

Министерство образования и науки Российской Федерации
Федеральное агентство по образованию
Государственное образовательное учреждение высшего профессионального образования
МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ ИМ. Н.Э. БАУМАНА
(МГТУ им. Н.Э. Баумана)
Факультет «Информатика и системы управления»
Кафедра «Компьютерные системы и сети»

«УТВЕРЖДАЮ»
Заведующий кафедрой ИУ-6
Д.т.н., профессор

_____ Сюзев В.В.
« ___ » _____ 2008 г.

Язык HTML

Методические указания по выполнению лабораторных работ
по курсу «Разработка Интернет-приложений»

Галямова Е.В.

2008 год,
Москва

Создание интерактивного Интернет-приложения

Цель лабораторной работы - освоение языка программирования HTML.

Задачами лабораторной работы являются:

1. Освоение основных компонент языка HTML: элементы, теги, атрибуты языка.
2. Оформление первой веб-страницы.
3. Написание первого скрипта
4. Создание сенсорных и анимированных изображений

1. Теоретическая часть

1. Элементы и теги

Язык HTML - это, по сути, множество *элементов*, специфицирующих формат описываемой страницы. Каждый элемент влияет на данные (текст или изображение), которые следуют за ним. Например, следующий элемент определяет, что текст будет выделен **полужирным** шрифтом:

```
<B> This text is bold</B>
```

Символ B - элемент языка, а обозначение называется *тег*, в частности, это тег конца элемента. Пары тегов обеспечивают возможность вложенного использования элементов:

```
<B> This text is bold and <I> This text is italicized </I></B>
```

Не все HTML-элементы могут вкладываться друг в друга, имеют теги начала и конца или требуют завершения их тегом конца, но большинство из них удовлетворяют этим правилам языка.

2. Атрибуты

HTML-элементы могут иметь и атрибуты. *Атрибуты* модифицируют характеристики элемента. Например, элемент P определяет начало абзаца текста. Он имеет один атрибут ALIGN, который позволяет определять выравнивание: LEFT, CENTER or RIGHT. Например:

```
<P ALIGN = CENTER> This sentence is centered. </P>
```

```
<P ALIGN = RIGHT> This sentence is aligned right. </P>
```

```
<P ALIGN = LEFT> This sentence is aligned left. </P>
```

3. Комментарии

Для комментариев предусмотрены теги комментариев <!-- и -->. Элемент комментария позволяет "спрятать" элементы, не поддерживаемые устаревшими браузерами (как - увидим дальше). Есть и альтернативный тег COMMENT из HTML версии 2.0. Примеры использования:

```
<!-- Это начало комментариев.
```

```
А это конец комментариев -->
```

```
<COMMENT>
```

```
Это начало комментариев с использованием тега версии 2.0.
```

```
</COMMENT>
```

4. Цвета

Некоторые HTML-элементы поддерживают атрибут COLOR. Его значение можно задать двумя способами:

- (RGB модель) Сочетанием шестнадцатеричных значений, соответствующих красной, зеленой и синей составляющим цвета
- прямым указанием названия цвета.

Примеры:

```
<body bgcolor="blue">
```

```
<h4><font color ="#0000FF" size = "2">
```

```
Last update: October 08, 2001 </font></h4>
```

```
<TABLE bgcolor="Ivory">
```

5. Элементы Заголовка

Некоторые элементы могут использоваться только между тегами начала **<HEAD>** и конца **</HEAD>** заголовка.

1.1. BASE

По умолчанию браузер использует **URL**, указанный для загрузки какого-либо документа, как полный, абсолютный адрес этого документа. Затем, при загрузке ссылок, встречающихся в документе, браузер вычисляет их абсолютные адреса на основании заданного базового адреса. Элемент **BASE** используется для явной установки абсолютного **URL**:

```
<!-- EDUCATION.HTM --- >
```

```
<BASE HREF = "http://education.vsnl.com/ruscult/education.htm">
```

Документы, в которых задействован элемент **BASE**, можно свободно перемещать из одного каталога в другой, поскольку ссылки, заданные относительно определенного абсолютного адреса, будут сохранять свою значимость. В приведенном выше примере при перемещении файла **EDUCATION.HTM** в другой каталог внутренние, относительные ссылки по-прежнему будут работать, поскольку браузер будет использовать **URL**, заданный в элементе **BASE**, а не новый адрес страницы.

