат которой должен соответствовать образцу (см. рис. 2). При определении

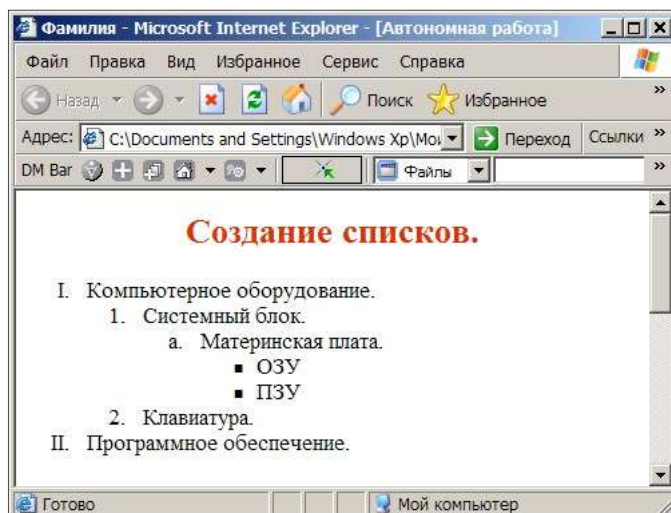


Рис. 1

формата таблицы необходимо указать следующее:

- i. Отображение границ и внешней рамки толщиной 10.
- ii. Минимальная ширина таблицы – 60% от ширины окна документа.
- iii. Центрирование таблицы по горизонтали страницы.
- iv. Заголовок таблицы, шрифт которого представлен в полужирном начертании, должен находиться сверху от таблицы.
- v. В первой строке должны быть записаны наименования столбцов в полужирном начертании на жёлтом фоне с центрированием по горизонтали ячейки.
- vi. Во второй и третьей строках размещаются ячейки обычного формата с информацией согласно образцу (см. рис.2).

25. После сохранения файла с программным кодом Table.htm обновить отображение содержимого Web-документа в Internet Explorer и убедиться в соответствии отображения созданной таблицы с образцом.

26. Изменяя размеры окна Internet Explorer с Web-документом, понаблюдать, как ведёт себя таблица и ответить на вопросы:

- i. Для чего устанавливается минимальная ширина таблицы?
- ii. Относительно чего устанавливается длина заголовка таблицы?

27. Файлы Link.htm и First.htm и Table.htm не удалять и сохранить на дискете для следующих работ. Ответить на следующие вопросы и подготовиться к защите.

### Контрольные вопросы:

10. Объяснить шестнадцатеричный формат значения устанавливаемого цвета и способы его получения.
11. Чем отличаются нумерованные, маркированные списки и списки определений?
12. Как создать маркированный список и установить и него нужный вид маркера?
13. Как создать нумерованный список и установить и него нужный способ нумерации?
14. Что нужно сделать, чтобы создать многоуровневый список ?
15. Для чего устанавливается минимальная ширина таблицы?
16. Относительно чего устанавливается длина заголовка таблицы при его выравнивании?
17. Как задаются центрирование по горизонтали и минимальная ширина таблицы?
18. Каким образом можно установить нужный цвет фона ячеек таблицы?
19. Как задаётся количество строк и столбцов таблицы?
20. Каким образом можно задать заголовок столбцов таблицы?
21. Какая по умолчанию устанавливается ширина между ячейками таблицы и как её изменить?
22. Как установить толщину рамки таблицы?
23. Как установить выравнивание содержимого ячейки таблицы по высоте?

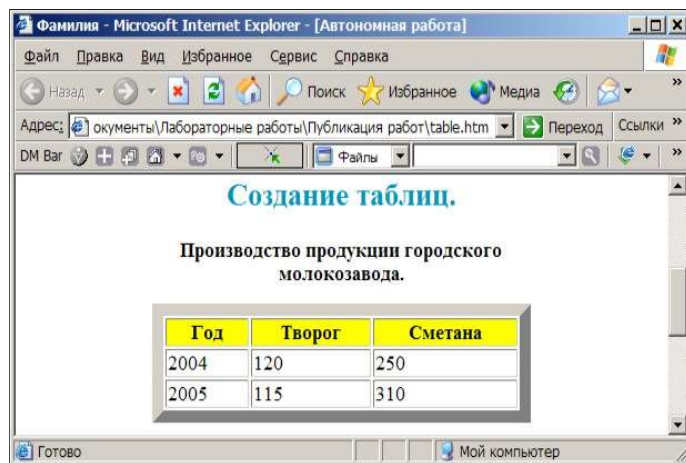


Рис.2

## 6. Colspan, rowspan, сложные таблицы

### ЛАБОРАТОРНАЯ РАБОТА №3

Тема работы: Создание сложных таблиц в HTML-документе.



