

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ  
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«ОРЕНБУРГСКИЙ ГОСУДАРСТВЕННЫЙ АГРАРНЫЙ УНИВЕРСИТЕТ»**

**МЕТОДИЧЕСКИЕ МАТЕРИАЛЫ ДЛЯ ОБУЧАЮЩИХСЯ  
ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ**

Б1.В.ДВ.05.02 WEB-программирование

**Направление подготовки** 09.03.01 Информатика и вычислительная техника

**Профиль образовательной программы** Автоматизированные системы обработки информации и управления

**Форма обучения** заочная

## СОДЕРЖАНИЕ

<b>1. Конспект лекций .....</b>	
<b>1.1 Лекция № 1 Обзор Web- технологий.....</b>	
<b>1.2 Лекция № 2 Язык разметки гипертекста – HTML.....</b>	
<b>2. Методические материалы по выполнению лабораторных работ .....</b>	
<b>2.1 Лабораторная работа № 1 ЛР-1 Проектирование Web - сайта.</b>	
<b>2.2 Лабораторная работа № 2 ЛР-2 Структура HTML документа.</b>	
<b>2.3 Лабораторная работа № 3 ЛР-3 Форматирование текста. Интерактивные формы HTML.</b>	
<b>2.4 Лабораторная работа № 4 ЛР-4 Введение в JavaScript. Синтаксис языка. Объектная модель JavaScript.</b>	
<b>2.5 Лабораторная работа № 5 ЛР-5 Обработка событий. Примеры эффективного программирования на JavaScript.</b>	

## 1. КОНСПЕКТ ЛЕКЦИЙ

### 1.1. Лекция № 1 (2 часа)

**Тема: «Обзор Web- технологий»**

#### 1.1.1. Вопросы лекции:

1. Основные понятия и определения.
2. Сдерживающие факторы развития web-технологий.
3. Клиентские Web-технологии.

#### 1.1.2 Краткое содержание вопросов:

1. Web-программирование - это отдельное направление в программировании, используемое для создания web-приложений. В связи с ускоренными темпами развития Интернет технологий постоянно появляются новые направления в этой области. Поэтому Web-программирование не сформировавшаяся наука, а скорее набор существующих программных технологий (клиентских и серверных), используемых для организации работы пользователя в сети Интернет. Сервер – компьютер, на котором находятся документы и приложения. Клиент – рабочая станция, используемая для просмотра документов и приложений, хранящихся на сервере.

2. Сдерживающие факторы развития web-технологий:

- ограничения, связанные с сетью Интернет (скорость);
- отсутствие централизованного управления Интернет (конкуренция браузеров);
- ограничения, связанные с технологиями программирования.

3 Интерпретатор команд обрабатывает код приложения и выдает браузеру сформированный документ. Преимуществом такой архитектуры являются возможность работы с динамическими документами (появляется интерактивность), а также возможность однажды загрузив приложение работать в нем, не обращаясь к серверу.

### 1.2. Лекция № 2 (2 часа)

**Тема: «Язык разметки гипертекста – HTML»**

#### 1.2.1. Вопросы лекции:

1. Элементы Web-страницы.
2. Общая структура простейшего документа html.

#### 1.2.2 Краткое содержание вопросов:

**Web-страница** представляет собой набор элементов.

**Тег (Tag)** — это стартовый и конечный маркеры элемента. Теги определяют границы действия элементов и отделяют элементы друг от друга. Теги заключаются в угловые скобки, например: **<HTML>**. Конечный тег, если он предусмотрен синтаксисом языка, снабжается косой чертой: **</HTML>**.

**Гиперссылка** (Hyperlinks) — фрагмент текста, который является указателем на другой файл или объект. Гиперссылки позволяют переходить от одного документа к другому.

**Фрейм** (Frame) — область гипертекстового документа со своими полосами прокрутки. Фреймы позволяют разбивать страницы на прямоугольные области, в каждой из которых отображается своя собственная страница. Вы можете разместить один или несколько

фреймов на странице (такая страница называется страницей фреймов или фреймсет (frameset)).

**Форма** (Form) — область гипертекстового документа, которая необходима для организации обратной связи с посетителями сайта.

**Апплет** (Applet) — небольшая прикладная программа, передаваемая на компьютер клиента в виде отдельного файла и запускаемая при просмотре Web-страницы.

**Скрипт** (Script) — программа, включенная в состав Web-страницы для расширения ее возможностей.

**Загрузка** (DownLoad) — копирование документа с Web-сервера на компьютер клиента.

**Размещение** (UpLoad) — копирование документа с компьютера клиента на Web-сервер — используется при создании собственной Web-страницы (т.е. при ее опубликовании).

