МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение
высшего образования
«Севастопольский государственный университет»

Создание статического сайта на HTML

Методические указания
к выполнению лабораторных работ
по дисциплине «Технологии Web - приложений»
для студентов очной формы обучения
по направлению 09.03.01 «Информатика и вычислительная техника»
Часть 1

Севастополь

2017

УДК 621.396.6

МЕТОДИЧЕСКИЕ УКАЗАНИЯ к выполнению лабораторных работ по дисциплине «Технологии Web - приложений» для студентов очной формы обучения по направлению09.03.01 «Информатика и вычислительная техника» / Сост. доцент кафедры ИТКС Лелеков С.Г., доцент кафедры ИТКС Шаталова Ю.Г. – Севастополь: Изд-во СГУ, 2017. – 41с.

Целью методических указаний является оказание помощи студентам в выполнении лабораторных работ по дисциплине «Технологии Web - приложений».

Приведены варианты заданий, рекомендации по выполнению, требования к оформлению работ.

Методические указания рассмотрены и утверждены на заседании кафедры информационных технологий и компьютерных систем, протокол
№ от 2017г.

Допущено учебно-методическим центром СГУ в качестве методических указаний.

Рецензенты: В. С. Чернега, к.т.н., доцент кафедры информационных систем;
Д.В. Моисеев, к.т.н., доцент кафедры информационных технологий и компьютерных систем

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

Выполнение лабораторных работ предусмотрено рабочим планом подготовки бакалавров по направлению «Информатика и вычислительная техника» для студентов очной формы обучения.

Выполнение цикла лабораторных работ (лабораторные работы 1-4) позволит студентам познакомиться с методами и средствами разработки статических Web- сайтов, языком разметки HTML, стилевому оформлению страниц с помощью CSS, табличной и блочной верстки, приобрести практические навыки разработки и оформления статического Web- сайта.

Методические указания по выполнению лабораторных работы содержат краткие теоретические сведения, пример выполнения заданий лабораторной работы, индивидуальное задание, требования к оформлению отчета. Все варианты индивидуальных заданий имеют примерно одинаковую трудоемкость. Способ выбора варианта определен в заданиях.

Перед выполнением лабораторной работы необходимо внимательно прочитать текст методических указаний, ознакомиться с рекомендуемой литературой. Предлагаемый пример выполнения задания желательно повторить, чтобы уже на его основе выполнять индивидуальное задание. При этом допускается использование других инструментальных средств и способов выполнения заданий, не рассмотренных в данных методических указаниях.

По результатам выполнения работы оформляется печатный отчет, который обязательно должен содержать:

– титульный лист;

– описание выполнения индивидуального задания;

Требования к описанию выполнения индивидуального задания в каждой работе свои и изложены в методических указаниях к ее выполнению.

1. ОБЩИЕ ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

WorldWideWeb (WWW)можно определить как интерактивную мультимедийную гипертекстовую среду, использующую язык разметки и поддерживающую множество протоколов Интернет. Обсудим основные моменты в этом определении.

Мультимедийная среда – возможность размещать в Сети не только и не столько текстовую информацию, но и графику, звук, видеоклипы.

Гипертекстовая среда – возможность использования в тексте ссылок на другие порции информации (текст, графику, звуковые и видеоклипы и т.д.), дополнительные источники. Многообразие таких ссылок и порождает Всемирную паутину WWW. В Web эти ссылки называют «Унифицированным указателем ресурсов URL». Каждый документ или файл в Интернет имеет собственный URL, что позволяет легко ссылаться на него из других документов.

Язык разметки HyperTextMarkupLanguage (HTML) является основой WWW. С помощью конструкций языка создаются Web страницы (сайты), которые можно просматривать с помощью специальных программ - обозревателей, или как их еще называют - браузеров. Наиболее известными браузерами являются Microsoft Internet Explorer, Mozilla Firefox, Opera, GoogleChrome.

Протоколы Internet. WWW использует собственный протокол связи между отдельными узлами: HTTP (HyperTextTransferProtocol). Любой протокол - это набор правил, которые используются компьютерами для обмена информацией. Кроме HTTP можно использовать и другие протоколы, получая доступ и другим приложениям Internet: UseNet - для доступа к группам новостей, FTP - для загрузки файлов; POP3, SMTP - электронной почте, Skype для телефонного общения и др.

WWW, в отличие от телевидения, интерактивная среда. Пользователи сами определяют, какой из узлов посещать, как долго изучать полученную информацию. Более того, на Web страницах часто можно ввести диалог с их авторами.

Web использует технологию "клиент - сервер". Это означает, что где – то в сети существует компьютер, на котором работает программное обеспечение Web - сервера, управляющее доступом к информации. Различные пользователи со всего мира являются клиентами сервера и получают от него информацию с помощью своих браузеров.

2. МЕТОДИЧЕСКИЕ УКАЗАНИЯ К ВЫПОЛНЕНИЮ ЛАБОРАТОРНЫХ РАБОТ

2.1. Лабораторная работа №1. Использование HTML для разработки простого статического Web– сайта. Часть 1.

Цель работы

Исследовать методы создания статического Web- сайта. Изучить структуру Html – документа, основные теги Html. Приобрести практические навыки создания статических Web - страниц. Создать страницу с информацией о себе.

Краткие теоретические сведения

Основным понятием языка HTML является понятие тег - инструкция обозревателю, как отображать текст. Независимо от того, как выглядит ваша домашняя страница, и какую информацию вы хотите отобразить, существует три тега, которые в соответствии со стандартами HTML и WWW должны присутствовать в каждой Web-странице:

– <html> Сообщает браузеру, что документ написан на языке HTML.
– <head> Отмечает служебную информацию HTML-документа.
– <body> Отмечает основной текст.

Эти теги необходимы Web-браузеру для определения различных частей HTML-документа, но они не оказывают прямого влияния на внешний вид Web-страницы.

Теги заголовка. В заголовке чаще всего размещают название страницы (тег <title>), служебную информацию для браузера: теги <meta>, <link>, <style>, <script>.

Название домашней страницы выводится браузером в строке заголовка окна. Название вводится с помощью тегов <title> и </title> размещаемых в заголовочной части между тегами <head> и </head>. Каждый документ может иметь только одно название.

Теги <body> и </body>, как и <head>, предназначены для обозначения отдельной части HTML-документа.

Теги<address>и</address> охватывают информацию о том, к кому нужно обращаться в отношении данной страницы в том случае, если у кого-нибудь возникнут вопросы или замечания. Они позволяют отделить эту информацию от основного блока.

Параметры тега <BODY>

В таблице 1.1. представлены атрибуты тега <BODY> предназначенные для изменения фона html-страницы.

Описанные в таблице параметры не являются обязательными, однако использование BGCOLOR рекомендуется по следующей причине: пользователь в настройках своего браузера может поставить любой цвет фона, а разработчик, полагая, что белый цвет является основным по умолчанию, может не указать этот параметр. В результате вместо подразумевающегося белого цвета, фон может оказаться черным, зеленым и т. д., что способно привести к нарушению оформления документа.

Таблица 1.1.Параметры фона

Атрибут	Функция	Примеры
BGCOLOR	Определение цвета фона	Задание белого фона <BODYBGCOLOR="white"> <BODY BGCOLOR="#FFFFFF"> <BODY BGCOLOR="255,255,255">
BACKGROUND	Указание фонового рисунка	<BODY BACKGROUND= "images/bg.gif">
BGPROPERTIES	Изменение свойств фона	BGPROPERTIES= "fixed" Прокручивая содержание документа, фоновое изображение остается в зафиксированном виде
TEXT	Определение цвета текста	TEXT = "green"

Также наряду с графическим изображением фона рекомендуется использовать и параметры цвета на тот случай, если рисунок не загрузится (тогда браузер отобразит цвет).

В таблице 1.2. представлены атрибуты тега <BODY> предназначенные для изменения положения информации (контента) на странице.

Таблица 1.2. Параметры границ документа

Атрибут	Функция	Примеры
TOPMARGIN	Определяет отступ от верхнего края документа	<BODY TOPMARGIN="5" BOTTOMMARGIN="5" LEFTMARGIN="10" RIGHTMARGIN="10" MARGINWIDTH="10" MARGINHEIGHT="5">
BOTTOMMARGIN	Определяет отступ от нижнего края документа	
LEFTMARGIN	Определяет отступ от левого края документа	
RIGHTMARGIN	Определяет отступ от правого края документа	
MARGINWIDTH	Определяет горизонтальный отступ	
MARGINHEIGHT	Определяет вертикальный отступ	

Примечание: Все значения параметров задаются в пикселах.

Заголовки

Существует шесть размеров шрифта заголовков (они пронумерованы от одного до шести, причем первый соответствует самому крупному шрифту).

Для ввода заголовка самым крупным первым номером, требуется поместить текст заголовка между тегами <h1> и </h1>. Если нужен заголовок размером поменьше, то необходимо воспользоваться тегами <h2> и </h2> и т.д.

Ввод текста и иной информации

Ввод текста - это, возможно, самый простой этап в процессе создания домашней страницы, поскольку текст можно набрать непосредственно в HTML - файле.

Однако, так как браузеры реагируют только на разметку HTML, то весь Ваш набранный текст будет отображаться в виде одного большого не отформатированного абзаца.

Для форматирования текста можно использовать следующие теги:

Тег нового абзаца - предписывает браузеру разделить два фрагмента текста пустой строкой. Для этого нужно установить курсор в то место, где Вы хотите вставить тег, и набрать . В конце абзаца поставьте тег .

Тег может содержать параметр ALIGN, отвечающий за тип горизонтального выравнивания текста в окне браузера.

ALIGN="left" — текст выровнен по левому краю (значение параметра, принятое по умолчанию);

ALIGN="center" — текст располагается посередине окна браузера;

ALIGN="right" — выравнивание текста по правому краю. Идеально подходит для создания эпиграфов, подписей, заголовков и пр.;

ALIGN="justify" — выравнивание по ширине окна браузера.

Например, Текст абзаца

Тег перевода строки
 - аналогичен тегу , только он не вводит пустой строки. После этого тега текст продолжает отображаться с начала следующей строки.

Тег горизонтальной линии <hr> помогает разделить страницу на смысловые части.

Различные виды линий можно получить, используя атрибуты тега <hr>:

width - задает ширину линии, например, width =”50%” определяет линию шириной в половину экрана;

align - задает выравнивание линии. Возможны следующие варианты: align=”left”, align=”center”, align=”right”;

size- задает толщину линии. Обычная ширина составляет 2 пикселя (size=”2”), но можно указать любую ширину от 1 до 175 пикселей.

