


МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное
образовательное учреждение высшего образования
«Тульский государственный университет»

Политехнический институт
Кафедра «Промышленная автоматика и робототехника»

Утверждено на заседании кафедры
«Промышленная автоматика
и робототехника»
«17» января 2023 г., протокол № 2

И.о. заведующего кафедрой

 О.А. Ерзин

**МЕТОДИЧЕСКИЕ УКАЗАНИЯ
ПО ВЫПОЛНЕНИЮ ЛАБОРАТОРНЫХ РАБОТ
по дисциплине (модулю)**

Web-дизайн

**основной профессиональной образовательной программы
высшего образования – программы бакалавриата**

по направлению подготовки
29.03.03 Технология полиграфического и упаковочного производства

с направленностью (профилем)
Технология полиграфического производства

Формы обучения: заочная

Идентификационный номер образовательной программы: 290303-01-23

Тула 2023 год

ЛИСТ СОГЛАСОВАНИЯ
методических указаний по выполнению лабораторных работ дисциплины (модуля)

Разработчик:

Яковлев Б.С., доцент, канд. техн. наук
(ФИО, должность, ученая степень, ученое звание)


(подпись)

СОДЕРЖАНИЕ

Лабораторная работа №1	
Структурирование изображений и текстов для Web-публикации	4
Лабораторная работа №2	
Построение базовой структуры <i>html</i> -документа	7
Лабораторная работа №3	
Заверстка макета заглавной и внутренней страниц web-сайта таблицами	14
Лабораторная работа №4	
Предварительная заверстка блоков изображений и текста макета заглавной страницы таблицами	18
Лабораторная работа №5	
Предварительная заверстка блоков изображений и текста макета внутренней страницы таблицами	23
Лабораторная работа №6	
Физическое форматирование текста	28
Лабораторная работа №7	
Логическое форматирование текста	33
Лабораторная работа №8	
Создание гиперссылок	37
Лабораторная работа №9	
Использование каскадных списков стилей CSS. Адаптация ранее сверстанных макетов	41
Лабораторная работа №10	
Заверстка фоновых изображений и создание стилей текста с использованием каскадных списков стилей	44
Лабораторная работа №11	
Проверка на кроссбраузерность	48
Лабораторная работа №12	
Размещение сайта на бесплатном хостинге	50
Лабораторная работа №13	
Тестирование Web-сайта	54

Лабораторная работа №1 (2 часа)
**СТРУКТУРИРОВАНИЕ ИЗОБРАЖЕНИЙ И ТЕКСТОВ
ДЛЯ WEB-ПУБЛИКАЦИИ**

1. ЦЕЛЬ И ЗАДАЧИ РАБОТЫ

Цель работы: усвоение принципов подготовки элементов дизайна и контента веб-сайта.

Задачи работы: подготовить изображения, используемые в дизайне сайта, подготовить тексты и изображения контента.

2. ОСНОВНЫЕ ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

Залогом успеха при создании любого проекта, в том числе и веб-сайта, является системный подход. Прежде, чем начинать верстку сайта на основе эскиза, следует заранее подготовить весь текстовый и иллюстративный материал.

Подготовка изображений (при условии того, что информационное содержание целиком и полностью устраивает), сводится к следующим этапам:

- 1) Кадрирование и изменение размера.
- 2) Сохранение с оптимизацией.
- 3) Создание понятной номенклатуры имен.

Кадрирование осуществляется с помощью программы Adobe Photoshop. Для этого следует выделить желаемую область изображения и применить команду меню «Изображение > Кадрировать» («Image > Crop»).

Изменение размера изображения производится с помощью меню «Изображение > Размер изображения» («Image > Image size»).

Перед сохранением изображения следует убедиться, что его параметры соответствуют параметрам, рекомендуемым для веб: **разрешение** – 72 dpi, **цветовая модель** RGB, **глубина цвета** – 8 бит.

Сохранение изображения производится с помощью команды меню «Файл > Сохранить как» («File > Save as»). Далее следует выбрать в выпадающем списке желаемый формат файла.

Еще одна полезная функция реализуется в меню «Файл>Сохранить для веб» («File>Save for Web»). Эта команда вызывает контекстное меню с рядом настроек (для каждого формата они свои). Данное действие позволяет значительно уменьшить объем памяти, занимаемый изображением, а значит, оно будет загружаться быстрее.

Формат - способ организации информации в файле. Для Web используются следующие основные форматы со сжатием информации:

BMP – для хранения рисунков и фотографий (без сжатия);

GIF - для хранения рисунков и анимаций (недеструктивное сжатие);

JPEG - для хранения фотографий (деструктивное сжатие);

PNG - для рисунков и фотографий, возможно с прозрачным фоном (недеструктивное сжатие).

Имя файла с изображением должно быть коротким и понятным. Так, например, изображение шапки сайта имеет смысл сохранить под именем «top.jpg», а в том случае, если шапка сайта состоит из двух фрагментов, назвать их «top_left.jpg» и «top_right.jpg».

Для дальнейшего удобства имеет смысл сохранять те изображения, которые являются элементами дизайна и те изображения, которые будут использоваться в контенте сохранять в разных папках.

При подготовке к публикации текста (здесь речь идет исключительно о контенте, так как использование значительных объемов текста в дизайне – случай из ряда вон выходящий) следует ориентироваться на следующую **последовательность действий**:

- 1) Подготовленный текст тщательно изучается и многократно перечитывается.
- 2) Удаляется лишнее. При этом удаляя лишнее, оставляют смысловую часть, в оставшейся части повторяющиеся слова на другие, не искажая смысла.
- 3) Оставшийся текстовый контент сайта еще раз тщательно изучается и многократно перечитывается.

Суть написания веб-текста: правка, сокращение и еще раз сокращение.

Веб-текст имеет следующую **специфику**:

- 1) Читать текст на экране монитора сложнее, чем на бумаге. Поэтому длину строки веб-текста целесообразнее делать немного короче. Веб-текст должен быть более лаконичным, по сравнению с обычным текстом на бумаге.
- 2) Информацию на сайте лучше подавать порциями, разделяя абзацы пробелами.
- 3) Следует помнить, что пользователи могут просматривать веб-страницу на мониторах с разным разрешением. Если компоновка веб-страницы адаптивная, т.е. подстраиваемая под разрешение монитора, то число строк в абзаце будет тем больше, чем меньше разрешение монитора. И наоборот. Поэтому длину строки и число строк в абзаце веб-текста желательно планировать под основной диапазон разрешения мониторов, на которых пользователи будут просматривать веб-страницу.

Для того, чтобы тексты на сайте производили благоприятное впечатление, нужно соблюдать некоторые правила:

- 1) Заголовок начинается с заглавной буквы. В конце заголовка точка не ставится. В конце подзаголовков точка не ставится.
- 2) В конце строки списка следует ставить либо точку, либо точку с запятой, либо не ставить. Во всех текстах сайта это правило должно быть одинаковым.
- 3) Текст должен быть разбит на абзацы. Если текст требует прокрутки - следует делать абзацы разной длины - так читателю будет легче ориентироваться в нем. Каждое новое предложение начинается с заглавной буквы. Между словами ставится только один пробел. Знаки препинания (точка, запятая, двоеточие, точка с запятой) пишутся слитно с тем словом, после которого они стоят. Между знаком препинания и следующим словом (если оно есть) ставится один пробел. Если скобки находятся в середине предложения, знаки препинания ставятся вне скобок. Если скобка заканчивает предложение, точка ставится также за скобкой. Эти же правила относятся к кавычкам. Не следует использовать перенос слов, если только верстка не фиксированная. Не следует использовать для текста подчеркивания, т.к. такое форматирование сбивает с толку посетителя и заголовок становится похож на ссылку. Не следует выделять текст разными цветами. Для выделения важных частей текста следует использовать жирное или курсивное начертание. Для текста обязательно должна выполняться проверка орфографии.

После того, как все недочеты текста исправлены, следует сохранить его в любом удобном формате, таким образом, чтобы в последствии можно было копировать его фрагменты. Имеет смысл контент различных страниц сохранять в различные текстовые файлы.

3. ОБОРУДОВАНИЕ И МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ

Компьютерные классы, оборудованные ПК с установленным программным обеспечением не ниже:

- 1) Операционная система Windows XP;
- 2) Текстовый редактор MS Word.
- 3) Графический редактор Adobe Photoshop.

4. ЗАДАНИЕ НА ЛАБОРАТОРНУЮ РАБОТУ

Произвести подготовку заранее отобранных изображений и текстов для веб-публикации.

5. ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ

- 1) Ознакомиться с содержанием лабораторной работы.
- 2) Сохранить на жестком диске все отобранные для использования в макете изображения.
- 3) Изменить их размер, согласно размерам, определенным в эскизе макета.
- 4) Сохранить изображения в отдельную папку в оптимальном в каждом конкретном случае формате, дав файлам понятные имена.
- 5) Сохранить на жестком диске все отобранные для использования в контенте изображения.
- 6) Единообразно изменить их размеры и пропорции.
- 7) Сохранить изображения в отдельную папку, дав файлам понятные имена.
- 8) Сохранить на жестком диске все отобранные для использования в контенте тексты.
- 9) Отредактировать их смысловую составляющую и поделить на абзацы.
- 10) Отредактировать тексты согласно правилам набора.
- 11) Сохранить тексты в отдельную папку, дав файлам названия соответственно именам будущих страниц..

6. СОДЕРЖАНИЕ ОТЧЕТА

Отчетом по данной лабораторной работе служит сохраненный в электронном виде набор подготовленных для веб-публикации изображений и текстов.

7. КОНТРОЛЬНЫЕ ВОПРОСЫ

- 1) Как изменить размер изображения?
- 2) Как изменить пропорции изображения?
- 3) Чем объясняется такое разнообразие графических форматов?
- 4) В чем суть редактирования смысловой части текстов для веб?
- 5) Какие правила текстового набора вы знаете?

8. БИБЛИОГРАФИЧЕСКИЙ СПИСОК

- 1) Курушин, В.Д. Дизайн и реклама : самоучитель / В.Д.Курушин .— М. : ДМК Пресс, 2006.— 272с. : ил. — ISBN 5-94074-299-8.
- 2) Пауэлл, Powell T. Web-дизайн : пер.с англ. / Т.Пауэлл .— 2-е изд. — СПб. : БХВ-Петербург, 2005 .— 1072с. : ил. — (В подлиннике) .— Парал.тит.л.англ. — ISBN 5-94157-389-8.


```

</head>
<body>
    ...
</body>
</html>

```

Тег <!DOCTYPE> предназначен для указания типа текущего документа -DTD (document type definition, описание типа документа).

Это необходимо, чтобы браузер понимал, как следует интерпретировать текущую веб-страницу, поскольку HTML существует в нескольких версиях, кроме того, имеется XHTML (EXtensible HyperText Markup Language, расширенный язык разметки гипертекста), похожий на HTML, но различающийся с ним по синтаксису. Чтобы браузер «не путался» и понимал, согласно какому стандарту отображать веб-страницу и необходимо в первой строке кода задавать <!DOCTYPE>. Закрывающий тег не требуется.

Синтаксис:

```

<!DOCTYPE [Элемент верхнего уровня] [Публичность]
"[Регистрация]//[Организация]//[Тип] [Имя]//[Язык]" "[URL]">

```

Атрибуты:

Элемент верхнего уровня = Html -указывает элемент верхнего уровня в документе, для HTML это атрибут html.

Публичность = Public | System -объект является публичным или системным, например, таким как локальный файл. Для HTML/XHTML указывается значение Public.

Регистрация = + | - -сообщает, что разработчик DTD зарегистрирован в международной организации по стандартизации (International Organization for Standardization, ISO). Принимает одно из двух значений: разработчик зарегистрирован в ISO или разработчик не зарегистрирован.

Организация = W3C -уникальное название организации, разработавшей DTD. Официально HTML/XHTML публикует W3C, это название и пишется в <!DOCTYPE>.

Тип = DTD - тип описываемого документа. Для HTML/XHTML значение указывается DTD.

Имя = ... -уникальное имя документа для описания DTD.

Язык = EN язык, на котором написан текст для описания объекта. Содержит две буквы, пишется в верхнем регистре. Для документа HTML/XHTML указывается английский язык (EN).

URL = http://... - адрес документа с DTD.

Пример:

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">

```

Документ HTML начинается открывающим тегом <html> и заканчивается закрывающим тегом </html>. Между данной парой контейнерных тегов располагаются две другие основные части HTML документа:

- 1) заголовок заключенный в контейнер <head>...</head>;
- 2) тело документа в контейнере <body>...</body>.

Каждый HTML документ должен содержать основную информацию о его происхождении.

Тег <head> предназначен для хранения других элементов, цель которых -помочь браузеру в работе с данными.

Внутри контейнера `<head>` допускается размещать следующие элементы: `<base>`, `<basefont>`, `<link>`, `<meta>`, `<script>`, `<style>`, `<title>`.

Синтаксис:

```
<head>
...
</head>
```

Пример:

```
<head>
  <title>htmlbook.ru - Как правильно писать заголовок
  страницы</title>
</head>
```

Тег `<base>` определяется внутри контейнера `<head>` и инструктирует браузер относительно полного базового адреса текущего документа. Тег `<base>` предназначен для документов, в которых используется относительный адрес и эти документы могут переноситься в другую папку или даже на другой компьютер без потери связи. Браузер ищет тег `<base>`, определяет полный адрес документа и корректно загружает его. Второе применение тега `<base>` - задание целевого окна для всех ссылок на текущей странице.

Синтаксис:

```
<base>
```

Атрибуты:

`href = "http://...."` - адрес, который должен использоваться для указания полного пути к файлам.

`target = "_blank | _self | _parent | _top"` - имя окна или фрейма, куда будет загружаться документ, открываемый по ссылке.

Пример:

Если адрес документа указан как

```
<base href="http://www.megasite.ru/hzchd/">
```

то при добавлении рисунков достаточно использовать относительный адрес ``

При этом полный путь к изображению будет `http://www.megasite.ru/hzchd/images/labuda.gif`, что позволяет браузеру всегда находить графический файл, независимо от того, где находится текущая веб-страница.

Тег `<basefont>` предназначен для задания шрифта, размера и цвета текста по умолчанию. Указанные значения будут использоваться во всем документе за исключением тега ``, в котором можно переопределить параметры оформления текста. Допускается использование тега в контейнере `<head>` или `<body>`, причем несколько раз. Это позволяет изменять вид шрифта для части документа.

Синтаксис:

```
<basefont>
```

Атрибуты:

`color = "название цвета | 16-ричный код цвета"` - устанавливает цвет текста.

`face = "название гарнитуры шрифта"` - определяет гарнитуру шрифта.

`size="1 | 2 | 3 | 4 | 5 | 6 | 7 | +1 | -1"` - задает размер шрифта в условных единицах.

Пример:

```
<basefont face="Arial, Helvetica, sans-serif" size="4"
color="maroon">
```

Тег <link> устанавливает связь с внешним документом вроде файла со стилями или со шрифтами. Тег <link> размещается всегда внутри контейнера <head> и не создает ссылку.

Синтаксис:

```
<link>
```

Атрибуты:

charset = "кодировка" - кодировка связываемого документа.

href = "http://..." - путь к связываемому файлу.

media = "all | braille | handheld | print | screen | speech | projection | tty | tv " - определяет устройство, для которого следует применять стилевое оформление.

rel = "stylesheet | alternate " - определяет отношения между текущим документом и файлом, на который делается ссылка.

sizes = "ширинаХвысота" - указывает размер иконок для визуального отображения.

type = "MIME-тип" - MIME-тип данных подключаемого файла.

Пример:

```
<link rel="stylesheet" type="text/css" href="ie.css">
<link rel="alternate" type="application/rss+xml"
title="Статьи с сайта htmlbook.ru"
href="http://htmlbook.ru/rss.xml">
<link rel="shortcut icon"
href="http://htmlbook.ru/favicon.ico">
```

Тег <style> применяется для определения стилей элементов веб-страницы. Тег <style> необходимо использовать внутри контейнера <head>. Можно задавать более чем один тег <style>.

Синтаксис:

```
<head>
  <style type="text/css">
    ...
  </style>
</head>
```

Атрибуты:

media - определяет устройство вывода, для работы с которым предназначена таблица стилей.

type="text/css" - сообщает браузеру, какой синтаксис использовать, чтобы правильно интерпретировать стили.

Пример:

```
<style type="text/css">
  H1 {
    font-size: 120%;
    font-family: Verdana, Arial, Helvetica, sans-serif;
    color: #333366;
```

```
}  
</style>
```

Тег <title> определяет заголовок документа. Элемент <title> не является частью документа и не показывается напрямую на веб-странице. В операционной системе Windows текст заголовка отображается в левом верхнем углу окна браузера. Допускается использовать только один тег <title> на документ и размещать его в контейнере <head>.

Синтаксис:

```
<head>  
  <title>Заголовок</title>  
</head>
```

Пример:

```
<title>htmlbook.ru - Как правильно писать заголовок  
страницы</title>
```

Метаинформация - информация о способах и методах переработки информации или о том, где найти информацию.

Тег <meta> определяет метатеги, которые используются для хранения информации предназначенной для браузеров и поисковых систем. Например, механизмы поисковых систем обращаются к метатегам для получения описания сайта, ключевых слов и других данных. Разрешается использовать более чем один метатег, все они размещаются в контейнере <head>. Как правило, атрибуты любого метатега сводятся к парам «имя=значение».

Синтаксис:

```
HTML <head>  
  <meta content="...">  
</head>
```

Атрибуты:

charset - задает кодировку документа.

content - устанавливает значение атрибута, заданного с помощью name или http-equiv.

http-equiv - предназначен для конвертирования метатега в заголовок HTTP.

name - имя метатега, также косвенно устанавливает его предназначение.

Пример:

```
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
```

С помощью метатегов веб-мастера могут предоставлять поисковым системам информацию о своих сайтах. Метатеги можно использовать для предоставления сведений самым разным клиентам, и каждая система обрабатывает метатеги только заданного формата, игнорируя остальные.

Поисковая система Google распознает следующие метатеги (и связанные с ними элементы):

```
<meta name="keywords" content="ключевые слова">
```

Этот тег предоставляет ключевые слова для поиска страницы.

```
<meta name="description" content="Описание страницы" />
```

Этот тег предоставляет краткое описание страницы. В некоторых случаях это описание отображается вместе с фрагментом текста в результатах поиска.

```
<meta name="robots" content="..., ..." />
```

```
<meta name="googlebot" content="..., ..." />
```

Эти метатеги позволяют контролировать сканирование и индексирование поисковыми системами. Метатег robots применяется ко всем поисковым системам, а googlebot - только к Google. По умолчанию задаются значения index, follow и их не надо определять.

```
<meta name="google" content="nottranslate" />
```

Обнаружив, что содержание страницы написано на языке, который может не подойти пользователю, Google добавляет в результат поиска ссылку, позволяющую сделать перевод на нужный язык. Данный метатег сообщает Google, что владелец сайта просит не переводить эту страницу на другие языки.

```
<meta name="google-site-verification" content="..." />
```

Этот тег можно использовать на странице самого верхнего уровня, чтобы подтвердить свои права собственности на сайт в Инструментах для веб-мастеров.

```
<meta http-equiv="refresh" content="...;url=..." />
```

Этот метатег отправляет пользователя на новый URL-адрес через определенный период времени. Иногда он используется как простая форма переадресации. Однако он поддерживается не всеми браузерами и может вводить пользователей в заблуждение.

Тег <body> предназначен для хранения содержания веб-страницы (контента), отображаемого в окне браузера, то есть того, которое будет визуализировано для пользователя. Информацию, которую следует выводить в документе, следует располагать именно внутри контейнера <body>. К такой информации относится текст, изображения, теги, скрипты JavaScript и т.д.

Тег <body> также применяется для определения цветов ссылок и текста на веб-странице. Подобная практика в HTML 4 осуждается и взамен для указания цветовой схемы рекомендуется использовать стили, применяя их к селектору BODY. Тем не менее, большинство атрибутов до сих пор поддерживается разными браузерами.

Часто тег <body> используется для размещения обработчика событий, например, onload, которое выполняется после того, как документ завершил загрузку в текущее окно или фрейм.

Синтаксис:

```
<body>
```

```
...
```

```
</body>
```

Атрибуты:

alink = "название цвета | 16-ричный код цвета" - устанавливает цвет активной ссылки.

background = "http://" - задает адрес фонового рисунка веб-страницы

bgcolor = "название цвета | 16-ричный код цвета" - цвет фона веб-страницы.

bgproperties = "fixed" - определяет, прокручивать фон совместно с текстом или нет. Если требуется, чтобы фон не фиксировался, следует удалить атрибут.

link = "название цвета | 16-ричный код цвета" - цвет ссылок на веб-странице.

scroll = "yes | no" - устанавливает, отображать полосы прокрутки или нет.

text = "название цвета | 16-ричный код цвета" - цвет текста в документе.

vlink = "название цвета | 16-ричный код цвета" - цвет посещенных ссылок.

3. ОБОРУДОВАНИЕ И МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ

Компьютерные классы, оборудованные ПК с установленным программным обеспечением не ниже:

- 1) Операционная система Windows XP;
- 2) Adobe Reader;
- 3) Текстовый редактор;
- 4) Браузер.

4. ЗАДАНИЕ НА ЛАБОРАТОРНУЮ РАБОТУ

Произвести описание базовой структуры страницы проектируемого сайта.

5. ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ

- 1) Ознакомиться с содержанием лабораторной работы.
- 2) Создать файл Index.html и вписать туда шаблон html-документа.
- 3) Добавить заголовочные теги, описывающие содержимое создаваемого сайта.
- 4) Добавить атрибуты тела документа, согласно разработанной ранее цветовой схеме.
- 5) Сохранить результат.
- 6) Подготовиться к ответу на контрольные вопросы.

6. СОДЕРЖАНИЕ ОТЧЕТА

Отчетом по данной лабораторной работе служит сохраненный html-код страницы.

7. КОНТРОЛЬНЫЕ ВОПРОСЫ

- 1) Как выглядит базовая структура html-документа?
- 2) Какие теги описываются в заголовочной части?
- 3) Какие атрибуты тела документа вы знаете?
- 4) Зачем нужна метаинформация?

8. БИБЛИОГРАФИЧЕСКИЙ СПИСОК

- 1) Пауэлл, Powell T. Web-дизайн : пер.с англ. / Т.Пауэлл .— 2-е изд. — СПб. : БХВ-Петербург, 2005 .— 1072с. : ил. — (В подлиннике) .— Парал.тит.л.англ. — ISBN 5-94157-389-8.
- 2) Вин, Д. Искусство web-дизайна : Самоучитель / Дж.Вин; Пер.с англ.О.Кузнецова .— СПб.и др. : Питер, 2002 .— 224с.
- 3) Джонсон, Johnson J. Web-дизайн:типичные ляпы и как их избежать : пер.с англ. / Д.Джонсон .— М. : КУДИЦ-ОБРАЗ, 2005 .— 400с.
- 4) Кирсанов, Д. Веб-дизайн:книга Дмитрия Кирсанова / Д.Кирсанов .— СПб. : Символ-Плюс, 2001 .— 376с.

Лабораторная работа №3 (4 часа)
**ЗАВЕРСТКА МАКЕТА ЗАГЛАВНОЙ И ВНУТРЕННЕЙ
СТРАНИЦ WEB-САЙТА ТАБЛИЦАМИ**

1. ЦЕЛЬ И ЗАДАЧИ РАБОТЫ

Цель работы: усвоение основных принципов табличной верстки.

Задачи работы: заверстать модульную сетку заглавной страницы сайта в таблицу, заверстать модульную сетку внутренней страницы сайта в таблицу.

2. ОСНОВНЫЕ ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

Для добавления таблицы на веб-страницу используется тег `<table>`. Этот элемент служит контейнером для элементов, определяющих содержимое таблицы. Любая таблица состоит из строк и ячеек. Таблица должна содержать хотя бы одну ячейку.

Тег `<table>` служит контейнером для элементов, определяющих содержимое таблицы. Любая таблица состоит из строк и ячеек, которые задаются с помощью тегов `<tr>` и `<td>`. Внутри `<table>` допустимо использовать следующие элементы: `<caption>`, `<col>`, `<colgroup>`, `<tbody>`, `<td>`, `<tfoot>`, `<th>`, `<thead>` и `<tr>`.

Атрибуты:

`align="left | center | right"` - определяет выравнивание таблицы.

`background = "http://...."` - задает фоновый рисунок в таблице.

`bgcolor = "название цвета | 16-ричный код цвета"` - цвет фона таблицы.

`border = "целое положительное число в пикселях"` - толщина рамки в пикселях.

`bordercolor = "название цвета | 16-ричный код цвета"` - цвет рамки.

`cellpadding = "целое значение в пикселях | целое число в процентах от доступного пространства"` - отступ от рамки до содержимого ячейки.

`cellspacing = "целое положительное число"` - расстояние между ячейками.

`cols = "целое положительное число"` - число колонок в таблице.

`frame=" void | border | above | below | hside | vside | rhs | lhs "` - сообщает браузеру, как отображать границы вокруг таблицы.

`height = "целое значение в пикселях | целое число в процентах от доступного пространства"` - высота таблицы.

`rules = " all | groups | cols | none | rows "` - сообщает браузеру, где отображать границы между ячейками.

`summary = "текст"` - краткое описание таблицы.

`width = "целое значение в пикселях | целое число в процентах от доступного пространства"` - ширина таблицы.

Тег `<tr>` служит контейнером для создания строки таблицы. Каждая ячейка в пределах такой строки может задаваться с помощью тега `< th>` или `<td>`.

Атрибуты:

`align = "left | center | right | justify | char"` - определяет выравнивание содержимого ячеек по горизонтали.

`bgcolor = "название цвета | 16-ричный код цвета"` - цвет фона ячеек.

`bordercolor = "название цвета | 16-ричный код цвета"` - цвет рамки.

`char = "текстовый символ"` - выравнивание содержимого ячеек относительно заданного символа.

`charoff = "любое целое число"` - смещение содержимого ячеек относительно указанного символа.

`valign = "top | middle | bottom | baseline"` - выравнивание содержимого ячеек по вертикали.

Тег <td> Предназначен для создания одной ячейки таблицы. Тег <td> должен размещаться внутри контейнера <tr>, который в свою очередь располагается внутри тега <table>.

Атрибуты:

`abbr = "текст"` - краткое описание содержимого ячейки.

`align = "left | center | right | justify | char"` - определяет выравнивание содержимого ячейки по горизонтали.

`axis = "текст"` - группирует ячейки связанные между собой похожей информацией. Значение атрибута - любой текст, представляющий собой произвольное имя ячейки. Для связывания ячеек между собой значения атрибута `axis` должны совпадать.

`background = "http://...."` - задает фоновый рисунок в ячейке.

`bgcolor = "название цвета | 16-ричный код цвета"` - цвет фона ячейки.

`bordercolor = "название цвета | 16-ричный код цвета"` - цвет рамки.

`char = "текстовый символ"` - выравнивание содержимого ячеек относительно заданного символа.

`charoff = "любое целое число"` - смещение содержимого ячеек относительно указанного символа.

`colspan = "целое положительное число больше 1"` - объединяет горизонтальные ячейки.

`height = "целое значение в пикселях | целое число в процентах от доступного пространства"` - высота ячейки.

`nowrap` - запрещает перенос строк.

`rowspan = "целое положительное число больше 1"` - объединяет вертикальные ячейки.

`valign = "top | middle | bottom | baseline"` - выравнивание содержимого ячейки по вертикали.

`width = "целое значение в пикселях | целое число в процентах от доступного пространства"` - ширина ячейки.

Тег <th> предназначен для создания одной ячейки таблицы, которая обозначается как заголовочная. Текст в такой ячейке отображается браузером обычно жирным шрифтом

и выравнивается по центру. Тег <th> должен размещаться внутри контейнера <tr>, который в свою очередь располагается внутри тега <table>.

Атрибуты: аналогично тегу <td>.

Синтаксис:

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

Для объединения двух и более ячеек в одну используются атрибуты colspan и rowspan тега <td>. Атрибут colspan устанавливает число ячеек объединяемых по горизонтали. Аналогично работает и атрибут rowspan, с тем лишь отличием, что объединяет ячейки по вертикали. Перед добавлением **следует проверять число ячеек в каждой строке**, чтобы не возникло ошибок. Так, если в одной строке

```
<td colspan="3">
```

заменяет три ячейки, то в следующей строке должно быть три тега <td>:

или конструкция вида

