

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ**
**Федеральное государственное бюджетное образовательное учреждение
высшего образования**
**«Пермский государственный национальный исследовательский
университет»**

Колледж профессионального образования

Web-дизайн

Методические рекомендации

для практических работ по изучению дисциплины
для студентов Колледжа профессионального образования
специальности

09.02.03 Программирование в компьютерных системах

Утверждено на заседании ПЦК

Информационных технологий

Протокол № 9 от 23.05.2018

председатель  Н.А. Серебрякова

Пермь 2018

Составители:

Серебрякова Наталья Александровна, преподаватель высшей квалификационной категории, преподаватель ПГНИУ

Web-дизайн: методические указания по выполнению практических заданий по дисциплине для студентов Колледжа профессионального образования специальности 09.02.03 Программирование в компьютерных системах, 2018.
– 33 с.

Предназначены для студентов Колледжа профессионального образования ПГНИУ специальности 09.02.03 Программирование в компьютерных системах

СОДЕРЖАНИЕ

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА	4
1. ТЕМАТИЧЕСКИЙ ПЛАН ДИСЦИПЛИНЫ	5
2. СОДЕРЖАНИЕ ЛАБОРАТОРНЫХ РАБОТ ЗАНЯТИЙ ПО ДИСЦИПЛИНЕ	8
3. ПЕРЕЧЕНЬ РЕКОМЕНДУЕМОЙ ЛИТЕРАТУРЫ И ИСТОЧНИ- КОВ	34

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

Цели и задачи преподавания дисциплины

Целью дисциплины является изучение методологических и концептуальных теоретических сведений о WEB-дизайне, формирование у студентов умения и навыков работы с WEB-страницами и эффективного комбинирования элементов мультимедиа, а также подготовка специалистов, умеющих применять современные методики разработки и сопровождения WEB-сайтов, используемых в дальнейшей профессиональной деятельности.

В задачи дисциплины входит изучение основ аппаратных средств WEB-дизайна, основных инструментальных средств, используемых для создания WEB-страниц, знакомство с возможностями создания базовых элементов WEB-страниц (текст, графические изображения, звук, анимация), с возможностями применения информационных технологий в сети Интернет.

Требования к уровню освоения содержания дисциплины

В результате изучения дисциплины «Web-дизайн» студент должен: знать:

- состояние развития современных web-технологий, их место и роль в работе компьютерных сетей Internet/Intranet;
- проблемы и направления развития web-технологий и программных средств, применяемых в web-технологиях;
- основные методы и средства автоматизации проектирования современных web- приложений;
- основы построения сложных web-узлов. уметь:
- применять различные инструментальные средства для разработки web-приложений;
- ориентироваться в современных информационных технологиях, их возможностях, перспективах развития;
- создавать различные элементы мультимедиа, используя при этом современные программно-аппаратные средства;

- осуществлять выбор средств и методов для решения поставленных профессиональных задач;
- проектировать логическую структуру веб-страниц, выбирать наиболее удобные решения подачи информации, а также реализовывать художественное оформление вебпроекта в соответствии с общепринятыми художественными и психологическими принципами.
- владеть:
 - инструментами создания web-страниц и перспективами развития современных мультимедийных инструментов, применяемых для создания таких web-страниц;
 - методикой формирования элементов мультимедиа с помощью современных программных средств;
 - навыками поиска сведений по программированию Интернет-приложений.

Место дисциплины и перечень дисциплин, знание которых необходимо для изучения курса.

Дисциплина «Web-дизайн» относится к дисциплинам по выбору цикла общепрофессиональных дисциплин.

1. ТЕМАТИЧЕСКИЙ ПЛАН ДИСЦИПЛИНЫ

№	Наименование разделов и тем	Максимальное количество часов	Аудиторных часов			Самостоятельная работа (час.)
			Лекции	Практические занятия	Лабораторные работы	
I.	<i>Раздел 1 Язык HTML</i>	94	14	0	46	34
	Тема 1.1. Основы HTML	12	2	0	6	4
	Тема 1.2. Графика на гипертекстовых страницах	16	2	0	6	8
	Тема 1.3. Работа с таблицами	14	2	0	6	6
	Тема 1.4. Объекты, формы и фреймы	18	2	0	12	4
	Тема 1.5. Стилизовое оформление HTML-документов	14	2	0	4	8
	Тема 1.6. Динамический HTML	10	2	0	6	2
	Тема 1.7. Основы верстки сайтов	10	2	0	6	2
II	Раздел 2. Инструментальные средства разработки Web- страниц.	12	4	0	6	4
	Тема 2.1. Работа в HTML – редакторе	12	4	0	6	4
III	Раздел 3. Системы управления контентом (CMS)	12	4	0	6	4
	Тема 3.1. Разработка корпоративного сайта	12	4	0	6	4
IV	Раздел 4. Публикация сайта в Интернет	16	6	0	6	4
	Тема 4.1 Продвижение сайта в сети Ин-	16	6	0	6	4

	тернет					
0		138	28		64	46

2. СОДЕРЖАНИЕ ЛАБОРАТОРНЫХ РАБОТ ПО ДИСЦИПЛИНЕ

Раздел 1 Язык HTML

Тема 1.1. Основы HTML

ЛАБОРАТОРНАЯ РАБОТА №1 (4 часа)

Цель работы: · Научиться создавать простейший гипертекстовый документ средствами текстового редактора Nod Pad

Научиться использовать теги форматирования шрифта и абзаца.

Задание на выполнение

1. Создать файл с гипертекстовым документом: · Запустить редактор, ввести текст: Приветствую Вас на моей первой web-страничке!
2. 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.
3. 2. Ввести теги, определяющие структуру html-документа: ·
4. 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
, 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.

5. Оформить фрагменты текста с помощью стилей Заголовков: ·

6. Первую строку документа оформить стилем Заголовок 1-го уровня с помощью парного тега <H1> ...</H1>.

7. 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.