Чтобы указать атрибут, его необходимо включить в тег <hr>, например, <hrsize=”5”width=”75%”align=”left”>.

Если в тексте требуется разместить несколько подряд идущих пробелов, то лучше использовать специальную комбинацию символов ** **столько раз подряд, сколько требуется пробелов.

Стилевое оформление текста

Управление стилем текста осуществляется парными тегами. Форматируемый текст заключается между тегом начала и тегом конца.

Теги служат для выравнивания текста по центру экрана.

Теги и позволяют отображать текст документа полужирным шрифтом.

Теги и позволяют отображать текст документа курсивом.

Теги <u> и </u> позволяют подчеркивать текст.

Теги <s> и </s> позволяют перечеркивать текст.

Также полезными являются теги для изменения размера текста в html-документе. В таблице 1.3.приведены их описание.

Таблица 1.3.Теги изменения размеров текста.

Тег	**Функция**
<BIG></BIG>	Отображение текста шрифтом большего, чем непомеченная часть текста размера
<SMALL></SMALL>	Отображение текста шрифтом меньшего размера
****	Сдвиг текста ниже уровня строки и вывод его (если возможно) шрифтом меньшего размера. Удобно использовать для математических индексов
****	Сдвиг текста выше уровня строки и вывод его (если возможно) шрифтом меньшего размера. Удобно использовать для задания степеней чисел
<FONT></FONT>	один из основных тегов форматирования текста, отображающий свойства шрифтов. Для него могут использоваться следующие параметры, приведенные в таблице 1.4.

Таблица 1.4.Атрибуты тега <FONT>

Атрибут	**Функция**
FACE	Служит для указания типа шрифта Например, **<FONTFACE="Verdana", "Arial", "Helvetica">**
SIZE	Служит для указания размеров шрифта в условных единицах от 1 до 7. Конкретный размер шрифта зависит от браузера. По умолчанию – норма равна 3. Размершрифта указывается абсолютной величиной, например SIZE=2, так иотносительной - SIZE=+1
COLOR	Служит для указания цвета шрифта. Значения могут быть заданы словами, числом в шестнадцатеричной системе, тройкой чисел RGB. Например, **<FONT COLOR="#FF0000"> Текст будет написан красным цветом </FONT>**

Для стилевого оформления документа в ряде тегов можно также использовать атрибут STYLE, например,

Текст

Более подробно использование элементов CSS будет рассмотрено в ЛР№3.

Описание лабораторной установки

Для выполнения лабораторной работы необходим персональный компьютер, подключенный к Интернет.

Необходимые программные средства для выполнения работы:

1. Любой Web-обозреватель (InternetExplorer, MozillaFirefox, Opera и др.).
2. Любой текстовый редактор (WindowsNotepad, MicrosoftWord идр.).

Одним из наиболее удобных для работы можно считать **Notepad++** (http://notepad-plus.sourceforge.net/ru/site.html).

Ход выполнения работы

Создадим простую статическую web – страницу, содержащую информацию, представляющую вас в Сети. Для этого выполним следующие шаги разработки.

Шаг 1.Откройте блокнот и наберите <html> в самом начале файла. Затем наберите его напарника – закрывающий тег </html>. Теперь весь текст, который появится между этими тегами, будет отмечен как текст формата HTML. Прямая косая черта используется для обозначения *закрывающих* тегов. Большинство тегов HTML парные, открывающий и закрывающий, они охватывают помечаемый ими текст.

Шаг 2. Наберите <head> на экране между тегами <html>, затем в следующей строке наберите его пару – тег </head>. Между ними можно будет разместить теги заголовка.

Шаг 3.Введите в вашу Web-страницу теги <body> и </body>. Текст, охваченный тегами <body>, является основной частью документа.

Документ примет следующий вид:

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

Шаг 4. Между тегами <body> и </body> наберите ваше имя и адрес электронной почты, например:

Петр Иванов-e-mail: Petr@Ivanov.com

Теперь введите тег <address> на строчке выше имени и адреса.

Введите закрывающий тег </address> после имени и адреса (вся контактная информация может занимать несколько строк).

Ваш HTML-файл теперь выглядит следующим образом:

```
<html>
        <head>
        </head>
        <body>
        <address>
              Петр_Иванов_e-mail:Petr@Ivanov.com
        </address>
        </body>
</html>
```

Шаг 5. Сохраните этот файл с расширением имени .html (Меню Файл/Сохранить, Тип файла выбрать – «Все файлы», Имя файла – first.html). Откройте его в Web-браузере.

Шаг 6. Добавьте в текст страницы тег <title>, например, так:

<title> Домашняя страница Петра Иванова </title>

Сохраните файл с расширением .html и откройте его в браузере.

Шаг 7. Добавьте в Вашу страницу между тегами <body></body> текст заголовка

Добро пожаловать на сайт Петра Иванова!

Обозначьте этот текст тегами <h1> и </h1>.

Введите подзаголовок

Приглашаются все желающие!

Обозначьте этот текст тегами <h2> и </h2>.

Сохраните файл и откройте или обновите Вашу страницу в браузере.

Шаг 8. Наберите, например, следующий текст: «Я рад приветствовать Вас на моей Web - странице. Я живу в славном городе Севастополе. Учусь в университете. С помощью Интернет хочу приобрести новых друзей.»

Сохраните файл и откройте Вашу страницу в браузере.

Шаг 9. Отформатируйте свой текст, используя теги <p>,
, <hr>. При этом горизонтальной линией отчеркните адрес электронной почты. Сохраните файл и откройте Вашу страницу в браузере.

Шаг 10. Отформатируйте свой текст, используя теги управления стилем. Сохраните файл и откройте Вашу страницу в браузере. Результат покажите преподавателю.

Задание на самостоятельную работу

1. Изучить материал данных методических указаний.

2. Реализовать фрагмент текста, соответствующий Вашему варианту, в виде html-документа без использования CSS.

Номер варианта определяется как (№ mod 12)+1, где № - номер по журналу академической группы. Варианты задания приведены в приложении А.

3. По результатам выполнения работы подготовить отчет.

Содержание отчета

1. Титульный лист.
2. Постановка задачи.
3. Описание реализации документа с использованием тегов HTML.

4.Описание тегов.
5. Скриншот окна браузера.
6. Выводы.
7. Список используемой литературы.

Контрольные вопросы

1. Что такое WorldWideWeb?
2. С помощью каких программ можно просматривать Web-сайты?
3. Какие Вы знаете программы-обозреватели?
4. Какие теги служат для создания html - документа?
5. Какие теги служат для размещения служебной информации?
6. Для чего используют тег <body>?
7. Какие атрибуты имеются у тега <body>?
8. Какие теги используют при вводе текста?
9. Как изменить стилевое оформление документа?
10. Поясните значение термина «Гипертекстовая среда»?

2.2. Лабораторная работа №2. Использование HTML для разработки простого статического Web– сайта. Часть2.

Цель работы

Приобрести практические навыки размещения на простом статическом сайте списков, таблиц и изображений.

Краткие теоретические сведения

Списки.

Список задает некоторое перечисление. Каждый элемент перечисления располагается с новой строки, нумеруется, или отмечается каким-нибудь символом (маркером). В первом случае список называется нумерованным, во втором - маркированным.

Чтобы задать маркированный список в HTML используется парный тег . Позиции списка отмечаются непарным тегом <li>. Допускается использование заголовка списка, который заключается в парный тег <lh></lh>. Например, Вы хотите список своих увлечений оформить в виде маркированного списка. Это можно сделать, например, так:

```
<ul><lh> Мои увлечения  </lh>
<li> Общение с друзьями
<li> Музыка
<li> Книги
</ul>
```

Чтобы задать нумерованный список в HTML используется парный тег <ol> . Позиции списка отмечаются также непарным тегом <li>.

Для нумерованного списка можно выбрать тип нумерации позиций. По умолчанию это арабские цифры, допускаются также буквы английского алфавита, римские цифры. Для указания типа используется атрибут type.

Например, тег <oltype=”a”> определяет нумерацию в виде букв английского алфавита, а <oltype=”1”> определяет нумерацию в виде арабских цифр.

Ниже представлен пример вложенных списков:

```
<OL>
<LI> Нумерованный
<UL>
<LI> Нумерованный
<LI> Ненумерованный
<LI> Список определений
</UL>
<LI> Ненумерованный
<UL TYPE=square>
<LI> Нумерованный
<LI> Ненумерованный
<LI> Список определений
</UL>
<LI> Вложенный
</OL>
```

Таблицы

Принцип создания таблиц в HTML таков: создаётся таблица, потом создаётся строка, потом все столбцы данной строки(т.е ячейки), потом очередная строка, снова все очередные столбцы данной строки и так далее.

Таблица создаётся с помощью тега <table>, а заканчивается тегом </table>.

Строка создаётся с помощью тега <tr>. а заканчивается тегом</tr>.

Внутри тега <tr> создают ячейки, которые создаются с помощью тегов <td> и </td>.

Внутри этих тегов находятся те элементы, которые должны быть расположены внутри данной ячейки.

Атрибуты тега <table>.

1) Атрибут "border", значение которого задаёт толщину рамки таблицы в пикселях. По умолчанию, рамки вообще нет.

2) Атрибуты "width" и "height" задают ширину и высоту таблицы соответственно. Размер может быть указан, как в абсолютных единицах (пиксели, px), так и в относительных (проценты, %). Относительный размер хорош тем, что он всегда подстроится под любое разрешение монитора пользователя и любой браузер. А абсолютные тем хороши, что при любых браузерах и любых разрешениях монитора не будет сюрпризов с дизайном, связанные, например, с растягиванием элементов (если монитор широкоэкранный, к примеру).

3)Атрибут "bgcolor" задает цвет фона таблицы. Можно использовать, например, следующие значения: aqua, black, blue, fuchsia, gray, grey, green, lime, maroon, navy, olive, purple, red, silver, teal, white, yellow.

4) Атрибут "style"позволяет задать форматирование таблицы в стиле CSS (см. ЛР№3).

Атрибуты тега <tr>.

Атрибут "height". Заметьте, что у тега <tr> нет атрибута "width", впрочем, это логично, ведь тег <tr> отвечает за строку, а, следовательно, за высоту. А за ширину отвечают столбцы.

Атрибуты тега <td>.

1) Атрибут "width". Объяснение то же, что и для атрибута тега <tr>. Соответственно, атрибута "height" нет.

2) Атрибут "colspan". Значение этого атрибута означает количество столбцов, которое занимает данная ячейка.

3) Атрибут "rowspan". Значение этого атрибута означает количество строк, которое занимает данная ячейка.