1.2. META

Элемент **META** используется для предоставления браузеру, поисковым серверам и другим приложениям высокоуровневой информации о документе. Имеет несколько полезных атрибутов, хотя применение их зависит от конкретного браузера.

1.2.1. Атрибут HTTP-EQUIV

Обеспечивает браузер сопроводительной информацией по данному документу. Позволяет проводить *клиентскую перезагрузку* (*client-pull*). Его используют для указания адреса документа, который должен автоматически загружаться в браузер. Например, при задании опции **Refresh** (**обновить**) браузер IE будет перезагружать с сервера либо текущую, либо заданную страницу. Следующий тег будет обеспечивать перезагрузку документа каждые 10 сек.:

```
<META HTTP-EQUIV = "Refresh" CONTENT = "10 ">
```

Причем, если задать **URL** другой страницы, то она будет загружаться вместо первоначальной (см. ниже).

1.2.2. Атрибут NAME

Строка, определяемая пользователем.

1.2.3. Атрибут CONTENT

Строка, определяемая пользователем. В сочетании с атрибутом **HTTP-EQUIV** может использоваться для автоматической перезагрузки документа через определенные интервалы времени.

1.2.4. URL

URL документа. Обычно используется с опцией **Refresh** атрибута **HTTP-EQUIV**:

```
<META HTTP-EQUIV="Refresh" URL=" http://education.vsnl.com/ruscult/education.htm ">
```

Элемент **META** используется для внесения в документ инфо, определяемой пользователем.

Например, редактор FrontPage Express вставляет тег **META** в каждый составляемый им документ. Атрибуты **NAME** и **CONTENT** используются как комментарии при описании приложения, генерирующего документ:

```
<META NAME = "GENERATOR" CONTENT = "MICROSOFT FrontPage 2.0 ">
```

Имейте в виду, что некоторые серверные приложения индексируют страницы узла, сканируя элементы **META**. Поэтому, имеет смысл помещать в них ключевые слова, характеризующие страницу, например:

```
<META NAME = "Keywords" CONTENT = "ActiveX, OLE, COM, MFC, Windows">
```

2. Создание сценариев (Элемент SCRIPT)

Элемент **SCRIPT** дает возможность создания сценариев - внедрение в страницу исполняемых элементов, что превращает ее в динамический документ. Элемент **SCRIPT** имеет несколько атрибутов. Наиболее употребительный - атрибут **LANGUAGE** - определяет язык, используемый для создания сценария (VBScript, Jscript Microsoft, или JavaScript Netscape).

Элементы основного раздела веб-страницы –BODY:

Элемент **HEAD** описывает характеристики документа на "верхнем уровне". Вторую важную часть HTML-документа составляет элемент **BODY**. Здесь содержатся рисунки и текст, отображаемые в рабочей области окна браузера. Большая часть элементов HTML используется в пределах элемента **BODY**.

Элемент **BODY** имеет несколько атрибутов, которые представлены таблице 1. Большинство из них обеспечивают задание неявных параметров страницы, которые могут быть переопределены заданием других элементов в теле документа. Приведем пример документа, использующего изображение в формате **JPEG** в качестве фона, белый цвет для просмотренных ссылок и красный для непросмотренных. Для левой и верхней границ установлено значение один пиксель.

Пример:

```
<background = "blusky.jpg"  
link="FF0000" vlink = "FFFFFF"  
topmargin="1" leftmargin="1">
```

Большинство элементов, рассматриваемых в этом разделе, используются в пределах элемента **BODY**. Другими словами, они должны быть обязательно заключены между тегами начала **< BODY >** и конца **< /BODY >** тела документа.

Таблица 1.