### Объединение ячеек

Параметр **colspan** – объединяет ячейки в ряду, а **rowspan** – объединяет ячейки в столбце. Эти параметры могут принимать значение от 2 и больше, т.е. наша ячейка может растягиваться на два и более столбца (ряда).

1x1	1x2	
2x1	2x2	2x3

В этом примере мы использовали параметр `colspan=2`, прописав его

для ячейки 1x1. Код будет выглядеть следующим образом:

```
<table>
<tr>
<td height=«35» bgcolor=«#FFCC33» colspan=«2» <center>1x1</center>
</td>
<td width=«50» bgcolor=«#336699» <center>1x2</center> </td>
</tr>
<tr>
<td height=«35» width=«50» bgcolor=«#336699» <center>2x1</center> </td>
<td width=«50» bgcolor=«#FFCC33» <center>2x2</center> </td>
<td width=«50» bgcolor=«#336699» <center>2x3</center> </td>
</tr>
</table>
```

Обратите внимание, на то, что параметр **width** для ячейки 1x1 в нашем примере не указан, если вас так тянет задавать этот параметр, то в нашем примере для ячейки 1x1 его надо было бы прописать равным 100 пикселям, т.к. все-таки ячейка 1x1 длиннее других в два раза.

Обратите внимание, в нашем примере нет ячейки 1x3, т.е. в первом ряду всего лишь две ячейки, т.к. ячейка 1x1 равна сама по себе двум ячейкам по длине (что мы и указали параметром `colspan`). Если бы мы прописали ячейку 1x3, тогда у нас получилась бы такая ерунда:

1x1	1x2	1x3
2x1	2x2	2x3

Теперь, когда мы разобрались с параметром `colspan`, разберемся с параметром `rowspan`. Например, надо создать следующую таблицу (обратите внимание, должна исчезнуть ячейка 2x3).

1x1	1x2	1x3
2x1	2x2	

Принцип действия тот же:

```

<table>
<tr>
<td height=«35» width=«50» bgcolor=«#FFCC33»> <center> 1x1 </center>
</td>
<td width=«50» bgcolor=«#336699»> <center> 1x2 </center> </td>
<td width=«50» bgcolor=«#FFCC33» rowspan=«2»> <center>1x3 </center>
</td>
</tr>

<tr>
<td height=«35» width=«50» bgcolor=«#336699» > <center> 2x1 </center>
</td>
<td width=«50» bgcolor=«#FFCC33»> <center> 2x2 </center> </td>
</tr>
</table>

```

### Границы ячеек

Итак, мы остановились на такой таблице:

1x1	1x2	
2x1	2x2	

Для изменения ширины пространства между ячейками таблицы используется атрибут `cellspacing`.

Чтобы избавиться от пространства между ячейками `cellspacing=0`

1x1	1x2	
2x1	2x2	

```

<table cellspacing=0>
<tr>
<td height=«35» bgcolor=«#FFCC33» colspan=«2» <center>1x1</center>
</td>
<td width=«50» bgcolor=«#336699» rowspan=«2» <center>1x2</center>
</td>
</tr>
<tr>
<td height=«35» width=«50» bgcolor=«#336699» <center>2x1</center> </td>
<td width=«50» bgcolor=«#FFCC33» <center>2x2</center> </td>
</tr>
</table>

```

Можно наоборот, увеличить пространство между ячейками, допустим пусть **cellspacing=5**, тогда получим такое:

1x1		1x2
2x1	2x2	

### Рамки

Рамка вводится параметром **border**. Зададим рамку равную 3 пикселям:

```
<table border=«3»>
```

Выглядеть это будет вот так:

таблица с рамкой
------------------

Рамке можно задать цвет. Пусть он будет черный в нашем примере:

```
<table border=«3» bordercolor=«#000000»>
```