4) Атрибут "align". Значение этого атрибута означает выравнивание элемента внутри ячейки по горизонтали. Бывают три значения: "left" (по левому краю), "center" (по центру), "right" (по правому краю). По умолчанию стоит выравнивание по левому краю.

5) Атрибут "valign". Значение этого атрибута означает выравнивание элемента внутри ячейки по вертикали. Снова имеются только три значения: "top" (по верху), "middle" (по середине), "bottom" (по низу). По умолчанию стоит значение "middle".

6) Атрибут "cellpadding". Данный атрибут определяет ширину пустого пространства между содержимым ячейки и ее границами, то есть задает поля внутри ячейки.

7) Атрибут "cellspacing". Определяет ширину промежутков между ячейками в пикселях. Если этот атрибут не указан, по умолчанию задается величина, равная двум пикселям.

Размещение изображений на Web - странице

В качестве рисунка может быть использован любой файл с расширением *.gif или *.jpg или *.png. Для размещения изображений на странице используется тег <imgsrc="Имя файла">. Например, <img SRC=”Tigers.gif” alt=”Здесь должен быть тигр”>. Так можно указывать только те файлы, которые расположены в той же папке, что и исходная страница.

Для выравнивания изображения относительно текста используется атрибут align.

Для масштабирования изображения можно использовать атрибуты width и height. Например, <imgsrc=”Tigers.gif” align=right width=300 height=200>. Здесь значения высоты и ширины изображения приводится в пикселях(точках). **Чтобы изображение сохранило свои пропорции при масштабировании надо использовать только один из этих атрибутов.**

Атрибут "alt" указывает текст, который описывает картинку. Этот же текст будет показываться в случае, если картинка по каким-либо причинам будет не найдена, либо у пользователя в браузере отключён показ картинок. Очень желательно этот атрибут ставить, так как это помогает оптимизации сайта.

Другие атрибуты тега:

1) Атрибут "ismap".Этот атрибут сообщает браузеру, что данное изображение позволяет пользователю выполнять какие-либо действия, щелкая мышью на определенном месте изображения.

2) Атрибут "border". Данный атрибут позволяет автору определить ширину рамки вокруг рисунка.

3) Атрибут "vspace". Позволяет установить размер в пикселах пустого пространства над и под рисунком, чтобы текст не наезжал на рисунок. Особенно это важно для динамически формируемых изображений, когда нельзя заранее увидеть документ.

4) Атрибут "hspace". То же самое, что и VSPACE, но только по горизонтали.

Ссылки на Web - страницах

Гиперссылки на Web - страницы - одно из основных свойств WWW.

Для создания гипертекстовой ссылки используется специальный парный тег

. Рассмотрим подробнее структуру гиперссылки на примере:

НовостиСГУ.

Здесь https://www.sevsu.ru/novosti- представляет URL-адрес файла в Internet.

«Новости СГУ» - текст, который видит пользователь на экране обозревателя. Этот текст обычно выделен синим цветом, и при наведении на него мыши появляется характерный указатель в виде указательного пальца.Очень внутри тега **<a>**располагают рисунок, тогда эта ссылка будет в виде картинки.

Если документ, на который ссылается данная ссылка, находится в той же самой директории, что и исходный документ, то в качестве URL-адреса можно просто указать имя файла, например,

Следующаястраница.

Разумеется, ссылаться можно не только на HTML-страницы, но и на любые файлы, будь то картинки, будь то фильмы, будь то музыка, будь то архивы, будь то ещё всё, что угодно.

Атрибут "title" задаёт текст, который будет виден при наведении мышки на ссылку.

Ссылки на разделы внутри документа

Допускается делать ссылки на различные участки или разделы одного и того же документа, используя специальный скрытый маркер(*якорь*) для этих разделов. Это позволяет быстро переходить от раздела к разделу внутри документа. Как только вы щелкнете на ссылке, браузер переместит вас на указанный раздел документа, а строка, в которой стоит якорь данного раздела (обычно, первая строка раздела или заголовок

раздела) будет размещена на первой строке окна браузера (если данная строка не присутствует уже на экране браузера).

Для якоря, как и для ссылки, используется тег <a>, но без атрибута href, а с атрибутом name:

** Текст **

Для создания ссылки на этот якорь из этой же страницы в атрибуте HREF ссылки укажем имя якоря после символа #:

Текст

Якорь может быть поставлен как в том же документе, который просматривается в текущий момент, так и в другом документе. Во втором случае в ссылке на этот якорь необходимо перед "#" указать полный URL документа. Тогда браузер осуществит загрузку указанного документа и перейдет к указанному якорем разделу. Например:

<A HREF="www.meta.ua/webprogramming/second.html#anchor">

Описание лабораторной установки

Для выполнения лабораторной работы необходим персональный компьютер, подключенный к Интернет.

Необходимые программные средства для выполнения работы:

1. Любой Web-обозреватель (InternetExplorer, MozillaFirefox, Opera и др.).
2. Любой текстовый редактор (WindowsNotepad, MicrosoftWord идр.).

Одним из наиболее удобных для работы можно считать **Notepad++** (http://notepad-plus.sourceforge.net/ru/site.html).

Ход выполнения работы

Продолжим создание личной страницы, работу над которой Вы начали на первом занятии. Необходимо выполнить следующие шаги.

Шаг 1. Добавьте на свою страницу список. Сохраните файл и откройте Вашу страницу в браузере.

Шаг 2. Добавьте на страницу таблицу, содержащую типовой распорядок Вашего выходного дня. Например:

№	Мероприятие	Время	
		Начало	Конец
1	Подъем	9-00	10-00
2	Утренняя гимнастика и водные процедуры	10-00	10-30
3	Завтрак	10-30	11-00
	. . .	. . .	. . .

Реализация имеет вид:

```
<h1><center>Мойраспорядокдня</h1></center>
<table width="75%" border="1" align=”center”bgcolor=”pink” style=”color:blue;”>
<tr>
  <td width="11%" align="center" rowspan="2">№ </td>
<td width="55%" align="center" rowspan="2">Мероприятие</td>
<td width="17%" align="center" colspan="2">Время</td>
</tr>
<tr>
<td width="17%" align="center">Начало</td>
<td width="17%" align="center">Конец</td>
```

```
</tr>
<tr>
<td width="11%" align="center">1</td>
<td width="55%">Подъем</td>
<td width="17%" align="center">9-00</td>
<td width="17%" align="center">10-00</td>
</tr>
<tr>
<td width="11%" align="center">2</td>
<tdwidth="55%">Утренняя гимнастика и водные процедуры</td>
<td width="17%" align="center">10-00</td>
<td width="17%" align="center">10-30</td>
</tr>
<tr>
<td width="11%" align="center">3</td>
<td width="55%">Завтрак</td>
<td width="17%" align="center">10-30</td>
<td width="17%" align="center">11-00</td>
</tr>
</table>
```

Продолжите создание таблицы самостоятельно.

Шаг 3. Добавьте на страницу какие-либо изображения, фотографии. Сохраните файл и откройте Вашу страницу в браузере.

Задание на самостоятельную работу

1. Изучить материалы методических указаний.
2. Реализовать группу Web-страниц, имеющую структуру, представленную на рисунке 2.1.

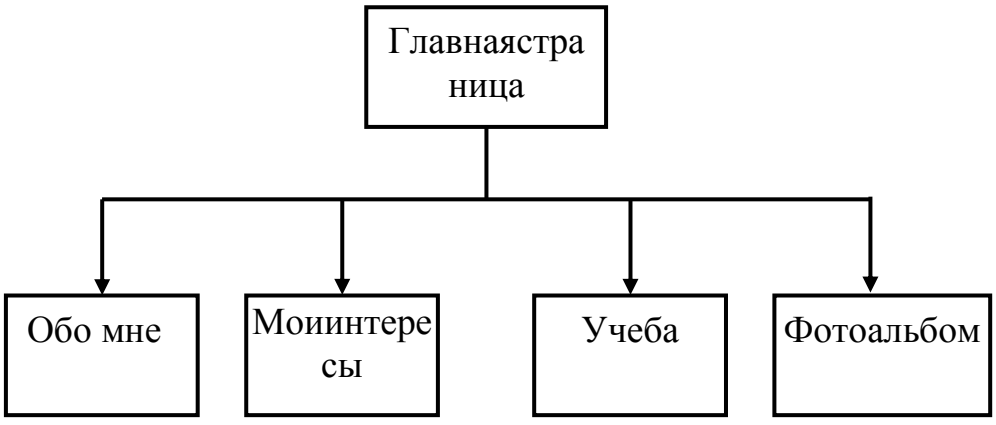


Рисунок 2.1. Необходимые Web-страницы

Рекомендуется следующее содержимое Web-страниц:

- **Главная страница**:
  - Меню, содержащее гиперссылки на все остальные страницы;
  - Фамилия Имя Отчество;
  - Фотография;

- Группа;

Необходимым является использование следующих элементов HTML: гиперссылки, таблицы, элементы форматирования текста, графические элементы.

- **Обо мне**:
  - Автобиография в произвольной форме (возможно использование и графического материала);
  - гиперссылка на главную страницу.

Необходимым является использование следующих элементов HTML: элементы форматирования текста, гиперссылки, стили.

- **Мои интересы**:
  - содержание страницы, выполненное в виде списка гиперссылок на якоря внутри этой страницы (например: мое хобби, мои любимые книги, моя любимая музыка, мои любимые фильмы и т.п.);
  - непосредственно содержимое каждого из разделов;
  - гиперссылка на главную страницу.

Необходимым является использование следующих элементов HTML: заголовки различных уровней, нумерованные и маркированные списки, таблицы, элементы форматирования текста, якоря и гиперссылки на якоря, стили.

- **Учеба**:
  - Полное название университета;
  - Полное название кафедры;
  - Перечень изучаемых дисциплин с 1 по 4 семестр, выполненный в виде таблицы в соответствии с вариантом задания (Приложение Б). Номер варианта выбирается как *(№ mod 2)+1, где № - номер по журналу академической группы*.
  - Успеваемость
  - гиперссылка на главную страницу.

Необходимым является использование следующих элементов HTML: таблицы, элементы форматирования текста, гиперссылки, стили.

- **Фотоальбом**:
  - 15 фотографий представленных в виде таблицы, по **N**[*)] фото в строке; каждое фото должно иметь всплывающую подсказку с его названием, а также подпись с названием под фото.
  - гиперссылка на главную страницу.

[*)]*N определяется как (№ mod 4)+3, где № - номер по журналу академической группы*

Необходимым является использование следующих элементов HTML: таблицы, графические элементы, элементы форматирования текста, гиперссылки.

