

ГОУВПО «Воронежский государственный
технический университет»

М.Ю. Сергеев

РАЗРАБОТКА WEB-САЙТОВ:
ПРАКТИКУМ

Утверждено Редакционно-издательским советом
университета в качестве учебного пособия

Воронеж 2010

УДК 681.3

Сергеев М.Ю. Разработка Web-сайтов: практикум: учеб. пособие / М.Ю. Сергеев. Воронеж: ГОУВПО «Воронежский государственный технический университет», 2010. 235 с.

В учебном пособии рассматриваются технологии разработки Web-сайтов с применением визуальных редакторов, языка гипертекстовой разметки HTML и каскадных таблиц стилей CSS.

Издание соответствует требованиям Государственного образовательного стандарта высшего профессионального образования по направлению 230100 «Информатика и вычислительная техника», специальности 230101 «Вычислительные машины, комплексы, системы и сети», дисциплине «Web-дизайн».

Учебное пособие предназначено для студентов первого курса очной формы обучения и третьего курса очной сокращенной формы обучения.

Табл. 1. Ил. 45. Библиогр.: 8 назв.

Научный редактор д-р техн. наук, проф. С.Л. Подвальный

Рецензенты: кафедра информационных и управляющих систем Воронежской государственной технологической академии (д-р техн. наук, проф. В.Ф. Лебедев);
д-р техн. наук, проф. Т.М. Леденева

© Сергеев М.Ю., 2010

© Оформление. ГОУВПО «Воронежский государственный технический университет», 2010

ВВЕДЕНИЕ

Одной из наиболее актуальных и важных областей применения информационных технологий является разработка web-страниц и web-сайтов различной тематики, направленности и степени сложности.

Главной целью данного практикума является изложение теоретического материала и практических заданий для выполнения лабораторных работ по дисциплине «Web-дизайн».

В первых главах рассматриваются принципы разработки web-сайтов с применением визуальных web-редакторов. В качестве инструментария разработки используется программа Adobe Dreamweaver CS3. Изложены вопросы разработки отдельных элементов web-страниц, принципы и методы создания web-страниц на основе шаблонов и табличной разметки.

Далее рассматриваются методы разработки web-страниц с использованием тегов языка гипертекстовой разметки HTML. Дано описание тегов, отвечающих за оформление заголовка и тела документа, текста, гиперссылок, таблиц, изображений и т.д.

Заключительные главы практикума посвящены разработке сайтов с использованием каскадных таблиц стилей CSS. Рассматриваются способы управления селекторами стилей, свойства стилей, отвечающие за оформление различных элементов web-страницы. Также приведены методы и способы создания web-страниц на основе блочной разметки.

Полученные теоретические и практические навыки позволят разрабатывать статическое содержимое web-страниц и web-сайтов с использованием современных методов и способов web-дизайна.

Практикум соответствует типовой программе по дисциплине «Web-дизайн» и предназначен для студентов очной полной и сокращенной форм обучения.

1. СОЗДАНИЕ МНОГОСТРАНИЧНОГО САЙТА СРЕДСТВАМИ ADOBE DREAMWEAVER

1.1. Запуск Adobe Dreamweaver и начальная настройка параметров

Запуск программы осуществляется стандартным образом: **Пуск, Программы, Adobe Dreamweaver CS3**. На экране появляется главное окно программы (рис. 1).

