

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
Политехнический колледж федерального государственного
бюджетного
образовательного учреждения высшего образования
«Майкопский государственный технологический университет»
в поселке Яблоновском

Методические указания
к выполнению практической работы
для студентов очной формы обучения
по дисциплине **ОП.12 Web-технологии.**
«Создание сайта»
специальность 09.02.03 -
«Программирование в компьютерных системах»
для студентов очной формы обучения

Яблоновский, 2018

УДК 004.91(07)

ББК 73

М-54

Web-технологии. «Создание сайта»: методические указания к выполнению лабораторной работы для студентов очной формы обучения специальности 09.02.03 - «Программирование в компьютерных системах»

Одобрено предметной (цикловой) комиссией информационных и математических дисциплин.

Протокол №1 т 31.08.2018 г.

Председатель предметной (цикловой) комиссии

А.А.Схаплок

Разработчик: Гониволк Н.В. - преподаватель политехнического колледжа филиал федерального государственного бюджетного образовательного учреждения высшего образования «Майкопский государственный технологический университет» в поселке Яблоновском.

Содержание

1. Цели работы.....	4
2. Теоретическая часть.....	4
2.1 HTML-редактор Adobe Dreamweaver.....	5
2.2. Создание сайта.....	7
2.2.1. Создание заглавной страницы.....	8
2.2.2 Форматирование HTML-страниц.....	9
2.2.3 Создание гипертекстовых ссылок.....	17
2.2.4 Создание и форматирование таблиц.....	19
3. Вставка изображений и других элементов.....	21
3.1 Использование графических материалов.....	21
3.2. Порядок выполнения работы.....	23
3.3. Основные требования к интернет-сайту.....	23
4. Список тем для сайтов.....	26
5. Вопросы для самопроверки.....	28
6.Список рекомендуемой литературы.....	28

1. Цели работы

Целями практической работы являются:

1. Изучение методики создания сайтов.
2. Овладение практическими навыками в работе с программой Adobe Dreamweaver.
3. Создание HTML страниц.

2. Теоретическая часть

В основу всемирной паутины положена идеология и технология гипертекста, технически реализованная при помощи языка HTML. HTML (от англ. HyperText Markup Language, язык гипертекстовой разметки) представляет собой язык, разработанный специально для создания web-документов. HTML-документы представляют собой обычные текстовые ASCII-файлы. Помимо текстового содержания, они включают в свой состав специальные управляющие конструкции языка - так называемые теги. При отображении документа в браузере сами теги на экран не выводятся. Они определяют структурные единицы внутри документа, задают параметры форматирования, включают в состав документа новые элементы (встроенные изображения, Java-апплеты и т.п.) и позволяют создавать гипертекстовые ссылки на другие документы и ресурсы сети Internet.

Для создания и редактирования HTML-документов достаточно простого текстового редактора, например Блокнот Windows. На практике обычно используют средства редактирования, разработанные специально для написания HTML - редакторы WYSIWYG (от англ. what you see is what you get, что видишь, то и получаешь) или, как их еще называют, средства визуального редактирования. Такие программные продукты имеют графические интерфейсы, делающие написание HTML-кода

подобным работе с программой редактирования текстов и разметки страниц. Подобные программные продукты позволяют повысить эффективность и значительно сократить время разработки, особенно если речь идет о средних и крупных сайтах. Специализированные HTML-редакторы также работают с исходным кодом web-страницы, но располагают различными вспомогательными инструментами, ускоряющими и упрощающими процесс редактирования документов, и дополнительными средствами для выполнения повторяющихся операций. Кроме того, наиболее мощные из них включают в свой состав средства автоматизации процесса разработки (работа с шаблонами, стилями и библиотеками элементов) и средства размещения файлов сайта на web-сервере. К данной группе редакторов относится большое количество программных продуктов, в том числе и распространяемых бесплатно.

2.1. HTML-редактор Adobe Dreamweaver

Dreamweaver HTML-редактор разработан и поддерживался компанией Macromedia в версиях до 8-й

(2005 год

http://ru.wikipedia.org/wiki/2005_%D0%B3%D0%BE%D0%B4>)

Следующие версии (начиная с Dreamweaver CS3 (2007)) выпускает Adobe. Богатый инструментарий, открытость приложения для всевозможных настроек, удобный интерфейс и другие особенности сделали Dreamweaver одним из наиболее популярных HTML-редакторов в мире. У данного редактора имеется и русскоязычный интерфейс. На сегодняшний момент последней версией является

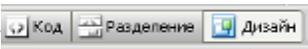
Dreamweaver CS4

<<http://www.adobe.com/ru/products/dreamweaver/>>.

Создатели программы Dreamweaver постарались сделать удобный и простой в использовании инструмент, облегчающий процесс создания HTML-страниц, их редактирования и управления большим количеством связанных друг с другом документов.

Каждый новый HTML-документ открывается в новой закладке. Для более удобной работы с программой предусмотрены система меню и набор палитр инструментов.

Вы можете настраивать рабочее пространство программы таким образом, чтобы ваша работа с ней была максимально удобной и эффективной. Программа Dreamweaver, являясь визуальным HTML-редактором, также позволяет переключаться в режим ручного редактирования кода, выполняя взаимосвязь между визуальным представлением страницы и ее исходным кодом. Для переключения между режимами отображения и редактирования HTML-документа предназначены три первые

кнопки  () на панели инструментов.