Атрибут	Описание
ALINK = цвет	Определяет цвет для активной гиперссылки
Background = url	URL, определяющий фоновое изображение для страницы
BGCOLOR = цвет	Определяет цвет фона страницы
BGPROPERTIES=FIXED	Отменяет многократное повторение фонового рисунка. Это позволяет создать эффект водяного знака
ID = значение	Идентификатор элемента в документе
LEFTMARGIN=n	Определяет отступ от левого края страницы в пикселях. Если n=0, то текст будет начинаться у левого края окна
LINK = цвет	Определяет цвет непросмотренных гиперссылок
STYLE = css- свойства	Определяет ссылку на страницу стилей
TEXT = цвет	Определяет цвет текста в документе
TOPMARGIN = n	Определяет отступ от верхнего края страницы в пикселях. Если n=0, то текст будет начинаться у верхнего края окна браузера (или фрейма)
VLINK = цвет	Определяет цвет отображения гиперссылок, которые уже просматривались

1. Абзацы и разрывы строки

В первую очередь WWW — это среда публикаций, хотя она и превращается (довольно быстро) в среду разработки приложений. Что представляла бы собой публикация без абзаца? Элемент P является наиболее часто употребляемым в HTML. Он используется для разделения текста на абзацы. Абзац имеет только один атрибут — ALIGN, позволяющий выравнивать текст по левому краю (LEFT), по правому краю (RIGHT) и по центру (CENTER). По умолчанию обычно принимается выравнивание по левому краю. Ниже приводится простой документ, состоящий из трех абзацев.

Пример:

```
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML//EN">
<html>
<head>
<meta http-equiv="Content-Type"
content="text/html; charset=koi8-r">
<meta name="GENERATOR" content="Microsoft FrontPage 2.0">
<title>The P and BR Elements</title>
</head>
<body bgcolor="#FFFFFF">
<h3>Элемент выделения параграфа </h3>
<p>Параграф один. Этот параграф расположен справа и использует атрибут ALIGN=LEFT элемента выделения параграфа. Чтобы сделать параграф подлиннее, использована дополнительная линия.
</p>
<p align="center"> Параграф два. Этот параграф расположен в центре окна и использует атрибут ALIGN=CENTER элемента выделения параграфа. Чтобы сделать параграф подлиннее, использована дополнительная линия. </p>
<p align="right"> Параграф три. Этот параграф расположен справа и использует атрибут ALIGN=RIGHT элемента выделения параграфа. Чтобы сделать параграф подлиннее, использована дополнительную линию. </p>
<hr>
</body>
</html>
```

Внимание!!! Особенность предыдущего примера — пустые строки, намеренно или случайно оставленные вами между абзацами, никак не влияют на окончательный вид страницы. Это же касается разрывов строк внутри абзаца. Текст, заключенный в теги абзаца, форматируется автоматически, и строки разрываются у границы окна браузера. Это станет Вам понятным, когда Вы создадите свою страницу на основании этого примера.

Если требуется разбить строки абзаца в определенной точке, следует использовать элемент разрыва строки (
). Этот элемент использует только один тег начала <BK>; поскольку тег конца не имеет смысла. Рассмотрим пример, построенный на основе предыдущего:

```
<h3>Разрыв строки</h3>
<p> Параграф один<br>
Этот параграф расположен справа и использует атрибут ALIGN=LEFT элемента выделения
```

параграфа.

Чтобы сделать параграф подлиннее, использована дополнительная линия. </p>

<p align="center"> Параграф два. Этот параграф расположен в центре окна и использует атрибут ALIGN=CENTER элемента выделения параграфа. Чтобы сделать параграф подлиннее, использована дополнительная линия. </p>

<p align="right">Параграф три.

Этот параграф расположен справа и использует атрибут ALIGN=RIGHT элемента выделения параграфа.

Чтобы сделать параграф подлиннее, использована дополнительная линия. </p>

2. Списки

Элементы списков позволяют создавать в web-страницах простые, нумерованные и вложенные списки.

Существует пять основных типов списков:

- перечисление,
- нумерованный,
- список определений,
- меню и
- каталог.