**Пиксел** (Pixel) — наименьшая цветная точка экрана монитора.

### **Общая структура простейшего документа html**

Общая структура простейшего документа HTML такова:

**<COMMENT>**Комментарий**</COMMENT>**

**<HTML>**

**<HEAD>**

**<TITLE>**Название документа**</TITLE>**

**</HEAD>**

**<BODY>**

Здесь расположен текст самого документа HTML.

**</BODY>**

**</HTML>**

Дадим пояснения указанным тегам документа HTML.

**<COMMENT>**— комментарий к документу. Не является обязательным.

**<HTML>**— идентификатор всего блока HTML-команд.

**<HEAD>**— идентификатор заголовка документа **HTML**.

**<TITLE>**— идентификатор заголовка окна просмотра.

**<BODY>**— идентификатор **HTML-команд** документа для просмотра.

Три основных тега **<HTML>**, **<HEAD>** и **<BODY>** передают браузеру основную информацию для идентификации и организации документа.

Все указанные теги — парные, т.е. каждый из них заканчивается конечным тегом с косой чертой. Все команды можно писать как с маленькой, так и с большой буквы.

**<META>**— непарный тег применяется для указания подробной информации о документе.

**<P>**— идентификатор конца абзаца, после которого добавляется пустая строка. В конце нужен тег **</P>**.

**<BR>**— идентификатор перевода строки.

**<HR>**— идентификатор изображения горизонтальной линии.

Эти теги одиночные, т.е. непарные, они не требуют тегов с косой чертой.

**<PRE>**— установка равноширинного шрифта. В конце нужен тег **</PRE>**.

### **Парные теги форматирования заголовков и подзаголовков документа**

**<H1>**, **<H2>**, **<H3>**, **<H4>**, **<H5>**, **<H6>**. При этом заголовки будут выведены большими буквами, причем размер букв у тега **<H1>** будет самый большой, у **<H2>** — меньше, у **<H3>** — еще меньше и т.д.

### **Парные теги форматирования символов текста**

**<B>**— идентификатор полужирного шрифта.

**<strong>**— идентификатор выделенного шрифта. **<i>**— идентификатор курсива. **<U>**— идентификатор подчеркивания. **<s>**— идентификатор перечеркивания.

**<tt>**— идентификатор равноширинного шрифта (телетайпного или курьера).

**<big>**— задает увеличенный размер шрифта. **<small>**— задает уменьшенный размер шрифта. **<center>**— задает центрирование текста. **<sub>**— задает нижний индекс.

**ТЕГИ ФОРМАТИРОВАНИЯ АБЗАЦЕВ** **<p align=left>**— выравнивание текста в абзаце по левому краю. **<p align=right>**— выравнивание текста в абзаце по правому краю.

**<p align=center>**— выравнивание текста в абзаце по центру. **<p align=justify>**— полное выравнивание по обоим краям экрана. **align** — атрибут выравнивания.

### **Теги списков**

Теги списков являются способом наглядного отображения структурированной информации. Упорядоченные списки служат для отображения последовательных операций или алгоритмов. Браузер автоматически генерирует номера для каждого пункта в списке.

**<OL>**— идентификатор упорядоченного списка. В конце**</OL>**.

Неупорядоченные списки служат для составления перечней, когда порядок следования пунктов несущественен. Перед каждым элементом будет стоять маркер.

**<UL>**— идентификатор неупорядоченного списка. В конце**</UL>**.

Отдельные элементы в упорядоченном и неупорядоченном списках помечают одиночным тегом **<LI>**, а элементы в списках определений **<DL>** тегами **<DT>** для термина и **<DD>** для значения термина.

**<LI>**— идентификатор элемента в упорядоченном и неупорядоченном списке.

Конечный тег **</LI>** может быть опущен.

Проектирование WEB – САЙТОВ.

Создание WEB- сайтов средствами языка HTML/ Теги списков определений. Теги цвета. Теги вставки изображений. Тег перехода к другому файлу. Тег перехода к другой странице. Тег вставки адреса Электронной почты. Тег вставки документа. Теги для создания бегущей строки. Список цветов символов HTML. Теги подключения звука. Теги фреймов. Создание таблиц в HTML.

### Теги списков определений

Списки определений обеспечивают специальное форматирование, как в словарях, для терминов и связанных с ними описаний.