```
<td colspan="2">...</td>
```

```
<td>...</td>
```

Если число ячеек в каждой строке не будет совпадать, появятся пустые **фантомные ячейки**.

3. ОБОРУДОВАНИЕ И МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ

Компьютерные классы, оборудованные ПК с установленным программным обеспечением не ниже:

- 1) Операционная система Windows XP;
- 2) Adobe Reader;
- 3) Текстовый редактор;
- 4) Браузер.

4. ЗАДАНИЕ НА ЛАБОРАТОРНУЮ РАБОТУ

Произвести табличную заверстку ранее разработанной модульной сетки макета заглавной и внутренней страниц.

5. ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ

- 1) Ознакомиться с содержанием лабораторной работы.
- 2) Рассмотрев эскиз модульной сетки заглавной страницы, определить, сколько ячеек будет в первоуровневой таблице и записать html-код таблицы с нужным количеством ячеек.
- 3) Определить, какие из ячеек должны быть объединены и внести коррективы в код.
- 4) Определить, требуются ли вложенные таблицы. Если требуются, сначала создать их по отдельности, а затем заверстать в соответствующие ячейки.
- 5) Задать размеры для тех элементов, для которых это необходимо.
- 6) Отредактировать границы ячеек, и присвоить им цвет, соответствующий разработанной ранее цветовой концепции.
- 7) Сохранить результат.
- 8) Повторить действия для макета внутренней страницы.

9) Подготовиться к ответу на контрольные вопросы.

6. СОДЕРЖАНИЕ ОТЧЕТА

Отчетом по данной лабораторной работе служит сохраненный html-код заглавной и внутренней страниц с заверстанной в таблицу модульной сеткой.

7. КОНТРОЛЬНЫЕ ВОПРОСЫ

- 1) Какие теги для создания таблиц вы знаете?
- 2) Какие атрибуты таблиц будут использоваться для создания дизайна?
- 3) Чем отличаются верстка «матрешками» и верстка с объединением ячеек?
- 4) Каким образом производится объединение ячеек?
- 5) Как задать фоновый цвет ячейки?

8. БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1) Пауэлл, Powell T. Web-дизайн : пер.с англ. / Т.Пауэлл .— 2-е изд. — СПб. : БХВ-Петербург, 2005 .— 1072с. : ил. — (В подлиннике) .— Парал.тит.л.англ. — ISBN 5-94157-389-8.

2) Вин, Д. Искусство web-дизайна : Самоучитель / Дж.Вин; Пер.с англ.О.Кузнецова .— СПб.и др. : Питер, 2002 .— 224с.

3) Джонсон, Johnson J. Web-дизайн:типичные ляпы и как их избежать : пер.с англ. / Д.Джонсон .— М. : КУДИЦ-ОБРАЗ, 2005 .— 400с.

4) Кирсанов, Д. Веб-дизайн:книга Дмитрия Кирсанова / Д.Кирсанов .— СПб. : Символ-Плюс, 2001 .— 376с.

ПРЕДВАРИТЕЛЬНАЯ ЗАВЕРСТКА БЛОКОВ ИЗОБРАЖЕНИЙ И ТЕКСТА МАКЕТА ЗАГЛАВНОЙ СТРАНИЦЫ ТАБЛИЦАМИ

1. ЦЕЛЬ И ЗАДАЧИ РАБОТЫ

Цель работы: усвоение основных принципов заверстки текстовых блоков и изображений.

Задачи работы: заверстать в таблицу текстовые и изобразительные элементы заглавной страницы.

2. ОСНОВНЫЕ ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

Для изменения вида текста существует достаточно большое количество различных тегов. Это и немудрено, ведь текст самый популярный вид информации.

Прежде чем редактировать код веб-страницы, следует принять во внимание некоторые особенности, которые присущи HTML при работе с текстом.

1) Любое количество пробелов идущих подряд, в браузере отображается как один. Сколько бы ни было пробелов между словами, это никак не повлияет на конечный вид текста. Это же правило относится к символам табуляции и переносу текста. Исключением из этого правила является тег `<pre>`, внутри которого любое число пробелов отображается именно так, как оно указано в коде.

2) HTML не поддерживает расстановку переносов в словах, как это делают текстовые редакторы, иначе говоря, все слова пишутся целиком без их разбиения. Это условие несущественно, пока не используется выравнивание текста по ширине. В этом случае блок текста выравнивается по левому и правому краю. Короткие строки при этом растягиваются за счет автоматического добавления пробелов между словами. Иногда пустые блоки между словами настолько велики, что портят внешний вид страницы и ухудшают читабельность текста.

3) Текст занимает ширину окна браузера или отведенного под него элемента макета. Если написать одну длинную строку в коде HTML, то в браузере она будет отформатирована, чтобы текст поместился по ширине в окно. Переносы текста будут добавлены автоматически в местах пробела или дефиса. Если в тексте нет ни того, ни другого символа, браузер не сможет создать переносы и отобразит текст одной строкой. Если она шире окна браузера, то неминуемо появится горизонтальная полоса прокрутки.

Тег `<p>` определяет текстовый абзац. Тег `<p>` является блочным элементом, всегда начинается с новой строки, абзацы текста идущие друг за другом разделяются между собой отбивкой. Величиной отбивки можно управлять с помощью стилей. Если закрывающего тега нет, считается, что конец абзаца совпадает с началом следующего блочного элемента.

Синтаксис:

`<p>Текст</p>`

Атрибуты:

`align="left | center | right | justify"` - определяет выравнивание текста.

Пример:

`<p align="center">Duis te feugifacilisi. </p>`

`<p align="right">Duis autem dolor in </p>`

Для того, чтобы управлять отступами от границ родительского элемента (ячейки таблицы) и выключкой текстового блока следует использовать атрибуты строк и ячеек таблицы (см. л/р № 3).

**Тег ** предназначен для отображения на веб-странице изображений в графическом формате GIF, JPEG или PNG. Этот тег имеет единственный обязательный атрибут `src`, который определяет адрес файла с картинкой. Если необходимо, то рисунок можно сделать ссылкой на другой файл, поместив тег `` в контейнер `<a>`. При этом вокруг изображения отображается рамка, которую можно убрать, добавив атрибут `border="0"` в тег ``.

Рисунки также могут применяться в качестве карт-изображений, когда картинка содержит активные области, выступающие в качестве ссылок. Такая карта по внешнему виду ничем не отличается от обычного изображения, но при этом оно может быть разбито на невидимые зоны разной формы, где каждая из областей служит ссылкой.

Синтаксис:

```

```

Атрибуты:

`align="bottom | left | middle | right | top"` - определяет как рисунок будет выравниваться по краю и способ обтекания текстом.

`alt="текст"` - альтернативный текст для изображения. Этот тег предназначен для того, чтобы пользователь даже если изображение не грузится, мог получить информацию о том, что на нем изображено.

`border="целое число в пикселях"` - толщина рамки вокруг изображения.

`height = "целое положительное число в пикселях или процентах"` - высота изображения.

`hspace="целое положительное число в пикселях"` - горизонтальный отступ от изображения до окружающего контента.

`ismap` - говорит браузеру, что картинка является серверной картой-изображением.

`longdesc = "URL"` - указывает адрес документа, где содержится аннотация к картинке.

`src="URL"` - путь к графическому файлу.

`vspace="целое положительное число в пикселях"` - вертикальный отступ от изображения до окружающего контента.

`width="целое положительное число в пикселях или процентах"` - ширина изображения.

`usemap="#идентификатор"` - ссылка на тег `<map>`, содержащий координаты для клиентской карты-изображения.

Пример:

```

```

Тег <map> служит контейнером для элементов `<area>`, которые определяют активные области для карт-изображений. Такие области устанавливают невидимые зоны на изображении, являющиеся ссылками на HTML-документы. Цель использования тега `<map>` — в связывании тега `` с клиентской картой-изображением. Эта связь определяется применением единого идентификатора как в теге ``, задаваемого атрибутом `usemap`, так и в теге `<map>`, устанавливаемого атрибутом `name`.

Синтаксис:

```
<map name="имя">
  <area атрибуты>
</map>
```

Атрибуты:

name="идентификатор" - имя карты-изображения.

Элемент <area> определяет активные области изображения, которые являются ссылками. Рисунок с привязанными к нему активными областями называется в совокупности картой-изображением. Такая карта по внешнему виду ничем не отличается от обычного изображения, но при этом оно может быть разбито на невидимые зоны разной формы, где каждая из областей служит ссылкой. Тег <area> задает форму области, ее размеры, устанавливает адрес документа, на который следует сделать ссылку, а также имя окна или фрейма, куда браузер будет загружать документ. Этот тег всегда располагается в контейнере <map>, который связывает координаты областей с изображением.

Несколько областей могут перекрывать друг друга, сверху будет та, которая в коде HTML располагается выше.

Синтаксис:

```
<area href="URL">
```

Атрибуты

accesskey = "клавиша" - переход к области с помощью комбинации клавиш.

alt="текст" - альтернативный текст для области изображения.

coords = "координата 1, координата 2, ..." - координаты активной области.

href="URL" - задает адрес документа, на который следует перейти.

hreflang = "код языка" - указывает язык документа, на который ведет ссылка.

nohref - область без ссылки на другой документ.

shape = "circle | default | poly | rect" - форма области.

tabindex="любое целое положительное число, начиная с нуля" - задает последовательность перехода между элементами с помощью клавиши Tab. Значения выстраиваются последовательно и переход между элементами происходит от меньшего значения к большему.

target="_blank | _self | _parent | _top" - имя окна или фрейма, куда браузер будет загружать документ.

type = "MIME тип" - устанавливает MIME-тип документа, на который ведёт ссылка.

Пример:

```
<map id="Navigation" name="Navigation">
  <area shape="poly" coords="113,24,211,24,233,0,137,0"
  href="inform.html" alt="Информация" />
  <area shape="poly" coords="210,24,233,0,329,0,307,24"
  href="activity.html" alt="Мероприятия" />
  <area shape="poly" coords="304,24,385,24,407,0,329,0"
  href="depart.html" alt="Отделения" />
</map></p>
```

3. ОБОРУДОВАНИЕ И МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ

Компьютерные классы, оборудованные ПК с установленным программным обеспечением не ниже:

- 1) Операционная система Windows XP;
- 2) Adobe Reader;
- 3) Текстовый редактор;
- 4) Браузер.

4. ЗАДАНИЕ НА ЛАБОРАТОРНУЮ РАБОТУ

Произвести заверстку текстовых блоков и изображений в ранее созданные таблицы макетов заглавной и внутренней страниц.

5. ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ

- 1) Ознакомиться с содержанием лабораторной работы.
- 2) Рассмотрев эскиз модульной сетки заглавной страницы и проанализировав код ее таблицы, определить, в каких ячейках будут располагаться изображения.
- 3) Заверстать изображения в ячейки, отрегулировав их размеры и выравнивание.
- 4) Определить, в каких ячейках будет располагаться текст и текстовые блоки.
- 5) Заверстать текстовые блоки и фрагменты (в том числе и те текстовые элементы, которые в дальнейшем будут ссылками отдельными абзацами. Заголовки, списки и т.д. также выделить отдельными абзацами.
- 6) Задать всем абзацам необходимые параметры выключки.
- 7) Сохранить результат.
- 9) Подготовиться к ответу на контрольные вопросы.

6. СОДЕРЖАНИЕ ОТЧЕТА

Отчетом по данной лабораторной работе служит сохраненный html-код заглавной страницы с заверстанными изображениями и текстовыми блоками.

7. КОНТРОЛЬНЫЕ ВОПРОСЫ

- 1) С помощью какого тега осуществляется заверстка изображения? Какие его атрибуты вы знаете?
- 2) Как устанавливать отступы от изображения до его родительского элемента?
- 3) Можно ли деформировать изображение непропорционально только с помощью тегов?
- 4) На ваш взгляд, следует ли сопровождать изображения альтернативными подписями?

8. БИБЛИОГРАФИЧЕСКИЙ СПИСОК

- 1) Пауэлл, Powell T. Web-дизайн : пер.с англ. / Т.Пауэлл .— 2-е изд. — СПб. : БХВ-Петербург, 2005 .— 1072с. : ил. — (В подлиннике) .— Парал.тит.л.англ. — ISBN 5-94157-389-8.
- 2) Вин, Д. Искусство web-дизайна : Самоучитель / Дж.Вин; Пер.с англ.О.Кузнецова .— СПб.и др. : Питер, 2002 .— 224с.
- 3) Джонсон, Johnson J. Web-дизайн:типичные ляпы и как их избежать : пер.с англ. / Д.Джонсон .— М. : КУДИЦ-ОБРАЗ, 2005 .— 400с.

4) Кирсанов, Д. Веб-дизайн: книга Дмитрия Кирсанова / Д.Кирсанов .— СПб. : Символ-Плюс, 2001 .— 376с.

ПРЕДВАРИТЕЛЬНАЯ ЗАВЕРСТКА БЛОКОВ ИЗОБРАЖЕНИЙ И ТЕКСТА МАКЕТА ВНУТРЕННЕЙ СТРАНИЦЫ ТАБЛИЦАМИ

1. ЦЕЛЬ И ЗАДАЧИ РАБОТЫ

Цель работы: усвоение основных принципов заверстки текстовых блоков и изображений.

Задачи работы: заверстать в таблицу текстовые и изобразительные элементы внутренней страницы.

2. ОСНОВНЫЕ ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

Для изменения вида текста существует достаточно большое количество различных тегов. Это и немудрено, ведь текст самый популярный вид информации.

Прежде чем редактировать код веб-страницы, следует принять во внимание некоторые особенности, которые присущи HTML при работе с текстом.

1) Любое количество пробелов идущих подряд, в браузере отображается как один. Сколько бы ни было пробелов между словами, это никак не повлияет на конечный вид текста. Это же правило относится к символам табуляции и переносу текста. Исключением из этого правила является тег `<pre>`, внутри которого любое число пробелов отображается именно так, как оно указано в коде.

2) HTML не поддерживает расстановку переносов в словах, как это делают текстовые редакторы, иначе говоря, все слова пишутся целиком без их разбиения. Это условие несущественно, пока не используется выравнивание текста по ширине. В этом случае блок текста выравнивается по левому и правому краю. Короткие строки при этом растягиваются за счет автоматического добавления пробелов между словами. Иногда пустые блоки между словами настолько велики, что портят внешний вид страницы и ухудшают читабельность текста.

3) Текст занимает ширину окна браузера или отведенного под него элемента макета. Если написать одну длинную строку в коде HTML, то в браузере она будет отформатирована, чтобы текст поместился по ширине в окно. Переносы текста будут добавлены автоматически в местах пробела или дефиса. Если в тексте нет ни того, ни другого символа, браузер не сможет создать переносы и отобразит текст одной строкой. Если она шире окна браузера, то неминуемо появится горизонтальная полоса прокрутки.

Тег `<p>` определяет текстовый абзац. Тег `<p>` является блочным элементом, всегда начинается с новой строки, абзацы текста идущие друг за другом разделяются между собой отбивкой. Величиной отбивки можно управлять с помощью стилей. Если закрывающего тега нет, считается, что конец абзаца совпадает с началом следующего блочного элемента.

Синтаксис:

`<p>Текст</p>`

Атрибуты:

`align="left | center | right | justify"` - определяет выравнивание текста.

Пример:

`<p align="center">Duis te feugifacilisi. </p>`

`<p align="right">Duis autem dolor in </p>`

Для того, чтобы управлять отступами от границ родительского элемента (ячейки таблицы) и выключкой текстового блока следует использовать атрибуты строк и ячеек таблицы (см. л/р № 3).

**Тег
** устанавливает перевод строки в том месте, где этот тег находится. В отличие от тега абзаца <p>, использование тега
 не добавляет пустой отступ перед строкой. Если текст, в котором используется перевод строки, обтекает плавающий элемент, то с помощью атрибута clear тега
 можно сделать так, чтобы следующая строка начиналась ниже элемента.

Синтаксис:

Текст
 текст

Атрибуты:

clear="all | left | right | none" - сообщает браузеру, как обрабатывать следующую строку, если текст обтекает плавающий элемент.

Пример:

```
<p>Р.Л. Стивенсон</p>
<p>Лето в стране настало,<br>
    Вереск опять цветет.<br>
    Но некому готовить<br>
    Вересковый мед.</p>
```

**Тег ** предназначен для отображения на веб-странице изображений в графическом формате GIF, JPEG или PNG. Этот тег имеет единственный обязательный атрибут src, который определяет адрес файла с картинкой. Если необходимо, то рисунок можно сделать ссылкой на другой файл, поместив тег в контейнер <a>. При этом вокруг изображения отображается рамка, которую можно убрать, добавив атрибут border="0" в тег .

Рисунки также могут применяться в качестве карт-изображений, когда картинка содержит активные области, выступающие в качестве ссылок. Такая карта по внешнему виду ничем не отличается от обычного изображения, но при этом оно может быть разбито на невидимые зоны разной формы, где каждая из областей служит ссылкой.

Синтаксис:

```

```

Атрибуты:

align="bottom | left | middle | right | top" - определяет как рисунок будет выравниваться по краю и способ обтекания текстом.

alt="текст" - альтернативный текст для изображения. Этот тег предназначен для того, чтобы пользователь даже если изображение не грузится, мог получить информацию о том, что на нем изображено.

border="целое число в пикселях" - толщина рамки вокруг изображения.

height = "целое положительное число в пикселях или процентах" - высота изображения.

hspace="целое положительное число в пикселях" - горизонтальный отступ от изображения до окружающего контента.

ismap - говорит браузеру, что картинка является серверной картой-изображением.

longdesc = "URL" - указывает адрес документа, где содержится аннотация к картинке.

src="URL" - путь к графическому файлу.

`vspace="целое положительное число в пикселях"` - вертикальный отступ от изображения до окружающего контента.

`width="целое положительное число в пикселях или процентах"` - ширина изображения.

`usemap="#идентификатор"` - ссылка на тег `<map>`, содержащий координаты для клиентской карты-изображения.

Пример:

```

```

Тег `<map>` служит контейнером для элементов `<area>`, которые определяют активные области для карт-изображений. Такие области устанавливают невидимые зоны на изображении, являющиеся ссылками на HTML-документы. Цель использования тега `<map>` — в связывании тега `` с клиентской картой-изображением. Эта связь определяется применением единого идентификатора как в теге ``, задаваемого атрибутом `usemap`, так и в теге `<map>`, устанавливаемого атрибутом `name`.

Синтаксис:

```
<map name="имя">  
  <area атрибуты>  
</map>
```

Атрибуты:

`name="идентификатор"` - имя карты-изображения.

Элемент `<area>` определяет активные области изображения, которые являются ссылками. Рисунок с привязанными к нему активными областями называется в совокупности картой-изображением. Такая карта по внешнему виду ничем не отличается от обычного изображения, но при этом оно может быть разбито на невидимые зоны разной формы, где каждая из областей служит ссылкой. Тег `<area>` задает форму области, ее размеры, устанавливает адрес документа, на который следует сделать ссылку, а также имя окна или фрейма, куда браузер будет загружать документ. Этот тег всегда располагается в контейнере `<map>`, который связывает координаты областей с изображением.

Несколько областей могут перекрывать друг друга, сверху будет та, которая в коде HTML располагается выше.

Синтаксис:

```
<area href="URL">
```

Атрибуты

`accesskey = "клавиша"` - переход к области с помощью комбинации клавиш.

`alt="текст"` - альтернативный текст для области изображения.

`coords = "координата 1, координата 2, ..."` - координаты активной области.

`href="URL"` - задает адрес документа, на который следует перейти.

`hreflang = "код языка"` - указывает язык документа, на который ведет ссылка.

`noreferrer` - область без ссылки на другой документ.

`shape = "circle | default | poly | rect"` - форма области.

`tabindex="любое целое положительное число, начиная с нуля"` - задает последовательность перехода между элементами с помощью клавиши Tab. Значения выстраиваются последовательно и переход между элементами происходит от меньшего значения к большему.

target="_blank | _self | _parent | _top" - имя окна или фрейма, куда браузер будет загружать документ.

type = "MIME тип" - устанавливает MIME-тип документа, на который ведёт ссылка.

Пример:

```
<map id="Navigation" name="Navigation">  
  <area shape="poly" coords="113,24,211,24,233,0,137,0"  
    href="inform.html" alt="Информация" />  
  <area shape="poly" coords="210,24,233,0,329,0,307,24"  
    href="activity.html" alt="Мероприятия" />  
  <area shape="poly" coords="304,24,385,24,407,0,329,0"  
    href="depart.html" alt="Отделения" />  
</map></p>
```

3. ОБОРУДОВАНИЕ И МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ

Компьютерные классы, оборудованные ПК с установленным программным обеспечением не ниже:

- 1) Операционная система Windows XP;
- 2) Adobe Reader;
- 3) Текстовый редактор;
- 4) Браузер.

4. ЗАДАНИЕ НА ЛАБОРАТОРНУЮ РАБОТУ

Произвести заверстку текстовых блоков и изображений в ранее созданные таблицы макетов заглавной и внутренней страниц.

5. ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ

- 1) Ознакомиться с содержанием лабораторной работы.
- 2) Рассмотрев эскиз модульной сетки внутренней страницы и проанализировав код ее таблицы, определить, в каких ячейках будут располагаться изображения.
- 3) Заверстать изображения в ячейки, отрегулировав их размеры и выравнивание.
- 4) Определить, в каких ячейках будет располагаться текст и текстовые блоки.
- 5) Заверстать текстовые блоки и фрагменты (в том числе и те текстовые элементы, которые в дальнейшем будут ссылками отдельными абзацами. Заголовки, списки и т.д. также выделить отдельными абзацами.
- 6) Задать всем абзацам необходимые параметры выключки.
- 7) Сохранить результат.
- 9) Подготовиться к ответу на контрольные вопросы.

6. СОДЕРЖАНИЕ ОТЧЕТА

Отчетом по данной лабораторной работе служит сохраненный html-код внутренней страницы с заверстанными изображениями и текстовыми блоками.

7. КОНТРОЛЬНЫЕ ВОПРОСЫ

- 1) Что такое карты изображений?
- 2) Какие особенности текста при заверстке в html вы знаете?
- 3) Каким образом следует форматировать объемные текстовые блоки?

4) Как осуществляется расстановка переносов?

8. БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1) Пауэлл, Powell T. Web-дизайн : пер.с англ. / Т.Пауэлл .— 2-е изд. — СПб. : БХВ-Петербург, 2005 .— 1072с. : ил. — (В подлиннике) .— Парал.тит.л.англ. — ISBN 5-94157-389-8.

2) Вин, Д. Искусство web-дизайна : Самоучитель / Дж.Вин; Пер.с англ.О.Кузнецова .— СПб.и др. : Питер, 2002 .— 224с.

3) Джонсон, Johnson J. Web-дизайн:типичные ляпы и как их избежать : пер.с англ. / Д.Джонсон .— М. : КУДИЦ-ОБРАЗ, 2005 .— 400с.

4) Кирсанов, Д. Веб-дизайн:книга Дмитрия Кирсанова / Д.Кирсанов .— СПб. : Символ-Плюс, 2001 .— 376с.

Лабораторная работа №6 (4 часа)
ФИЗИЧЕСКОЕ ФОРМАТИРОВАНИЕ ТЕКСТА

1. ЦЕЛЬ И ЗАДАЧИ РАБОТЫ

Цель работы: усвоение основных принципов физического форматирования текста.

Задачи работы: задать текстовым элементам размер, гарнитуру шрифта, цвет, начертание и оформление.

2. ОСНОВНЫЕ ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

**Тег ** представляет собой контейнер для изменения характеристик шрифта, таких как размер, цвет и гарнитура. Хотя этот тег до сих пор поддерживается всеми браузерами, он считается устаревшим и от его использования рекомендуется отказаться в пользу стилей.

Синтаксис:

`Текст`

Атрибуты:

`color = "название цвета | 16-ричный код цвета"` - устанавливает цвет текста.

`face = "название гарнитуры шрифта"` - определяет гарнитуру шрифта.

`size="1 | 2 | 3 | 4 | 5 | 6 | 7 | +1 | -1"` - задает размер шрифта в условных единицах.

Пример:

`<p>Первая буква этого предложения написана шрифтом Arial, выделена красным цветом и увеличена в размерах.</p>`

**Тег ** устанавливает жирное начертание шрифта. Допустимо использовать этот тег совместно с другими тегами, которые определяют начертание текста.

Синтаксис:

`Текст`

Пример:

`<p>Lorem ipsum dolor sit amet</p>`

`<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam </p>`

Тег <i> устанавливает курсивное начертание шрифта. Допустимо использовать этот тег совместно с другими тегами, которые определяют начертание текста.

Синтаксис:

`<i>Текст</i>`

Пример:

`<p><i>Lorem ipsum dolor sit amet</i></p>`

`<p><i>Lorem</i> ipsum dolor sit amet, consectetur adipiscing elit</p>`

Тег <u> Добавляет подчеркивание к тексту. Этот тег осуждается спецификацией HTML, взамен рекомендуется использовать стили.

Синтаксис:

`<u>Текст</u>`

Пример:

`<p><u>Меню</u></p>`

```
<p>Пахлава</p>
<p>Курабье</p>
<p>Шакер-бура</p>
```

Тег `<s>` отображает текст как перечеркнутый. Этот тег аналогичен тегу `<strike>`, но в отличие от него имеет сокращенную форму записи подобно тегам ``, `<i>` и `<u>`. Взамен этого тега рекомендуется использовать стили.

Синтаксис:

```
<strike>Текст</strike>
```

Пример:

```
<p><b>Список должников</b></p>
<p><strike>Чебурашка</strike><br>
<s>Крокодил Гена</s><br>
Шапокляк</p>
```

Тег `<marquee>` создает бегущую строку на странице. На самом деле содержимое контейнера `<marquee>` не ограничивается строками и позволяет перемещать (скролировать) любые элементы веб-страницы - изображения, текст, таблицы, элементы форм и т.д. Перемещение можно задать не только по горизонтали, но и вертикали, в этом случае указываются размеры области, в которой будет происходить движение. Первоначально тег `<marquee>` был предназначен только для браузера Internet Explorer, но современные версии других браузеров также понимают и поддерживают этот тег. Этот тег не входит в спецификацию HTML.

Синтаксис:

```
<marquee>...</marquee>
```

Атрибуты:

`behavior = "alternate | scroll | slide"` - задает тип движения содержимого контейнера `<marquee>`.

`bgcolor = "название цвета | 16-ричный код цвета"` - задает цвет фона под содержимым элементом.

`direction="down | left | right | up"` - указывает направление движения содержимого контейнера `<marquee>`.

`height="целое положительное число в пикселях | в процентах"` - высота области прокрутки.

`hspace="целое положительное число в пикселях"` - горизонтальные поля вокруг контента.

`loop="число повторений"` - задает, сколько раз будет прокручиваться содержимое.

`scrollamount="целое положительное число в пикселях"` - скорость движения контента.

`scrolldelay="целое положительное число в миллисекундах"` - величина задержки в миллисекундах между движениями.

`trueSpeed` - отменяет встроенный ограничитель скорости при низких значениях атрибута `scrollDelay`.

`vspace="целое положительное число в пикселях"` - вертикальные поля вокруг содержимого.

`width="целое положительное число в пикселях | в процентах"` - ширина области прокрутки.

Пример:

```
<marquee behavior="alternate" direction="left" bgcolor="#ffcc00"> Lorem ipsum dolor sit amet... </marquee>
```

Тег <tt> отображает текст моноширинным шрифтом.

Синтаксис:

```
<tt>Текст</tt>
```

Пример:

```
<tt>Lorem ipsum dolor sit amet</tt>
```

Тег <Hx>. HTML предлагает шесть заголовков разного уровня, которые показывают относительную важность секции, расположенной после заголовка. Так, тег <h1> представляет собой наиболее важный заголовок первого уровня, а тег <h6> служит для обозначения заголовка шестого уровня и является наименее значительным. По умолчанию, заголовок первого уровня отображается самым крупным шрифтом жирного начертания, заголовки последующего уровня по размеру меньше. Теги <h1>,...,<h6> относятся к блочным элементам, они всегда начинаются с новой строки, а после них другие элементы отображаются на следующей строке. Кроме того, перед заголовком и после него добавляется пустое пространство.

Синтаксис:

```
<h1>Заголовок первого уровня</h1>
```

Атрибуты:

align = "left | center | right | justify" - определяет выравнивание заголовка.

Пример:

```
<h1>Lorem ipsum dolor sit amet</h1>
```

```
<p>Lorem ipsum dolor sit amet, consectetur </p>
```

Тег <big> увеличивает размер шрифта на единицу по сравнению с обычным текстом. В HTML размер шрифта измеряется в условных единицах от 1 до 7, средний размер текста, используемый по умолчанию, принят 3. Таким образом, добавление тега <big> увеличивает текст на одну условную единицу. Допускается применение вложенных тегов <big>, при этом размер шрифта будет больше с каждым уровнем.

На размер шрифта влияет не только заданный атрибут size тега , но и выбор гарнитуры шрифта. Шрифт Arial выглядит крупнее, чем шрифт Times, а шрифт Verdana чуть больше шрифта Arial.

Синтаксис:

```
<big>Текст</big>
```

Пример:

```
<p><big>Lorem</big> ipsum dolor sit amet</p>
```

Тег <small> уменьшает размер шрифта на единицу по сравнению с обычным текстом.

Синтаксис:

```
<small>Текст</small>
```

Пример:

```
<p>Из семейства кошачьих самая большая кошка это совсем не  
<small>лев</small>, как можно было бы подумать. </p>
```

Тег <sub> отображает шрифт в виде нижнего индекса. Текст при этом располагается ниже базовой линии остальных символов строки и уменьшенного размера.

Синтаксис:

_{Текст}

Пример:

<p> CH₃CHO + NH₂OH = CH₃C
(=NOH)H + H₂O</p>

Тег <sup> отображает шрифт в виде верхнего индекса. Шрифт при этом отображается выше базовой линии текста и уменьшенного размера.

Синтаксис:

^{Текст}

Пример:

<p><i>X ^{<small>n</small>} + Y ^{<small>n</small>}
n</small></sup>= Z ^{<small>n</small>}</i></p>

Тег <hr> рисует горизонтальную линию, которая по своему виду зависит от используемых параметров, а также браузера. Тег <hr> относится к блочным элементам, линия всегда начинается с новой строки, а после нее все элементы отображаются на следующей строке.

Синтаксис:

<hr>

Атрибуты:

align = "left | center | right" - определяет выравнивание линии.

color = "название цвета | 16-ричный код цвета" - цвет линии.

noshade - рисует линию без трехмерных эффектов.

size = "целое положительное число в пикселях" - толщина линии.

width = "целое положительное число в процентах" - ширина линии.

Пример:

<hr noshade size="4" width="75%">

<hr noshade size="2" width="60%">

3. ОБОРУДОВАНИЕ И МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ

Компьютерные классы, оборудованные ПК с установленным программным обеспечением не ниже:

- 1) Операционная система Windows XP;
- 2) Adobe Reader;
- 3) Текстовый редактор;
- 4) Браузер.

4. ЗАДАНИЕ НА ЛАБОРАТОРНУЮ РАБОТУ

Произвести физическое форматирование текстовых элементов.

5. ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ

- 1) Ознакомиться с содержанием лабораторной работы.
- 2) Проанализировав содержание текстовых блоков заглавной страницы, выделить элементы, нуждающиеся в визуальном акценте – названия пунктов меню, заголовки, подзаголовки и т.д.
- 3) Определить, каким образом будут расставляться акценты – изменением гарнитуры, начертания, размера, цвета шрифта и т.д.
- 4) Выбрать соответствующие теги физического форматирования и применить к элементам.

- 5) Сохранить результат.
- 6) Повторить действия для внутренней страницы.
- 7) Подготовиться к ответу на контрольные вопросы.

6. СОДЕРЖАНИЕ ОТЧЕТА

Отчетом по данной лабораторной работе служит сохраненный html-код заглавной и внутренней страниц с заверстанными изображениями и текстовыми блоками с выделенными смысловыми единицами.

7. КОНТРОЛЬНЫЕ ВОПРОСЫ

- 1) Как сделать шрифт полужирным?
- 2) Как сделать шрифт курсивным?
- 3) Какие атрибуты тега <marquee> вы знаете?
- 4) Можно ли применять тег <marquee> только к тексту, или к изображениям тоже?
- 5) Как создаются разноуровневые заголовки?
- 6) Какими тегами регулируется размер шрифта?

8. БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1) Пауэлл, Powell T. Web-дизайн : пер.с англ. / Т.Пауэлл .— 2-е изд. — СПб. : БХВ-Петербург, 2005 .— 1072с. : ил. — (В подлиннике) .— Парал.тит.л.англ. — ISBN 5-94157-389-8.

2) Вин, Д. Искусство web-дизайна : Самоучитель / Дж.Вин; Пер.с англ.О.Кузнецова .— СПб.и др. : Питер, 2002 .— 224с.

3) Джонсон, Johnson J. Web-дизайн:типичные ляпы и как их избежать : пер.с англ. / Д.Джонсон .— М. : КУДИЦ-ОБРАЗ, 2005 .— 400с.

4) Кирсанов, Д. Веб-дизайн:книга Дмитрия Кирсанова / Д.Кирсанов .— СПб. : Символ-Плюс, 2001 .— 376с.

Лабораторная работа №7 (4 часа)
ЛОГИЧЕСКОЕ ФОРМАТИРОВАНИЕ ТЕКСТА

1. ЦЕЛЬ И ЗАДАЧИ РАБОТЫ

Цель работы: усвоение основных принципов логического форматирования текста.

Задачи работы: создать логическую разметку текста – отформатировать списки, цитаты и т.д., в нужных местах вставить спецсимволы.

2. ОСНОВНЫЕ ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

Тег <blockquote> предназначен для выделения длинных цитат внутри документа. Текст, обозначенный этим тегом, традиционно отображается как выровненный блок с отступами слева и справа (примерно по 40 пикселей), а также с отбивкой сверху и снизу.

Синтаксис:

```
<blockquote>Текст</blockquote>
```

Пример:

```
<blockquote> <p>Lorem ipsum </p> </blockquote>
```

Тег <cite> помечает текст как цитату или сноску на другой материал. Такое выделение удобно для изменения стиля текста через CSS, а также применяется для разделения кода HTML на структурные элементы. Браузеры обычно устанавливают текст внутри контейнера <cite> курсивом.

Синтаксис:

```
<cite>Текст</cite>
```

Пример:

```
<p><cite>Айзек Азимов</cite></p>
```

Тег <dfn> . Как правило, в документе, когда упоминается новый термин, он выделяется курсивом и дается его определение. При использовании этого термина в дальнейшем, он считается уже известным читателю. Тег <dfn> применяется для выделения таких терминов при их первом появлении в тексте.

Браузеры отображают содержимое контейнера <dfn> с помощью курсивного начертания.

Синтаксис:

```
<dfn>Текст</dfn>
```

Пример:

```
<p><dfn>Капиталью</dfn> в типографике называется текст,  
набранный прописными буквами уменьшенного размера.</p>
```

**Тег ** предназначен для акцентирования текста. Браузеры отображают такой текст курсивным начертанием.

Синтаксис:

```
<em>Текст</em>
```

Пример:

```
<p><em>Lorem ipsum dolor sit amet</em></p>
```

**Тег ** предназначен для акцентирования текста. Браузеры отображают такой текст жирным начертанием.

Синтаксис:

```
<strong>Текст</strong>
```

Пример:

```
<p><strong>Lorem ipsum dolor sit amet</strong></p>
```

Для отображения символов, которых нет на клавиатуре применяются специальные знаки, начинающиеся с амперсанда (&) и заканчивающиеся точкой с запятой (;).

Пример:

```
<p> Не следует умножать сущности сверх необходимости &copy; </p>
```

Списком называется взаимосвязанный набор отдельных фраз или предложений, которые начинаются с маркера или цифры. Списки предоставляют возможность упорядочить и систематизировать разные данные и представить их в наглядном и удобном для пользователя виде.

Маркированный список определяется тем, что перед каждым элементом списка добавляется небольшой маркер, обычно в виде закрашенного кружка. Сам список формируется с помощью контейнера ``, а каждый пункт списка начинается с тега ``, как показано ниже. Отступы сверху, снизу и слева от списка добавляются автоматически.

Синтаксис:

```
<ul>
  <li>Первый пункт</li>
  <li>Второй пункт</li>
  <li>Третий пункт</li>
</ul>
```

Атрибуты :

`type="disc | circle | square"` - устанавливает вид маркера списка.

Атрибуты :

`type="disc | circle | square"` - устанавливает вид маркера списка.

Пример:

```
<ul type="square">
  <li>Чебурашка</li>
  <li>Крокодил Гена</li>
  <li>Шапокляк</li>
  <li>Крыса Лариса</li>
</ul>
```

Нумерованный список представляет собой набор элементов с их порядковыми номерами. Вид и тип нумерации зависит от атрибутов тега ``, который и применяется для создания списка. Каждый пункт нумерованного списка обозначается тегом ``, как показано ниже.

Синтаксис:

```
<ol>
  <li>Первый пункт</li>
  <li>Второй пункт</li>
  <li>Третий пункт</li>
</ol>
```

Атрибуты :

`type="A | a | I | i | 1"` - устанавливает вид маркера списка.

`reversed` - нумерация в списке становится по убыванию (3,2,1).

`start = "целое положительное число"` - задаёт число, с которого будет начинаться нумерованный список.

Атрибуты :

type="disc | circle | square" - устанавливает вид маркера списка.

value="целое положительное число" - число, с которого будет начинаться нумерованный список.

Пример:

```
<ol type="I" start="8">
  <li>Король Магнум XLIV</li>
  <li>Король Зигфрид XVI</li>
  <li>Король Сигизмунд XXI</li>
  <li>Король Хусбрандт I</li>
</ol>
```

Список определений состоит из двух элементов - термина и его определения. Сам список задается с помощью контейнера <dl>, термин - тегом <dt>, а его определение - с помощью тега <dd>.

Синтаксис:

```
<dl>
  <dt>Термин 1</dt>
  <dd>Определение 1</dd>
  <dt>Термин 2</dt>
  <dd>Определение 2</dd>
</dl>
```

Пример:

```
<dl>
  <dt>Тег</dt>
  <dd>Тег - это специальный символ разметки, который
применяется для вставки различных элементов на веб-страницу.
</dd>
  <dt>HTML-документ</dt>
  <dd>Обычный текстовый файл, который может содержать в себе
текст, теги и стили. </dd>
</dl>
```

3. ОБОРУДОВАНИЕ И МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ

Компьютерные классы, оборудованные ПК с установленным программным обеспечением не ниже:

- 1) Операционная система Windows XP;
- 2) Adobe Reader;
- 3) Текстовый редактор;
- 4) Браузер.

4. ЗАДАНИЕ НА ЛАБОРАТОРНУЮ РАБОТУ

Произвести логическое форматирование текстовых элементов.

5. ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ

- 1) Ознакомиться с содержанием лабораторной работы.
- 2) Проанализировав содержание текстовых блоков заглавной страницы, выделить элементы, нуждающиеся в логическом форматировании – цитаты, определения, акцентированный текст, списки и т.д.

- 3) Выбрать соответствующие теги логического форматирования и применить к элементам.
- 4) Сохранить результат.
- 5) Повторить действия для внутренней страницы.
- 6) Подготовиться к ответу на контрольные вопросы.

6. СОДЕРЖАНИЕ ОТЧЕТА

Отчетом по данной лабораторной работе служит сохраненный html-код заглавной и внутренней страниц с заверстанными изображениями и текстовыми блоками с логическим форматированием.

7. КОНТРОЛЬНЫЕ ВОПРОСЫ

- 1) Как выделить цитату?
- 2) Как создать маркированный список?
- 3) Как создать нумерованный список?
- 4) Как создать список определений?
- 5) Как вставить спецсимвол ©?

8. БИБЛИОГРАФИЧЕСКИЙ СПИСОК

- 1) Пауэлл, Powell T. Web-дизайн : пер.с англ. / Т.Пауэлл .— 2-е изд. — СПб. : БХВ-Петербург, 2005 .— 1072с. : ил. — (В подлиннике) .— Парал.тит.л.англ. — ISBN 5-94157-389-8.
- 2) Вин, Д. Искусство web-дизайна : Самоучитель / Дж.Вин; Пер.с англ.О.Кузнецова .— СПб.и др. : Питер, 2002 .— 224с.
- 3) Джонсон, Johnson J. Web-дизайн:типичные ляпы и как их избежать : пер.с англ. / Д.Джонсон .— М. : КУДИЦ-ОБРАЗ, 2005 .— 400с.
- 4) Кирсанов, Д. Веб-дизайн:книга Дмитрия Кирсанова / Д.Кирсанов .— СПб. : Символ-Плюс, 2001 .— 376с.

1. ЦЕЛЬ И ЗАДАЧИ РАБОТЫ

Цель работы: усвоение основных принципов создания ссылок.

Задачи работы: проанализировать страницы на предмет необходимости наличия ссылок, создать текстовые ссылки, создать изображения ссылок.

2. ОСНОВНЫЕ ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

Ссылки являются основой гипертекстовых документов и позволяют переходить с одной веб-страницы на другую. Особенность их состоит в том, что сама ссылка может вести не только на HTML-файлы, но и на файл любого типа, причем этот файл может размещаться совсем на другом сайте. Главное, чтобы к документу, на который делается ссылка, был доступ. Иными словами, если путь к файлу можно указать в адресной строке браузера, и файл при этом будет открыт, то на него можно сделать ссылку.

Для создания ссылки необходимо сообщить браузеру, что является ссылкой, а также указать адрес документа, на который следует сделать ссылку.

Тег <a> является одним из важных элементов HTML и предназначен для создания ссылок. В зависимости от присутствия атрибутов name или href тег <a> устанавливает ссылку или якорь. Для создания ссылки необходимо сообщить браузеру, что является ссылкой, а также указать адрес документа, на который следует сделать ссылку. В качестве значения атрибута href используется адрес документа (URL, Universal Resource Locator, универсальный указатель ресурсов), на который происходит переход.

Синтаксис:

`...`

`...`

Атрибуты:

`accesskey = "Цифра (0-9) | латинская буква (a-z)"` - позволяет активировать ссылку с помощью некоторого сочетания клавиш с заданной в коде ссылки буквой или цифрой. Браузеры при этом используют различные комбинации клавиш. Например, для `accesskey="s"` работают следующие сочетания: Internet Explorer: Alt + S; Chrome: Alt + S; Opera: Shift + Esc, S; Safari: Alt + S; Firefox: Shift + Alt + S.

`charset="utf-8 | windows-1251 | ..."` - указывает кодировку текста, на который ведет ссылка.

`coords="координаты"` - устанавливает координаты активной области. Набор координат определяется формой «горячей области», которая задается атрибутом `shape`. Отсчет координат обычно ведется от левого верхнего угла объекта или изображения и указывается в пикселах. Для прямоугольника (`shape="rect"`) определяется четыре координаты — X1, Y1, X2, Y2. Для окружности (`shape="circle"`) определяется три координаты — координаты центра окружности (X, Y) и ее радиус (R). Для полигона (многоугольника) (`shape="poly"`) последовательно указываются координаты каждой вершины (X1, Y1, X2, Y2, ...).

`href = "Http://..."` - задает адрес документа, на который следует перейти.

`hreflang=" en | rus | ..."` - идентифицирует язык текста по ссылке.

`media = "all | braille | handheld | print | screen | speech | projection | tty | tv "` - указывает тип носителя документа, на который ведёт ссылка.

`name="любой текст с учетом регистра"` - устанавливает имя якоря внутри документа.

`rel = " answer | chapter | co-worker | colleague | contact | details | edit | friend | question | ... "` - отношения между ссылаемым и текущим документами.

`rev = "любая текстовая строка"` - отношения между текущим и ссылаемым документами. Обратен атрибуту `rel`.

`shape = "circle | default | poly | rect "` - задает форму активной области ссылки для изображений.

`tabindex = "любое целое положительное число, начиная с нуля "` - определяет последовательность перехода между ссылками при нажатии на кнопку `<Tab>`. Значения выстраиваются последовательно и переход между ссылками происходит от меньшего значения к большему.

`target = " _blank | _self | _parent | _top "` - имя окна или фрейма, куда браузер будет загружать документ. (`_blank` — загружает страницу в новое окно браузера; `_self` — загружает страницу в текущее окно (это значение задается по умолчанию); `_parent` — загружает страницу во фрейм-родитель, если фреймов нет, то это значение работает как `_self`; `_top` — отменяет все фреймы и загружает страницу в полном окне браузера, если фреймов нет, то это значение работает как `_self`).

`title="текст"` - добавляет всплывающую подсказку к тексту ссылки.

`type="text/css"` - сообщает браузеру, какой синтаксис использовать, чтобы правильно интерпретировать стили. Указывает MIME-тип документа, на который ведёт ссылка.

Пример:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
<html>
<head>
<meta http-equiv="content-type" content="text/html;
charset=utf-8">
<title>Ссылки на странице</title>
</head>
<body>
<p><a href="dog.html">Собаки</a></p>
<p><a href="cat.html">Кошки</a></p>
</body>
</html>
```

Атрибут `href` определяет URL (Universal Resource Locator, универсальный указатель ресурса), иными словами, адрес документа, на который следует перейти, а содержимое контейнера `<a>` является ссылкой. Текст, расположенный между тегами `<a>` и ``, по умолчанию становится синего цвета и подчеркивается.