Списки меню и каталогов очень похожи на нумерованный список, поэтому здесь мы коснемся только форматов трех типов списков.

Список-перечисление создается с помощью элементов **UL** (unordered list - неупорядоченный список) и **LI** (Line Item — строковый элемент).

Пример:

```
<ul>
  <li> Пункт 1</li>
  <li> Пункт 2</li>
  <li> Пункт 3</li>
  <li> Пункт 4</li>
</ul>
```

Нумерованный список начинается с элемента **OL**. Один из пяти видов нумерации элементов задается с помощью атрибута **TYPE**: большими и малыми буквами, большими и малыми римскими цифрами и десятичными числами. Ниже приводится пример простого нумерованного списка, в котором элементы нумеруются числами. В этом примере показано также действие атрибута **START**, который позволяет начинать нумерацию элементов списка с произвольной цифры или буквы.

Пример:

```
<ol type="1" start="10">
  <li> Пункт Номер 1</li>
  <li> Пункт Номер 2</li>
  <li> Пункт Номер 3</li>
</ol>
```

Список-определение используется для представления текста в обычном формате определения терминов. Список начинается с элемента **DL**. Термины определяются с помощью элемента **DT** (definition term — термин определения), а соответствующее определение обозначается элементом **DD**.

Пример:

```
<dl>
  <dt>ActiveX</dt>
  <dd>Понятие ActiveX принято фирмой Microsoft для описания ее
  новых Web- технологий, основанных на компонентной модели объекта
  (COM). </dd>
  <dt>COM </dt>
  <dd>Component Object Model - Компонентная модель объекта -
  Технология системного уровня, созданная фирмой Microsoft для
  облегчения разработки компонентного программного обеспечения.
  </dd>
</dl>
```

Форматирование текста

Язык HTML содержит множество элементов, предназначенных для форматирования текста (см. табл. 7.1). Например, элементы заголовка (от H1 до H6)

обеспечивают целых шесть уровней форматирования стиля заголовка, которые в основном изменяют размер шрифта. Заголовок первого уровня (H1) имеет наибольший размер, заголовок шестого уровня (H6) — наименьший.

Таблица 7.1 Элементы форматирования текста

Элемент	Тег	Эффект
Абзац	P	Обозначает начало и конец абзаца. Текст, находящийся между этими тегами, автоматически разбивается браузером на строки
Разрыв строки	BR	Явно разрывает строку
Отмена разрыва строки	NOBR	Запрещает разрыв строки
Разбивка на подразделы	DIV	Играет роль контейнера для последовательностей элементов. Например, позволяет установить элемент ALIGN для нескольких абзацев сразу
Заголовок	H1 - H6	Стили заголовков. H1 использует наибольший шрифт, а H6 —наименьший. Элементы H задают также выделение текста полужирным шрифтом
Полужирный	B or STRONG	Выделяет текст полужирным шрифтом
Курсив	I or EM	Выделяет текст курсивом.
Подчеркивание	U	Выделяет текст подчеркиванием
Отступы	BLOCKQUOTE	Задаёт отступы справа и слева
Верхний или нижний индекс	SUB and SUP	Преобразуют текст в верхние и нижние индексы соответственно
Шрифт меньшего или большего размера	SMALL or BIG	SMALL задает размер шрифта на один пункт меньше текущего значения, а BIG — на один пункт больше. Теги аналогичны элементам and
Моноширинный шрифт, HTML-теги не обрабатываются	PLAINTEXT	Полезен для отображения на экране HTML-программы. Текст отображается моноширинным шрифтом, а HTML-код, заключенный между тегами, не обрабатывается
Моноширинный Шрифт	CODE	Элемент CODE задает использование моноширинного шрифта. Это полезно при отображении текстов программ
Моноширинный Шрифт	PRE	Элемент PRE определяет, что следующий участок текста уже отформатирован и должен отображаться "как есть"
Шрифт по умолчанию	BASE FONT	Задаёт тип, размер и цвет шрифта по умолчанию
Шрифт	FONT	Задаёт тип, размер и цвет шрифта для фрагмента текста

Задание характеристик шрифта

В браузере Internet Explorer реализованы и усовершенствованы несколько расширений NetScare, относящихся к шрифтам. Элемент BASE FONT позволяет установить тип шрифта, его размер и цвет по умолчанию. Под типом шрифта понимается его название. Это может быть любой шрифт, установленный на локальном компьютере, которым можно отобразить данную страницу. Размер шрифта представляет собой целое число от 1 (наименьший) до 7 (наибольший размер). Цвет шрифта является стандартным HTML - цветом. Можно задать несколько видов шрифтов, но использован будет первый

найденный на данном компьютере.