таблица с рамкой
------------------

### Ход работы.

В процессе выполнения работы все промежуточные изменения программного кода (окно Блокнот с фамилией) и промежуточные результаты (окно браузера с фамилией) необходимо копировать (Alt + PrtScreen) и вставлять в свой отчетный файл формата Word:

**Задание 1.** Запустите стандартную программу Блокнот (Notepad) и наберите следующий текст с элементами форматирования (в строке заголовка Web-документа к тексту добавить – свою фамилию, при сохранении файла в имени тоже добавить свою фамилию):



```
<html>
<body>
<table border=2>
<tr><td> Программирование </td></tr>
<tr><td> Императивное </td></tr>
<tr><td> функциональное </td></tr>
</table>
</body>
</html>
```

Для просмотра созданной Web-страницы используйте браузер Microsoft Internet Explorer.

1. Вставьте в тег <table> следующие атрибуты:  
`width=50% align="center" bgcolor="yellow" bordercolor="blue".`

Просмотрите обновленный документ в браузере.

2. Добавьте в тег <table> атрибут:  
`cellspacing=5.`

Просмотрите обновленный документ в браузере.

3. Измените значение последнего атрибута на 10, 30 и посмотрите что произойдет с таблицей, верните обратно это значение.

4. Добавьте между второй парой тегов <tr>:</tr> тег  
`<td> Декларативное </td>`

Просмотрите полученный документ в окне браузера.

5. Добавьте между третьей парой тегов <tr>:</tr> тег  
`<td> логическое </td>`

Просмотрите обновленный документ в браузере.

6. Добавьте в первой паре тегов <tr>:</tr> в теге <td> атрибут  
`colspan=3.`

Просмотрите полученный документ в окне браузера.

7. Добавьте в первой паре тегов <tr>:</tr> в теге <td> еще атрибут  
`align="center"`

Просмотрите полученный документ в окне браузера.

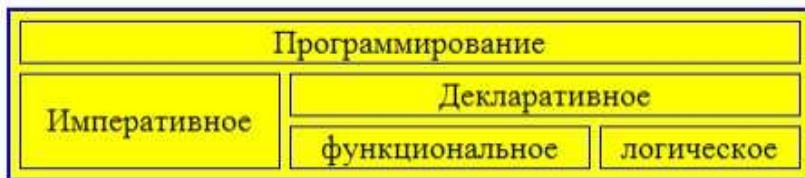
8. Добавьте во второй паре тегов <tr>:</tr> в первом теге <td> атрибут  
`rowspan=2`

Просмотрите полученный документ в окне браузера.

9. Добавьте во второй паре тегов <tr>:</tr> во втором теге <td> атрибут  
`colspan=2.`

Просмотрите полученный документ в окне браузера.

10. Отцентрируйте самостоятельно текст в ячейках и остальное, как показано в образце.



**Задание 2.** Добавьте ниже следующие таблицы оформив их по образцам (в последней таблице цветовое оформление установить самостоятельно):

**Основные характеристики IP-адресов классов А,В и С.**

Характеристика	Класс		
	А	В	С
Номер сети	W	W.X	W.X.Y
Номер узла	X.Y.Z	Y.Z	Z
Возможное количество сетей	126	16 384	2 097 151
Возможное количество узлов	16 777 214	65 534	254
	Особые адреса		
Запись адреса сети в целом	W.0.0.0	W.X.0.0	W.X.Y.0
Широковещательный адрес в сети	W.255.255.255	W.X.255.255	W.X.Y.255

**Кабельные системы**

Тип кабеля	Характеристика	
	Максимальное расстояние передачи	Максимальная скорость передачи
Коаксиальный кабель	185 – 500 м	10 Мбит/с
"Витая пара"	30 – 100 м	10 Мбит/с – 1 Гбит/с
Оптоволоконный кабель	2 км	10 Мбит/с – 2 Гбит/с

**Стандартные номера портов для основных сервисов.**

Компонент службы	Номер порта	Транспортные протоколы
<b>Электронная почта</b>		
SMTP-сервер	25	TCP
POP3-сервер	110	
IMAP-сервер	143	
<b>Телеконференции</b>		
NNTP-сервер	119	TCP
<b>FTP</b>		
FTP-сервер	20, 21	TCP
<b>Telnet</b>		
Telnet-сервер	23	TCP