Если указана ссылка на файл, которого не существует, например, его имя в атрибуте href набрано с ошибкой, то такая ссылка называется битая. Битых ссылок следует категорически избегать, поскольку они вводят посетителей сайта в заблуждение.

По умолчанию, при переходе по ссылке документ открывается в текущем окне или фрейме. При необходимости, это условие может быть изменено атрибутом target тега <a>.

Изображение так же может быть ссылкой.

Пример:

```
<a href="images/xxx.jpg">Посмотрите на мою фотографию!</a>
```

Любая ссылка является встроенным элементом, поэтому для нее действуют те же правила, что и для встроенных элементов. А именно, нельзя размещать внутри тега <a> блочные элементы, но допустимо делать наоборот, и вкладывать ссылку в блочный контейнер.

3. ОБОРУДОВАНИЕ И МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ

Компьютерные классы, оборудованные ПК с установленным программным обеспечением не ниже:

- 1) Операционная система Windows XP;
- 2) Adobe Reader;
- 3) Текстовый редактор;
- 4) Браузер.

4. ЗАДАНИЕ НА ЛАБОРАТОРНУЮ РАБОТУ

Расставить на заглавной и внутренней страницах гиперссылки.

5. ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ

- 1) Ознакомиться с содержанием лабораторной работы.
- 2) Проанализировав содержание заглавной и внутренней страницы, выделить те элементы, которые должны быть ссылками. Не следует забывать, что ссылка – это не обязательно текст. Так, например, обычно логотип сайта является ссылкой на заглавную страницу.
- 3) Создать гиперссылки. В частности, гиперссылки с названий пунктов меню на соответствующие внутренние страницы и т.д.
- 4) Создать соответствующие гиперссылкам внутренние страницы, которые пока будут оставаться пустыми, и сохранить их в том же каталоге, в котором располагаются все остальные элементы.
- 4) Сохранить результат.
- 5) Повторить действия для создания ссылок с внутренней страницы.
- 6) Подготовиться к ответу на контрольные вопросы.

6. СОДЕРЖАНИЕ ОТЧЕТА

Отчетом по данной лабораторной работе служит сохраненный html-код заглавной и внутренней страниц с текстовыми и рисуночными гиперссылками, а так же предварительная структура сайта, сформированными из пустых внутренних страниц с соответствующими названиями.

7. КОНТРОЛЬНЫЕ ВОПРОСЫ

- 1) Как сделать ссылкой текст?
- 2) Как сделать ссылкой изображение?
- 3) Что определяет атрибут target?
- 4) Можно ли внутри тега <A> размещать блочные элементы?

8. БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1) Пауэлл, Powell T. Web-дизайн : пер.с англ. / Т.Пауэлл .— 2-е изд. — СПб. : БХВ-Петербург, 2005 .— 1072с. : ил. — (В подлиннике) .— Парал.тит.л.англ. — ISBN 5-94157-389-8.

2) Вин, Д. Искусство web-дизайна : Самоучитель / Дж.Вин; Пер.с англ.О.Кузнецова .— СПб.и др. : Питер, 2002 .— 224с.

3) Джонсон, Johnson J. Web-дизайн:типичные ляпы и как их избежать : пер.с англ. / Д.Джонсон .— М. : КУДИЦ-ОБРАЗ, 2005 .— 400с.

4) Кирсанов, Д. Веб-дизайн:книга Дмитрия Кирсанова / Д.Кирсанов .— СПб. : Символ-Плюс, 2001 .— 376с.

ИСПОЛЬЗОВАНИЕ КАСКАДНЫХ СПИСКОВ СТИЛЕЙ CSS. АДАПТАЦИЯ РАНЕЕ СВЕРСТАННЫХ МАКЕТОВ

1. ЦЕЛЬ И ЗАДАЧИ РАБОТЫ

Цель работы: усвоение основных принципов использования каскадных списков стилей.

Задачи работы: проанализировать страницы на предмет наличия элементов с одинаковым стилем. Создать эти стили в виде каскадного списка и заменить в html-коде повторяющиеся виды форматирования на форматирование с помощью описания стилей.

2. ОСНОВНЫЕ ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

Стилем или CSS (Cascading Style Sheets, каскадные таблицы стилей) называется набор параметров форматирования, который применяется к элементам документа, чтобы изменить их внешний вид. Возможность работы со стилями издавна включают в развитые издательские системы и текстовые редакторы, тем самым позволяя одним нажатием кнопки придать тексту заданный, заранее установленный вид. Теперь это доступно и создателям сайта, когда цвет, размеры текста и другие параметры хранятся в определенном месте и легко «прикручиваются» к любому тегу. Еще одним преимуществом стилей является то, что они предлагают намного больше возможностей для форматирования, чем обычный HTML.

Пример:

Использование страницы index.html с кодом

```
<!DOCTYPE html PUBLIC
"-//W3C//DTD XHTML 1.0 Transitional//EN"
http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html>
<head>
  <title>Пример </title>
  <meta http-equiv="Content-Type"
    content="text/html; charset=windows-1251" />
</head>
<body>
  <font color="#e14ccb"><b>Название раздела</b></font>
  <P> Текст</P>
</body>
</html>
```

Полностью аналогично использованию страниц index.html с кодом

```
<!DOCTYPE html PUBLIC
"-//W3C//DTD XHTML 1.0 Transitional//EN"
http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html>
<head>
  <title>Пример </title>
  <meta http-equiv="Content-Type"
    content="text/html; charset=windows-1251" />
  <link rel="stylesheet" href="style.css" type="text/css">
```

```
</head>
<body>
  <P class="razdel">Название раздела</P>
  <P> Текст</P>
</body>
</html>
```

И стилового файла style.css с кодом

```
P.razdel{color: #e14ccb;
  font-weight: bolder;
}
```

Так же можно описывать стилями параметры не для конкретного случая форматирования, а для всех элементов определенного типа. Так, например, для того, чтобы сделать начертание заголовков отличным от умолчального, в стилевом файле можно прописать:

```
h1 {font-family: 'Times New Roman', Times, serif;
  font-size: 250%;
}
```

3. ОБОРУДОВАНИЕ И МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ

Компьютерные классы, оборудованные ПК с установленным программным обеспечением не ниже:

- 1) Операционная система Windows XP;
- 2) Adobe Reader;
- 3) Текстовый редактор;
- 4) Браузер.

4. ЗАДАНИЕ НА ЛАБОРАТОРНУЮ РАБОТУ

Адаптировать ранее сверстанные макеты для использования каскадных списков стилей.

5. ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ

- 1) Ознакомиться с содержанием лабораторной работы.
- 2) Проанализировав содержание заглавной и внутренней страницы, выделить те элементы, которые имеют одинаковое форматирование – например пункты меню, фотографии в контенте, ссылки в тексте, заголовки и т.д.
- 3) Выписать параметры каждого из выделенных классов форматирования: цвет, размер и т.д.
- 4) Для дальнейшего удобства обозначить эти элементы в html-коде комментариями. Как любой язык, HTML позволяет вставлять в тело документа комментарии, которые сохраняются при передаче документа по сети, но не отображаются браузером. Синтаксис комментария:

```
<!-- Это комментарий -->
```
- 5) Сохранить результат.
- 6) Подготовиться к ответу на контрольные вопросы.

6. СОДЕРЖАНИЕ ОТЧЕТА

Отчетом по данной лабораторной работе служит сохраненный html-код заглавной и внутренней страниц с комментариями к тем элементам, которым в дальнейшем понадобится описание стилей, а так же список стилевых классов.

7. КОНТРОЛЬНЫЕ ВОПРОСЫ

- 1) Зачем нужны каскадные списки стилей?
- 2) Нужно ли для каждого элемента, форматирование которого отличается от умолчального создавать свой стиль?
- 3) Можно ли создавать стили не для отдельных классов для для какого-то конкретного тега в целом?
- 4) Как прописать в код сайта использование стилей?

8. БИБЛИОГРАФИЧЕСКИЙ СПИСОК

- 1) Пауэлл, Powell T. Web-дизайн : пер.с англ. / Т.Пауэлл .— 2-е изд. — СПб. : БХВ-Петербург, 2005 .— 1072с. : ил. — (В подлиннике) .— Парал.тит.л.англ. — ISBN 5-94157-389-8.
- 2) Вин, Д. Искусство web-дизайна : Самоучитель / Дж.Вин; Пер.с англ.О.Кузнецова .— СПб.и др. : Питер, 2002 .— 224с.
- 3) Джонсон, Johnson J. Web-дизайн:типичные ляпы и как их избежать : пер.с англ. / Д.Джонсон .— М. : КУДИЦ-ОБРАЗ, 2005 .— 400с.
- 4) Кирсанов, Д. Веб-дизайн:книга Дмитрия Кирсанова / Д.Кирсанов .— СПб. : Символ-Плюс, 2001 .— 376с.

ЗАВЕРСТКА ФОНОВЫХ ИЗОБРАЖЕНИЙ И СОЗДАНИЕ СТИЛЕЙ ТЕКСТА С ИСПОЛЬЗОВАНИЕМ КАСКАДНЫХ СПИСКОВ СТИЛЕЙ

1. ЦЕЛЬ И ЗАДАЧИ РАБОТЫ

Цель работы: усвоение основных принципов создания каскадных списков стилей и адаптации ранее созданных макетов.

Задачи работы: создать каскадные списки стилей и прописать стили ранее созданным текстовым, изобразительным и разметочным элементам.

2. ОСНОВНЫЕ ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

Далее приведен список стилей по категориям.

Размеры, позиции и отступы.

`margin: [значение | проценты | auto] {1,4} | inherit`

Устанавливает величину отступа от каждого края элемента. Разрешается использовать одно, два, три или четыре значения, разделяя их между собой пробелом.

`margin-bottom: значение | auto | inherit`

Устанавливает величину отступа от нижнего края элемента.

`margin-left: значение | auto | inherit`

Устанавливает величину отступа от левого края элемента.

`margin-right: значение | auto | inherit`

Устанавливает величину отступа от правого края элемента.

`margin-top: значение | auto | inherit`

Устанавливает величину отступа от верхнего края элемента.

`bottom: значение | проценты | auto | inherit`

Устанавливает положение нижнего края содержимого элемента без учета толщины рамок и отступов.

`left: значение | проценты | auto | inherit`

Устанавливает положение левого края содержимого элемента.

`position: absolute | fixed | relative | static | inherit`

Устанавливает способ позиционирования элемента относительно окна браузера или других объектов на веб-странице.

`right: значение | проценты | auto | inherit`

Устанавливает положение правого края содержимого элемента.

`top: значение | проценты | auto | inherit`

Устанавливает положение верхнего края содержимого элемента.

`padding: [значение | проценты] {1, 4} | inherit`

Устанавливает значение полей вокруг содержимого элемента. Разрешается использовать одно, два, три или четыре значения, разделяя их между собой пробелом.

`padding-bottom: значение | inherit`

Устанавливает значение поля от нижнего края содержимого элемента.

`padding-left: значение | inherit`

Устанавливает значение поля от левого края содержимого элемента.

`padding-right: значение | inherit`

Устанавливает значение поля от правого края содержимого элемента.

`padding-top: значение | inherit`

`height: значение | проценты | auto | inherit`

Устанавливает высоту блочных или заменяемых элементов.

width: значение | проценты | auto | inherit

Устанавливает ширину блочных или заменяемых элементов.

vertical-align: baseline|bottom|middle|sub|super|text-bottom|text-top|top|inherit | значение | проценты

Выравнивает элемент по вертикали относительно своего родителя, окружающего текста или ячейки таблицы.

Таблицы.

border-collapse: collapse | separate | inherit

Устанавливает, как отображать границы вокруг ячеек таблицы.

border-spacing: значение1 [значение2]

Задаёт расстояние между границами ячеек в таблице.

Текст и шрифт.

letter-spacing: значение | normal | inherit Определяет интервал между символами в пределах элемента.

line-height: множитель | значение | проценты | normal | inherit

Устанавливает интерлиньяж (межстрочный интервал) текста, отсчет ведется от базовой линии шрифта.

text-align: center | justify | left | right | inherit

Определяет горизонтальное выравнивание текста в пределах элемента.

text-decoration: [blink | line-through | overline | underline]
| none | inherit

Добавляет оформление текста в виде его подчеркивания, перечеркивания, линии над текстом и мигания.

text-indent: <значение> | <проценты> | inherit

Устанавливает величину отступа первой строки блока текста.

text-transform: capitalize | lowercase | uppercase | none | inherit

Управляет преобразованием текста элемента в заглавные или прописные символы.

white-space: normal | nowrap | pre | pre-line | pre-wrap | inherit

Устанавливает, как отображать пробелы между словами.

word-spacing: значение | normal | inherit

Устанавливает интервал между словами.

font-family: имя шрифта [, имя шрифта[, ...]] | inherit

Устанавливает семейство шрифта, которое будет использоваться для оформления текста содержимого. Serif — шрифты с засечками (антиквенные), типа Times; sans-serif — рубленые шрифты (шрифты без засечек или гротески), типичный представитель — Arial; cursive — курсивные шрифты; fantasy — декоративные шрифты; monospace — моноширинные шрифты, ширина каждого символа в таком семействе одинакова (шрифт Courier).

font-size: абсолютный размер | относительный размер | значение | проценты | inherit

Определяет размер шрифта элемента.

font-style: normal | italic | oblique | inherit

Определяет начертание шрифта — обычное, курсивное или наклонное.

font-variant: normal | small-caps | inherit

Определяет, как нужно представлять строчные буквы — оставить их без модификаций или делать их все прописными уменьшенного размера.

font-weight:

bold|bolder|lighter|normal|100|200|300|400|500|600|700|800|900

Устанавливает насыщенность шрифта.

list-style-image: none | url('путь к файлу') | inherit

Устанавливает адрес изображения, которое служит в качестве маркера списка.

list-style-position: inside | outside

Определяет, как будет размещаться маркер относительно текста.

list-style-type: circle | disc | square | armenian | decimal | decimal-leading-zero | georgian | lower-alpha | lower-greek | lower-latin | lower-roman | upper-alpha | upper-latin | upper-roman | none | inherit

Изменяет вид маркера для каждого элемента списка.