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 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 (использо-

вать копирование текста из документа 1_name.htm): <HTML> <HEAD>
<TITLE> Фамилия </TITLE> </HEAD> <BODY> Приветствую Вас на моей
второй web-страничке!
 Монолог Гамлета </BODY>
</HTML> · Выровнять текст по центру. · Ввести текст: Быть иль не быть -
вот в чем вопрос. Что благороднее: сносить удары неистой судьбы - иль
против моря невзгод вооружиться, в бой вступить. И все покончить разом... ·
Оформить выравнивание абзаца по ширине. · Ограничить абзац горизонталь-
ными 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 Неистой
судьбы - иль против моря Невзгод вооружиться, в бой вступить И все покон-
чить разом... · Сохранить документ. · Просмотреть документ в окне браузера,
изменяя размер окна. 7. Выполнить оформление списков: · Создать новый
документ 3_name.htm, сохранить его в той же рабочей папке жесткого диска.
· Ввести текст: <HTML> <HEAD> <TITLE> Фамилия </TITLE> </HEAD>
<BODY> Приветствую Вас на моей третьей web-страничке! </BODY> </
HTML> · Дополнить текст документа (между тегами <BODY>...</BODY>)
следующим текстом: Я знаю как оформлять: Шрифты, Заголовки, Абзацы ·
Оформить три последние строки как список нумерованный. Для этого ис-
пользовать следующую конструкцию тегов: Шрифты,
Заголовки,
 Абзацы · Поменять оформление списка на список марки-
рованный. Использовать теги , · Создать «смешанный» список:
Я знаю как оформлять: 1. Шрифты · Размер · Цвет · Гарнитуру · Индексы 2.
Заголовки · От 1-го до 6-го уровня
3. Абзацы · Выравнивание · Разрыв строк внутри абзаца · С использованием
переформатирования.

8. Предъявить результаты работы преподавателю

Тема 1.2. Графика на гипертекстовых страницах

Лабораторная работа № 2 (6 часов)

Цель работы: ·

Научиться выполнять вставку рисунков в HTML-документ ·

Научиться создавать закладки и гиперссылки

1. Скопировать из Интернета или какой-либо папки в личную папку файлы три графических файла (например, Arrows1.wmf, Arrows2.wmf, Arrows3.wmf). Убедиться, что созданные ранее документы 1_name.htm, 2_name.htm и 3_name.htm также находятся в вашей папке на жёстком диске.
2. Вставка рисунков в документ. ·

2. Открыть в редакторе документ 2_name.htm. · Вставить рисунок Arrows1.wmf в начало документа 2_name.htm. Для вставки использовать тег IMG с параметрами WIDTH и HEIGHT для установки размеров рисунка 50 пикселей по горизонтали и по вертикали. · Сохранить документ под именем 4_name.htm. · Просмотреть в браузере полученный результат. · Ввести в тег рисунка параметр ALIGN для выравнивания рисунка по правому краю. Просмотреть результат в браузере. · Вставить рисунок Arrows2.wmf в конец документа 4_name.htm перед, подобрать тип выравнивания рисунка на свое усмотрение. Установить размер рисунка 100 пикселей по горизонтали и по вертикали. С помощью параметра ALT создать всплывающую подсказку «Рисунок 2», появляющуюся при наведении курсора мыши на рисунок. · Просмотреть в браузере полученный результат.
3. Создание гиперссылок и закладок. · В документе 3_name.htm закрепить гиперссылки за следующими элементами списка: За словом Шрифт – гиперссылка на документ 1_name.htm. За словом Заголовки – на документ 1_name.htm. За словом Абзацы - на документ 2_name.htm. · Создать закладку в документе 1_name.htm перед фразой «Нас утро встречает прохладой». Дать ей имя «Morning». · Изменить первую гиперссылку (слово Шрифт) так, чтобы она указывала на закладку «Morning» в документе 1_name.htm. · Создать закладку в начале текущего документа 3_name.htm. Присвоить ей имя «Hello». · Изменить вторую гиперссылку (на слове Заголовки), определив для неё переход в начало текущего документа на установленную закладку «Hello». · Создать закладку в документе 2_name.htm перед фрагментом монолога. Присвоить ей имя «Mono». · Установить на слово переформатирования гиперссылку на закладку «Mono». · Проверить правильность переходов по всем гиперссылкам.
4.

Закрепить гиперссылки за графическими файлами: · Отредактировать тег вставки рисунка Arrows1.wmf, ввести в тег атрибут ALT для отображения текста подсказки «Вернуться». Просмотреть в браузере как реагирует рисунок на наведение курсора мыши. · Закрепить за рисунком Arrows1.wmf в документе 4_name.htm гиперссылку на документ 3_name.htm. Выполнить переход между документами.

5. Предъявить результат преподавателю.

Лабораторная работа № 3 (6 часов)

Тема 1.3. Работа с таблицами

Цель работы: Научиться создавать таблицы при помощи тэга `<table></table>`.

Задачи работы:

1. Овладеть методикой работы по созданию таблиц .
2. Получить навыки работы с командой `<Table></Table>`, атрибутами `<tr></tr>`, `<td></td>`

Обеспечивающие средства: Сборник описаний практических работ; операционная система , программа Браузер; программа «Блокнот»; персональный компьютер.

Требования к отчету: Итоги практической работы представить в виде файла lab6.html .

Технология работы:

Теоретическая часть :

При создании сайтов таблицы используются очень часто. Таблица задается тэгом: `<table></table>`

Таблица состоит из строк и столбцов (ячеек), поэтому нам надо еще и указать их.

`<tr></tr>` - строчка таблицы

`<td></td>` - столбец (ячейка) таблицы

Зададим таблицу состоящую из двух строк и трех столбцов (ячеек). Для наглядности ячейки таблицы выделены разными цветами. Границы таблицы не заданы, поэтому вы их не видите

```
<table>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
```

```
<td></td>
</tr>
</table>
```

Заполните получившийся каркас цифрами:

```
<table>
<tr>
<td>1x1</td>
<td>1x2</td>
<td>1x3</td>
</tr>
<tr>
<td>2x1</td>
<td>2x2</td>
<td>2x3</td>
</tr>
</table>
```

Фон задается параметром bgcolor="цвет_фона". Фон можно задать для таблицы в целом, для ряда, для столбца (в пределах одного ряда). Задаем фон для каждого столбца. В параметрах height и width - вы можете их задать для всей таблицы, для одного ряда, для ячейки (столбца).

```
<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"> <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>
<td width="50" bgcolor="#336699"> <center> 2x3 </center> </td>
</tr>
</table>
```

Параметры colspan и rowspan. Colspan - определяет количество столбцов, на которые простирается данная ячейка, а rowspan - количество рядов (эти параметры могут принимать значение от 2 и больше, т.е. наша ячейка может растягиваться на два и более столбца (ряда)).