Все основные действия при создании и редактировании HTML-страницы выполняются в поле документа. Как известно, кроме текстовой информации, HTML-документ может содержать различные графические и мультимедийные элементы. Для удобства вставки таких элементов предназначена панель объектов **Вставка** (рис. 1). С помощью кнопок, расположенных на этой панели, можно быстро вставить различные объекты (изображения, таблицы, Flash-анимация) в редактируемый документ.

2.2.1. Создание заглавной страницы

Для создания страницы необходимо открыть новый, пустой документ. Это можно сделать различными способами, например, выбрав команду **File=>New** в главном меню, а затем выбрав HTML из списка. Первое, что необходимо сделать после создания документа, сохранить файл с необходимым именем (иначе при создании гипертекстовых ссылок, вставке изображений и т.п. могут быть использованы неверные адреса). Главная страница сайта должна иметь специальное имя - index.html (или index.htm). После того как странице присвоено имя и она сохранена, можно приступить к наполнению ее информационным содержанием. Начать следует с установки параметров страницы. Для этого необходимо из главного меню выбрать команду **Modify=>Page Properties....** Основное, что необходимо задать в открывшемся диалоговом окне, это название (заголовок) страницы - поле **Title** в категории **Title/Encoding** (рис. 2). Каждая страница сайта должна иметь продуманное, кратко и четко сформулированное название. Визуально оно будет отображено в заголовке окна браузера и, кроме того, будет использоваться поисковыми системами и т.п.

Кроме заголовка, в категории **Appearance** этого диалогового окна можно установить следующие параметры страницы:

- **Background image** - изображение, которое будет использовано в качестве фона страницы (графический файл с изображением можно выбрать, нажав кнопку **Browse...**);

- **Background / Text** - соответственно цвета фона страницы и текста, расположенного на ней (можно выбрать цвет из палитры, нажав на кнопке рядом);

- **Links / Visited links / Active links** - соответственно цвета ссылок / посещенных ссылок / активных ссылок;

• **Left margin / Top margin / Margin width / Margin height** - задают отступы (поля) по бокам и в начале, и конце страницы.

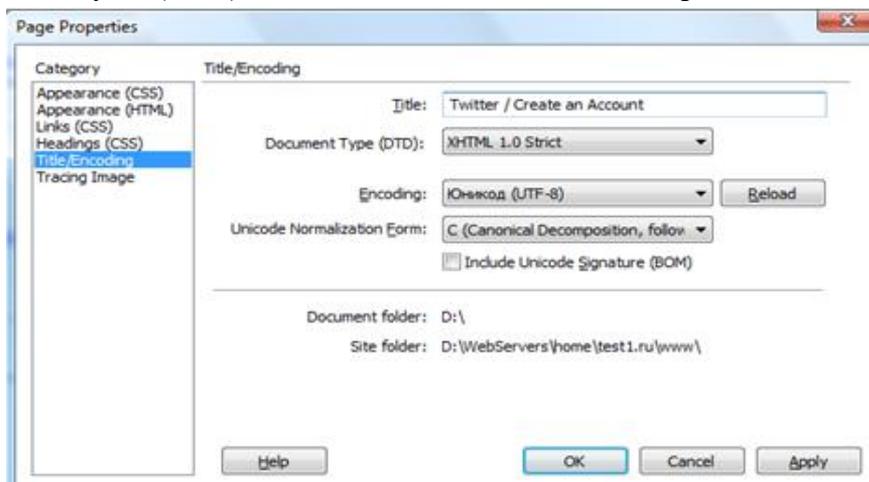


Рис. 2. Настройка параметров HTML-страницы

2.2.2. Форматирование HTML-страниц

В HTML-документе необходимо разделять данные документа и их представление. **Содержание документа задается в виде тэгов HTML.** Для определения же визуального представления документа следует использовать **таблицы стилей (CSS - Cascading Style Sheets)** (через CSS задаются интервалы между строками текста, отступы, цвета, используемые для текста и фона, размер и стиль шрифтов и другое оформление).

Существует несколько основных тэгов HTML, которые должны присутствовать в тексте любой web-страницы. Каждая web-страница обязана содержать тэг `<HTML>`, располагаемый в самом начале (каждый из тэгов имеет открывающую (например, `<HTML>`) и закрывающую части (например, `</HTML>`)). Этот тэг описывает документ как web-страницу, выполненную в формате

HTML. Непосредственно за тэгом <HTML> обычно следует тэг <HEAD>, который указывает на наличие текста, содержащего наименование страницы и дополнительные сведения о ней. В раздел HEAD обычно вложен тэг <TITLE>, служащий для обозначения наименования страницы, (наименование страниц обычно отображается в строке заголовка окна браузера). Затем следует тэг <BODY>, который указывает на начало "тела" web-страницы. В этом разделе размещается весь текст, отображаемый браузером, графика, таблицы и другие элементы содержимого страницы, которые увидит посетитель, обратившийся к сайту.