<b>WWW</b>		
HTTP-сервер	80	TCP
<b>DNS</b>		
DNS-сервер	53	TCP, UDP

**Задание 3.** Проверить свою работу и подготовиться к её защите.

## 7. Фрэймы, построение сайта на основе фреймов

### ЛАБОРАТОРНАЯ РАБОТА 4

#### «Создание WWW-странички с фреймами»

**1. Создайте четыре файла: a.htm, b.htm, c.htm, d.htm, которые будут в дальнейшем заполнять поля фреймов.**

##### Файл a.htm

```
<HTML>
<HEAD><TITLE>Страничка A</TITLE></HEAD>
<BODY bgcolor=red text=white>
<p>Страничка A</p>
</BODY>
</HTML>
```

##### Файл b.htm

```
<HTML>
<HEAD><TITLE>Страничка B</TITLE></HEAD>
<BODY bgcolor=yellow text=black>
<p>Страничка B</p>
</BODY>
</HTML>
```

##### Файл c.htm

```
<HTML>
<HEAD><TITLE>Страничка C</TITLE></HEAD>
<BODY bgcolor=green text=white>
<p>Страничка C</p>
</BODY>
</HTML>
```

##### Файл d.htm

```
<HTML>
<HEAD><TITLE>Страничка D</TITLE></HEAD>
<BODY bgcolor=blue text=white>
<p>Страничка D</p>
</BODY>
</HTML>
```

**2. Создайте файлы WWW-страничек с различными вариантами расположения фреймов.**

##### Файл index-2-frames.htm

```
<HTML>
<HEAD>
<TITLE>ФРЕЙМЫ-2</TITLE></HEAD>
```

**<!-- Пример создания двух вертикальных фреймов одинаковой ширины -->**

```
<FRAMESET cols="50%,50%">
```

```
<FRAME src="a.htm">
```

```
<FRAME src="b.htm">
```

```
</FRAMESET>
```

```
</HTML>
```

**Файл index-4-frames-cols.htm**

```
<HTML>
```

```
<HEAD>
```

```
<TITLE>ФРЕЙМЫ-4</TITLE></HEAD>
```

**<!-- Пример создания четырех вертикальных фреймов. Из них 3 фрейма одинаковой ширины, а один – занимает всю оставшуюся площадь экрана -->**

```
<FRAMESET cols="20%,20%,20%,*">
```

```
<FRAME src="a.htm">
```

```
<FRAME src="b.htm">
```

```
<FRAME src="c.htm">
```

```
<FRAME src="d.htm"> </FRAMESET></HTML>
```

**Файл index-4-frames-rows.htm**

```
<HTML>
```

```
<HEAD>
```

```
<TITLE>ФРЕЙМЫ-4</TITLE></HEAD>
```

```
<FRAMESET rows="20%,20%,20%,*">
```

**<!-- Пример создания четырех горизонтальных фреймов. Из них 3 фрейма одинаковой ширины, а один – занимает всю оставшуюся площадь экрана -->**

```
<FRAME src="a.htm">
```

```
<FRAME src="b.htm">
```

```
<FRAME src="c.htm">
```

```
<FRAME src="d.htm">
```

```
</FRAMESET>
```

```
</HTML>
```

**Файл index-3-frames-cols-coot.htm**

```
<HTML>
```

```
<HEAD>
```

```
<TITLE>ФРЕЙМЫ-4</TITLE></HEAD>
```

**<!--Пример создания фреймов различной ширины путем задания соотношения между ними-->**

```
<FRAMESET cols="100,*,2*">
```

```
<FRAME src="a.htm">
```

```
<FRAME src="b.htm">
```

```
<FRAME src="c.htm">
```

```
<FRAME src="d.htm">
```

```
</FRAMESET>
```

```
</HTML>
```

**Файл index-4-frames-cols-rows.htm**

```
<HTML>
```

```
<HEAD>
```

```
<TITLE>ФРЕЙМЫ-4</TITLE></HEAD>
```

```
<!--Пример создания двух вертикальных и двух горизонтальных фреймов (третий фрейм делится пополам на строки) -->
```

```
<FRAMESET cols="20%,20%,*">
```

```
<FRAME src="a.htm">
```

```
<FRAME src="b.htm">
```

```
<FRAMESET rows="50%,50%">
```

```
<FRAME src="c.htm">
```

```
<FRAME src="d.htm">
```

```
</FRAMESET>
```

```
</HTML>
```