Используем параметр colspan=2, прописав его для ячейки 1x1. Код следующий:

```
<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>

```

Попробуйте сами написать код для такой таблицы (у вас должна исчезнуть ячейка 2x3). Для закрепления напишите код для таблицы:

Можно избавиться от пространства между ячейками таблицы.

Это достигается с помощью атрибута `cellspacing`, равного нулю:

```

<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 увеличить пространство между ячейками, допустим пусть `cellspacing=5`, тогда получим такое:

Обычно атрибут `cellspacing`, рассматривается в руководствах и учебниках в паре с атрибутом `cellpadding`, который добавляет свободное пространство между содержимым ячейки и ее границами. Чтобы было видно нагляднее я для начала прижму текст ячеек первого ряда к верху, в нижнего - к низу, используя атрибут `valign`

```

<table cellpadding=5>
<tr>
<td height="35" bgcolor="#FFCC33" colspan="2" valign="top">
<center>1x1</center> </td>
<td width="50" bgcolor="#336699" rowspan="2" valign="top">
<center>1x2</center> </td>
</tr>
<tr>
<td height="35" width="50" bgcolor="#336699" valign="bottom">
<center>2x1</center> </td>
<td width="50" bgcolor="#FFCC33" valign="bottom">
<center>2x2</center> </td>
</tr>
</table>

```

Выполните на компьютере:

Задание 1

1. Создайте файл lab6.html оформите как документ, в котором, в заголовке окна браузера должна быть надпись «Практическая б».

2. Практическая №6- заголовок (по центру и соответствующим шрифтом).

3. Создайте таблицы согласно заданию. Задать фон ячейкам желтый, зеленый, согласно рисунку:

1) 2)

1. Создать таблицу из трех строк и четырех столбцов, ширина таблицы составляет 60% от ширины экрана.

2. Ширина левого столбца составляет 30% от ширины таблицы.

3. Задать цвет фона для элементов таблицы, установить цвет рамки.

4. Создать общий заголовок: Работа с таблицами (по центру и соответствующим шрифтом);

9. Сохранить файл как lab6.txt в блокноте и как lab6.html для просмотра в браузере.

Лабораторная работа № 4 (12часов)

Тема 1.4. Объекты, формы и фреймы

Цель работы: Научиться создавать многооконные документы.

Задачи работы:

1. Овладеть методикой работы с тегом `<frameset></frameset>` и параметрами `rows`, `cols`.
2. Научиться присваивать имена окнам при помощи атрибута `name` команды `<frame>`.

Обеспечивающие средства: Сборник описаний практических работ; операционная система , программа Браузер; программа «Блокнот»; персональный компьютер.

Требования к отчету: Итоги практической работы представить в виде файла lab7.html .

Технология работы:

1. Создать документ, в котором в заголовке окна браузера должна быть надпись "Практическая 7".
2. С использованием команд создания многооконных документов сформировать 4 файла по указанному заданию. Дать им имя:

фрейм 1.html, фрейм 2.html и т.д.



3. Пример для 1-го фрейма:

```
<html>
<head>
<title>Л</title>
<frameset rows="100,*,150">
<frame src="logo.html">
<frame src="content.html">
<frame src="menu.html">
</frameset>
</head>
</html>
```

1. Пример для 2-го фрейма:

```
<html>
<head>
<title>Хождение по фреймам</title>
<frameset rows="100,*">
<frame src="logo.html">
<frameset cols="150,*">
<frame src="menu.html">
<frame src="content.html">
</frameset>
</frameset>
</head>
</html>
```

1. Пример для 3-го фрейма: параметр cols делит окно браузера на колонки<frameset cols="100,150,*">
2. Остальные фреймы создайте согласно рисунку, самостоятельно.
8. Сохранить файл как lab4.txt в блокноте и как lab4.html для просмотра в браузере.

Лабораторная работа № 5 (4 часов)

Тема 1.5.Стилевое оформление HTML-документов

Цель работы: Научиться оформлять веб - страницы с использованием каскадных таблиц стилей CSS.

Задачи работы:

- Овладеть методикой работы с атрибутом STYLE.
- 2. Получить навыки работы замены тегов на стилевое описание.

Обеспечивающие средства: Сборник описаний практических работ; операционная система , программа Браузер; программа «Блокнот»; персональный компьютер.

Требования к отчету: Итоги практической работы представить в виде файла lab9.html .

Технология работы:

- Создайте стиль следующего образца:

Для команды скопируйте текст из практической работы №4 (файл lab4.html)

- Пустое поле сверху и снизу во всем документе
- Пустое поле слева - 5; пустое поле справа-2
- Цвет фона - светло-серый, цвет текста - черный.
- Установите в данном тексте цвет непосещаемых гиперссылок – коричневый, посещаемых –зеленый, стиль шрифта – черный.
- Абзацы: на белом фоне. У первого абзаца – вертикальная одинарная красная черта слева; у второго абзаца – вертикальная двойная синяя черта справа; у третьего абзаца никаких линий нет
- Текст абзацев идет с отступом от края белого фона, с увеличенным интервалом между строками и с заданием «красной строки»
- В первом заголовке получите объемный эффект. Для этого:
 - с помощью свойств margin-top и margin-left сделайте наложение двух строк текста
 - с помощью свойства font-size установите большой размер строк
 - с помощью свойства color - сделайте, чтобы вторая строка выглядела тенью
 - шрифт заголовка - Arial.

В первом абзаце слова "информационной Вселенной" идут жирным курсивом и с разрядкой.

Второй заголовок: желтого цвета на зеленом фоне, причем фон лишь немного шире заголовка, но уже основного текста. Буквы заголовка подчеркнуты

ты и надчеркнуты.

Во втором абзаце, в первом предложении часть слов расположена выше основного текста, а часть слов - ниже. Это должно быть сделано средствами CSS.

Лабораторная работа №6 (6 часов)

Тема 1.6. Динамический HTML

Цель работы: Научиться. создавать формы.

Задачи работы:

1. Овладеть методикой работы по созданию форм с помощью команды FORM.
2. Получить навыки по созданию форм, кнопок командами: INPUT, SELECT, TYPE.

Обеспечивающие средства: Сборник описаний практических работ; операционная система, программа Браузер; программа «Блокнот»; персональный компьютер.

Требования к отчету: Итоги практической работы представить в виде файла lab10.html .
персональный компьютер.

Технология работы:

1. Форма — это инструмент, с помощью которого HTML-документ может послать некоторую информацию в некоторую заранее определенную точку внешнего мира, где информация будет некоторым образом обработана. Форма открывается меткой <FORM> и заканчивается меткой </FORM>. **Создайте в окне «содержание» практической работы №8 следующие формы:**