Пример HTML-документа:

```
<html>  
<head><title> Online обучение: HTML</title></head>  
<body>
```

Текст сайта

```
</body>  
</html>
```

Любой текст разделяется на последовательность абзацев. Для того чтобы начать новый абзац (при редактировании страницы в режиме дизайна), как и в большинстве текстовых редакторов, достаточно нажать клавишу **Enter**. При этом в исходный текст HTML-документа автоматически вставляются открывающий и закрывающий теги <P>. Для создания принудительного перехода на новую строку используется комбинация клавиш **Shift+Enter**. В исходный код в этом случае будет добавлен тег
.

Как правило, текст документа должен содержать заголовки. Причем это будет не просто выделенный крупным шрифтом, цветом или каким-либо другим образом текст, а именно структурная единица - заголовок. В языке HTML существует шесть уровней заголовков: тег <H1> задает наиболее важный

(заголовок первого уровня) и <H6> - наименее важный (заголовок шестого уровня). При использовании этих элементов следует соблюдать их иерархию.

Общим заголовком документа должен быть заголовок первого уровня. Не следует пропускать уровни, используя для заголовков разделов сразу заголовки третьего или четвертого уровня.

Для каждого блока текста можно установить его тип, то есть сделать этот текст заголовком любого уровня или форматированным текстом. Проще всего это делать, используя панель свойств объекта (рис. 3). Если панели нет на экране, то можно ее открыть, выполнив команду **Window=>Properties** (кроме того, указать тип блока текста (абзаца) можно в меню **Text=>Paragraph Format** или в разделе **Paragraph Format** контекстного меню, появляющегося при щелчке правой кнопкой мыши в поле документа).



Рис. 3. Панель свойств объектов

Выделив блок текста или просто установив курсор в любое место абзаца, можно выбрать его тип в поле **Format**. В соответствии с элементами языка HTML возможны следующие варианты: **Paragraph** - обычный абзац (ограничен тегами <P>), **Heading 1...6** - заголовки соответствующего уровня (ограничены тегами <H1>...<H6>), **Preformatted** - предварительно форматированный текст (тег <PRE>) или **None** - блок текста перестает быть абзацем.

Использование тегов визуального форматирования является нежелательным.

В Dreamweaver для форматирования текста можно использовать каскадные таблицы стилей (CSS, Cascade Style Sheet). Преимущество CSS - удобство редактирования дизайна всего сайта путем изменения всего одного файла CSS и сокращения объема сайта из-за выноса оформления в отдельный файл CSS.

Таблица стилей (CSS) состоит из набора *правил*. Каждое правило, в свою очередь, состоит из одного или нескольких *селекторов*, разделённых запятыми, и *блока определений*. Блок определений обрамляется фигурными скобками и состоит из набора свойств и их значений. Схематически это можно показать так:

```
селектор {  
    свойство1: значение;  
    свойство2: значение;  
    ...  
    свойствоn: значение;  
}
```

Dreamweaver позволяет редактировать правила CSS с использованием специальных панелей и диалоговых окон, которые представляют правила в наглядной форме.

Для отображения панели стилей нужно выполнить команду **Window=>CSS Styles** или нажать на кнопку **CSS Styles** в навигаторе панелей (рис. 4).

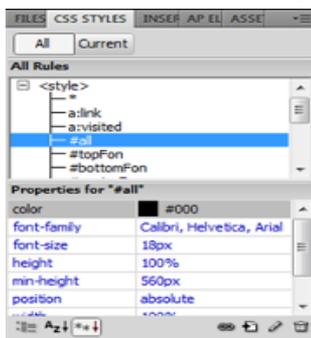


Рис. 4. Панель стилей

Для создания нового стиля (правила) надо нажать кнопку **New CSS Rule** на панели стилей или выбрать команду **New...** из контекстного меню, появляющегося при щелчке правой кнопкой мыши в любом месте панели стилей.

В открывшемся диалоговом окне (рис. 5) необходимо указать тип создаваемого стиля - список **Selector Type. Class (can apply to any HTML element)** создает именованный стиль, который может быть применен к различным блокам текста. При этом в поле **Selector Name** необходимо указать имя - идентификатор стиля (CLASS). Оно должно начинаться с символа точка, являющегося идентификатором класса. **Tag (redefine an HTML element)** переопределяет параметры форматирования определенного тега, который необходимо выбрать в списке **Tag. ID (applies to only one HTML element)** позволяет определить форматирование для тегов, содержащих особый идентификатор. Идентификаторы используются аналогично именованным стилям, но предназначены для форматирования элементов, требующих индивидуальной обработки. В остальных случаях предпочтительным является использование именованных стилей (CLASS).

Далее необходимо определить, где будет расположено описание стилей - список **Rule Definition**. При выборе **New Style Sheet File** таблица стилей будет сохранена в отдельном файле, ссылка на который будет вставлена особым образом в заголовок создаваемого HTML-документа (если с редактируемой страницей уже связаны какие-либо файлы с описанием стилей, то их имена будут перечислены в списке). При таком подходе одна и та же таблица стилей может использоваться в различных документах. Это особенно удобно, когда надо определить элементы форматирования, общие для всех страниц сайта. Во втором случае (**This Document Only**) описание стилей будет сохранено непосредственно в редактируемом документе (и соответственно может быть использовано только в его пределах). Такой вариант

удобен для создания стилей, специфичных для данной страницы.