### Файл index-3-frames-rows-cols.htm

```
<HTML>
```

```
<HEAD>
```

```
<TITLE>ФРЕЙМЫ-4</TITLE></HEAD>
```

```
<!--Пример создания одного горизонтального (заголовочного) и двух вертикальных фреймов-->
```

```
<FRAMESET rows="20%,*">
```

```
<FRAME src="a.htm">
```

```
<FRAMESET cols="20%,*">
```

```
<FRAME src="b.htm">
```

```
<FRAME src="c.htm">
```

```
</FRAMESET>
```

```
</FRAMESET>
```

```
</HTML>
```



### 3. Создайте фрейм с графическим изображением

Создайте файл с именем **index\_2\_grafic.htm**

```
<HTML>
<HEAD>
<TITLE>ФРЕЙМЫ С ГРАФИКОЙ</TITLE></HEAD>
<FRAMESET cols="*, 70%">
<FRAME src="image_1.jpg" marginwidth=10 marginheight=10 scrolling=auto>
<!--Атрибуты marginwidth marginheight управляют расстоянием от изображения до
границ рамки фрейма, атрибут scrolling ( = auto/ yes/ no) управляет наличием полос
прокрутки -->
<FRAME src="b.htm">
</FRAMESET>
</HTML>
```

### 4. Установка связи между фреймами.

Измените содержимое файла **a.htm** следующим образом:

```
<HTML>
<HEAD><TITLE>Страничка А</TITLE></HEAD>
<BODY bgcolor=red text=white>
<p>Страничка А</p>
<B>A link to <A href="c.htm">c.htm</A></B>
</BODY>
</HTML>
```

Сохраните файл с именем **a1.htm**.

Загрузите базовый пример из файла **index-2-frames.htm**. Измените имя вызываемого во фрейм файла с **a.htm** **a1.htm**. Сохраните файл с именем **index-2-frames01.htm**

```
<HTML>
<HEAD>
<TITLE>ФРЕЙМЫ</TITLE></HEAD>
<FRAMESET cols="50%,50%">
<FRAME src="a1.htm">
<FRAME src="b.htm">
</FRAMESET>
</HTML>
```

Активизируйте ссылку **c.htm**. При активизации ссылки содержимое страницы c.htm отображается внутри фрейма А.

*Попробуйте отобразить содержимое страницы С внутри фрейма В.*

Загрузите базовый пример из файла **index-2-frames.htm**. Введите во фрейм, выводящий на экран файл В атрибут name - "имя фрейма": name="FRAME\_B". Это имя в последующем будет использовано в качестве параметра атрибута target тега <A> при создании файла a2.htm. Сохраните файл с именем **index-2-frames02.htm**

```
<HTML>
<HEAD>
<TITLE>ФРЕЙМЫ</TITLE></HEAD>
<FRAMESET cols="50%,50%">
<FRAME src="a2.htm">
<FRAME src="b.htm" name="FRAME_B">
```

```
</FRAMESET>
```

```
</HTML>
```

Измените содержимое файла **a1.htm** следующим образом:

```
<HTML>
```

```
<HEAD><TITLE>Страничка А</TITLE></HEAD>
```

```
<BODY bgcolor=red text=white>
```

```
<p>Страничка А</p>
```

```
<B>А link to <A href="c.htm" target="FRAME_B">c.htm</A></B>
```

```
</BODY>
```

```
</HTML>
```

Сохраните файл с именем **a2.htm**.

Загрузите файл **index-2-frames02.htm** Активизируйте ссылку **c.htm**. При активизации ссылки содержимое страницы c.htm отображается внутри фрейма В.