Примечания: Заголовок каждой HTML страницы должен содержать название данной страницы (например: «Иванов Иван. Главная страница.»). Все страницы должны иметь одинаковый цвет фона (или фоновое изображение) и быть выполненными в одном стиле.

ПРИ ВЫПОЛНЕНИИ РАБОТЫ ЗАПРЕЩАЕТСЯ ПОЛЬЗОВАТЬСЯ ГРАФИЧЕСКИМИ HTML-РЕДАКТОРАМИ ИЛИ КОНВЕРТЕРАМИ В ФОРМАТ HTML!

3. По результатам работы подготовьте отчет.

Содержание отчета

1. Титульный лист.
2. Постановка задачи.
3. HTML – тексты, разработанных страниц.
4.Описание тегов и их атрибутов.
5. Скриншоты страниц.
6. Выводы.
7. Список используемой литературы.

Контрольные вопросы

1. Какие теги служат для создания списков?
2. Какие теги служат для создания таблиц?
3. Как разместить в документе изображение?
4. Что такое гиперссылки, как их создать?
5. Как создать ссылку на раздел внутри документа?
6. Какие атрибуты для тега <tr> Вы знаете?
7. Назовите атрибуты тега <table>.

2.3. Лабораторная работа №3. Использование CSS при разработке Web -сайта. Основы блочной верстки

Цель работы

Изучить методы CSS, получить практические навыки использования блочной верстки для создания web- страницы с динамическим меню навигации при помощи списков и правил CSS.

Краткие теоретические сведения

CSS (CascadingStyleSheets, каскадные таблицы стилей) – технология описания внешнего вида документа, написанного языком разметки. Преимущественно используется как средство оформления веб-страниц в формате HTML и XHTML, но может применяться с любыми видами документов в формате XML, включая SVG и XUL.

CSS используется создателями веб-страниц для задания цветов, шрифтов, расположения и других аспектов представления документа. До появления CSS оформление веб-страниц осуществлялось непосредственно внутри содержимого документа (см. лабораторную работу №1). Однако, с появлением CSS стало возможным принципиальное разделение содержания (написанного на HTML или другом языке разметки) и представления (написанного на CSS) документа. Это разделение может увеличить доступность документа, предоставить большую гибкость и возможность управления его представлением, а также уменьшить сложность и повторяемость в структурном содержимом. Кроме того, CSS позволяет представлять один и тот же документ в различных стилях или методах вывода, таких как экранное представление, печать, чтение голосом (специальным голосовым браузером или программой чтения с экрана) и др.

Селекторы

Чтобы применить CSS-оформление используются *селекторы* – специальные указатели на HTML-объекты, к которым мы планируем применить css-правило.

CSSправило состоит из селектора и списка свойств и их значений, заключенного в фигурные скобки.

Вот три основных вида селекторов.

1) HTML селекторы.

Это простейший случай – в качестве селектора используется имя того html-элемента, который требуется отформатировать. Например, для тега селектором будет strong. Соответственно, для тега <h1> селектором будет h1, и так далее. Теперь можно переопределить внешний вид всех таких элементов в документе:

```
strong{font-weight:normal; color:red;}
h1 { font:bold10ptverdana;}
```

2) Селекторы класса.

«Класс» - это некое имя, строка, которое можно применить к любым HTML-тегам, чтобы впоследствии ссылаться на них по имени класса. Удобство таких селекторов в том, что можно присвоить одно имя класса множеству html-тегов в документе и затем управлять их внешним видом, обращаясь к ним по имени класса:

Пусть есть фрагмент HTML – кода, содержащий несколько абзацев, например:

```
<p class=”myClass”>Текст1</p>
<p>Текст2  </p>
<p class=”myClass”>Текст3</p>
```

Тогда, если применить стиль к классу (обратите внимание на точку перед именем класса в записи селектора!)

.myClass{font:bold10ptverdana;}, тошрифт изменится, только у первого и третьего абзаца.

3) ID селекторы (или идентификаторы)

Любой идентификатор (ID) – это некое имя, которое можно применить, так же, как и в случае с классами, к любому HTML-тегу. Основное отличие – ID должен быть уникален в рамках html-документа:

#myObj{margin:1em;}/* изменяем поля для элемента, у которого id=”myObj” ***/**
span#today{margin:1em;}/* изменяем поля для элемента span, у которого id=”today” ***/**

Список основных свойств, используемых для CSS правил, приведен в Приложении В.

Блочная верстка

Работа по созданию сайта начинается с разработки макетов страниц. Этот этап называется версткой. Различают блочную и табличную верстки.

Блочная верстка базируется на том, что блочные элементы HTML, такие как , <p>, <h1>и др., как правило, располагаются по вертикали, сверху вниз друг за другом в том порядке, в котором они встречаются в HTML коде. Кроме этого блокам можно задавать свойство плавучести (float:left | right | none | inherit). Если блоку указать свойство float:left, то он будет выровнен по левому краю, а все остальные блоки будут игнорировать его, как будто этого блока нет, за исключение текста, остальные блоки, которым задано это же свойство будут обтекать его справа, на сколько это позволяет ширина экрана или элемента внутри которого они находятся. Следует заметить, что любой элемент можно сделать блочным, заданием ему свойства display:block.

Свойство float:rightвыровняет блок по правому краю, а все остальные блоки будут игнорировать его, либо обтекать, если им задано это же свойство и если в коде идут подряд два или несколько блоков с указанным свойством, то первым вправо встанет тот блок, который идет первым в коде, остальные обтекают его слева.

Свойство float:none отменяет эффект плавучести для блока, но это не значит что блок будет располагаться как обычно сверху вниз, если выше расположен блок с эффектом плавучести, то нижний блок будет игнорировать верхний и встанет под него, чтобы этого не было нужно задать этому блоку свойство clear:both.

При необходимости произвольного задания положения блока используются разные способы позиционирования.

Существует четыре основных типа позиционирования:

1. Статическое (Static)

Используется как расположение «по умолчанию». В этом случае браузер просматривает html код, разделяет его на элементы и составляет из них страницу. Получается последовательность из ряда элементов. Выводятся они в том порядке, в котором указаны в html коде.

Применение параметров left, top, right и bottom не приводит к каким-либо результатам. Необходимо помнить о статическом позиционировании, когда используется относительное расположение.

2. Абсолютное (Absolute)

С помощью абсолютного позиционирование задаются координаты левого верхнего угла блока. При этом отсчет координат происходит относительно расположения родительского элемента. Если родительским элементом является окно браузера, тогда выравнивание блока происходит относительно него. Если существует другой элемент,

внутри которого расположен блок, тогда выравнивание происходит уже относительно этого элемента.

3. Фиксированное (Fixed)

Уже из названия становится ясно, что в данном случае элемент фиксируется. Он располагается в определенном месте веб-страницы и никуда не сдвигается. Подобное выравнивание часто применяется по отношению к всплывающим окнам, когда они фиксируются по центру и не смещаются при прокрутке страницы.

4. Относительное (Relative)

Расположение элемента в этом случае происходит относительно его же места в статическом положении.

Проще говоря, вы указываете браузеру, что необходимо передвинуть элемент на столько-то пикселей, относительно того места, где он расположен по умолчанию.

Если родительский элемент имеет относительное позиционирование, а вложенный в него элемент абсолютное, отсчет координат дочернего элемента будет производиться относительно родительского элемента, с учетом его смещения, если оно имеет место.

Для указания координат применяются четыре свойства:

1. Top - положительное значение (например, 20px) смещает блок на 20 пикселей вниз. Отрицательное значение (-20px) смещает элемент на 20 пикселей верх. Все это происходит относительно левого верхнего угла родительского элемента.

2. Left - смещение влево или вправо, в зависимости от знака. Относительно левого верхнего угла.

3. Right - смещение вправо и влево, смотря какой знак. Относительно правого верхнего угла.

4. Bottom - смещение вверх или вниз, зависит от знака. Происходит относительно левого нижнего угла.

Пример:

HTML:

```
<html>
<body>

<div id="q1">
<div id="q11"></div>
</div>
<div id="q2"></div>
</body>
</html>
```

CSS:

```
#q1 {
position:relative;
top:100px;
left:100px;
width:500px;
height:500px;
background-color:#CCCCCC;
}
#q11 {
background-color:#003399;
position:absolute;
bottom: -30px;
right: -50px;
```

```
width:100px;
height:100px;
}
#q2 {
background-color:#990066;
width:200px;
height:300px
}
```

Достоинства и недостатки блочной верстки:

Меньший объем страницы (+);

Реализация сложного дизайна с перекрывающимися блоками (+);

Трудно освоить, табличная верстка проще (-);

Чаще приходится решать вопросы кроссбраузерности. Блоки могут перекрываться при изменении разрешения экрана, масштабировании (-).

Описание лабораторной установки

Для выполнения лабораторной работы необходим персональный компьютер, подключенный к Интернет.

Необходимые программные средства для выполнения работы:

1. Любой Web-обозреватель (InternetExplorer, MozillaFirefox, Opera и др.).
2. Любой текстовый редактор (WindowsNotepad, MicrosoftWord идр.).

Одним из наиболее удобных для работы можно считать **Notepad++** (http://notepad-plus.sourceforge.net/ru/site.html).

Ход выполнения работы

Используем блочную верстку для создания web- страницы с динамическим меню навигации при помощи списков и правил CSS.

Необходимость в таком меню с набором гиперссылок - одна из часто встречающихся задач в практике веб-дизайнера. Меню двухуровневое: ссылки обычно группируют по категориям, а в основное меню включают только категории. При выборе категории должны отобразиться соответствующие ссылки (или подкатегории).

Как правило, такие динамические меню принято создавать средствами языка Javascript, позволяющего совершать любой сложности манипуляции с элементами веб-страницы. Однако, существует и решение на CSS - довольно простое и красивое.

Рассмотрим его пошагово.

Шаг 1. Создайте новый документ HTML.

Шаг 2. Внутри элемента head введите подходящий заголовок страницы(тег <title>).

Шаг 3. Также внутри элемента head введите тэги <style></style> .