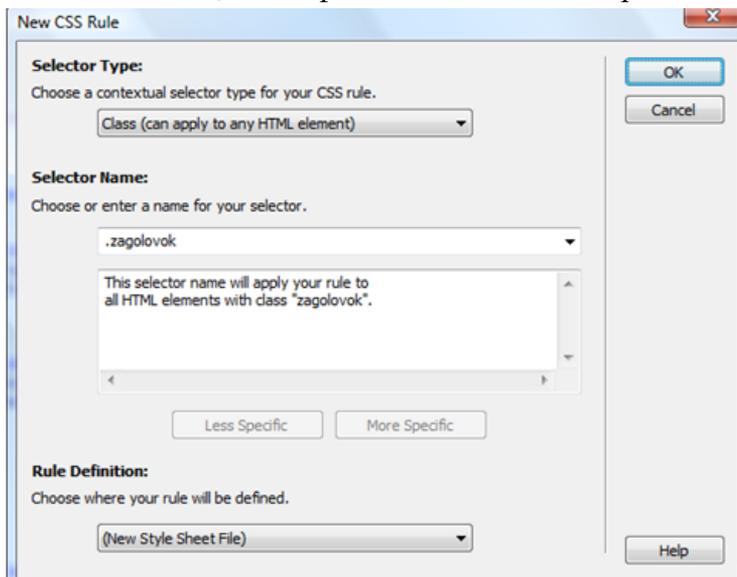


Рис. 5. Создание нового стиля

После того как все параметры определены, можно нажать ОК, после чего откроется диалоговое окно определения параметров нового стиля (рис.6). В левой его части в списке **Category** можно выбирать группы параметров форматирования, которые необходимо определить.

Type - параметры форматирования шрифта. В поле **Font-family** можно задать гарнитуру шрифта. Поле **Font-size** позволяет определить размер шрифта. Он может быть задан числовой величиной, тогда в соседнем поле необходимо указать единицы измерения (наибольший интерес здесь представляют пиксель - **pixels** и традиционная типографская единица измерения пункт - **points**). Кроме того, размер может быть задан при помощи одной из predetermined относительных единиц измерения или в процентах от унаследованного размера.

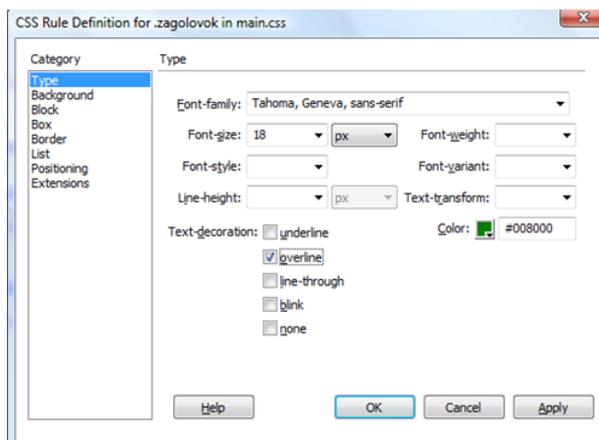


Рис. 6. Диалоговое окно определения параметров нового стиля

В поле **Font-style** можно выбрать один из вариантов начертания шрифта: **normal** - обычное (используется по умолчанию), **italic** - курсив, **oblique** - наклонное. Параметр **Line Height** позволяет задать расстояние между базовыми линиями смежных строк текста - интерлиньяж. Свойство **Text-decoration** применяет к тексту подчеркивание (**underline**), надчеркивание (линия над текстом, **overline**), зачеркивание (**line-through**), эффект мигания (**blink**), **none** используется по умолчанию для обычного текста.

Параметр **Font-weight** задает вес, или жирность, шрифта. Свойство **Font-variant** используется для отображения элемента маленькими прописными буквами (значение **small-caps**). Параметр **Text-transform** оказывает влияние на использование заглавных букв в элементе (**capitalize** - отображает первую букву каждого слова в верхнем регистре, **uppercase** - отображает все символы в верхнем регистре, **lowercase** - отображает все символы в нижнем регистре). Параметр **Color** используется для задания цвета текста.

Background - параметры свойства фона, применяемые к "холсту" позади элемента. **Background Color** и **Background Image**

определяют соответственно цвет фона и фоновое изображение элемента. Если задано фоновое изображение, то свойство **Repeat** определяет, как оно будет повторяться (**repeat** - в вертикальном и горизонтальном направлениях, **repeat-x** - только по горизонтали, **repeat-y** - только по вертикали, **no-repeat** - без повторов, один раз). Параметром **Attachment** определяется, будет ли прокручиваться фоновое изображение вместе с документом (по умолчанию, **scroll**) или нет (**fixed**). Свойства **Horizontal Position** и **Vertical Position** определяют начальное положение фонового изображения относительно прямоугольника, окружающего содержимое элемента.

Block - параметры выравнивания и взаимного расположения фрагментов текста в элементе. Свойство **Word Spacing** задает интервал между словами. **Letter Spacing** определяет размер интервала между символами. Параметр **Vertical-alignment** влияет на вертикальное выравнивание элемента. **Text-align** задает горизонтальное выравнивание текстовых элементов. Свойство **Text-indent** определяет размер отступа перед первой строкой текста элемента. Параметр **Whitespace** определяет, каким образом используются пробелы, заданные в исходном документе: **normal** - трактует текст обычным образом, заменяя множественные пробелы на один; **pre** -аналог тега <pre> в HTML (но при этом не используется шрифт с фиксированной шириной); **nowrap** - предотвращает перенос элементов текста на следующую строку, если не задан тег
.