### 5. Открытие WWW-страницы из сети Интернет внутри фрейма.

Откройте файл с именем **index\_2\_grafic.htm**. Внесите в html-код соответствующие изменения для того, чтобы во втором фрейме открылся сайт «Нижний Новгород Online».

```
<HTML>
```

```
<HEAD>
```

```
<TITLE>ФРЕЙМЫ С ГРАФИКОЙ</TITLE></HEAD>
```

```
<FRAMESET cols="*, 70%">
```

```
<FRAME src="image_1.jpg" marginwidth=10 marginheight=10 scrolling=auto>
```

```
<FRAME src="http://www.nn.ru/">
```

```
</FRAMESET>
```

```
</HTML>
```

### 6. Открытие WWW-страницы в отдельном окне.

Создайте файл **Index\_Demo.htm**

```
<HTML>
```

```
<HEAD>
```

```
<TITLE>ФРЕЙМЫ</TITLE>
```

```
</HEAD>
```

```
<frameset cols="50%,50%">
```

```
<frame src="a3.htm">
```

```
<frame src="b.htm" name="FRAME_B">
```

```
</frameset>
```

```
</HTML>
```

Измените содержимое файла **a2.htm** следующим образом:

```
<HTML>
```

```
<HEAD><TITLE>Страничка А</TITLE></HEAD>
```

```
<BODY bgcolor=red text=white>
```

```
<p>Страничка А</p>
```

```
<B>Ссылка на страничку<A href="c.htm" target=_blank> c.htm</A></B>
```

```
</BODY>
```

```
</HTML>
```

Сохраните файл с именем **a3.htm**.

Загрузите файл **Index\_Demo.htm**. Активизируйте ссылку **c.htm**. При активизации ссылки содержимое страницы c.htm отображается в новом пустом окне.

---

Примечания:

**Значения атрибута target** тега <A>

target=\_self - содержимое страницы, заданной ссылкой, загружается в окно, которое содержит ссылку.

target=\_parent - содержимое страницы, заданной ссылкой, загружается в окно, являющееся непосредственным владельцем фреймов.

target=\_top – содержимое страницы, заданной ссылкой, загружается в окно, игнорируя используемые фреймы.

---

## 7. «Плавающие фреймы».

Плавающий фрейм может появляться в любом месте экрана. Текст, расположенный на главной странице, как бы обтекает плавающий фрейм.

Для создания плавающих фреймов применяется специальный тег <IFRAME></IFRAME>

**Формат тега:**

<IFRAME src=n name=m> N </IFRAME>

**n** – адрес или имя открываемой рамки;

**m** – название самой рамки;

**N** – тот ресурс, который будет выводиться на экран в случае использования пользователем браузера не поддерживающего плавающих фреймов.

Создайте файл **Index\_swim.htm**.

(При создании плавающей рамки укажем ее размер в 50% экрана, высоту 300 пх. Зададим выравнивание рамки фрейма справа от текста с помощью атрибута align.)

```
<HTML>
```

```
<HEAD>
```

```
<TITLE>Главная страница</TITLE>
```

```
</HEAD>
```

```
<BODY bgcolor="FFCCFF">
```

```
<IFRAME align=right width=50% height=300 src="swim.htm">
```

```
</IFRAME>
```

```
<p>
```

Плавающий фрейм может появляться в любом месте экрана.

Текст, расположенный на главной странице, как бы обтекает плавающий фрейм.

Для создания плавающих фреймов

применяется специальный тег <IFRAME> </IFRAME> <br>

Формат тега:<br>

<IFRAME src=n name=m> N </IFRAME>, где <br>

n - адрес или имя открываемой рамки;<br>

m - название самой рамки;<br>

N - тот ресурс, который будет выводиться на экран

в случае использования пользователем браузера

не поддерживающего плавающих фреймов.

```
</P>
</BODY>
</HTML>
```

Создайте файл, отображаемый внутри «плавающего фрейма» **swim.htm**.

```
<HTML>
<HEAD>
<TITLE>Внутренняя страница</TITLE>
</HEAD>
<BODY bgcolor="ccccff">
<H2> <CENTER> ПРИМЕР «ПЛАВАЮЩЕГО ФРЕЙМА» </CENTER></H2>
<H4> <CENTER> Здесь будет расположен текст плавающего фрейма.</CENTER></H4>
</BODY>
</HTML>
```

## 8.Формы

### ЛАБОРАТОРНАЯ РАБОТА 5

Цель: Получение практических навыков в работе с формами HTML

#### 1) Создание простой формы

Теги `<form>` и `</form>` задают начало и конец формы. Начинаящий форму тег `<form>` содержит два атрибута: *action* и *method*. Атрибут *action* содержит адрес URL сценария, который должен быть вызван для обработки сценария. Атрибут *method* указывает браузеру, какой вид HTTP запроса необходимо использовать для отправки формы; возможны значения *POST* и *GET*.