Шаг 4. Создайте в теле документа набор категорий в виде неупорядоченного списка (ul), каждый элемент которого содержит список ссылок.

```
    <ulid="MainMenu">
    <li><a href="#" title="Популярные блюда">Популярные
блюда</a>
    <ul>
    <li><a href="/Italian.htm">Итальянскаякухня</a></li>
    <li><a href="/Snack.htm">Закуски</a></li>
    <li><a href="/Breakfast.htm">Закуски</a></li>
    <li><a href="/Sweets.htm">Сласти</a></li>
    <li><a href="/Fruits.htm">Фрукты</a></li>
    </ul>
```

```
    </li>
    <li><a href="#" title="Подарки к празднику">Подарки к
празднику</a>
    <ul>
    <li><a href="/Anniversary.htm">Юбилеи</a></li>
    <li><a href="/Baby.htm">Малышам</a></li>
    <li><a href="/Birthday-Food.htm">Деньрождения</a></li>
    <li><a href="/Congratulations.htm">Поздравляем!!!</a></li>
    </ul>
    </li>
    <li><a href="#" title="Выбор по цене">Выбор по цене</a>
    <ul>
    <li><a href="Under-1000.htm">Меньше 1000 руб</a></li>
    <li><a href="1000-To-1500.htm">От 1000 до 1500
руб</a></li>
    <li><a href="1500-To-2000.htm">От 1500 до 2000
руб</a></li>
    <li><a href="2000-To-3000.htm">От 2000 до 3000
руб</a></li>
    <li><a href="3000-up.htm">Дорого</a></li>
    </ul>
    </li>
    <li><a href="#" title="ФруктыиОвощи">ФруктыиОвощи</a>
    <ul>
    <li><a href="/Dried-Fruits-
Nuts.htm">СухофруктыиОрехи</a></li>
    <li><a href="/Fruit-Baskets.htm">Корзиныфруктов</a></li>
    <li><a href="/Fruit-Towers.htm">Горыфруктов</a></li>
    <li><a href="/Healthy-Food.htm">Здороваяпища</a></li>
    <li><a href="/Organic-Food.htm">Безпестицидов</a></li>
    </ul>
    </li>
    <li><a href="#" title="Садовые и комнатные
цветы">Цветы</a>
    <ul>
    <li><a href="/Fresh-Flowers.htm">Живыецветы</a></li>
    <li><a href="/Plants-And-Dish-
Gardens.htm">Садовыецветы</a></li>
    <li><a href="/Sympathy-Flowers.htm">"Говорящие"
цветы</a></li>
    </ul>
    </li>
    <li><a title="Корпоративныеподарки" href="/Corporate-
Food.htm">
    Корпоративные подарки</a></li>
                </ul>
```

Названия ссылок и категорий придумайте самостоятельно. Важно, чтобы атрибут id внешнего списка имел значение MainMenu - далее ему будет назначен особый стиль по этому идентификатору).

Шаг 5. В таблице стилей добавьте следующее правило:

```
#MainMenu> li {
       float: left;
       list-style-type: none;
      }
```

После применения указанного стилевого правила пункты внешнего списка (li) расположились горизонтально (за счёт обтекания):

Шаг 6. Пусть вложенные списки ссылок будут невидимыми (добавьте им (#MainMenuliul) свойство display:none;) и появляются только при наведении курсора на название соответствующей категории. Следующее правило с селектором псевдокласса hover имеет такой смысл: у списка (ul), вложенного в пункт списка (li), на который наведён указатель (:hover) и который вложен в элемент с id=#MainMenu, способ отображения следует сделать блочным (а не невидимым):

```
#MainMenuli:hoverul {
display:block;
   }
```

Шаг 7. Сохраните разрабатываемый документ, откройте его в браузере и убедитесь, что меню работает правильно.

Шаг 8. Принципиально механизм работает - осталась эстетическая сторона.

- Назначьте якорям любого уровня вложенности в меню (правило #MainMenu a) желаемый цвет (color), гарнитуру (font), а также уберите подчёркивание (text-decoration).
- Назначьте элементам списка категорий (правило #MainMenu>li) фоновый цвет (background), внутренний отступ (padding) и рамку справа (border-right).
- Назначьте элементам вложенного списка ссылок (правило #MainMenulili) такой же фоновый цвет, как и в списке категорий, а также небольшой отступ и рамку снизу. Кроме того, уберите маркировку списка (list-style-type).
- Уберите у списка ссылок (правило #MainMenuliul) поля и отступы (margin и padding).

Небольшой проблемой является то, что некоторые элементы списка категорий увеличиваются в ширину при наведении на них указателя. Это является следствием того, что ширина элемента списка определяется шириной всего содержимого - включая вложенный список. Однако если выбросить вложенный список из нормального потока и позиционировать его абсолютно, то его ширина более не будет влиять на ширину родительского элемента. Поэтому укажите для списка ссылок абсолютное позиционирование, а для элементов списка категорий - относительное. Проверьте в браузере. Затем уточните положение вложенного списка относительно его контейнера путём задания значения свойств left и top (не опускайте выпадающий список слишком низко, иначе он станет пропадать при попытке выбрать в нём ссылку).

Зачем нужно было позиционировать элементы списка категорий относительно? Уберите соответствующее правило, и вам всё станет ясно.

Шаг 9. Добавьте последний штрих: пусть элементы обоих списков при наведении указателя немного изменяют цвет фона (правило #MainMenuli:hover).

Окончательный результат должен быть подобен представленному на рисунке 3.1:

Рисунок. 3.1. Меню с выпадающими списками ссылок в действии

Задание на самостоятельную работу

Самостоятельная работа включает 3 задания. Размер внешнего блока 500px x400px.

Задание 1. Создать шаблон в соответствии с рисунком, используя для позиционирования свойства float, margin и не используя свойство position. Все размеры приведены в пикселях.

Номер варианта определяется как (№ mod 12)+1, где № - номер по журналу академической группы. Варианты приведены в таблице 3.1.

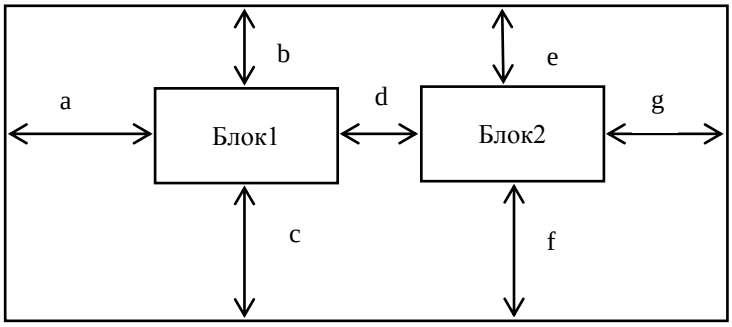
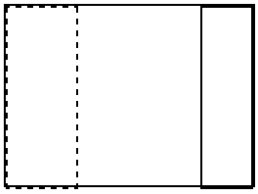


Таблица 3.1. Варианты заданий к лабораторной работе №3

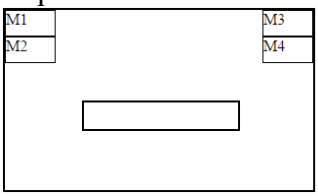
№ Варианта	a	b	c	d	e	f	g	Высота Б1	Ширина Б1	Высота Б2	Ширина Б2
1	10	20	-	-	10	-	10	100	150	100	100
2	50	50	-	20	40	-	-	50	200	50	150
3	-	50	-	50	50	-	30	150	100	120	120
4	20	-	50	0	-	40	-	120	120	150	100
5	50	50	-	-10	-	-	-	100	150	120	75
6	-	-	100	20	-	70	30	150	250	50	50
7	40	-	70	-	-	50	20	80	120	120	80
8	30	40	-	90	50	-	-	60	140	100	60
9	-	80	-	-	70	-	100	90	150	120	100
10	-	-	70	90	-	70	80	100	120	130	110
11	40	50	-	-	50	-	100	80	120	90	180
12	-	100	-	30	40	-	50	120	120	100	50

задание 2. Создать шаблон в соответствии с рисунком, используя свойство position. Вариант 1. Ячейка1 по правому краю основного блока, ячейка2 – по левому.

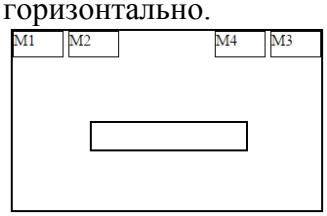


Вариант 2. Ячейка1 по нижнему краю основного блока, ячейка2 – по левому.
Вариант 3. Ячейка1 по левому краю основного блока, ячейка2 – по нижнему.
Вариант 4. Ячейка1 по верхнему краю основного блока, ячейка2 – по нижнему.
Вариант 5. Ячейка1 по правому краю основного блока, ячейка2 – по нижнему.
Вариант 6. Ячейка1 по нижнему краю основного блока, ячейка2 – по нижнему.
Вариант 7. Ячейка1 по левому краю основного блока, ячейка2 – по левому.
Вариант 8. Ячейка1 по верхнему краю основного блока, ячейка2 – по левому.
Вариант 9. Ячейка1 по правому краю основного блока, ячейка2 – по верхнему.
Вариант 10. Ячейка1 по нижнему краю основного блока, ячейка2 – по верхнему.
Вариант 11. Ячейка1 по левому краю основного блока, ячейка2 – по верхнему.
Вариант 12. Ячейка1 по верхнему краю основного блока, ячейка2 – по верхнему.

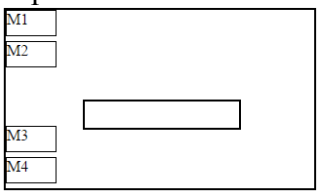
Задание 3. Создать шаблон в соответствии с рисунком, используя свойства position, float.
Вариант 1. Элементы меню вверху вертикальными парами



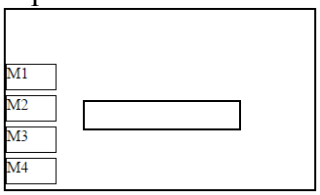
Вариант 2. Элементы меню внизу вертикальными парами.
Вариант 3. Два элемента меню слева вверху горизонтально, два справа вверху горизонтально.



Вариант 4. Элементы меню слева вертикально. Два элемента меню вверху, два внизу.



Вариант 5. Элементы меню слева вертикально вверху.
Вариант 6. Элементы меню слева вертикально внизу.



Вариант 7. Элементы меню справа вертикально вверху.
Вариант 8. Элементы меню справа вертикально внизу.
Вариант 9. Элементы меню справа вертикально. Два элемента меню вверху, два внизу.
Вариант 10. Элементы меню справа внизу горизонтально.

Вариант 11. Два элемента меню слева внизу горизонтально, два справа внизу горизонтально.
Вариант 12. Элементы меню слева внизу горизонтально.

Содержание отчета

1. Титульный лист.
2. Постановка задачи.
3. Исходные коды страниц
4. Скриншоты страниц.
5. Выводы.
6. Список используемой литературы.

Контрольные вопросы

1. В чем заключается задача web верстки?
2. В чем преимущество использования языка CSS?
3. Что такое селектор?
4. Какие виды селекторов Вам известны?
5. Какие способы позиционирования элементов существуют?
6. Перечислите достоинства и недостатки блочной верстки.
7. Какие свойства применяются для указания координат?