Цвет и фон.

background-attachment: fixed | scroll | inherit

Устанавливает, будет ли прокручиваться фоновое изображение вместе с содержимым элемента.

background-clip: [padding-box | border-box | content-box] [, [padding-box | border-box | content-box]]

Определяет, как цвет фона или фоновая картинка должна выводиться под границами.

background-color: цвет | transparent | inherit

Определяет цвет фона элемента.

background-image: url(путь к файлу) | none | inherit

Устанавливает фоновое изображение для элемента.

background-origin: [padding-box | border-box | content-box] [, [padding-box | border-box | content-box]]

Определяет область позиционирования фонового рисунка.

background-position: [left | center | right | <проценты> | <значение>] || [top | center | bottom | <проценты> | <значение>] | inherit

Задаёт начальное положение фонового изображения, установленного с помощью свойства background-image.

background-position-x: left | center | right | <проценты> | <значение>

Задаёт положение фонового изображения внутри элемента по горизонтали.

background-position-y: top | center | bottom | проценты | значение

Задаёт положение фонового изображения внутри элемента по вертикали.

background-repeat: no-repeat | repeat | repeat-x | repeat-y | inherit

Определяет, как будет повторяться фоновое изображение, установленное с помощью свойства background-image.

background-size: [<значение> | <проценты> | auto]{1,2} | cover | contain

Масштабирует фоновое изображение согласно заданным размерам.

color: цвет | inherit

Определяет цвет текста элемента.

После того, как каскадные списки созданы, можно приступить к созданию и заполнению остальных внутренних страниц.

3. ОБОРУДОВАНИЕ И МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ

Компьютерные классы, оборудованные ПК с установленным программным обеспечением не ниже:

- 1) Операционная система Windows XP;
- 2) Adobe Reader;
- 3) Текстовый редактор;
- 4) Браузер.

4. ЗАДАНИЕ НА ЛАБОРАТОРНУЮ РАБОТУ

Создать каскадный список стилей и адаптировать под него ранее сверстанный макет.

5. ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ

- 1) Ознакомиться с содержанием лабораторной работы.
- 2) Создать каскадный список стилей для ранее определенных элементов.
- 3) Добиться идентичности воспроизведения элементов.
- 4) Заменить html-форматирование описанием соответствующих классов.
- 5) Сохранить результат.
- 6) Подготовиться к ответу на контрольные вопросы.

6. СОДЕРЖАНИЕ ОТЧЕТА

Отчетом по данной лабораторной работе служит сохраненный html-код заглавной и внутренней страниц с каскадными списками стилей и файл стилей.

7. КОНТРОЛЬНЫЕ ВОПРОСЫ

- 1) Какие из стилевых свойств можно применить к изображениям?
- 2) Какие из стилевых свойств можно применить к тексту?
- 3) Какие из стилевых свойств можно применить к заголовкам?
- 4) Какие из стилевых свойств можно применить к таблицам?
- 5) Какие из стилевых свойств можно применить к документу в целом?

8. БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1) Пауэлл, Powell T. Web-дизайн : пер.с англ. / Т.Пауэлл .— 2-е изд. — СПб. : БХВ-Петербург, 2005 .— 1072с. : ил. — (В подлиннике) .— Парал.тит.л.англ. — ISBN 5-94157-389-8.

2) Вин, Д. Искусство web-дизайна : Самоучитель / Дж.Вин; Пер.с англ.О.Кузнецова .— СПб.и др. : Питер, 2002 .— 224с.

3) Джонсон, Johnson J. Web-дизайн:типичные ляпы и как их избежать : пер.с англ. / Д.Джонсон .— М. : КУДИЦ-ОБРАЗ, 2005 .— 400с.

4) Кирсанов, Д. Веб-дизайн:книга Дмитрия Кирсанова / Д.Кирсанов .— СПб. : Символ-Плюс, 2001 .— 376с.

Лабораторная работа №11 (4 часа)
ПРОВЕРКА НА КРОССБРАУЗЕРНОСТЬ

1. ЦЕЛЬ И ЗАДАЧИ РАБОТЫ

Цель работы: усвоение основных принципов проверки на кроссбраузерность.

Задачи работы: оценить корректность отображения сайта в браузерах Mozilla Firefox, Opera и Internet Explorer. Исправить недочеты.

2. ОСНОВНЫЕ ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

Кроссбраузерность - свойство сайта отображаться и работать во всех популярных браузерах идентично. Под идентичностью понимается отсутствие развалов верстки и способность отображать материал с одинаковой степенью читабельности. Понятие «кроссбраузерность» очень часто путают с попиксельным соответствием, что на самом деле является разными понятиями.

Кроссбраузерность остается одной из самых сложных проблем в веб-разработке. Следование веб-стандартам само по себе дает достаточно высокий уровень совместимости, но не все браузеры совершенны. Перед разработчиком стоит задача, обеспечения максимальной совместимости сайта с наиболее популярными браузерами, чтобы каждый посетитель мог с удобством его использовать.

Веб-дизайнер не может знать, каким браузером пользуется очередной посетитель сайта, а различия в поведении и причудах браузеров все еще велики, чтобы можно было игнорировать их. К примеру, различные браузеры и операционные системы используют различные способы вывода шрифтов (Win vs. Mac on handling fonts). Размеры шрифтов тоже не одинаковы в различных системах, а некоторые шрифты могут просто отсутствовать на компьютере пользователя.

Практически невозможно постоянно помнить обо всех возможных проблемах каждого конкретного браузера, детальное тестирование в различных браузерах и операционных системах - это единственный способ обеспечить нормальное функционирование сайта.

Для тестирования на кроссбраузерность кроме того, что можно просто установить несколько браузеров, существуют веб-сервисы, которые либо предоставляют удаленный доступ к браузеру через VPN либо позволяют просмотреть скриншоты отображения сайта в различных браузерах (семейство Mozilla, Internet Explorer, Opera, Safari, мобильные браузеры), в различных операционных системах (Mac OS, Linux, Win). Эффект от внесенных в разметку сайта изменений можно увидеть сразу же несмотря на то, что на компьютере дизайнера нет множества браузеров. К таким сервисам относятся <http://browsershots.org>, <http://netrenderer.com>, <http://broserlab.adobe.com>, <http://www.browsrcamp.com>.

3. ОБОРУДОВАНИЕ И МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ

Компьютерные классы, оборудованные ПК с установленным программным обеспечением не ниже:

- 1) Операционная система Windows XP;
- 2) Adobe Reader;
- 3) Текстовый редактор;
- 4) Браузеры Mozilla Firefox, Opera и Internet Explorer.

4. ЗАДАНИЕ НА ЛАБОРАТОРНУЮ РАБОТУ

Проверить сайт на кроссбраузерность и исправить недочеты.

5. ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ

- 1) Ознакомиться с содержанием лабораторной работы.
- 2) Поочередно открыть сайт в браузерах Mozilla Firefox, Opera и Internet Explorer.
- 3) Зафиксировать недочеты отображения сайта в каждом из браузеров.
- 4) Проверить синтаксис html-верстки и описания каскадных списков стилей.
- 5) Проверить – можно ли реализовать заверстку и описание стилей тех элементов, которые отображаются некорректно другим способом.
- 6) Исправить недочеты.
- 7) Сохранить результат.
- 8) Подготовиться к ответу на контрольные вопросы.

6. СОДЕРЖАНИЕ ОТЧЕТА

Отчетом по данной лабораторной работе служит сохраненный html-код заглавной и внутренней страниц с каскадными списками стилей и файл стилей, корректно отображающиеся минимум в трех браузерах.

7. КОНТРОЛЬНЫЕ ВОПРОСЫ

- 1) Для чего нужна проверка на кроссбраузерность?
- 2) В каком браузере ваш сайт отображался некорректно? Почему?
- 3) В каких случаях сайты отображаются некорректно в Mozilla Firefox?
- 4) В каких случаях сайты отображаются некорректно в Opera?
- 5) В каких случаях сайты отображаются некорректно в Internet Explorer?

8. БИБЛИОГРАФИЧЕСКИЙ СПИСОК

- 1) Пауэлл, Powell T. Web-дизайн : пер.с англ. / Т.Пауэлл .— 2-е изд. — СПб. : БХВ-Петербург, 2005 .— 1072с. : ил. — (В подлиннике) .— Парал.тит.л.англ. — ISBN 5-94157-389-8.
- 2) Вин, Д. Искусство web-дизайна : Самоучитель / Дж.Вин; Пер.с англ.О.Кузнецова .— СПб.и др. : Питер, 2002 .— 224с.
- 3) Джонсон, Johnson J. Web-дизайн:типичные ляпы и как их избежать : пер.с англ. / Д.Джонсон .— М. : КУДИЦ-ОБРАЗ, 2005 .— 400с.
- 4) Кирсанов, Д. Веб-дизайн:книга Дмитрия Кирсанова / Д.Кирсанов .— СПб. : Символ-Плюс, 2001 .— 376с.

Лабораторная работа №12 (4 часа)
РАЗМЕЩЕНИЕ САЙТА НА БЕПЛАТНОМ ХОСТИНГЕ

1. ЦЕЛЬ И ЗАДАЧИ РАБОТЫ

Цель работы: получение навыков размещения сайтов в Интернет.

Задачи работы: оценить корректность отображения сайта в браузерах Mozilla Firefox, Opera и Internet Explorer. Исправить недочеты.

2. ОСНОВНЫЕ ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

Хостинг (англ. hosting) - услуга по предоставлению вычислительных мощностей для физического размещения информации на сервере, постоянно находящемся в сети (обычно Интернет). Хостингом также называется услуга по размещению оборудования клиента на территории провайдера с обеспечением подключения его к каналам связи с высокой пропускной способностью.

Обычно под понятием услуги хостинга подразумевают как минимум услугу размещения файлов сайта на сервере, на котором запущено ПО, необходимое для обработки запросов к этим файлам (веб-сервер). Как правило, в услугу хостинга уже входит предоставление места для почтовой корреспонденции, баз данных, DNS, файлового хранилища на специально выделенном файл-сервере и т. п., а также поддержка функционирования соответствующих сервисов.

Основные бесплатные хостинги – это Narod.ru и Ucoz.ru.

Далее приведены выдержки из официального руководства пользователя хостинга **Narod.ru**.

Яндекс.Народ—это сервис для создания и бесплатного хостинга сайтов, хранения и обмена файлами. Для того чтобы создать сайт или загрузить файлы на Народе, требуется доступ к персональной части Яндекса. Если вы уже зарегистрированы на Яндексе, введите свои логин и пароль, перейдя по ссылке «Войти», расположенной в правой верхней части страницы.

Загрузка файлов на сайт

Загрузка файлов осуществляется в **Мастерской**, в разделе «**Загрузка файлов на сайт**». Нажмите на кнопку «Browse» или «Обзор» и выберите на вашем жестком диске файл для загрузки в текущую папку, он должен появиться в поле рядом с кнопкой. Выберите до десяти таких файлов (размер каждого файла не должен превышать 5 Мб) и нажмите на кнопку «Загрузить файлы». Вмена загружаемых файлов не должны содержать недопустимые символы, например, символы «пробел», русские буквы и т.д. Имя файла должно состоять из латинских букв, цифр, знаков дефис и подчеркивание, может содержать точку.

Вы также можете загружать файлы на свой сайт при помощи FTP. Чтобы воспользоваться протоколом FTP, понадобится специальная программа, так называемый FTP-клиент. FTP-клиентом может служить интернет-браузер (MS Internet Explorer, Netscape Navigator), программа управления файлами FAR, специализированная программа CuteFTP и другие. FTP-клиент встроен также в некоторые HTML-редакторы, такие как HomeSite, Dreamweaver.

Сразу после загрузки файлов вы продолжаете видеть старую версию вашей мастерской. Обновить изображение можно при помощи кнопок Refresh (Reload) в меню вашего браузера, или клавиш Ctrl+R на клавиатуре. Эта команда перезагружает странички непосредственно с сервера, а не из кэша вашего компьютера.

Файлы, размещаемые на Народе, должны быть в кодировке Windows-1251. Кроме того, не надо использовать meta-тег charset.

Управление файлами на сайте

Все управление файлами осуществляется с помощью файл-менеджера в разделе **«Управление файлами» Мастерской**. Здесь вы сможете загружать готовые файлы или удалять их, создавать html-файлы самостоятельно, редактировать или переименовывать уже созданные, создавать и редактировать страницы по шаблону и т. д.

Открыв файл-менеджер, вы увидите список ваших файлов, для каждого из которых будет указано имя, описание, размер и дата последнего изменения. По умолчанию файлы разделены по типам: вначале выводятся страницы, созданные по шаблонам, затем нешаблонные html-страницы, далее — файлы-картинки и файлы иных видов. Кроме того, в самом верху, над списком файлов, выводится список подпапок текущей папки. Каждый тип объекта для удобства обозначен своим значком.

Все операции в файл-менеджере можно условно разделить на три группы: индивидуальные действия с объектами (файлами и папками), групповые операции и создание новых объектов. Индивидуальные действия (задание имени и описания объекта, управление счетчиками, редактирование html-кода) реализуются с помощью кнопки «Свойства», которая присутствует в каждой строчке файл-менеджера. При ее нажатии открывается всплывающее окно, напоминающее окно свойств в Windows. Отдельным свойствам файлов соответствуют закладки, отображаемые в левой части этого окна. При выборе закладки в правой части окна появляется форма для управления соответствующим свойством файла. Для сохранения измененных свойств служит кнопка «Сохранить» при ее нажатии страница с файл-менеджером обновляется.

Групповые действия — копирование, перемещение и удаление — выполняются с помощью одноименных кнопок, которые размещаются в верхней и нижней частях файл-менеджера; при этом для осуществления операций предварительно требуется выделить нужные файлы или папки, отметив соответствующие флажки. Создание новых папок, страниц по шаблону и html-файлов реализуется с помощью ссылок, которые расположены сразу над шапкой файл-менеджера. Там же находится ссылка «Настройки», которая открывает раздел, где можно изменить внешний вид и правила отображения файлов в файл-менеджере.

Работа с папками

Для перехода в одну из папок нажмите ссылку с ее именем. На экран будет выведено содержимое выбранной папки. Чтобы вернуться на один уровень вверх, нажмите значок с изображением открытой папки. При нахождении в одной из внутренних папок есть возможность перейти в любую из папок вышележащих уровней: в файл-менеджере всегда указывается полный путь к выбранной папке, причем каждый элемент этого пути является ссылкой, нажав которую, вы перейдете в соответствующую папку.

Чтобы создать новую папку, выберите ссылку «Создать папку», после чего укажите имя папки. Для переименования существующей папки воспользуйтесь кнопкой «Свойства», затем выберите закладку «Имя и описание», задайте новое имя (здесь же можно задать и словесное описание папки — для удобства) и нажмите кнопку «Сохранить».

Для удаления одной или нескольких папок отметьте нужные папки флажками и нажмите желтую кнопку «Удалить» (расположена на серой полосе над и под списком

файлов). Аналогичным образом можно скопировать или переместить нужные папки в другую папку, при этом папки копируются и перемещаются со всем их содержимым.