#### Замечание

Главное отличие методов *POST* и *GET* заключается в способе передачи информации. В методе *GET* параметры передаются через адресную строку, т.е. по сути в HTTP-заголовке запроса, в то время как в методе *POST* параметры передаются через тело HTTP-запроса и никак не отражаются на виде адресной строки.

```
<form method="post" action="../admin/add_story.php">
```

```
</form>
```

#### 2) Флажок (checkbox)

Флажки *checkbox* предлагают пользователю ряд вариантов, и разрешает выбор нескольких из них.

```
<input name="Имя переключателя" type="Тип" value="Значение">
```

Группа флажков состоит из элементов `<input>`, имеющих одинаковые атрибуты *name* и *type(checkbox)*. Если вы хотите, чтобы элемент был отмечен по умолчанию необходимо пометить его как *checked*. Если элемент выбран, то сценарию поступит строка *имя=значение*, в противном случае в обработчик формы не *придет ничего*, т.е. не выбранные флажки вообще никак не проявляют себя в переданном наборе данных.

Пример:

```
<input name="mycolor" type="checkbox" value="red" checked>Красный(выбран по умолчанию)
```

```
<input name="mycolor" type="checkbox" value="blue">Синий
```

```
<input name="mycolor" type="checkbox" value="black">Черный
```

```
<input name="mycolor" type="checkbox" value="white">Белый
```

### 3) Переключатель(**radio**)

Переключатели *radio* предлагают пользователю ряд вариантов, но разрешает выбрать только один из них.

```
<input name="Имя переключателя" type="Тип" value="Значение">
```

Переключатель (*radio*) имеет атрибуты *name*, *type* и *value*. Атрибут *name* задает имя переключателя, *type* задает тип *radio*, а атрибут *value* задает значение. Если пользователь выберет переключатель, то сценарию будет передана строка *имя=значение*. При необходимости можно указать параметр *checked*, который указывает на то, что переключатель будет иметь фокус (т.е. будет отмечен по умолчанию) при загрузке страницы. Переключатели также можно объединять в группы, для этого они должны иметь одно и то же имя.

Пример:

```
<input name="mycolor" type="radio" value="white"> Белый  
<input name="mycolor " type="radio" value="green" checked> Зеленый (выбран по умолчанию)  
<input name="mycolor " type="radio" value="blue"> Синий  
<input name="mycolor " type="radio" value="red"> Красный  
<input name="mycolor " type="radio" value="black"> Черный
```

### 4) Кнопка сброса формы(**Reset**)

```
<input type="Тип" name="Имя кнопки" value="Надпись на кнопке">
```

При нажатии на кнопку сброса (*reset*), все элементы формы будут установлены в то состояние, которое было задано в атрибутах по умолчанию, причем отправка формы *не производится*.

Пример:

```
<input type="reset" name="Reset" value="Очистить форму">
```

### 5) Выпадающий список (**select**)

Тэг *<select>* представляет собой выпадающий или раскрытый список, при этом одновременно могут быть выбраны одна или несколько строк.

Список начинается с парных тегов *<select></select>*.

Теги *<option></option>* позволяют определить содержимое списка, а параметр *value* определяет значение строки. Если в теге *<option>* указан параметр *selected*, то строка будет изначально выбранной. Параметр *size* задает, сколько строк будет занимать список. Если *size* равен 1, то список будет выпадающим. Если указан атрибут *multiple*, то разрешено выбирать несколько элементов из списка (при *size = 1* не имеет смысла).

```
<select name="Имя списка" size = "Размер" multiple>  
<option value="Значение">Отображаемый текст в списке</option>  
</select>
```

При передаче данных выпадающего списка сценарию передается строка *имя=значение*, а при раскрытом списке передается строка *имя=значение1&имя=значение2&имя=значениеN*.

### 6) Текстовое поле (**text**)

Позволяет пользователям вводить различную информацию.

```
<input type="Тип" name="Имя поля" size="Размер" maxlength="Макс. количество символов">
```

При создании обычного текстового поля размером *size* и максимальной допустимой длины *maxlength* символов, атрибут *type* принимает значение *text*. Если указан параметр *value*, то поле будет содержать отображать *value*-текст. При создании поля не забывайте указывать имя поля, т.к. этот атрибут является обязательным.