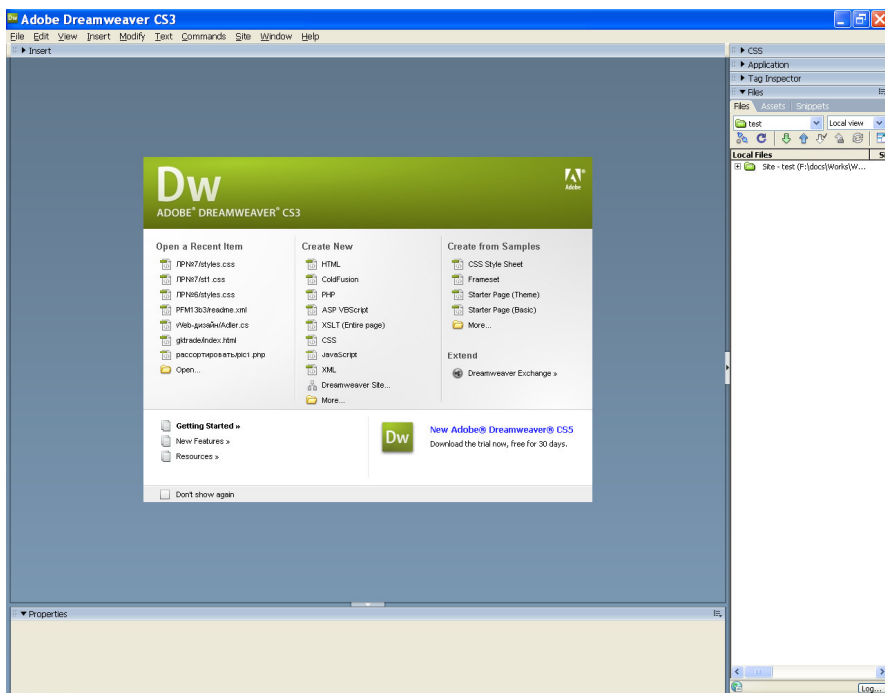


Рис. 1. Главное окно программы Adobe Dreamweaver

В верхней части окна находится главное меню программы. В средней части окна находится так называемый стартовый экран, с помощью которого можно быстро создать новый доку-

мент или открыть уже существующий. При желании данное окно можно отключить, включив опцию **Don't show again** в нижней части окна.

Перед созданием страниц web-сайта следует настроить параметры создаваемых web-страниц. Для этого следует выбрать пункты меню **Edit, Preferences** или нажать комбинацию клавиш CTRL+U. Далее в списке вкладок следует выбрать пункт New Document, после чего откроется окно настройки параметров создаваемого web-документа (рис.2).

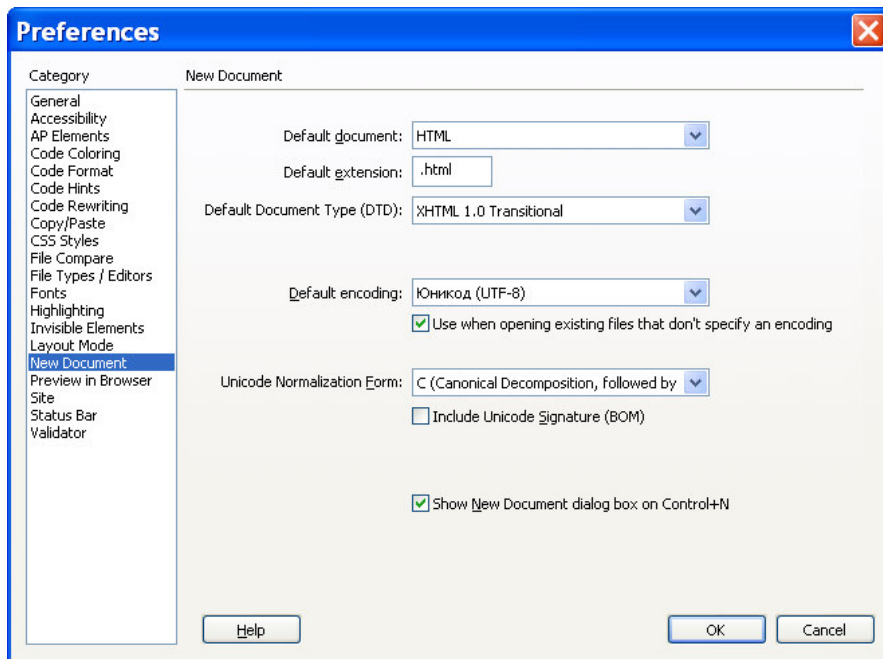


Рис. 2. Окно настройки параметров создаваемого web-документа

В списке **Default Document Type** следует выбрать тип создаваемого web-документа. Обычно это XHTML 1.0 Transitional или HTML 4.01 Strict. В списке Default Encoding

следует выбрать кодировку. Если это XHTML-документ следует выбрать Юникод (UTF-8), если это HTML-документ на русском языке, то – Кириллица (Windows). После выбора всех необходимых параметров следует щелкнуть мышью по кнопке ОК.

Для создания новой web-страницы через стартовое окно следует выбрать из списка **Create New** вариант **HTML**.