Элемент FONT дает возможность задать шрифт определенного фрагмента текста. Он поддерживает те же атрибуты, что и BASEFONT (Face, Size, Color).

```
<BASEFONT FACE ="Courier New" Size= 3 Color = Blue>
```

Этот текст напечатан шрифтом Courier New синим цветом


```
<BASEFONT FACE ="Arial" Size= 3 Color = Red>
```

Этот текст напечатан шрифтом Arial красным цветом (размер=3)


```
<FONT SIZE =+1 >
```

Шрифт увеличен на один размер (размер = 4)


```
<FONT SIZE =+1 > Шрифт не изменился, потому что он меняется по отношению к BASEFONT (остался размер = 4)<BR>
```

```
<FONT SIZE =-1 >
```

Шрифт уменьшился сразу на два размера (размер = 2), потому что он меняется по отношению к BASEFONT


```
<FONT SIZE =5 > Размер шрифта стал = 5 <BR>
```

```
<font color="#000000" size="5" face="Times New Roman">А теперь мы изменили цвет и стиль шрифта на "Times New Roman"<br>
```

Практическая часть

1. Создание первой веб-страницы

Запустите NotePad и наберите (или скопируйте в него) следующий текст:

```
<HTML>
<HEAD>
<TITLE>Моя первая Веб-страница </TITLE>
</HEAD>
<BODY>
<H1> А здесь - текст о себе </H1>
</BODY>
</HTML>
```

Сохраните этот файл под именем SIMPLE.HTM.

Запустите Internet Explorer. Как только он будет запущен, введите путь к Вашему файлу в поле **Address** и нажмите клавишу **Enter**. Ваша первая страница отобразится в окне Internet Explorer.

2. Использование элементов

Полужирный текст и курсив

Вставьте в текст программы строки

```
<B> Этот текст напечатан полужирным шрифтом</B>
```

```
<B> Этот текст напечатан полужирным шрифтом и <I> курсивом</I></B>
```

3. Использование атрибутов

Вставьте в текст программы строки:

```
<P ALIGN=CENTER>Это предложение расположено по центру страницы </P>
```

```
<P ALIGN = RIGHT> Это предложение расположено справа. </P>
```

```
<P ALIGN = LEFT> Это предложение расположено слева. </P>
```

4. Использование комментариев

Все новые строки в программе сопровождайте комментариями.

Для ввода комментариев в текст программы, используя два способа

1 способ

```
<!-- Это начало комментариев.
```

```
А это конец комментариев -- >
```

2 способ

<COMMENT>

Это начало комментариев с использованием тега версии 2.0.

</COMMENT>

5. Использование цвета

Введите строку, обозначающую СИНИЙ цвет фона Вашей страницы, в начало основного тела программы (после элемента <BODY>):

```
<body bgcolor="blue">  
<h4><font color="#0000FF" size="2">
```

```
Last update: October 08, 2001 </font></h4>
```

```
<TABLE bgcolor="Ivory">
```

Варианты заданий:

1. Оформите свою веб-страницу в соответствии с Вашим вариантом (для справки см. таблицу 1)