Пример:

```
<input type="text" name="txtName" size="10" maxlength="5" value="Текст по умолчанию">
```

### 7) Поле для ввода пароля (password)

Полностью аналогичен текстовому полю, за исключением того что символы, набираемые пользователем, не будут отображаться на экране.

Пример:

```
<input type="password" name="txtName" size="10" maxlength="5">
```

### 8) Многострочное поле ввода текста (textarea)

Многострочное поле ввода текста позволяет отправлять не одну строку, а сразу несколько. По умолчанию тег создает пустое поле шириной в 20 символов и состоящее из двух строк.

```
<textarea name="Имя поля" cols="Ширина поля" rows="Число строк">Текст</textarea>
```

Многострочное поле ввода текста начинается с парных тегов `<textarea>``</textarea>`. Тэг `name` задает имя многострочного поля. Также можно указать ширину поля (`cols`) и число строк (`rows`). При необходимости можно указать атрибут `readonly`, который запрещает редактировать, удалять и изменять текст, т.е. текст будет предназначен только для чтения. Если необходимо чтобы текст был изначально отображен в многострочном поле ввода, то его необходимо поместить между тэгами `<textarea>``</textarea>`.

Пример:

```
<textarea name="txtArea" cols="15" rows="10" readonly> Текст, который изначально будет отображен в многострочном поле ввода и который нельзя изменять, т.к. указан атрибут readonly </textarea>
```

### 9) Скрытое текстовое поле

Позволяет передавать сценарию какую то служебную информацию, не отображая её на странице.

```
<input name="Имя" type="Тип" value="Значение">
```

Скрытое поле начинается с тега `<input>`, атрибуты которого являются `name`, `type` и `value`. Атрибут `name` задает имя поля, `type` определяет тип поля, а атрибут `value` задает значение поля.

Пример:

```
<input name="email" type="hidden" value="spam@nosпам.ru">
```

### 10) Кнопка отправки формы (submit)

Служит для отправки формы сценарию.

```
<input type="Тип" name="Имя кнопки" value="Текст кнопки">
```

При создании кнопки для отправки формы необходимо указать 2 атрибута: `type="submit"` и `value="Текст кнопки"`. Атрибут `name` необходим если кнопка не одна, а несколько и все они созданы для разных операций, например кнопки "Сохранить", "Удалить", "Редактировать" и т.д. После нажатия на кнопку сценарию передается строка `имя=текст кнопки`.

### 11) Кнопка для загрузки файлов (browse)

Служит для реализации загрузки файлов на сервер. Объект `browse` начитается с парных тегов `<form>``</form>`. Начинаящий тэг `<form>` содержит необходимый атрибут `enctype`. Атрибут `enctype` принимает значение `multipart/form-data`, который извещает сервер о том, что вместе с обычной информацией посылается и файл. При создании текстового поля также необходимо указать тип файла – "file".

```
<form enctype="multipart/form-data" action="upload.php" method="post">  
  Загрузить файл: <input name="my_file" type="file">  
  <input type="submit" value="Отправить">  
</form>
```

## 12) Рамка (fieldset)

Объект *fieldset* позволяет вам нарисовать рамку вокруг объектов. Имеет закрывающий тэг *</fieldset>*. Заголовок указывается в тэгах *<legend></legend>*. Основное назначение объекта – задание различных стилей оформления.

Пример:

```
<fieldset>  
<legend>Программное обеспечение(заголовок рамки)</legend>  
Текст, который будет помещен внутри рамки.</fieldset>
```

**Задание:** Разработать страницу HTML с использованием всех элементов форм в виде анкеты на произвольную тему. Минимальное количество вопросов- 25

## Список источников

- 1.100% самоучитель. Создание Web-страниц и Web-сайтов. HTML и JavaScript/Автор: Гаевский, А.Ю.; Романовский, В.А. Год: 2008 Издание: Триумф
2. HTML, XHTML и CSS. Библия пользователя/Автор: Пфаффенбергер, Брайан; Шафер, Стивен; Уайт, Чак и др. Год: 2007 Издание: Вильямс; Издание 3-е

## Дополнительные ресурсы

1. <http://www.w3.org/>
2. <http://validator.w3.org/>
3. <http://jigsaw.w3.org/css-validator/>