```
<HTML>
<HEAD>
<TITLE>Пример создания форм</TITLE>
</HEAD>
<H1>Несколько более сложная форма </H1>
<FORM ACTION="http://206.31.82.215/hp/nc/fd-win.pht" METHOD=post>
<H2>Расскажите немного о себе...</H2>
<P>Указывать подлинные данные совсем не обязательно.
Для целей демонстрации вполне подойдут и вымышленные. </P>
<P>Имя: <INPUT TYPE=text SIZE=40 NAME=fn><BR>
Фамилия: <INPUT TYPE=text SIZE=40 NAME=ln><BR>
Пол: <INPUT TYPE=radio NAME=gender VALUE="male" checked>мужской
<INPUT TYPE=radio NAME=gender VALUE="female">женский<BR>
Возраст: <INPUT TYPE=text SIZE=5 NAME=age> лет<BR>
```

```
<INPUT TYPE=submit VALUE="Запустить обработчик"></P>
```

```
<SELECT NAME="selection">
```

```
<OPTION VALUE="option1" checked>Вариант 1
```

```
<OPTION VALUE="option2">Вариант 2
```

```
<OPTION VALUE="option3">Вариант 3
```

```
</SELECT>
```

```
</FORM>
```

```
</BODY>
```

```
</HTML>
```

Метка `<OPTION>` определяет элемент меню. Обязательный атрибут `VALUE` устанавливает значение, которое будет передано обработчику, если выбран этот элемент меню. Метка `<OPTION>` может включать атрибут `checked`, показывающий, что данный элемент отмечен по умолчанию.

Оформите: окна практической работы №8 в единой цветовой гамме `BGCOLOR="FFF0FF"`, окно "main" на тон темнее; правой верхней части окна «head» вставьте «логотип»-HTML-прямоугольник 10x100, другие оформите по своему усмотрению.

2. Создайте все формы (Пример использования), приведенные в теоретической части. Покажите выполненное преподавателю.

Теоретическая часть:

Типы элементов формы

TYPE="TEXT"

Представляет собой однострочное текстовое поле (рис. 10.1), физический размер которого можно установить с использованием атрибута `SIZE`, а первоначально введенную в него символьную последовательность — с помощью атрибута `VALUE`.



Рис. 10.1. Элемент формы TEXT

Примером использования данного элемента в коде HTML может служить следующая строка:

```
<INPUT TYPE="TEXT" SIZE="40" NAME="user_name" VALUE="Введите  
ваше имя">TYPE="PASSWORD"
```

Тип, полностью аналогичный предыдущему, за исключением того, что вводимая пользователем в текстовое поле информация будет отображаться на экране символами «*», чтобы скрыть текст от постороннего зрителя.

TYPE="CHECKBOX"

Элемент, представляющий собой простую форму выбора, принимающую одно из двух устойчивых состояний: «отмечено» — «не отмечено» (рис. 10.2).

□ Описание

Рис. 10.2. Элемент формы CHECKBOX

Данный элемент оперирует так называемыми булевыми переменными, то

есть переменными, каждая из которых может принимать значение «ИСТИНА» или «ЛОЖЬ». Каждый элемент формы CHECKBOX создает логическую пару значений вида «имя_элемента—состояние» и передает их на сервер. Для установки первоначального состояния элемента (отмечен, либо нет) используют атрибут CHECKED.

Пример использования:

```
<INPUT TYPE="CHECKBOX" CHECKED NAME="C01" VALUE="yes">
```

TYPE="RADIO"

Так называемая радиокнопка (рис. 10.3) применяется в случае, когда какая-либо логическая переменная может принимать только одно значение из множества возможных.

• Описание

Рис. 10.3. Элемент формы RADIO

Все элементы RADIO одной формы обозначаются одним и тем же значением атрибута NAME. Использование радиокнопок требует явного указания значений атрибута VALUE, одна из кнопок должна быть обязательно выделена атрибутом CHECKED.

Пример использования:

```
<P ALIGN="CENTER">Пожалуйста, укажите ваш возраст</P>
```

```
<CENTER>
```

```
<INPUT TYPE="RADIO" NAME="user-age" VALUE="0-12">
```

```
<INPUT TYPE="RADIO" NAME="user-age" VALUE="13-17">
```

```
<INPUT TYPE="RADIO" NAME="user-age" VALUE="18-25">
```

```
<INPUT TYPE="RADIO" NAME="user-age" VALUE="26-35" CHECKED>
```

```
<INPUT TYPE="RADIO" NAME="user-age" VALUE="36-">
```

```
</CENTER>
```

TYPE="BUTTON"

Отображает обыкновенную кнопку (рис. 10.4), нажатие на которую приводит к какому-либо действию сервера.



Рис. 10.4. Элемент формы BUTTON

С помощью атрибута NAME данному элементу присваивается уникальное имя, атрибут VALUE позволяет изменять надпись, отображаемую на кнопке.

Пример использования:

```
<INPUT TYPE="BUTTON" VALUE="Button" NAME="B1">TYPE="SUBMIT"
```

Определяет кнопку, по нажатию которой данные из формы передаются серверу. Так же как и в предыдущем случае, надпись, отображаемая на кнопке, задается атрибутом VALUE.

Пример использования:

```
<INPUT TYPE="SUBMIT" VALUE="Отправить!">
```

TYPE="RESET"

Создает кнопку, которая очищает неправильно заполненную текстовую

форму. Параметры и значения этой кнопки не передаются на сервер вместе с другими данными формы. Пример использования:

```
<INPUT TYPE="RESET" VALUE="Очистить форму">TYPE="FILE"
```

Генерирует на экране кнопку, по нажатию на которую на экране появляется Проводник Windows, позволяющий присоединить к отсылаемым на сервер данным любой файл с локального компьютера пользователя. Данный элемент применяется в основном в формах отправки с сервера сообщений электронной почты для организации вложений, а также для загрузки изображений на сервер. Обычно рядом с кнопкой отображается небольшое текстовое поле, куда автоматически заносится имя отсылаемого файла и путь к нему на локальном диске.

Пример использования:

```
<INPUT TYPE="FILE" NAME="picture">TYPE="IMAGE"
```

Создает кнопку отсылки, аналогичную элементу SUBMIT, но с использованием графического изображения, созданного пользователем. Обычно применяется в случаях, когда стандартная серая прямоугольная кнопка «не вписывается» в дизайн сайта. URL к изображению, играющему роль кнопки, указывается атрибутом SRC, причем сам элемент может содержать собственные атрибуты, аналогичные применяемым в теге (см. урок 5), в том числе ALIGN, ALT и др. Атрибуты NAME и VALUE трактуются так же, как аналогичные атрибуты элемента SUBMIT.

Пример использования:

```
<INPUT TYPE="IMAGE" SRC="/images/button.gif ALIGN="BOTTOM"  
NAME="submit" VALUE="Отправить!">  
TYPE="HIDDEN"
```

Данный элемент является скрытым и не отображается на экране монитора пользователя. Применяется он для хранения и передачи на сервер информации о текущем состоянии формы: при нажатии кнопки SUBMIT элемент HIDDEN формирует логическую пару переменных типа «имя—значение», которые будут отосланы серверу с помощью протокола HTTP. Элементы HIDDEN служат доступной альтернативой файлам cookies — специальным файлам, в которых сохраняются индивидуальные настройки пользователя и позволяющим, например, восстановить последнее состояние формы при повторном посещении пользователем содержащей эту форму страницы.

Пример использования:

```
<INPUT TYPE="HIDDEN" NAME="form1" VALUE="c3576-236-2113">
```

Однако с помощью тега <INPUT> можно передать далеко не все доступные web-дизайнеру элементы форм. Например, текстовые поля, задаваемые атрибутами TEXT и PASSWORD, весьма ограничены по количеству символов, которые можно ввести в эти поля. Если перед web-мастером встала задача создать такие текстовые поля, чтобы пользователь мог вводить большое количество символов, используется самостоятельный тег <TEXTAREA>, формирующий на экране одноименный элемент формы (рис. 10.5).

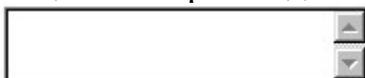


Рис. 10.5. Элемент формы TEXTAREA

Синтаксис записи данного тега выглядит следующим образом:

```
<TEXTAREA NAME="имя элемента" ROWS="целое число" COLS="целое число">
```

Текст, выводимый в текстовом поле по умолчанию</TEXTAREA>

С атрибутом NAME, определяющим индивидуальное имя каждого элемента формы, вы уже знакомы. Атрибуты ROWS и COLS указывают соответственно максимально допустимое количество строк вводимого текста и символов в строке. В случае если набираемый пользователем текст не помещается в видимую часть текстового контейнера, по краям поля появляются вертикальные и горизонтальные полосы прокрутки для скроллинга содержимого элемента TEXTAREA.

Пример использования:

```
<TEXTAREA NAME="message" ROWS="25" COLS="40">
```

Введите сюда текст сообщения

```
</TEXTAREA>
```

Для создания меню выбора в форме (рис. 10.6) web-мастер может воспользоваться тегом <SELECT>.

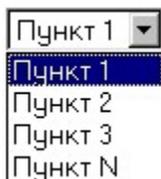


Рис. 10.6. Элемент формы SELECT

Синтаксис записи данного тега в общем виде выглядит так:

```
<SELECT NAME="имя" MULTIPLE SIZE="целое число">
```

```
<OPTION VALUE="Пункт 1">Пункт 1</OPTION>
```

```
<OPTION VALUE="Пункт 2">Пункт 2</OPTION>
```

```
<OPTION VALUE="Пункт 3">Пункт 3</OPTION>
```

```
OPTION VALUE="Пункт N">Пункт N</OPTION>
```

```
</SELECT>
```

Атрибут MULTIPLE определяет для пользователя возможность отметить не одну, а сразу несколько позиций из предложенного списка. Если данный атрибут задан, можно воспользоваться атрибутом SIZE, определяющим количество позиций, видимых на экране одновременно.

Пример использования:

```
<P ALIGN="CENTER">Пожалуйста, укажите ваш город</P>
```

```
<CENTER>
```

```
<SELECT NAME="City" MULTIPLE SIZE="3">
```

```
OPTION VALUE="a">Санкт-Петербург</OPTION>
```

```
OPTION VALUE="b">Москва</OPTION>
```

```
<OPTION VALUE="c">Екатеринбург</OPTION>
```

```
</SELECT>
```

```
</CENTER>
```

3. Создайте все формы из теоретической части и сохраните их в файле form_10.txt, form_10.html

4. Сохранить файл как lab10.txt в блокноте и как lab10.html для просмотра в браузере.

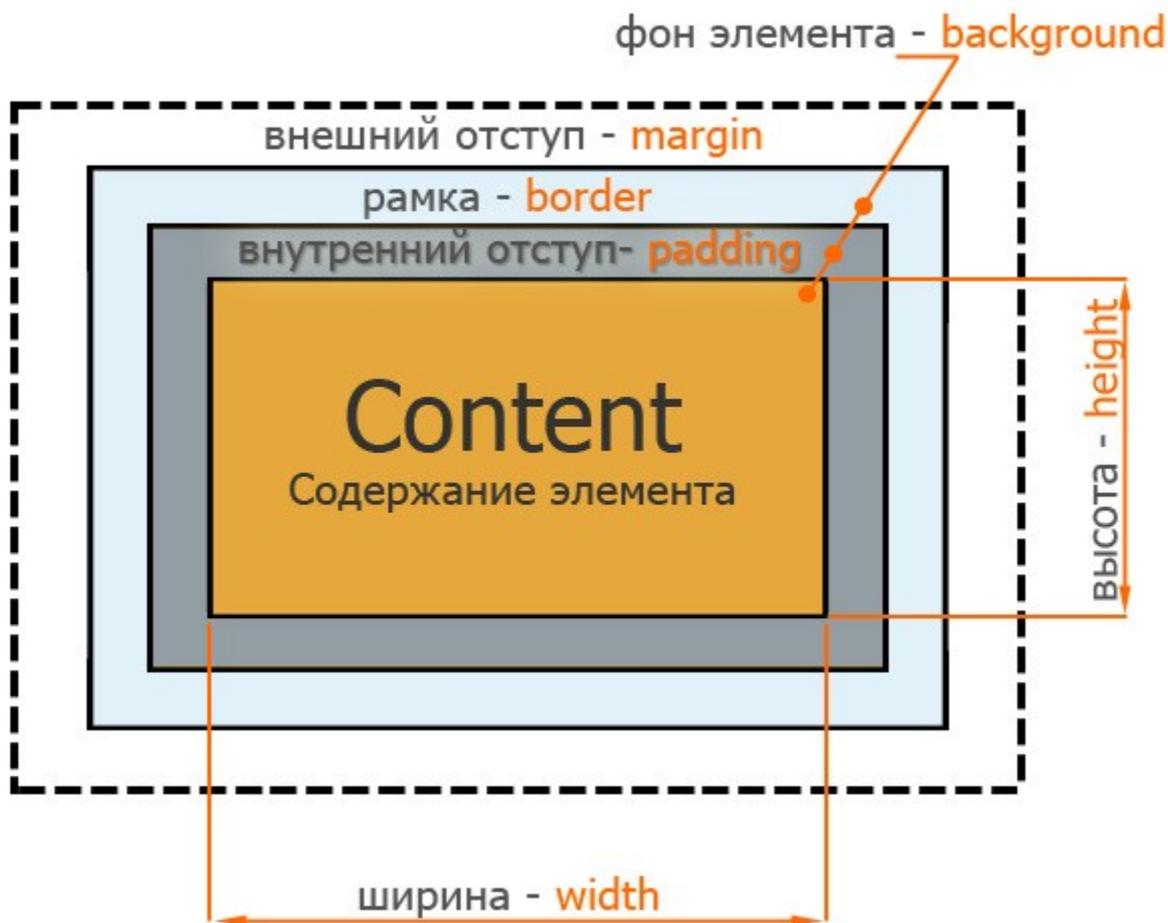
Лабораторная работа № 7 (6 часов)

Тема 1.7. Основы верстки сайтов

Задание сверстать макет сайта



Селектор	width	height
#header	960	100
#nav	960	100
#section	?	700
.article		100
#aside	300	700
#footer	960	100



Бордюры

Описание

Универсальное свойство `border` позволяет одновременно установить толщину, стиль и цвет границы вокруг элемента. Значения могут идти в любом порядке, разделяясь пробелом, браузер сам определит, какое из них соответствует нужному свойству. Для установки границы только на определенных сторонах элемента, воспользуйтесь свойствами `border-top`, `border-bottom`, `border-left`, `border-right`.

Синтаксис