Как и в традиционных файл-менеджерах, которые служат для управления файлами на вашем компьютере, файл-менеджер Narod.ru позволяет удалять файлы, копировать, перемещать, переименовывать и редактировать. Удаление, копирование и перемещение реализованы как групповые операции. Это означает, что вначале следует выбрать один или несколько файлов, а затем нажать желтую кнопку с названием нужной операции. Переименование и редактирование файлов — индивидуальные операции, для их выполнения следует нажать кнопку «Свойства» напротив нужного файла и выбрать закладку, соответствующую требуемому действию. Закладка «Имя и описание» позволяет указать новое имя файла и задать ему словесное описание.

Редактирование файлов

Отредактировать html-код файла можно с помощью закладки «Текстовый редактор». Для того чтобы отредактировать страницу, созданную по шаблону, воспользуйтесь кнопкой «Редактировать».

Для того чтобы посмотреть любой из файлов, представленных в файл-менеджере, просто нажмите на ссылку с его именем — файл при этом откроется в новом окне.

Вы можете изменить html-код своих страниц прямо на сайте, воспользовавшись онлайн-текстовым редактором. Чтобы вызвать его, нажмите кнопку «Свойства» напротив нужного файла и затем выберите закладку «Текстовый редактор». Чтобы вызвать редактор для страницы, созданной не по шаблону, или текстового файла, достаточно нажать кнопку «Редактировать», которая находится слева от имени файла.

Редактор представляет собой текстовое поле, в котором отображается html-код или текст страницы и содержимое которого можно изменять (в этом и состоит редактирование), и под этим полем — несколько кнопок с действиями. Для того чтобы было удобнее работать с длинными строками, окно с текстовым редактором, как и любое другое окно в Windows, можно раскрыть на весь экран. После того как вы внесли изменения в текст, нажмите кнопку «Сохранить», и отредактированная страница запишется на ваш сайт.

По ходу внесения изменений удобно проверять, какой вид приобретает страница, — для этого служит кнопка «Пред. просмотр» страница при этом открывается в новом окне.

Если вы хотите отменить только что внесенные правки, пока еще не была нажата кнопка «Сохранить», нажмите кнопку «Отменить изменения», и тогда текст в поле станет таким, каким он был до начала редактирования.

3. ОБОРУДОВАНИЕ И МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ

Компьютерные классы, оборудованные ПК с установленным программным обеспечением не ниже:

- 1) Операционная система Windows XP;
- 2) Adobe Reader;
- 3) Текстовый редактор;
- 4) Браузер.

4. ЗАДАНИЕ НА ЛАБОРАТОРНУЮ РАБОТУ

Разместить сайт на бесплатном хостинге и открыть доступ пользователям.

5. ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ

- 1) Ознакомиться с содержанием лабораторной работы.
- 2) Открыть страницу <http://narod.yandex.ru/> и зарегистрироваться.
- 3) Загрузить все файлы проектируемого сайта на хостинг.
- 4) Проверить корректность работы сайта.
- 7) Сохранить результат.
- 8) Подготовиться к ответу на контрольные вопросы.

6. СОДЕРЖАНИЕ ОТЧЕТА

Отчетом по данной лабораторной работе является размещенный в открытом доступе сайт.

7. КОНТРОЛЬНЫЕ ВОПРОСЫ

- 1) Что такое хостинг?
- 2) Какие бесплатные хостинги вы знаете?
- 3) Какими способами можно загружать файлы на сервер?
- 4) Можно ли корректировать файлы на сервере?
- 5) Имеет ли смысл сначала корректировать html-код на локальном компьютере, а лишь потом загружать исправленные файлы на сервер, или это слишком сложное решение?

8. БИБЛИОГРАФИЧЕСКИЙ СПИСОК

- 1) Пауэлл, Powell T. Web-дизайн : пер.с англ. / Т.Пауэлл .— 2-е изд. — СПб. : БХВ-Петербург, 2005 .— 1072с. : ил. — (В подлиннике) .— Парал.тит.л.англ. — ISBN 5-94157-389-8.
- 2) Вин, Д. Искусство web-дизайна : Самоучитель / Дж.Вин; Пер.с англ.О.Кузнецова .— СПб.и др. : Питер, 2002 .— 224с.
- 3) Джонсон, Johnson J. Web-дизайн:типичные ляпы и как их избежать : пер.с англ. / Д.Джонсон .— М. : КУДИЦ-ОБРАЗ, 2005 .— 400с.
- 4) Кирсанов, Д. Веб-дизайн:книга Дмитрия Кирсанова / Д.Кирсанов .— СПб. : Символ-Плюс, 2001 .— 376с.

Лабораторная работа №13 (4 часа)

ТЕСТИРОВАНИЕ WEB-САЙТА

1. ЦЕЛЬ И ЗАДАЧИ РАБОТЫ

Цель работы: получение навыков тестирования сайта.

Задачи работы: оценить корректность отображения сайта при разных размерах окна браузера, проверить корректность гиперссылок, проверить содержание страниц. Исправить недочеты.

2. ОСНОВНЫЕ ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

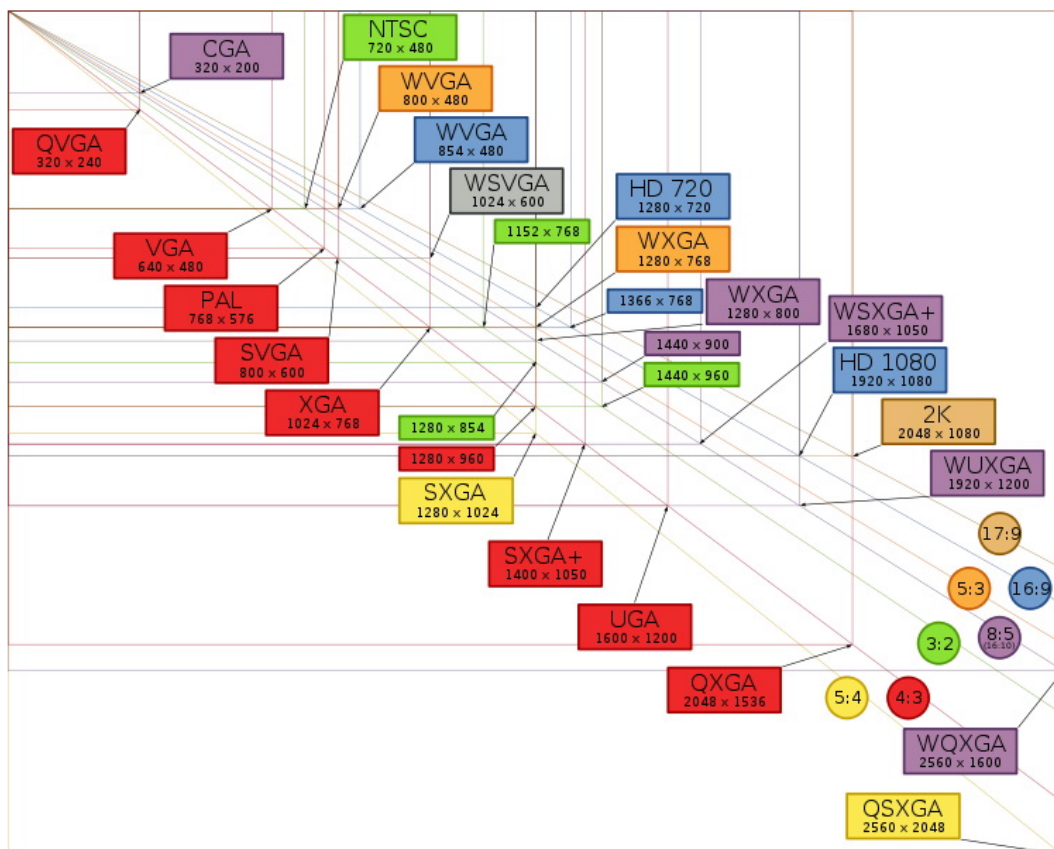
Всегда следует учитывать, что для просмотра сайта может использоваться широкий диапазон устройств – мониторы различного разрешения, планшетные компьютеры, телефоны и т.д.

Учёт разрешения монитора, на котором посетители будут просматривать сайт, - важный составной момент как разработки сайта, так и редизайна сайта. В техническом задании на разработку сайта или редизайн сайта желательно оговорить отдельным пунктом основной диапазон разрешения мониторов, на которых должен просматриваться сайт.

Увеличение среднего разрешения экрана монитора у посетителей сайта - одна из основных причин необходимости периодического редизайна сайта. Ежегодно среднее разрешение экрана монитора увеличивается примерно на 25-30 % (1/2 дюйма в год). Кроме того, в мировом Интернете диапазон разрешения мониторов несколько выше, по сравнению с российским Интернет-сегментом. Основные причины этого в том, что стоимость мониторов с высоким разрешением и портативных устройств достаточно высока. Средний уровень жизни населения страны во многом определяет основные, наиболее популярные приобретаемые мониторы. Чем больше пользователей Интернет, тем больше диапазон разрешения мониторов. Российский Интернет по числу пользователей составляет приблизительно 3% от мирового. Далее приведены наиболее распространенные параметры видимости на различных устройствах вывода.

Таблица 5. Минимальная видимая область при различных разрешениях мониторов

Минимальная видимая область (пикс.)				
Browser	640 x 480	800 x 600	1024 x 768	1280 x 1024
Windows				
Internet Explorer 6	620 x 303	780 x 423	1004 x 591	1260 x 847
Internet Explorer 5+	620 x 303	780 x 423	1004 x 591	1260 x 847
Netscape 6	618 x 301	778 x 421	1002 x 589	1258 x 845
Netscape 4.7	620 x 286	780 x 406	1004 x 574	1260 x 830
Macintosh				
Internet Explorer 5	591 x 309	751 x 429	975 x 597	1231 x 853
Internet Explorer 4.5	592 x 316	752 x 436	976 x 604	1232 x 860
Netscape 6	607 x 322	767 x 442	991 x 610	1247 x 866
Netscape 4.7	613 x 307	773 x 427	997 x 595	1253 x 851



Естественно если компьютер веб-дизайнер не поддерживает большое разрешение экрана, то посмотреть как выглядит какой-либо сайт при разрешении 1920 на 1200 у него не получится. При этом может помочь сервис <http://viewlike.us/>.

Также дизайнер не должен забывать о том, что многие посетители сайта имеют не слишком высокую скорость соединения с Интернет. Даже наличие "выделенной" линии не всегда гарантирует высокую скорость соединения с Интернет. Игнорирование этого факта при строительстве сайта приводит к тому, что не все из потенциальных клиентов могут просмотреть сайт.

Существует ряд специальных сервисов проверки битых ссылок, которые позволяют найти все ссылки на сайте, которые ведут на несуществующие страницы или файлы. Не требуется установка какой-либо программы проверки ссылок или регистрации - сканирование сайта и просмотр результатов производится в режиме онлайн. Сервис полностью обойдет весь сайт по внутренним ссылкам и соберет подробную информацию о его структуре и ошибках, а также позволит просмотреть полученные результаты в формате Microsoft Excel, а также сгенерировать файл карты сайта sitemap.xml в формате Sitemaps для поисковых систем Google и Яндекс.

К таким сервисам относятся <http://www.keeplinks.ru>, linkchecker и др.

3. ОБОРУДОВАНИЕ И МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ

Компьютерные классы, оборудованные ПК с установленным программным обеспечением не ниже:

- 1) Операционная система Windows XP;
- 2) Adobe Reader;
- 3) Текстовый редактор;
- 4) Браузер.

4. ЗАДАНИЕ НА ЛАБОРАТОРНУЮ РАБОТУ

Протестировать сайт в различных разрешениях. Проверить корректность ссылок. Исправить недочеты.

5. ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ

- 1) Ознакомиться с содержанием лабораторной работы.
- 2) Открыть проектируемый сайт в окне браузера.
- 3) Изменяя размер окна браузера, проверить корректность отображения сайта – отсутствие горизонтальных полос прокрутки, налезания элементов, искажения дизайна.
- 4) Исправить недочеты.
- 5) Проверить все ссылки на работоспособность. В случае возникновения ошибок создать недостающие страницы, либо исправить ссылки.
- 6) Сохранить результат.
- 7) Подготовиться к ответу на контрольные вопросы.

6. СОДЕРЖАНИЕ ОТЧЕТА

Отчетом по данной лабораторной работе размещенный в открытом доступе кроссплатформенный сайт с полностью корректными гиперссылками.

7. КОНТРОЛЬНЫЕ ВОПРОСЫ

- 1) Зачем нужна проверка на кроссплатформенность?
- 2) Как проверить корректность отображения сайта для различных разрешений?
- 3) Какой сайт будет искажаться с большей вероятностью – с резиновым дизайном или с фиксированным?
- 4) Как проверить правильность гиперссылок?
- 5) В каких случаях гиперссылками могут стать нерабочими?

8. БИБЛИОГРАФИЧЕСКИЙ СПИСОК

Основная литература

1. Пантюхина Е.В., Котляров В.С., Пантюхин О.В. Перспективные технологии изготовления пищевой упаковки: учебник. Тула: Изд-во ТулГУ, 2018. 212 с.
2. Серова В.Н. Материаловедение в полиграфическом и упаковочном производствах: учебное пособие / Серова В.Н. — Казань : Казанский национальный исследовательский технологический университет, 2017. — 332 с.
3. Мочалова Е.Н. Материаловедение и основы полиграфического и упаковочного производств : учебное пособие / Мочалова Е.Н., Мусина Л.Р.. — Казань : Казанский национальный исследовательский технологический университет, 2017. — 148 с.

Дополнительная литература

1. Ильина О.В. Дизайн-конструирование тары и упаковки : учебное пособие / Ильина О.В.. — Санкт-Петербург : Санкт-Петербургский государственный университет промышленных технологий и дизайна, 2017. — 48 с.
2. Ильина О.В. Конструирование и дизайн упаковки : учебное пособие / Ильина О.В. — Санкт-Петербург: Санкт-Петербургский государственный университет промышленных технологий и дизайна, 2018. — 98 с.
3. Тара и упаковка [электронный ресурс]: журнал. — ISSN 0868-5568. Режим доступа: https://elibrary.ru/title_about.asp?id=9731 - eLibrary.ru, по паролю.

Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины (модуля)

1. <https://tsutula.bibliotech.ru/> - Электронный читальный зал “БИБЛИОТЕХ”: учебники авторов ТулГУ по всем дисциплинам. Режим доступа: по паролю.- Загл. с экрана
2. <https://www1.fips.ru> – Федеральный институт промышленной собственности [Электронный ресурс]. Режим доступа: свободный.- Загл. с экрана
3. <http://window.edu.ru> - Единое окно доступа к образовательным ресурсам: портал [Электронный ресурс]. Режим доступа: свободный.- Загл. с экрана
4. <http://cyberleninka.ru/> - НЭБ КиберЛенинка научная электронная библиотека открытого доступа. Режим доступа: свободный.- Загл. с экрана.