2.4. Лабораторная работа №4. Основы табличной верстки

Цель работы

Изучить особенности табличной верстки сайтов, приобрести практические навыки использования приемов табличной верстки.

Краткие теоретические сведения

В основу табличной верстки положено представление сайта в виде таблицы. Как известно подавляющее большинство сайтов имеют типовую структуру, которую можно детализировать под конкретную задачу. Весь макет представляется некоторой таблицей, а каждая ячейка наполняется своим информационным содержанием.

Достоинства и недостатки табличной верстки:
Проста в понимании, не требует более глубоких знаний HTML и CSS (+);
Интуитивно понятна при построении, минимум CSS правил (+);
Трудно разбираться в HTML коде при более сложной структуре сайта (-);
Пока вся таблица не загрузиться она не будет показана на экране (-);
Сложный дизайн с перекрытием элементов не реализуем (-);
Много лишнего кода (-).

Для того, чтобы применить таблицу стилей к HTML-документу, можно избрать один из трёх способов, либо комбинировать их:

1. Применить *внешние стили* (в виде отдельного текстового .*css*-файла) с помощью тега <link>.
2. *Встроить стили* непосредственно в HTML-документ (в виде блока css-текста) с помощью тега <style>.
3. Применить *inline-стиль*, то есть назначить стиль конкретному HTML-элементу непосредственно в документе, с помощью HTML-атрибута style (см. ЛР№1).

Наиболее распространенными являются первый и второй способы.

Описание лабораторной установки

Для выполнения лабораторной работы необходим персональный компьютер, подключенный к Интернет.

Необходимые программные средства для выполнения работы:

1. Любой Web-обозреватель (InternetExplorer, MozillaFirefox, Opera и др.).
2. Любой текстовый редактор (WindowsNotepad, MicrosoftWord идр.).

Одним из наиболее удобных для работы можно считать **Notepad++** (http://notepad-plus.sourceforge.net/ru/site.html).

Ход выполнения работы

Для иллюстрации приемов табличной верстки возьмем задачу разработки сайта об автомобилях. Сайт преднзначен для посещения автолюбителей, интересующихся историей появления автомобиля, новинками автомобилестроения и т.п.

Сайт будет состоять из нескольких страниц, структуру которых надо спроектировать заранее. Для его реализации необходимо выполнить перечисленные ниже шаги.

Собственно проект страницы может выглядеть следующим образом (рис.4.1):

<table>
<tr><td colspan="2">Шапка страницы</td></tr>
<tr><td>Главное (Top) меню</td><td>Форма авторизации</td></tr>
<tr><td>Содержимое страницы (Контент)</td><td>Банеры</td></tr>
<tr><td colspan="2">Подвал</td></tr>
</table>

Рис.4.1. Верстка страниц сайта об автомобилях

Шаг 1. Создайте в блокноте файл index.php и подготовьте заголовок.

```
<html>
<head>
  <title> Сайт об автомобилях </title>
  <meta http-equip="Content-type" content="text/html; charset=1251" />
  <link rel="stylesheet" type="text/css" href="styles/main.css" />
</head>
<body>
</body>
</html>
```

Применим внешние стили, с описанием стилей в отдельном файле, расположенном в папке styles с именем main.css.

Шаг 2. Основу шаблона сайта составляет таблица, добавьте ее в тег <body>.

```
  <table id=”main”>
        <tr>
              <td colspan="2" id="header">
              <h1> Сайт об автомобилях</h1>
              <p>
                    <img src= "images/header.png" alt ="Шапка сайта" />
              </p>
              </td>
        </tr>
        <tr>
              <td colspan="2">
                    <hr />
              </td>
        </tr>
</table>
```

Подберите картинку заголовка и разместите её в папке images под именем header.png.

Проверьте результаты своей работы, открыв файл index.php в браузере. Так как он имеет расширение имени *php*, то автоматически он не откроется!

Шаг 3**.** Создайте папку *styles*. В ней создайте файл *main.css*.
В файле main.css для тега <body> обнулим отступы (margin, padding):

```
body {
  margin: 0;
  padding: 0;
}
```

Обнулим отступы для прямой (тег <hr>):

```
hr {
  margin: 0;
}
```

Используя ID селекторы, растянем основную часть на весь экран и сделаем выравнивание заголовка по центру.

```
#main {
  width: 100%;
}
#header {
  text-align: center;
}
```

Дополнительно можно убрать отступы в заголовке (использованы контекстные селекторы):

```
#header h1, #header p {
  margin: 0 ;
}
```

Проверьте результаты своей работы, открыв файл index.php в браузере.

Закончив с заголовком, переходим к следующей части таблицы: верхнего меню и форме авторизации (рис.4.2).

Шаг 4. В соответствии с проектом страницы добавьте в файле index.php в таблицу следующую строку и разбейте ее на две ячейки:

```
<tr>
  <td>  </td>
  <td>  </td>
</tr>
<tr>    <!-- прямая линия -->
        <tdcolspan="2"><hr />
        </td>
</tr>
```

В первой ячейке будет меню из четырех пунктов: Главная, Регистрация, Статьи, Гостевая книга. Оформите его в виде таблицы из одной строки.

```
<tableid="topmenu">
        <tr>
                <td>
                        <a href ="index.php"> Главная </a>
                </td>
```

```
                <td>
                    <a href ="reg.php"> Регистрация </a>
                </td>
                <td>
                    <a href ="articles.php"> Статьи </a>
                </td>
                <td>
                    <a href ="guestbook.php"> Гостевая книга </a>
                </td>
            </tr>
        </table>
```

Файлы reg.php, articles.php, guestbook.php пока отсутствуют.
Проверьте результаты своей работы, открыв файл index.php в браузере.

Следующей является форма авторизации, которая находится в ячейке таблицы рядом с главным меню (рис.4.2).

Содержимым формы будут два поля для ввода E-mail и пароля и ниже кнопки «Войти».

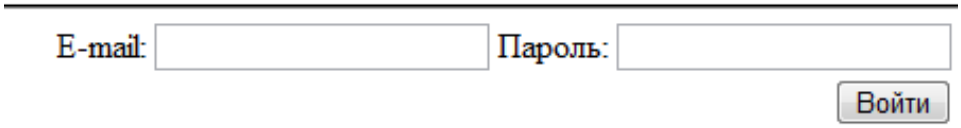


Рис.4.2. Форма авторизации

Шаг 5. Во вторую ячейку поместите таблицу формы авторизации.

```
<table>
        <tr>    <!-- 1-я строка формы авторизации -->
                <td>E-mail:</td>
                <td>
                    <input type="text" name="email" />
                </td>
                <td>Пароль: </td>
                <td>
                    <input type="password" name="password" />
                </td>
        </tr>
        <tr>    <!-- 2-я строка формы авторизации -->
                <td colspan="4">
                    <input  type="submit" name="button_auth" value="Войти"
/>
                </td>

        </tr>

    </table>
```

Проверьте результаты своей работы, открыв файл index.php в браузере.

Строка меню и формы авторизации готова. Внизу проведена ограничительная горизонтальная линия.

Теперь займемся оформлением главного меню и формы авторизации, используя стили.

Шаг 6. Откройте папку styles и в ней файл main.css. Добавьте следующие стили.

```
#topmenu{
  font-size:150%;
  width:100%
}
#topmenutd{text-align:center;}
.right{float:right;}
.righttd{text-align:right;}
```

Проверьте результаты своей работы, открыв файл index.php в браузере.

Следующая часть разметки страницы Content и Banners.

Шаг 7. Вновь откройте файл index.php. Добавьте после горизонтальной черты отделяющей верхнее меню и форму авторизации еще одну строку и ячейку. В ячейке поместите таблицу.

```
<tr>
  <td colspan=”2”>
        <table cellpadding=”0” cellspacing=”0” id=”content”>
        </table>
  </td>
</tr>
```

Содержимым таблицы будет строка из двух ячеек.

```
<tr>
  <td></td><!—1-я ячейка -->
  <td></td><!—2-я ячейка -->
</tr>
```

В первой ячейке будет располагаться основной текст об автомобиле и его фотография.

```
  <td><!—1-ая ячейка -->
        <div class=”article”>
              <h1> Автомобиль </h1>
              <p class=”article_img”>
                    <imgsrc=”images/main.png” alt=”Статья1”/>
              </p>
              <p>Текст…</p>
              <p>Текст…</p>
              <p>Текст…</p>
              <p>Текст…</p>
        </div>
  </td>
```

Во второй ячейке будут располагаться баннеры.

```
<td id="banners_240"><!--2-я ячейка -->
  <div class=”banner”>
        <a href=”#”>
              <imgsrc=”images/banner_1.jpg” alt=”Баннер1”/>
```

```
        </a>
  </div>
  <hr />
  <div class=”banner”>
        <a href=”#”>
              <imgsrc=”images/banner_2.jpg” alt=”Баннер2”/>
        </a>
  </div>
  <hr />
</td>
```

Ссылки на реальные сайты пока отсутствуют.

Проверьте результаты своей работы, открыв файл index.php в браузере.

Шаг 8. Для оформления стилей откройте файл main.css. Для контента и подвала добавим в него:

```
#content {
  width: 100%;
  padding: 0;
  margin: 0;
}
#contenttd{
  vertical-align: top;
}
.article h1, .article_img {
  text-align: center;
}
#banners_240 {
  border-left: 1px solid  #000;
  width: 240px;
}
#footer p {
  text-align: center;
}
.article h1, .article h2, .article_img {
  text-align: center;
}
```

Проверьте результаты своей работы, открыв файл index.php в браузере.

Последней строкой является Подвал(Footer).

Шаг 9. Отделите подвал чертой и укажите авторские права.

```
<tr>
        <tdcolspan="2"><hr /></td>
</tr>
<tr>
  <tdcolspan="2" id="footer">
        <p>Все права защищены &copy; "Сайт об автомобилях"    </p>
  </td>
</tr>
```

Проверьте результаты своей работы, открыв файл index.php в браузере.

Задание на самостоятельную работу.

1. На основе примера создания сайта об автомобилях создать собственный сайт согласно варианту, который определяется следующим образом: найти остаток от деления № студента в списке академической группы на 19; к полученному остатку добавить единицу. Варианты заданий приведены в таблице 4.1.

Таблица 4.1. Варианты заданий к лабораторной работе №4

№ варианта	**Организация, для которой выполняется проектирование учебного сайта**
1	магазин;
2	ресторан;
3	санаторий;
4	заказ такси;
5	гостиница;
6	поликлиника;
7	офис по продаже недвижимости;
8	бюро по трудоустройству и профориентации
9	атлетический клуб
10	больница (стационар)
11	автомобильные грузоперевозки
12	продажа железнодорожных билетов
13	малое производственное предприятие;
14	деканат ВУЗа
15	автосервис
16	туристическая фирма
17	парикмахерская
18	троллейбусный парк (организация движения)
19	библиотека

2. Макет сайта создать в одном из двух вариантов: с использованием табличной или блочной верстки.