```
border: [border-width || border-style || border-color] | inherit
```

Значения

Значение `border-width` определяет толщину границы. Для управления ее видом предоставляется несколько значений `border-style`. Их названия и результат действия представлен на рис. 1.



Рис. 1. Стили рамок

`border-color` устанавливает цвет границы, значение может быть в любом допустимом для CSS формате.

Пример: `border: 4px double black`; Можно задавать стиль одной строкой как показано в примере.

`float:left`

В случае, если необходимо разместить два блока *div* в одну линию друг за другом, у первого блока указывается стиль *float:left* (это означает что своей левой границей данный блок должен прилипнуть к предыдущему блоку в потоке). Первый блок оказывается прижатым, например к левой границе окна браузера. Если следующему блоку в потоке указать тоже самое значение в стилях, то два блока будут выводиться на одной линии. Первый блок будет прилипать к левой границе экрана, а второй, своей левой границей, к правой границе предыдущего блока.

Если у второго блока указать значения стиля *float:right*, то оба блока все так же окажутся расположенными на одной линии, но теперь первый блок будет прилипать к левой границе окна браузера, а правый — своей правой стороной к правой границе окна браузера.

Clear- Запрет на обтекание.

Устанавливает, с какой стороны элемента запрещено его обтекание другими элементами. Если задано обтекание элемента с помощью свойства *float*, то *clear* отменяет его действие для указанных сторон.

Синтаксис

clear: none | left | right | both | inherit

Значения

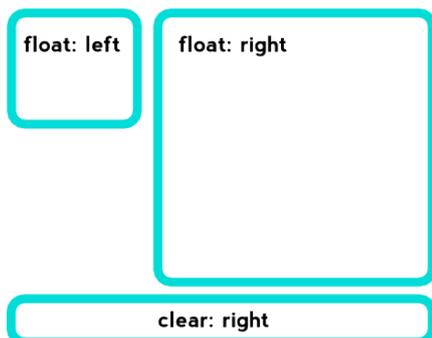
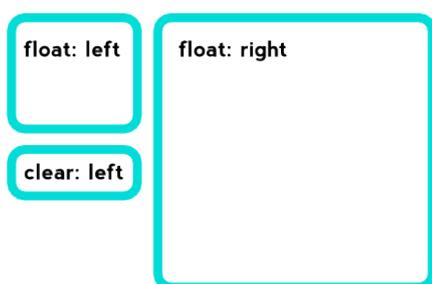
None — Отменяет действие свойства *clear*, при этом обтекание элемента происходит, как задано с помощью свойства *float* или других настроек.

Both — Отменяет обтекание элемента одновременно с правого и левого края. Это значение рекомендуется устанавливать, когда требуется снять обтекание элемента, но неизвестно точно с какой стороны.

Left — Отменяет обтекание с левого края элемента. При этом все другие элементы на этой стороне будут опущены вниз, и располагаться под текущим элементом.

Right — Отменяет обтекание с правой стороны элемента.

Inherit — Устанавливает значение родителя.



Блочная верстка или основы анатомии скелета сайтов

Сайты тоже имеют свой скелет. Но о его особенностях спрашивать врачей бесполезно. Да и ветеринары тоже не в курсе строения сайта. Об этом ведомо лишь верстальщикам. Именно от них зависит строение скелета будущего ресурса. А главным способом создания костей его скелета является блочная верстка.

Верстка сайта – ремесло для посвященных

Есть в верстке сайта что-то таинственное. Но это до тех пор, пока не познакомишься с этим ремеслом поближе. Начинаем наше посвящение:

Следующим этапом разработки сайта после создания его макета является верстка. Задача верстальщика перенести с помощью html кода и таблиц css скелет будущего сайта в виртуальный мир. Проще говоря, перенести размеры и пропорции ресурса в форму, понятную для браузера.

В процессе верстки кодом html происходит разбивка «скелета» сайта на части. А с помощью css(*каскадных таблиц стилей*) задаются размеры его «костей», цвет и расположение.

Различают несколько видов верстки:

I. Табличная – ранее была основным способом верстки. В табличной верстке для задания структуры сайта используется тег <table> и его дочерние теги. Верстка с помощью таблиц позволяет наиболее пропорционально расположить все элементы дизайна относительно друг друга. Но в тоже время такой код получается слишком объемным:

Также к основным недостаткам табличного кода относится его долгая загрузка и плохая индексация содержимого поисковыми системами.

Содержимое страницы, сверстанной на основе таблиц, не будет отображено до тех пор, пока не загрузятся все данные. Блочная верстка позволяет отображать каждый загруженный элемент отдельно.

Плохая индексация табличных страниц объясняется большими промежутками между блоками текста, расположенного в разных ячейках таблицы.

Теперь табличная верста редко используется в качестве основного метода создания сайтов. Сейчас ее применяют лишь для структурирования табличных данных и расположения графических изображений.

II. Блочная – в данный момент является основным способом верстки. В отличие от табличной блочная верстка обладает рядом преимуществ:

- Отделение стиля элементов от кода html;
- Возможность наложения одного слоя на другой – такая возможность во многом облегчает позиционирование элементов.
- Лучшая индексация поисковиками;
- Высокая скорость загрузки страницы, состоящей от взаимно независимых элементов;
- Легкость создания визуальных эффектов (*выпадающих меню, списков, всплывающих подсказок*).

Основным недостатком блочной верстки является некая «двусмысленность» понимания ее кода различными браузерами. Поэтому часто html страницы приходится «доводить» путем использования специальных хаков.

С появлением блочной верстки родилось такое понятие, как «кроссбраузерность». Из-за различия отображения одного и того же элемента в разных

браузерах верстальщикам приходится вставлять в основной html целые куски кода (хаки).

Действие хака является узкоспециализированным и решает проблему некорректного отображения лишь в одном браузере.

Основным элементом, применяемым в блочной верстке, является тег <div>. Участок кода, отделенный этим тегом, называется слоем. Все стилевые решения вынесены за границы кода html в каскадные таблицы стилей. Доступ к ним осуществляется через идентификаторы или классы css:

Как происходит блочная верстка?

Перед началом верстки готовый psd макет сайта в графическом редакторе разрезают на блоки (*слои*). В отдельную папку помещают вырезанные фоновые картинки, которые будут прикрепляться отдельно к каждому слою:

Для примера возьмем вот такой макет сайта, созданный в Photoshop. Сначала в текстовом редакторе с помощью div задаем структуру будущего ресурса и присваиваем каждому слою свой селектор id. Получается такая структура:

Затем к готовой структуре сайта на html строкой <link rel=»stylesheet» type=»text/css» href=»style.css» /> прикрепляем файл css. После чего добавляем в него стилевое описание каждого слоя, позиционирование относительно других элементов и его размеры.

Более подробно ознакомиться со всеми свойствами css можно из технической документации к языку.

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
  <title>Пример блочной верстки</title>
<link rel="stylesheet" type="text/css" href="style.css" />
</head>

<body>

<div id="container">
  <div id="header">
    <h2>Шапка</h2>
  </div>

  <div id="navigation">
    <h2>Блок навигации</h2>
  </div>

  <div id="menu">
    <h2>Меню</h2>
  </div>

  <div id="content">
    <h2>Контент</h2>
  </div>

  <div id="clear">

</div>

<div id="footer">
  <h2>Подвал сайта</h2>
```

```
</div>
</div>

</body>
</html>
Содержимое файла style.css:
body {
    background: #f3f2f3;
    color: #000000;
    font-family: Trebuchet MS, Arial, Times New Roman;
    font-size: 12px;
}

#container {
    background:#99CC99;
    margin: 30px auto;
    width: 900px;
    height: 600px;
}

#header {
    background: #66CCCC;
    height: 100px;
    width: 900px;
}

#navigation {
    background: #FF9999;
    width: 900px;
    height: 20px;
}

#menu {
    background: #99CC99;
    float: left;
    width: 200px;
    height: 400px;
}

#content {
    background: #d2d0d2;
    float: right;

    width: 700px;
    height: 400px;
}

#clear {
    clear:both;
}

#footer {
    background: #0066FF;
    height: 80px;
    width: 900px;
}
```

Лабораторная работа № 8 (6 часов)

Тема 2.1. Работа в HTML – редакторе

Создание Web-страниц средствами различных программных средств разработки (Microsoft FrontPage, CoffeCup, Macromedia Dreamweaver, TopStyle, HomeSite, Adobe Golive).

Создать Web-сайт “Моя домашняя страница”. На сайте представить информацию по следующим разделам: “Моя семья”, “Друзья”, “Учеба”, “Хобби”, “Любимые животные”.

На сайте необходимо реализовать форматирование текста, абзацев, списков, заголовков, также работу с цветом, мультимедиа, видео. Осуществить добавление элементов дизайна рисунки, таблицы. Реализовать работу с картами-изображений. Реализовать все типы гиперссылок (относительные, абсолютные, закладки). При разработке Web-сайта связать между собой 10-15 Web-страниц. В исходном html-коде использовать комментарии каждого тега.

Лабораторная работа № 9 (6 часов)

Тема 3.1. Разработка корпоративного сайта

Создать Web-страницу в HTML-редакторе на тему “Рекламный сайт фирмы”.

Необходимо реализовать средствами редактора форматирование текста, списков, заголовков, абзацев.

Помимо этого Web-сайт должен содержать такие элементы дизайна как рисунки, таблицы, фреймы, карты изображений, Web-компоненты.

С помощью элемента дизайна форма реализовать гостевую страницу пользователей. Использовать элементы динамического HTML.

Реализовать подключение внешних таблиц стилей - CSS.

Реализовать все типы гиперссылок.

Основной файл index.html должен быть связан с 10-15 страницами.

Главная страница должна содержать следующие пункты: “О нас”, “История фирмы”, “Контакты”, “Прайс”, “Сотрудники”.