№№ пп	Текст на странице	Шрифт	Цвет фона	Расположение текста на странице
1	ФИО, группа	Полужирный	Серый	В центре сверху
2	ФИО, группа	Курсив	Зеленый	В центре посередине
3	ФИО, группа	Нормальный	Синий	В центре внизу
4	ФИО, группа	Подчеркнутый	Белый	Слева сверху
5	ФИО, группа	Полужирный	Бежевый	Справа сверху
6	ФИО, группа	Курсив	Фиолетовый	Слева внизу
7	ФИО, группа	Нормальный	Серый	Справа внизу
8	ФИО, группа	Подчеркнутый	Зеленый	В центре внизу
9	ФИО, группа	Полужирный	Синий	В центре сверху
10	ФИО, группа	Курсив	Белый	В центре посередине
11	ФИО, группа	Нормальный	Бежевый	В центре внизу
12	ФИО, группа	Подчеркнутый	Фиолетовый	Слева сверху
13	ФИО, группа	Полужирный	Серый	Справа сверху
14	ФИО, группа	Курсив	Зеленый	Слева внизу
15	ФИО, группа	Нормальный	Синий	Справа внизу
16	ФИО, группа	Подчеркнутый	Белый	В центре внизу

ВНИМАНИЕ: Текст программы обязательно должен содержать комментарии.

1. Перемещение веб-узла с одного сервера на другой

Чтобы переместить веб-узел с одного сервера на другой, можно сделать так:

```
<!DOCTYPE HTML PUBLIC "-//IETF/DTD HTML //EN">
```

```
<HTML>
```

```
<HEAD>
```

```
<META HTTP-EQUIV="Refresh"
```

```
CONTENT="3"; URL=http://www.blackboard.com/>
```

```
<TITLE> We've moved </TITLE>
```

```
</HEAD>
```

```
<h2 align="center"> We've moved</h2>
```

Веб-страница курса "Web-технологии для создания Интернет-приложений поменяла сервер в Интернете. Пожалуйста, исправьте Ваши ссылки на новый адрес. </h2>

```
<a
```

```
href="http://coursesites.blackboard.com/bin/common/course.pl?course_id=_17938_1&frame=top">
```


<p align="center">
http://coursesites.blackboard.com/bin/common/course.pl?course_id=_17938_1&frame=top</p>

 <BODY>
 </HTML>

В этом примере указывается, что узел переместился, и поэтому используется опция **Refresh**, с помощью которой, после 3-х секундного ожидания, клиентский браузер будет переадресован на другой узел, указанный атрибутом **URL**.

Замечания:

- атрибут **URL** здесь вставлен в атрибут **CONTENT**, а адрес нового местоположения узла указан также в гиперссылке, заданной в документе, что дает возможность посетителям моментально перейти на новый узел одним щелчком мыши.
- Освоить этот прием нам необходимо еще и потому, что есть планы сменить сервер нашего виртуального курса.

Задание: Разберитесь с текстом предложенного примера, подставьте свои данные, добавьте комментарии (в соответствии с вариантом) Варианты заданий

№	Текст в заголовке	Старый URL	Новый URL	Расположение текста на странице
1	ФИО, группа	URL=http://www.blackboard.com/	http://education.vsnl.com/ruscult/education.htm	В центре сверху
2	ФИО, группа	URL=http://www.blackboard.com/	Свой адрес	В центре посередине
3	ФИО, группа	URL=http://www.blackboard.com/	http://www.anriintern.com/computer/itb/intro.asp	В центре внизу
4	ФИО, группа	http://education.vsnl.com/ruscult/education.htm	Свой адрес	Слева сверху
5	ФИО, группа	Свой адрес	http://education.vsnl.com/ruscult/education.htm	Справа сверху
6	ФИО, группа	http://education.vsnl.com/ruscult/education.htm	URL=http://www.blackboard.com/	Слева внизу
7	ФИО, группа	http://education.vsnl.com/ruscult/education.htm	http://www.anriintern.com/computer/itb/intro.asp	Справа внизу
8	ФИО, группа	Свой адрес	http://www.anriintern.com/computer/itb/intro.asp	В центре внизу
9	ФИО, группа	http://www.anriintern.com/computer/itb/intro.asp	Свой адрес	В центре сверху
10	ФИО, группа	Свой адрес	Свой адрес	В центре посередине
11	ФИО, группа	Свой адрес	Свой адрес	В центре внизу
12	ФИО, группа	Свой адрес	Свой адрес	Слева сверху
13	ФИО, группа	Свой адрес	Свой адрес	Справа сверху
14	ФИО, группа	Свой адрес	Свой адрес	Слева внизу
1	ФИО, группа	Свой адрес	Свой адрес	Справа внизу

5	группа			
1	ФИО,	Свой адрес	Свой адрес	В центре внизу
6	группа			

ВНИМАНИЕ: Текст программы обязательно должен содержать комментарии.