3. По результатам выполнения работы подготовить отчет.

Содержание отчета

1. Титульный лист.
2. Постановка задачи.
3. Структурная схема сайта.
4. Исходные тексты.
5. Скриншоты страниц.
5. Выводы.
6. Список используемой литературы.

Контрольные вопросы

1. В чем заключаются преимущества табличной верстки?
2. Какие недостатки табличной верстки?
3. Какие Вы знаете способы применения таблицы стилей к HTML-документу?
4. Для чего используются атрибуты margin и padding?
5. Как, используя ID селекторы, растянуть основную часть сайта на весь экран?

3.СПИСОК РЕКОМЕНДОВАННОЙ ЛИТЕРАТУРЫ

1. Хоган Б.Книга веб-программиста: секреты профессиональной разработки веб-сайтов [Текст]/ Б.Хоган, К.Уоррен.– СПб.: Питер, 2013. – 288с.
2. Дуванов А.А. Web-конструирование. HTML [Текст]/ А.А. Дуванов. – СПб.: БХВ-Петербург, 2003. — 325 с.
3. Хольцшлаг М. Э. Использование HTML 4: Пер. с англ.: Уч. пос. [Текст]/М. Э. Хольцшлаг. – М: Издательский дом «Вильямс», 2000. — 1008 с.
4. Храмцов П.Б. Основы WEB-технологий [Текст]/П.Б. Храмцов, С.А.Брик, А.М. Русак, А.И.Сурин. – М.: ИТУИТ.РУ, 2003. – 512 с.

ПРИЛОЖЕНИЕ А. Варианты заданий к лабораторной работе №1

Вариант 1

Типы компьютерных сетей

Одним из наиболее простых типов ВС являются ЛВС (LAN – LocalAreaNetwork). Локальные ВС служат для объединения компьютеров, расположенных на незначительном расстоянии друг от друга, причем физическое соединение компьютеров ЛВС, выполняемое с помощью специальных кабелей, обычно не выходит за пределы одного помещения (здания).

С введением ЛВС быстро возникла потребность в том, чтобы в пределах предприятия объединить *такие сети в единую* систему электронной обработки данных и оставить их в качестве небольших систем коллективного **пользования**в отдельных подразделениях.

Таким образом, корпоративной сетью называют совокупность ЛВС, мощных ЭВМ и терминальных систем, использующих общую информационную магистраль для обмена. Как правило, подобная информационная магистраль обладает на порядок большей производительностью (скоростью передачи данных), чем магистраль ~~ЛВС~~.

Вариант 2

Типы компьютерных сетей

Одним из наиболее простых типов ВС являются ЛВС (LAN – LocalAreaNetwork). Локальные ВС служат для объединения компьютеров, расположенных на незначительном расстоянии друг от друга, причем физическое соединение компьютеров ЛВС, выполняемое с помощью специальных кабелей, обычно не выходит за пределы одного помещения (здания).

С введением ЛВС быстро возникла *потребность* в том, чтобы в пределах предприятия объединить такие сети в единую систему электронной обработки данных и оставить их в качестве небольших систем коллективного **пользования** в отдельных подразделениях.

Таким образом, ~~корпоративной сетью~~ называют совокупность ЛВС, мощных ЭВМ и терминальных систем, использующих общую информационную магистраль для обмена. Как правило, подобная информационная магистраль обладает на порядок большей производительностью (скоростью передачи данных), чем магистраль ЛВС.

Вариант 3

Типы компьютерных сетей

Одним из наиболее простых типов ВС являются ЛВС (LAN – LocalAreaNetwork). Локальные ВС служат для объединения компьютеров, расположенных на незначительном расстоянии друг от друга, причем физическое соединение компьютеров ЛВС, выполняемое с помощью специальных кабелей, обычно не выходит за пределы одного помещения (здания).

С введением ЛВС быстро возникла *потребность* в том, чтобы в пределах предприятия объединить такие сети в единую систему электронной обработки данных и оставить их в качестве небольших систем коллективного **пользования** в отдельных подразделениях.

Таким образом, корпоративной сетью называют совокупность ЛВС, мощных ЭВМ и терминальных систем, ~~использующих~~ общую информационную магистраль для обмена. Как правило, подобная информационная магистраль обладает на порядок большейпроизводительностью (скоростью передачи данных), чем магистраль ЛВС.

Вариант 4

Типы компьютерных сетей

Одним из наиболее простых типов ВС являются ЛВС (LAN – LocalAreaNetwork). Локальные ВС служат для объединения компьютеров, расположенных на незначительном расстоянии друг от друга, причем физическое соединение компьютеров ЛВС, выполняемое с помощью специальных кабелей, обычно не выходит за пределы одного помещения (здания).

С введением **ЛВС** быстро возникла *потребность* в том, чтобы в пределах предприятия объединить такие сети в единую систему электронной обработки данных и оставить их в качестве небольших систем коллективного пользования в отдельных подразделениях.

Таким образом, корпоративной сетью называют совокупность ЛВС, мощных ЭВМ и терминальных систем, использующих общую информационную ~~магистраль для обмена~~. Как правило, подобная информационная магистраль *обладает на порядок большей*производительностью (скоростью передачи данных), чем магистраль ЛВС.

Вариант 5

Типы компьютерных сетей

Одним из наиболее простых типов ВС являются ЛВС (LAN – LocalAreaNetwork). Локальные ВС служат для объединения компьютеров, расположенных на незначительном расстоянии друг от друга, причем физическое соединение компьютеров ЛВС, выполняемое с помощью специальных кабелей, обычно не выходит за пределы одногопомещения (здания).

С введением ЛВС быстро возникла потребность в том, чтобы в пределах предприятия объединить такие сети в *единую систему*электронной обработки данных и оставить их в качестве небольших систем коллективного **пользования** в отдельных подразделениях.

Таким образом, корпоративной сетью называют совокупность ЛВС, мощных ЭВМ и терминальных систем, использующих общую информационную магистраль для обмена.

Как правило, подобная информационная магистраль обладает на порядок большей производительностью (~~скоростью передачи~~ данных), чем магистраль ЛВС.

Вариант 6

Типы компьютерных сетей

Одним из наиболее простых типов ВС являются ЛВС(LAN – LocalAreaNetwork). Локальные ВС служат для объединения компьютеров, расположенных на незначительном расстоянии друг от друга, причем физическое соединение компьютеров ЛВС, выполняемое с помощью специальных кабелей, обычно не выходит за пределы одногопомещения (здания).

С введением ЛВС быстро возникла потребность в том, чтобы в пределах **предприятия объединить**такие сети в единую систему электронной обработки *данных и оставить* их в качестве небольшихсистем коллективного пользования в отдельных подразделениях.

Таким образом, ~~корпоративной сетью~~ называют совокупность ЛВС, мощных ЭВМ и терминальных систем, использующих общую информационную магистраль для обмена. Как правило, подобная информационная магистраль обладает на порядок большей производительностью (~~скоростью передачи~~ данных), чем магистраль ЛВС.

Вариант 7

Интерфейс ODBC

Сегодня большинство информационных систем в той или иной степени используют базы данных. Не составляют исключение и системы, основанные на веб-технологиях. Поэтому организация взаимодействия веб-приложений с СУБД является неотъемлемой составной частью веб-технологий.

До начала 90-х годов существовало несколько разных поставщиков баз данных, *каждый из которых* имел собственный интерфейс. Если приложению было необходимо обмениваться данными с несколькими источниками данных, **для взаимодействия** с каждой из баз данных было необходимо написать отдельный код. С целью решения этой проблемы Майкрософт и ряд других компаний создали стандартный интерфейс для получения и отправки данных ~~источникам данных~~различных типов. Этот интерфейс получил название opendatabaseconnectivity(**ODBC**).

С помощью *ODBC прикладные программисты*смогли разрабатывать приложения с использованием единого интерфейса доступа к данным, не учитывая тонкости взаимодействия с различными источниками данных.

Вариант 8

Интерфейс ODBC

Сегодня большинство информационных систем в той или иной степени используют базы данных. Не составляют исключение и системы, основанные на веб-технологиях. Поэтому организация взаимодействия веб-приложений с ***СУБД*** является неотъемлемой составной частью веб-технологий.

До начала 90-х годов существовало несколько разных поставщиков баз данных, *каждый из которых* имел собственный интерфейс. Если приложению было необходимо обмениваться данными с несколькими источниками данных, для взаимодействия с каждой из баз данных было необходимо написать отдельный код. С целью решения этой проблемы ~~*Майкрософт*~~ и ряд других компаний создали стандартный интерфейс для получения и отправки данных источникам данных различных типов. Этот интерфейс получил название *opendatabaseconnectivity*(ODBC).

С помощью ODBC прикладные программисты смогли разрабатывать приложения с использованием единого интерфейса доступа к данным, не учитывая тонкости взаимодействия с различными источниками данных.

Вариант 9

Интерфейс ODBC

Сегодня большинство информационных систем в той или иной степени используют базы данных. Не составляют исключение и системы, основанные *на веб-технологиях*. Поэтому организация взаимодействия веб-приложений с **СУБД** является неотъемлемой составной частью веб-технологий.

До начала 90-х годов существовало несколько разных поставщиков баз данных, *каждый из которых* имел собственный интерфейс. Если приложению было необходимо обмениваться данными с несколькими источниками данных, для взаимодействия с каждой из баз данных было необходимо написать отдельный код. С целью решения этой проблемы **Майкрософт** и ряд других компаний создали стандартный интерфейс для получения и отправки данных источникам данных различных типов. Этот интерфейс получил название *opendatabaseconnectivity*(ODBC).

С помощью ODBC прикладные ~~программисты~~ смогли *разрабатывать* приложения с использованием единого интерфейса *доступа* к данным, не учитывая тонкости взаимодействия с различными источниками данных.

Вариант 10

Интерфейс ODBC

Сегодня большинство *информационных систем* в той или иной степени используют базы данных. Не составляют исключение и системы, основанные *на веб-технологиях*. Поэтому организация взаимодействия веб-приложений с **СУБД** является неотъемлемой составной частью веб-технологий.