Тематика:

- 1) Книгомир,
- 2) Фаренгейт,
- 3) Галактика,
- 4) Интерьер,
- 5) Золотая середина,
- 6) Пирамида,
- 7) Обувь-Класс,
- 8) Beefree,
- 9) Мир кожи и меха Cressida,
- 10) Эльдorado, 11) М-видео,
- 12) Рив Гош,
- 13) Лэтуаль,
- 14) Оранжевич,
- 15) Компьютерная база,
- 16) Столплит,
- 17) Никс,
- 18) Гелиос,
- 19) Автосалон RENAULT,
- 20) Консул,
- 21) Диаманд,
- 22) Жемчуг,
- 23) Оренбургские авиалинии,
- 24) Галерея обуви,
- 25) Турагентство Regas.

При разработке Web-сайта связать между собой 10-15 Web-страниц. В исходном html-коде использовать комментарии каждого тега

Лабораторная работа № 10 (6 часов)

Тема 4.1 Продвижение сайта в сети Интернет (6 часов)

Размещение сайта на хостинге.

Проанализировать бесплатные хостинги для размещения сайтов. Выбрать хостинг для своего сайта – обосновать выбор. Разместить сайт на хостинге

Краткая теория:

Хостинг – это место, где будет размещён сайт. Хостинг бывает двух типов: бесплатный и платный.

При размещении на бесплатном хостинге ваш сайт будет автоматически использован как рекламная площадка для этого хостинга. Выразаться эта реклама может разными способами. Например, при заходе на сайт появится всплывающее окно с рекламой, либо на каждой странице вашего сайта будет добавлен сверху большой баннер¹. Довольно часто на бесплатных хостингах бывают разные "неполадки с сервером". Ваш сайт могут спокойно удалить из базы данных, ничего при этом не объясняя. На бесплатном хостинге ваш сайт будет называться примерно так: `your_name.имя_хостинга.ru` (`your_name` – имя вашего сайта). На платном хостинге вам дадут своё доменное имя, например, `your_name.ru` (`com`, `net` и т.д.).

Самый популярный бесплатный хостинг – `www.narod.ru`. Этот хостинг предоставляет 100 Мбайт под сайт. Из рекламы только небольшое окошко в правом верхнем углу, да и то с крестиком.

Следующий хостинг – `www.boom.ru`. Место под сайт – 50 Мбайт. Главный недостаток – навязчивая реклама. При заходе на сайт в нижней части окна появляется большой баннер, который при прокрутке страницы остаётся неподвижным. Баннер с крестиком, но всё равно мешает.

Хостинг `www.by.ru`. Под сайт даётся неограниченное место. При заходе на сайт выскакивает отдельное небольшое окошко `pop-up` и вверху страницы добавляется надпись, что сайт размещён именно на этом хостинге. Возникают довольно частые сбои на сервере, из-за чего сайт некоторое время не представляется возможным обновить.

Хостинг `www.newmail.ru`. Под сайт дают 16 Мбайт. Реклама необязательна. Предоставляет несколько доменных имён `your_name.newmail.ru`, `your_name.orc.ru`, `your_name.hotmail.ru`, `your_name.nm.ru`, `your_name.nightmail.ru`.

Существуют и другие бесплатные хостинги. Здесь приведены самые популярные и надёжные.

Рассмотрим пример регистрации на бесплатном хостинге www.narod.ru:

Выбрать бесплатный хостинг www.narod.ru.

Щёлкнуть ЛКМ на ссылке Регистрация, которая размещена на главной странице в левом верхнем углу.

В предлагаемых формах нужно указать ваш логин, пароль для доступа, фамилию, имя, e-mail, а также имя сайта. Его нужно напечатать латинскими буквами (это то самое `your_name`, после которого идёт точка и название хостинга narod.ru). Имя сайта должно легко запоминаться и иметь непосредственное отношение к теме вашего сайта. В процессе регистрации необходимо следовать предлагаемым инструкциям.

После регистрации обычно высылается письмо на ваш электронный почтовый ящик, который вы указали. В нём обычно повторно указывается ваш логин и пароль (чтобы вы не забыли) и даётся ссылка на панель управления сайтом. Щёлкните по этой ссылке.

На открывшейся странице (Мастерская) нужно найти ссылку Загрузка файлов на сайт.

Нажмите рядом с каждым пустым полем кнопку Обзор, найдите папку в которой вы сохранили страницы вашего сайта, а в ней тот файл, который вы хотите загрузить на сервер. В каждом поле можно указать только один файл, но папку `images` пока не трогайте.

После того, как загрузите все файлы и получите подтверждение о том, перейдите на страницу Управление файлами.

На этой странице нажмите на ссылку Создать папку, которая находится вверху страницы. Папку назовите `images`. После создания папки вы автоматически попадаете на страницу Управление файлами.

Зайдите в папку `images` и нажмите ссылку Загрузить файлы и повторите процедуру с загрузкой файлов.

Примечание: на некоторых хостингах можно закачивать файлы по FTP (File Transfer Protocol). Этот способ значительно удобнее, так как можно будет закачать одновременно все файлы методом Копировать|Вставить, а не закачивать на сервер по одному файлу.

Таким же способом (закачкой файлов) вы в дальнейшем сможете обновлять свой проект.

Теперь можно зайти по адресу вашего сайта и посмотреть, что получилось. Следует проверить, все ли страницы на месте и все ли картинки отображаются. Если какая-то страница "404 Page not found" (т.е. страница не найдена), а вместо картинки квадратик с красным крестиком, значит либо вы неправильно указали путь к ссылке или файлу, либо просто забыли загрузить этот файл на сайт.

Нужно выполнить отладку работы сайта.

Результат предъявите преподавателю.

Перечень рекомендуемой литературы и источников

1. HTML5, CSS3 и JavaScript. Исчерпывающее руководство, Роббинс Дж., 2014
2. Лучанинов Д. В. Основы разработки web-сайтов образовательного назначения: Учебное пособие/Лучанинов Д. В.-Саратов: Ай Пи Эр Медиа, 2018, ISBN 978-5-4486-0174-3.-105.
3. Храмцов П. Б. Основы Web-технологий: Учебное пособие/Храмцов П. Б.-Москва, Саратов: Интернет-Университет Информационных Технологий (ИНТУИТ), Вузовское образование, 2017, ISBN 978-5-4487-0068-2.-375.
4. Интеграция приложений на основе WebSphere MQ: учебный курс
5. Севостьянов И. О. Поисковая оптимизация: практическое руководство по продвижению сайта в Интернете/Иван Севостьянов.-Санкт-Петербург [и др.]: Питер, 2016, ISBN 978-5-496-01968-2.-272.
6. <http://htmlbook.ru/> - материалы для разработчиков сайтов