Параметры из раздела **Border** позволяют управлять выводом и параметрами рамок вокруг элементов.

List - параметры форматирования списков.

Positioning - параметры позволяют преобразовать тег или выделенный блок текста в слой. Использование ограничено тем, что эти свойства поддерживаются только Internet Explorer.

Extensions - параметры, не поддерживаемые большинством

браузеров.

После того как все необходимые параметры стиля определены, он появится в списке в панели стилей. Выбранный в списке стиль применяется к текущему выделенному элементу документа.

Чтобы включить созданные стили в HTML-страницу, необходимо в разделе <HEAD> поместить строку со ссылкой на файл стилей и с указанием об использовании CSS:

```
<LINK          REL=STYLESHEET          TYPE="text/css"  
HREF="styles.css">
```

где "styles.css" - имя файла с созданными стилями CSS.

2.2.3. Создание гипертекстовых ссылок

Важнейшим элементом любого web-документа являются гипертекстовые ссылки. В языке HTML они реализуются с использованием тега <A>. Для создания ссылки в редакторе Dreamweaver необходимо выделить объект (блок текста, изображение или другой элемент страницы), а затем на панели свойств объекта (см. рис. 3) в поле **Link** указать URL файла, на который будет осуществляться гипертекстовый переход. Если это внутренняя ссылка (то есть файл входит в состав данного сайта), то адрес перехода проще всего указать с использованием инструмента **Browse for File**, который вызывается по кнопке, расположенной рядом с полем **Link**. В отрывшемся стандартном диалоговом окне необходимо выбрать файл, на который будет осуществляться переход. Если же это внешняя ссылка, то необходимо указать полный URL ресурса, начиная с имени протокола (например, http://spb.fio.ru/). Поле **Target** указывает, где должен быть открыт документ, на который указывает ссылка. Это может быть одно из четырех предустановленных значений:

- **_blank** - документ загружается в новом неименованном

окне;

- **_parent** - документ загружается в родительском окне;
- **_self** - документ загружается в то же окно, что и исходный;
- **_top** - документ открывается в том же окне браузера, но все существующие фреймы при этом удаляются.

По умолчанию все ссылки открываются в том же окне (то есть используется установка **_self**).

Кроме ссылок на файлы целиком, существует возможность создания ссылки на определенное место в документе - *якорь*. Для этого сначала необходимо создать в документе сами якоря. Это можно сделать, выполнив команду **Insert=>Named Anchor** или нажав кнопку **Named Anchor** на закладке **Common** панели объектов (предварительно поставьте курсор в то место, где надо сделать якорь).

В открывшемся диалоговом окне в поле **Anchor name** необходимо указать имя якоря, которое должно состоять из латинских символов и/или цифр и не может содержать пробелов. Для создания ссылки на якорь в поле **Link** после имени файла необходимо вписать символ **#** и указать после него имя якоря (например, filename.html#part1). Если якорь находится в этом же файле, то имя файла можно опустить (например, #part1).

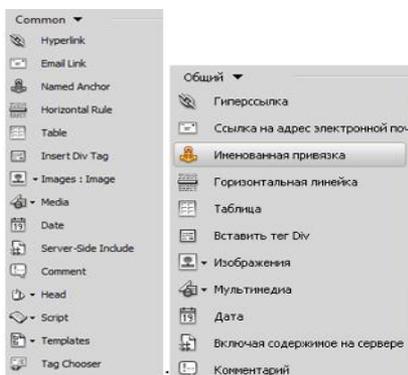


Рис. 7. Панель Common

Отдельный случай представляют собой ссылки на адреса электронной почты. Для создания таких ссылок можно воспользоваться кнопкой **Email Link** на закладке **Common** панели объектов (рис. 7) или выполнить команду **Insert=>Email Link**. В открывшемся диалоговом окне в поле **Text** необходимо указать тот текст, с которого будет идти ссылка, а в поле **E-Mail** - адрес электронной почты, на который эта ссылка будет указывать.

2.2.4. Создание и форматирование таблиц

Одним из наиболее широко используемых в HTML средств являются таблицы. Причем очень часто они используются не только традиционно - как метод представления табличных данных, но и как средство форматирования web-страниц.

Основное назначение HTML-таблиц - упорядочить данные (текст, форматированный текст, изображения, ссылки, формы, поля форм, другие таблицы) и разместить их в отдельных ячейках.

Одним из главных принципов модели таблиц HTML является то, что размеры ячеек браузер определяет автоматически в зависимости от содержимого. Размеры таблиц могут изменяться динамически в соответствии с текущими размерами окна. Тем не менее в языке HTML предусмотрена возможность явного задания размеров ячеек таблиц.

Добавить таблицу в редактируемый документ в Dreamweaver можно несколькими способами. Самый простой способ - это выбрать кнопку **Table** на панели объектов. Также можно воспользоваться командой **Insert=>Table**.