2. Создание сценария (Элемент **SCRIPT**)

Рассмотрим пример простой страницы с частью программы, написанной на VBScript и содержащейся в элементе **SCRIPT** (Её выполнение Вы можете посмотреть в приложении файл FirstScript.htm):

```

<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML//EN">
<html>
<head>
<meta http-equiv="Content-Type"
content="text/html; charset=windows-1251">
<meta name="GENERATOR" content="Microsoft FrontPage 2.0">
<title>Our First Script</title>
</head>
<body bgcolor="#FFFFFF">
<script language="VBScript">
<!--
document.write "<CENTER>"
document.write "<H2>" & "Здесь некоторая информация о Вашем браузере:" & "</H2>"
document.write "Наименование: " & Window.Navigator.AppName & "<BR>"
document.write "Версия: " & Window.Navigator.AppVersion & "<BR>"
document.write "Обозначение: " & Window.Navigator.AppCodeName & "<BR>"
document.write "Пользователь: " & Window.Navigator.UserAgent
document.write "</CENTER>"
document.close
-->
</script>
</body>
</html>

```

Замечание: Здесь нужно обратить внимание на несколько вещей. Первое: в элемент **HEAD** добавлена (с помощью элемента **SCRIPT**) **VBScript**-программа. Фактическая программа сценария заключена между тегами комментария, чтобы более ранние браузеры, не поддерживающие обработку сценариев, не выводили текст сценария на экран. Следует также заметить, что в теле документа не содержатся никакие данные, но при его загрузке в Internet Explorer появится информация, описывающая Ваш браузер.

Задание:

1. Наберите текст приведенного примера в редакторе (например, в Блокноте) и сохраните его под именем **FirstScript.htm**
2. Запустите **Internet Explorer** и откройте из него файл **FirstScript.htm**
3. Прочитайте полученную информацию о Вашем браузере.

При загрузке страницы начинает выполняться сценарий, динамически генерирующий **HTML**-данные и передающий их в браузер. В процессе загрузки программа также собирает информацию о браузере, в котором работает. Затем эти данные отображаются на экране. Специфику оператора **document.write** мы рассмотрим в одной из следующих лекций. А сейчас мы лишь хотим посмотреть, каким образом путем создания сценариев можно генерировать динамические Web-страницы.

Данные о Вашем браузере генерировались непосредственно при загрузке страницы. Однако, если появилась возможность писать программы, выполняемые при загрузке Web-страницы, хорошо бы еще иметь и пользовательские объекты, внедряемые в страницу. Язык **HTML** 3.2 позволяет внедрять в документы пользовательские объекты с помощью тегов (**OBJECT** и **APPLET**). Мы рассмотрим эти возможности несколько позже.

Задания:

1. Разберитесь с текстом предложенных примеров, подставьте свои данные, добавьте комментарии (в соответствии с вариантом)
2. Создайте веб-страничку со всеми рассмотренными примерами форматирования текста.
3. Посмотрите, какие команды реализованы в файле **Prim7-1.htm**. (чтобы увидеть HTML-код программы, надо в браузере IE открыть последовательно окна "Вид", "В виде HTML...". Используя атрибуты шрифтов, описанные в таблице 7.1., а также пример, приведенный в приложении к лекции в файле **format.htm**, создайте свой файл, в котором продемонстрируйте все возможности изменения характеристик шрифтов и элементов форматирования текста.

4. Преобразуйте лекцию №7 в веб-страницу, используя все освоенные атрибуты форматирования текста. Таблицы пока в веб-страницу не включать!