**<DL>**— идентификатор списка определений. В конце**</DL>**.

**<DT>**— идентификатор термина в списке определений.

**<DD>**— идентификатор значений термина в списке определений. **<DL>**

**<DT>**название термина 1

**<DD>**определение термина 1

**<DD>**другое определение термина 1

**</DL>**

### Теги цвета

**<BODY bgcolor=«teal» text=«aqua» link=«red»>**— задает цвет текста(*text*), цвет фона(*bgcolor*) и цвет гипертекстовой ссылки(*link*).

**<BODY background=«back.jpg»>**— задает фоновый рисунок (обои) на Web-странице в файле back.jpg.

**<BODYbackground=«\windows\tartan.bmp»>**— фоновый рисунок (обои) взят из файла tartan.bmp, причем указан путь этого файла.

**<font color=«yellow» size=5>**— задает цвет символов текста(*color*) и их размер(*size*)- В конце нужен тег **</font>**.

еги линий

**<hr color=«lime»>**— задает цвет горизонтальной линии.

**<hr color=«red» size=3 width=220 align=center>**— атрибут **size=N** задает толщину линии в пикселях; атрибут **width=M** задает длину линии в пикселях; атрибут **align=center** (или **left**, или **right**) задает смещение линии (ее размещение в центре, или смещение влево, или вправо).

### Теги для вставки изображений

Графика на страницах — это средство выражения мысли, подчеркивания идеи.

Изображения можно сохранять в нескольких форматах. Тип файлов **GIF**— графический формат обмена, стал первым типом файлов, которые поддерживались в **WWW**. Затем был разработан формат **JPEG**— объединенная группа экспертов фотографии. Это форма эффективнее для представления больших изображений.

Приобрел популярность формат **PNG**— персональная сетевая графика, который заменяет **GIF**.

При публикации сайта рисунок должен находиться в папке, где хранятся файлы страниц, а **URL-адрес** должен быть реальным.

**JPEG** использует специальную технику компрессии изображений, что является преимуществом для рисунков и фотографий.

**<IMG SRC=«lycos.gif»>**— вставка графического изображения в виде анимационного файла **lycos.gif**. Можно также использовать и графические файлы формата **\*.jpg** (или **\*.jpeg**), **\*.bmp**.

**<IMG SRC=«echomsk.gif» width=«403» height=«263»>**— если при вставке изображения использованы атрибуты **width** (ширина в пикселях) и **height** (высота в пикселях), то при загрузке изображения браузер покажет сначала рамку, где должно быть изображение, а затем уже само изображение, поскольку оно загружается дольше. Таким образом, резервируется место на экране под изображение.

## 2. МЕТОДИЧЕСКИЕ МАТЕРИАЛЫ ПО ВЫПОЛНЕНИЮ ЛАБОРАТОРНЫХ РАБОТ

### 2.1 Лабораторная работа № 1 (2 часа).

**Тема:** «Проектирование Web - сайта»

#### 2.1.1 Задание для работы:

1. Способы создания Web-страниц.

#### 2.1.2 Краткое описание проводимого занятия:

**Web-сайт** —набор**связанных** между собой, близких по смыслуWeb-страниц и файлов. На каждомWeb-сайте существует однаWeb-страница называемая домашней (homepage) или главной страницей. Все посетителиWeb-сайта сначала попадают на нее, а потом с помощью гиперссылок попадают на другие страницыWeb-сайта,

Web– страница (документHTML) представляет собой текстовый файл на языкеHTMLформата ".htmlили \*.html, размещенный вWWW.Web-страница кроме текста может содержать графику, анимации, видеоклипы, музыку, а также гипертекстовые ссылки, при помощикоторых можно переходить к другимWeb-страницам и просматривать их.

Один из способов создания Web-страниц (или документовHTML) состоит в использовании текстового редактора **Блокнот** (NotePad), встроенного в Windows. В этом редакторе создается файл Web-страницы, который сохраняется с расширением \*.htm с последующим просмотром результатов с помощью браузера. После сохранения файла и выхода из **Блокнота** для просмотра отредактированной страницы надо выполнить команду **Файл — Открыть** в программе **Internet Explorer** и указать путь к созданной Web-странице с помощью кнопки **Обзор**.

### 2.2 Лабораторная работа № 2 (2 часа).

**Тема:** «Структура HTML документа»

#### 2.2.1 Задание для работы:

1. Структура документа;
2. Структура и параметры тегов.

#### 2.2.2 Краткое описание проводимого занятия:

1. HTML-документ заключается в теги **<html>** и **</html>**. Между этими тегами располагаются два раздела: раздел заголовка (элемент **head**) и раздел тела документа (элемент **body** для простого документа либо элемент **frameset**, задающий набор кадров). Все указанные элементы имеют начальный и конечный тег.