До начала 90-х годов существовало несколько разных поставщиков баз данных, *каждый из которых* имел собственный интерфейс. Если приложению было необходимо обмениваться данными с несколькими источниками данных, для взаимодействия с каждой из баз данных было необходимо ***написать отдельный*** код. С целью решения этой проблемы Майкрософт и ряд других компаний создали стандартный интерфейс для получения и отправки данных источникам данных различных типов. Этот интерфейс получил название opendatabaseconnectivity(ODBC).

С помощью ~~ODBC~~прикладныепрограммистысмогли разрабатывать приложения с использованием единого *интерфейсадоступа* к данным, не учитывая тонкости взаимодействия с различными источниками данных.

Вариант 11

Интерфейс ODBC

Сегодня большинство информационных систем в той или иной ***степени*** используют базы данных. Не составляют исключение и системы, основанные *на* веб-технологиях. Поэтому организация взаимодействия веб-приложений с СУБД является неотъемлемой составной частью веб-технологий.

До начала 90-х годов существовало несколько разных поставщиков баз данных, *каждый* ***из которых*** имел собственный интерфейс. Если приложениюбыло необходимо обмениваться данными с несколькими источниками данных, для взаимодействия с каждой из баз данных было необходимо *написать отдельный* код. С целью решения этой проблемы Майкрософт и ряд других компаний создали стандартный интерфейс для получения и отправки данных источникам данных различных типов. ~~Этот интерфейс~~ получил название opendatabaseconnectivity(ODBC).

С помощью ODBCприкладныепрограммистысмогли разрабатывать приложения с использованием единого интерфейсадоступа к данным, не учитывая тонкости взаимодействия с различными источниками данных.

Вариант 12

Интерфейс ODBC

Сегодня большинство информационных систем в той или иной степени используют базы данных. Не составляют исключение и системы, основанные *на* веб-технологиях. Поэтому организациявзаимодействия веб-приложений с СУБД является неотъемлемой составной частью веб-технологий.

До начала 90-х годов ***существовало несколько*** разных поставщиков баз данных, каждый из которых имел собственный интерфейс. Если приложению было необходимо обмениваться данными с несколькими источниками данных, для взаимодействия с каждой из баз **данных было** необходимо написать отдельныйкод. С целью решения этой проблемы Майкрософт и ряд других компаний создали стандартный интерфейс для получения ~~и отправки данных~~ источникам данных различных типов. Этот интерфейс получил название opendatabaseconnectivity(ODBC).

С помощью ODBCприкладныепрограммистысмогли *разрабатывать* приложения с использованием единого интерфейсадоступак данным, не учитывая тонкости взаимодействия с различными источниками данных.

ПРИЛОЖЕНИЕ Б. Варианты задания к лабораторной работе №2

Вариант 1

ПЛАН УЧЕБНОГО ПРОЦЕССА								
№	**Дисциплина**	**Кафедра**	**Всего часов**					
			Всего	**Ауд**	**Лк**	**Лб**	**Пр**	**СРС**
1	История	ФСН	54	27	18	0	9	27
2	Математика	ВМ	540	282	141	0	141	258
3	Электротехника, электроника и схемотехника	СПЭМС	108	54	18	0	36	54
4	Дискретная математика	ИТиКС	216	139	87	0	52	77
5	Программирование	ИТиКС	405	210	105	87	18	195
6	Компьютерная логика	ИТиКС	54	27	18	0	9	27
7	Теория вероятностей и математическая статистика	ИТиКС	162	72	54	18	0	90
8	Физика	Физики	324	194	106	88	0	130
9	Системное программирование	ИТиКС	108	72	36	18	18	36
10	Алгоритмы и методы вычислений	ИТиКС	189	89	36	36	17	100
11	Иностранный язык	ИЯ	216	104	52	35	17	112

Вариант 2

№	**Дисциплина**	**Часов в неделю (Лекций, Лаб.раб, Практ. раб)**											
		1 курс						**2 курс**					
		1 сем			**2 сем**			**3 сем**			**4 сем**		
1	*История*							**1**	**0**	**1**			
2	*Математика*	**3**	**0**	**3**	**3**	**0**	**3**	**2**	**0**	**2**			
3	*Электротехника, электроника и схемотехника*	**1**	**0**	**2**									
4	*Дискретная математика*	**2**	**0**	**1**	**3**	**0**	**2**						
5	*Программирование*	**3**	**2**	**0**	**3**	**3**	**0**	**0**	**0**	**1**			
6	*Компьютерная логика*							**1**	**0**	**0**			
7	*Теория вероятностей и математическая статистика*							**3**	**1**	**0**			
8	*Физика*	**2**	**2**	**0**	**2**	**2**	**0**	**2**	**1**	**0**			
9	*Системное программирование*							**2**	**1**	**1**			
10	*Алгоритмы и методы вычислений*							**2**	**2**	**0**	**0**	**0**	**1**
11	*Иностранный язык*							**1**	**1**	**0**	**2**	**1**	**1**

ПРИЛОЖЕНИЕ В. Перечень основных свойств CSS в соответствии со спецификацией CSS2.1

Свойства	Значение
Фон	
background	Сокращенныйвариантзаписидлясвойствbackground-color, background-image, background-repeat, background-attachment и background-position.
background-attachment	Устанавливает, должна ли фоновая картинка скролиться или должна быть зафиксирована в окне браузера (scroll/fixed).
background-color	Устанавливает цвет фона для элемента.
background-image	Устанавливает фоновую картинку для элемента(`url('...файл...')`).
background-position	Устанавливает первоначальное положение для фоновой картинки(процент% процент%)
background-repeat	Управляет циклическим повторением фоновой картинки(repeat/repeat-x/repeat-y)
Рамка (граница, бордюр)	
border	Краткий вариант записи для свойств border-width, border-style и border-color. Влияет на все четыре границы элемента.
border-color	Устанавливает цвет рамки со всех сторон элемента.
border-width	Устанавливает толщину рамки со всех сторон элемента.
border-style	Определяет стиль оформления рамки вокруг элемента.
border-collapse	Указывает ячейкам таблицы, иметь ли собственный бордюр или общий с соседней ячейкой(collapse/separate)
border-spacing	Устанавливает расстояние между ячейками таблицы.
Шрифт	
font	Краткийвариантзаписисвойств font-style, font-variant, font-weight, font-size, line-height и font-family.
font-family	Определяет шрифт(ы) для отображения текста.
font-size	Управляет размером шрифта.
font-style	Управляет наклоном шрифта (normal/italic).
font-variant	Управляет внешним видом строчных букв (строчные как прописные, "капитель").
font-weight	Управляет толщиной (насыщенностью) шрифта(normal/bold).
Позиционирование	
position	Определяет порядок, в соответствии с которым элемент отображается на веб-странице(absolute/relative/fixed).
bottom	Сдвигает элемент относительно нижнего края. Используется совместно со свойством position.
left	Сдвигает элемент относительно левого края. Используется совместно со свойством position.
top	Сдвигает элемент относительно верхнего края. Используется совместно со свойством position.
right	Сдвигает элемент относительно правого края. Используется совместно со свойством position.
z-index	Определяет порядок, в соответствии с которым элементы накладываются друг на друга, если необходимо отобразить их на одном месте(z-idex:2;).

Форматирование	
clear	Запрещает/разрешает элементу обтекать "floated" объекты(left/right/both/none).
clip	Определяет область элемента, которая должна отображаться на странице (rect (Y1,X1,Y2,X2))
display	Изменяет базовые свойства элементов(Значение none позволяет скрыть объект).
float	Сдвигает элемент к правому или левому краю (left/right).
height	Определяет высоту прямоугольной области вокруг элемента.
overflow	Определят как отображать блочный элемент в случае, если его содержимое выходит за рамки родительского элемента(visible/scroll/hidden).
visibility	Управляет настройкой видимости элемента(none/hidden/collapse).
width	Определяет ширину прямоугольной области вокруг элемента.
Списки	
list-style	позволяет одновременно задать три параметра для маркеров пунктов списка: list-style-type, list-style-position и/или list-style-image.
list-style-image	Устанавливает изображение, которое будет использоваться для маркировки пунктов списка, например, ul{list-style-image:url('/images/rhomb.gif')}
list-style-position	Определяет, как отобразить на странице маркер пункта в списке: внутри того же прямоугольника, в котором располагается элемент списка или вне его.
list-style-type	Определяет, какой вид будет иметь маркер пункта в списке..
Текст	
direction	Применяется при создании сайтов на языках, в которых чтение страницы идет справа налево.
letter-spacing	Определяет длину интервала между буквами.
page-break-inside	Определяет размер межстрочного интервала.
text-align	Выравнивает содержимое блочного элемента (текст или изображение) относительно границ блока, а так же содержимое ячеек таблицы по горизонтали.
text-decoration	Определяет, какой оформительский прием нужно применить к тексту(underline/overline/line-through/none).
text-indent	Определяет размер отступа первой строки в тексте.
text-transform	Управляет написанием прописных или строчных букв в тексте(uppercase/lowercase/capitalize/none).
vertical-align	Определяет высоту содержимого внутри инлайн элемента или внутри ячейки таблицы.
white-space	Определяет как отображать пробелы, символы табуляции и пустой строки (pre).
word-spacing	Определяет расстояние между словами.
Поля	
padding	Сокращенный способ задать следующие параметры: padding-top, padding-right, padding-bottom и/или padding-left.
Прочее	
caption-side	Определяет, где будет отображаться заголовок таблицы: над

	ней или под ней (top/bottom).
color	Устанавливает цвет текста элемента.
cursor	Определяет вид курсора при наведении мышки на некий элемент.
empty-cells	Определяет, нужно ли отображать границы и фон ячейки, если в ней нет содержимого (show/hide).
margin	Сокращенныйспособзадатьследующиепараметры: margin-top, margin-right, margin-bottom и/или margin-left
margin-bottom	Определяет ширину внешнего пространства между нижним бордюром и невидимой границей прямоугольника.
margin-left	Определяет ширину внешнего пространства между левым бордюром и невидимой границей прямоугольника.
margin-right	Определяет ширину внешнего пространства между правым бордюром и невидимой границей прямоугольника.
margin-top	Определяет ширину внешнего пространства между верхним бордюром и невидимой границей прямоугольника.
max-height	Определяет максимальную высоту элемента.
max-width	Определяет максимальную ширину элемента.
min-height	Определяет минимальную высоту элемента.
min-width	Определяет минимальную ширину элемента.
outline	Это быстрый способ задать следующие параметры: outline-width, outline-style и/или outline-color.Свойство аналогичное border.
outline-color	Определяет цвет контура вокруг элемента.
outline-style	Определяет вид контура вокруг элемента.
outline-width	Определяет ширину контура вокруг элемента.
quotes	Определяет вид открывающей и закрывающей кавычки в тексте.
table-layout	Определяет ширину столбцов в таблице.