После этого на экране появляется диалоговое окно **Insert Table**, в котором предлагается указать параметры будущей таблицы (рис 8). В разделе **Table size**, в полях **Rows** и **Columns** задается количество строк и столбцов соответственно, в поле **Table width** - ширина таблицы (может быть относительной в процентах

или абсолютной в пикселях). Параметр **Border thickness** задает толщину рамки вокруг всей таблицы в целом и каждой из ее ячеек. Если указано нулевое значение, то рамка не будет отображаться.

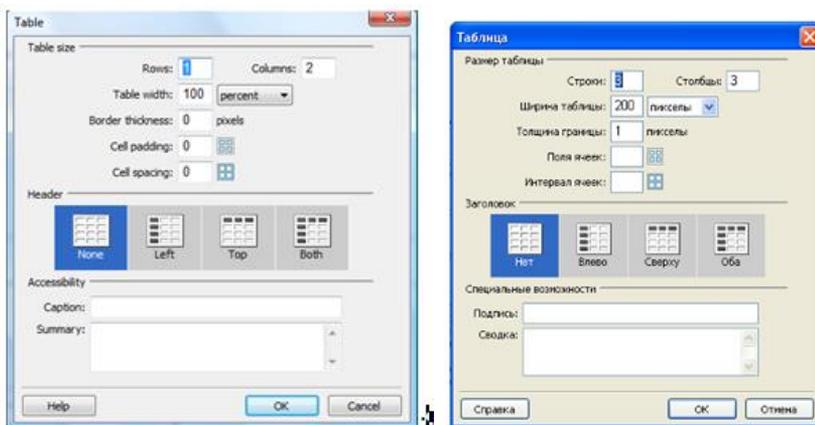


Рис. 8. Диалоговое окно параметров вставки таблицы

Параметр **Cell padding** задает расстояние между содержимым ячейки и ее границами (по умолчанию 1 пиксель), а **Cell spacing** - расстояние между соседними ячейками таблицы (по умолчанию 2 пикселя).

В разделе **Header** можно задать "шапку" и выделенный первый столбец таблицы. В этом наборе доступны варианты **None** (нет ни "шапки", ни выделенного первого столбца), **Left** (выделенный первый столбец), **Top** ("шапка") и **Both** (и "шапка", и выделенный первый столбец).

Притом ячейки, составляющие "шапку" и выделенный столбец, будут оформлены как ячейки заголовка, а текст, который мы введем в них, будет автоматически выровнен по центру и выделен полужирным шрифтом.

В разделе **Accessibility** в поле **Caption** задается название таблицы. Это название будет располагаться рядом с создаваемой таблицей. Раскрывающийся список **Align caption** позволяет задать местоположение и выравнивание названия.

После того как таблица создана, можно наполнять ее содержанием. Кроме того, можно изменять параметры таблицы: добавлять и удалять строки и столбцы, объединять и делить ячейки, менять размеры всей таблицы и отдельных строк и столбцов, цвет фона. Для того чтобы изменить свойства всей таблицы в целом, необходимо ее выделить, щелкнув левой клавишей мыши в левом верхнем углу или по нижней или правой границе таблицы. Также можно выполнить команду **Modify=>Table=>Select Table** (в этом случае курсор должен находиться в любой ячейке таблицы). Можно изменить количество строк и столбцов (**Rows** и **Cols**), ширину таблицы (**W**), параметры отступов (**CellPad** и **CellSpace**), толщину рамки (**Border**). Кроме того, можно указать выравнивание таблицы относительно остальных элементов абзаца (**Align**), цвета фона и рамки таблицы (**Bg color** и **Brdr color**), а также задать фоновое изображение (**Bg Image**). Эти же действия можно выполнить и через панель свойств при нахождении текстового курсора в таблице (рис. 3).

3. Вставка изображений и других элементов

3.1 Использование графических материалов

Язык HTML обладает возможностью вставки иллюстраций в web-документ. Для того чтобы вставить изображение в редактируемый документ, необходимо нажать кнопку **Images** на панели объектов и в открывшемся списке выбрать **Image** (рис. 9) или выполнить команду **Insert=>Image**. В открывшемся диалоговом окне необходимо выбрать графический файл (в формате GIF, JPEG или PNG). Следует обратить внимание на то, что документ перед вставкой графических и большинства других элементов необходимо сохранить, иначе путь к подключаемому файлу может быть абсолютным (начиная от корня диска).

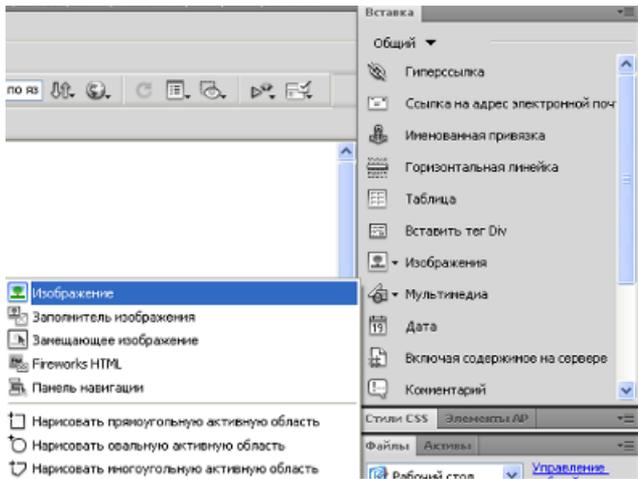


Рис. 9. Вставка изображения

Если выделить изображение (для этого достаточно щелкнуть по нему левой кнопкой мыши), то на панели свойств будут отображены его свойства (рис. 10).