2. Секция заголовка содержит описание параметров, используемых при отображении документа, но не отражающихся непосредственно в окне браузера. Секция тела документа содержит текст, предназначенный для отображения браузером и элементы, указывающие на способ форматирования текста, определяющие графическое оформление документа, задающие параметры гиперссылок и так далее.

### 2.3 Лабораторная работа № 3 (2 часа).

**Тема:** «Форматирование текста. Интерактивные формы HTML»

#### 2.3.1 Задание для работы:

1. Форматирование текста: представление текстовой информации;
2. Escape-последовательности;
3. Формы HTML: организация формы;
4. Основные параметры формы.



### 2.3.2 Краткое описание проводимого занятия:

1. Все компьютерные файлы можно разделить на две категории: текстовые файлы (*plaintext*) и двоичные (бинарные) файлы. Текстовые файлы содержат обычные символы, такие как буквы, цифры, знаки препинания и некоторое количество управляющих символов, таких как знаки табуляции и перевода строки. Символы обычно кодируются с помощью *ASCII-кодировки* или её расширений для различных языков. *Кодировка* – совокупность всех уникальных символов, которые способна распознать операционная система.

2. Сочетания символов, состоящих из косой черты (\), за которой следует буква или набор цифр, называются *escape-последовательностями*. Для представления знака новой строки, одиночной кавычки или некоторых других символов в символьной константе, необходимо использовать *escape-последовательности*. *Escape-последовательность* рассматривается как один символ и, следовательно, является допустимой символьной константой. *Escape-последовательности* обычно используются для указания действий, например возврата каретки или табуляции, на терминалах и принтерах. Они также используются для предоставления буквенных представлений непечатаемых символов, а также символов, которые обычно имеют специальное значение, например двойных кавычек ("). В следующей таблице перечислены *escape-последовательности ANSI* и представляемые ими значения. Обратите внимание, что вопросительный знак, перед которым стоит обратная косая черта (\?), обозначает литерал вопросительного знака в таких случаях, когда последовательность символов может быть ошибочно интерпретирована как триграф. Дополнительные сведения см. в разделе Триграфы.

Форма (англ. form) в HTML — раздел документа, позволяющий пользователю вводить информацию для последующей обработки системой. Синтаксически форма в HTML задаётся с помощью элемента `<form>` и в дополнение к разметке обычных элементов содержит разметку для элементов управления (англ. controls), надписей.

Элементы управления служат для взаимодействия пользователя с формой. Атрибут `name` определяет имя элемента управления с областью видимости внутри данной формы.

С каждым элементом формы связано начальное и текущее значение. За некоторыми исключениями (`<textarea>`, `<object>`), начальное значение может быть задано атрибутом `value`. Значения, соответствующие элементам, могут изменяться при взаимодействии пользователя или скриптов (например, на JavaScript) с формой. При очистке (англ. reset) формы элементы приобретают начальные значения. Данные всех действующих (англ. successful) элементов формы отправляются (англ. submit) на обработку в виде пар имя-значение.

### 2.4 Лабораторная работа № 4 (2 часа).

**Тема: «Введение в JavaScript. Синтаксис языка. Объектная модель JavaScript»**

#### 2.4.1 Задание для работы:

1. Объектная модель: модель DOM. Синтаксис языка JavaScript.

#### 2.4.2. Краткое описание проводимого занятия:

Большинство действий в javascript выполняется с HTML-страницей. В javascript страница представлена в виде объектной модели DOM (DocumentObjectModel).

Любые действия со страницей требуют вызова соответствующего метода DOM.

Понимание, как работать с документом в модели DOM - краеугольный камень в javascript-программировании.

Введение. DOM в примерах.

Основным инструментом работы и динамических изменений на странице является DOM (DocumentObjectModel) - объектная модель, используемая для XML/HTML-документов.

Согласно DOM-модели, документ является иерархией. Каждый HTML-тег образует отдельный элемент-узел, каждый фрагмент текста - текстовый элемент, и т.п.

## **2.5 Лабораторная работа № 5 (2 часа).**

**Тема: «Обработка событий. Примеры эффективного программирования на JavaScript»**

### **2.5.1 Задание для работы:**

1. Примеры эффективного программирования на JS.

### **2.5.2.Краткое описание проводимого занятия:**

JavaScript обычно используется как встраиваемый язык для программного доступа к объектам приложений. Наиболее широкое применение находит в браузерах как язык сценариев для придания интерактивности веб-страницам (встраивается в HTML)

Основные архитектурные черты JavaScript: динамическая типизация, слабая типизация, автоматическое управление памятью, прототипное программирование, функции как объекты первого класса.