Варианты заданий

№№ пп	Текст на странице	Шрифт	Цвет фона	Расположение текста на странице
1	ФИО, группа	Полужирный	Серый	В центре сверху
2	ФИО, группа	Курсив	Зеленый	В центре посередине
3	ФИО, группа	Нормальный	Синий	В центре внизу
4	ФИО, группа	Подчеркнутый	Белый	Слева сверху
5	ФИО, группа	Полужирный	Бежевый	Справа сверху
6	ФИО, группа	Курсив	Фиолетовый	Слева внизу
7	ФИО, группа	Нормальный	Серый	Справа внизу
8	ФИО, группа	Подчеркнутый	Зеленый	В центре внизу
9	ФИО, группа	Полужирный	Синий	В центре сверху
10	ФИО, группа	Курсив	Белый	В центре посередине
11	ФИО, группа	Нормальный	Бежевый	В центре внизу
12	ФИО, группа	Подчеркнутый	Фиолетовый	Слева сверху
13	ФИО, группа	Полужирный	Серый	Справа сверху
14	ФИО, группа	Курсив	Зеленый	Слева внизу
15	ФИО, группа	Нормальный	Синий	Справа внизу
16	ФИО, группа	Подчеркнутый	Белый	В центре внизу

ВНИМАНИЕ: Текст программы обязательно должен содержать комментарии.

Литература

1. Том Армстронг. ActiveX: создание Web-приложений. Пер. с англ. Н.Б.Шишкиной – ВНУ, Киев, 1998 – 590 стр.
2. Галямова Е.В. Лекция №5 по курсу "WEB-технологии для создания Интернет-приложений"

Справочный материал для выполнения лабораторной работы:

1. Таблица цветов в файле [all_named_color.htm](#) (информация, предоставленная студентом Рывкиным Я.)
2. Таблица безопасных цветов в файле [all_color.htm](#) (информация, предоставленная студентом Рывкиным Я. Взято из <http://www.design.ru/free/colors/> - в 2001 так называлась студия Артемия Лебедева, сейчас их адрес <http://www.artlebedev.ru/tools/>)
3. Список атрибутов см. в Таблице 1

Таблица 1. Элементы форматирования текста

Элемент	Тег	Эффект
Абзац	P	Обозначает начало и конец абзаца. Текст, находящийся между этими тегами, автоматически разбивается браузером на строки
Разрыв строки	BR	Явно разрывает строку
Отмена разрыва строки	NOBR	Запрещает разрыв строки
Разбивка на подразделы	DIV	Играет роль контейнера для последовательностей элементов. Например, позволяет установить элемент

		ALIGN для нескольких абзацев сразу
Заголовок	H1 - H6	Стили заголовков. H1 использует наибольший шрифт, а H6 —наименьший. Элементы H задают также выделение текста полужирным шрифтом
Полужирный	B or STRONG	Выделяет текст полужирным шрифтом
Курсив	I or EM	Выделяет текст курсивом.
Подчеркивание	U	Выделяет текст подчеркиванием
Отступы	BLOCKQUOTE	Задаёт отступы справа и слева
Верхний или нижний индекс	E	
Верхний или нижний индекс	SUB and SUP	Преобразуют текст в верхние и нижние индексы соответственно
Шрифт меньшего или большего размера	SMALL or BIG	SMALL задает размер шрифта на один пункт меньше текущего значения, а BIG — на один пункт больше. Теги аналогичны элементам and
Моноширинный шрифт, HTML-теги не обрабатываются	PLAINTEXT	Полезен для отображения на экране HTML-программы. Текст отображается моноширинным шрифтом, а HTML-код, заключенный между тегами, не обрабатывается
Моноширинный Шрифт	CODE	Элемент CODE задает использование моноширинного шрифта. Это полезно при отображении текстов программ
Моноширинный Шрифт	PRE	Элемент PRE определяет, что следующий участок текста уже отформатирован и должен отображаться "как есть"
Шрифт по умолчанию	BASE FONT	Задаёт тип, размер и цвет шрифта по умолчанию
Шрифт	FONT	Задаёт тип, размер и цвет шрифта для фрагмента текста