Рис. 10. Свойства графического объекта

Параметры **W** и **H** определяют соответственно ширину и высоту изображения. По возможности следует изменять размеры самого графического файла (используя графические редакторы), оставляя параметры **W** и **H** равными исходным размерам. Если одна из этих величин не соответствует истинному значению размеров изображения, она будет выделена жирным шрифтом. Для того чтобы вернуть параметрам их действительные значения, достаточно щелкнуть левой кнопкой мыши по подписи к полю (буквам **W** или **H**) или воспользоваться кнопкой **Reset Size**.

В поле **Src** указан путь к графическому файлу: при

необходимости его можно отредактировать вручную или выбрать файл заново, нажав расположенную рядом кнопку **Browse for File** . Параметр **Align** задает выравнивание графического объекта относительно других элементов абзаца или строки.

3.2. Порядок выполнения работы

Работа выполняется студентом самостоятельно и состоит из следующих этапов:

- изучение методических указаний по выполнению лабораторной работы;
- создание сайта на выбранную тематику с использованием Adobe Dreamweaver;
- защита лабораторной работы.

Конечным результатом выполнения лабораторной работы является работоспособный сайт. Исходный код сайта представляется в электронном виде. После выполнения лабораторной работы преподаватель проверяет качество оформления сайта и правильность функционирования.

3.3. Основные требования к интернет-сайту

Тема сайта выбирается студентом из списка и утверждается преподавателем. Приступить к созданию сайта необходимо после утверждения темы и согласования с преподавателем структуры сайта. Данные процедуры проводятся в течение недели после выдачи задания на лабораторную работу.

Структура размещения информации на сайте должна исходить из удобства ее просмотра пользователями.

При создании сайта необходимо придерживаться трех основных принципов: удобство и быстрота нахождения требуемой информации, удобство восприятия информации сайта, скорость

загрузки страниц.

Объем сайта должен составлять не менее 10 страниц формата А4. Излагаемый материал должен быть логично структурирован, оформлен для удобства восприятия, включать рисунки и таблицы. Информация сайта, скопированная из других источников, не должна нарушать авторские и другие права пользователей на эту информацию.

Основой разработки страниц сайта является язык HTML. HTML-страницы должны обязательно включать конструкции языка HTML, относящиеся к каждой из групп:

- 1) тэги для управления шрифтом в документе;
- 2) тэги для управления форматом документа;
- 3) тэги для описания элементов перечислений;
- 4) тэги для построения таблиц;
- 5) тэги для организации гиперсвязей;
- 6) тэги для включения в документ изображений.

Одна или несколько страниц сайта должны быть выполнены с использованием языков XML и XSL.

Сайт должен иметь уникальный законченный дизайн, разработанный на основе творческого подхода (использовать стандартные шаблоны дизайна не допускается). Оформление дизайна сайта должно быть выполнено с использованием технологии CSS.

Повторяющиеся элементы на страницах сайта должны быть оформлены в соответствии с технологией SSI (т.е. должны быть вынесены в отдельные файлы и включаться в страницу только во время ее загрузки пользователем).

Сайт должен включать не менее двух сценариев на языке JavaScript, которые будут расширять функциональные возможности сайта для пользователя (упрощать доступ к информации (элементы поиска информации и навигации), информировать о данных текущего сеанса работы с сайтом

(время работы, идентификационная информация, просмотренные страницы), защищать конфиденциальные данные сайта, а также добавлять другие функции, не мешающие восприятию материала. Допустимо использовать стандартные скрипты, имеющиеся в Интернете.

Создание сайта может быть также выполнено с использованием технологий AJAX, jQuery, PHP, CGI, MySQL, Flash, Silver light и др.

На сайте могут применяться элементы связи с пользователями - форма для отправки электронных писем, тематический форум, блог, чат и др.

Все отступления от основных требований должны быть согласованы с преподавателем.

Сайт может быть разработан с использованием одного из стандартных средств: Adobe Dreamweaver, MS FrontPage, Amaya и др. Тестирование сайта может быть проведено, например, с использованием локального web-сервера denwer (denwer.ru) или IIS, входящего в Windows. Корректность работы сайта должна быть проверена в большинстве стандартных браузеров (в том числе и их версиях): IE, FireFox, Opera, Safari, Chrome.

После создания сайт необходимо разместить в Интернете на одном из сайтов, предоставляющих бесплатный хостинг (holm.ru, narod.ru и др.) и протестировать. Сайт обязательно должен быть зарегистрирован не менее чем в трех средствах информационного поиска и каталогах, а также хотя бы в одной системе учета статистики сайта (например, на hotlog.ru) (отличным результатом регистрации сайта в Интернете является выдача поисковой системой ссылки на созданный сайт в первой десятке адресов после проведения поиска по предложенным ключевым словам).

Процесс создания сайта контролируется преподавателем. Вначале студент должен согласовать структуру сайта (1-я неделя). Далее (второй этап) необходимо согласовать дизайн основных

страниц и их контент (информационное наполнение) (2-3 недели). На третьем этапе согласовывается полностью готовый сайт перед размещением его в Интернете (2-3 недели). После выполнения всех требований студент допускается к защите работы (8-я неделя).

При защите работы сдается электронная копия сайта.

4. Список тем

Темы сайтов

- . История создания Интернета.
- . Принципы работы Интернета.
- . Структура сети Интернет.
- . Ресурсы интернета.
- . Поисковые системы Интернета.
- . Электронная коммерция.
- . Электронные платежные системы.
- . Баннерные сети.
- . Счетчики и рейтинги
<<http://yaca.yandex.ru/yca/ungrp/cat/Computers/Internet/Ratings/>>.
- . Прикладные протоколы Интернета.
- . Транспортные протоколы Интернета.
- . Языки программирования для Интернета.
- . Язык JavaScript.
- . Язык PHP.
- . Язык Python.
- . Программные платформы для Интернета.
- . Электронная почта.
- . Защита от спама.
- . Технология World Wide Web.
- . Технология Semantic Web.
- . Технология Web 2.0.
- . Виртуальная реальность в Интернете.

- . Безопасность корпоративных систем Интернета и Интранета.
- . Системы обнаружения атак.
- . Аутентификация в Интернете.
- . Электронная цифровая подпись.
- . Несанкционированный доступ к информации на компьютерах в Интернете и Интранете.
- . Защита информации передаваемой по открытым каналам связи.
- . Язык HTML.
- . Каскадные таблицы стилей CSS.
- . Анимация в HTML-документе.
- . Виртуальные частные сети.
- . Направление развития Интернет-технологий.
- . Хостинг.
- . Доступ в Интернет: последняя миля.
- . Беспроводные сети.
- . Браузеры.
- . Источники, риски и формы атак на информацию.
- . Компьютерная преступность.
- . Базы данных в Интернете.
- . Аппаратная база доступа в Интернет.
- . Телеконференции в Интернете.
- . IP-телефония.
- . Сетевая этика.
- . Общение в Интернете.
- . Прокси-сервер.
- . DNS-сервер.
- . Тема, предложенная студентом, должна соответствовать изучаемому лекционному курсу, специальности, на которой ведется обучение, и иметь техническую направленность.

5. Вопросы для самопроверки

1. Что такое язык HTML и где и как он используется?
2. Какие элементы входят в рабочее пространство Adobe Dreamweaver?
3. Какие правила построения сайтов вы знаете?
4. Назовите уровни форматирования HTML-документа? Как они используются?
5. Что такое CSS-технология и где эта технология применима?
6. Назовите способы вставки и форматирования таблиц.
7. Перечислите этапы создания сайта.
8. Что входит в свойства графического объекта?

6. Список рекомендуемой литературы

1. Adobe Dreamweaver CS4: офиц. учебный курс. - М.: Эксмо, 2009. - 304 с.
2. Александров, А.В. Создание web-страниц и web-сайтов / А.В. Александров, Г.Г. Сергеев, С.П. Костин. - М.: Триумф, 2008. - 288 с.
3. Дронов, В.А. Adobe Dreamweaver CS4 (+ CD-ROM) / В.А. Дронов. - М.: BHV, 2009. - 832 с.
4. Кисленко, Н.П. HTML. Самое необходимое (+ CD-ROM) / Н.П. Кисленко. - СПб: БХВ-Петербург, 2008. - 352 с.
5. Коэн, И.Л., Коэн, И.Д. Полный справочник по HTML, CSS и JavaScript: [пер. с англ.] / И.Л. Коэн, И.Д. Коэн. - М.: ЭКОМ, 2007. - 1168 с.
6. Мейер, Э. CSS - каскадные таблицы стилей: подробное руководство / Э. Мейер. - М.: Символ-Плюс, 2008. - 576 с.
7. Хольцшлаг, М. Языки HTML и CSS для создания Web-

- сайтов: официальный учебный курс: [пер. с англ.] / М. Хольцшлаг. - М.: Триумф, 2007. - 304 с.
8. <http://pro.net.ru/> - Записки web-мастера.
9. Бурсов, М.В. Основы работы с HTML-редактором Dreamweaver: учеб.-метод. Пособие / М.В. Бурсов, В.М. Домненко, Д.А. Гаврилин, Д.Г. Николаев. - СПб: ГИТМО (ТУ), 2002. - 104 с.
10. Дригалкин, В.В. HTML в примерах. Как создать свой Web-сайт: самоучитель / В.В. Дригалкин. - М.: Диалектика, 2003. - 192 с.
11. Кузнецов, М.В. PHP 5. Практика разработки Web-сайтов / М.В. Кузнецов, И.В. Симдянов, С.В. Гольшев. - СПб.: БХВ-Петербург, 2005. - 960с. ил.
12. Пауэрс, Д. Adobe Dreamweaver, CSS, Ajax и PHP / Д. Пауэрс. - СПб: BHV-СПб, 2009. - 1056 с.
13. Ташков, П. Веб-мастеринг на 100%. HTML, CSS, JavaScript, PHP, CMS, графика, раскрутка / П. Ташков. - СПб.: Питер, 2009. - 512 с.
14. Экслер, А. Создание и раскрутка сайтов в Интернете / А. Экслер. - М.: Пресс, 2007. - 288 с.