Если создавать новую страницу через главное меню следует выбрать пункты **File, New...** (также можно нажать комбинацию клавиш CTRL+N). После этого появится диалоговое окно **New Document** (рис. 3).

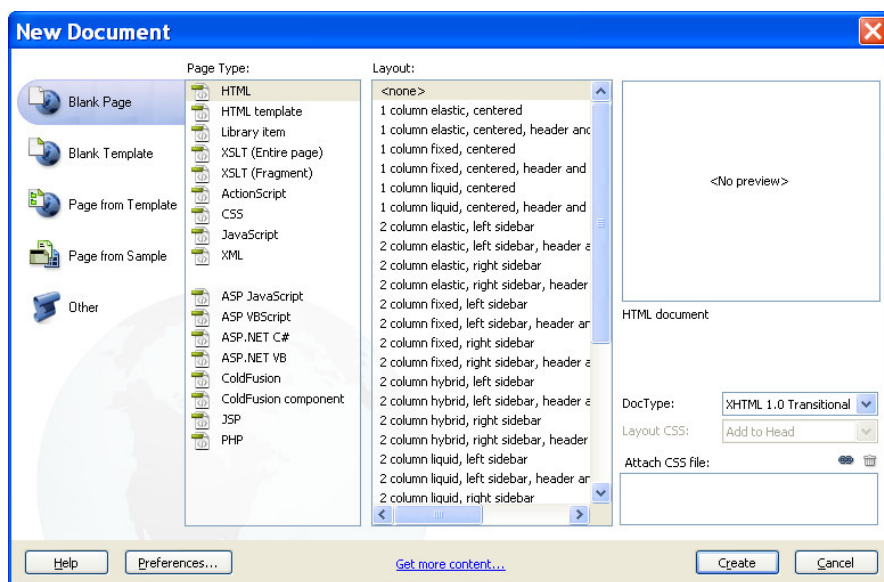


Рис. 3. Диалоговое окно New Document

В диалоговом окне **New Document** во вкладке **Blank Page** следует выбрать в списке **Page Type** опцию **HTML** и щелкнуть мышью по кнопке **Create**. На экране появится окно нового документа (рис. 4).

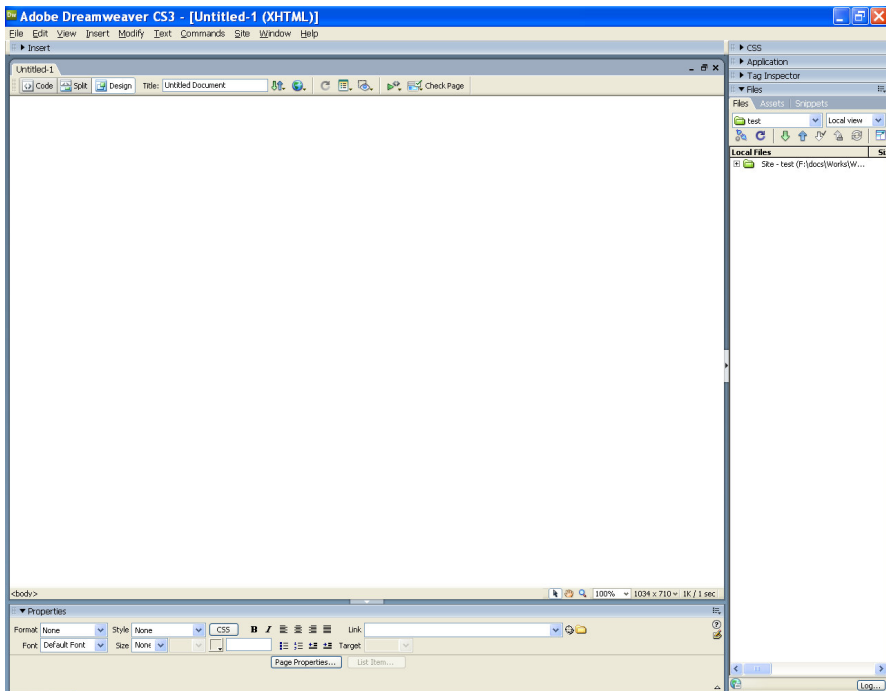


Рис. 4. Окно нового документа

Открытие для правки уже существующей web-страницы осуществляют с помощью пунктов меню **File, Open** или комбинации клавиш CTRL+O.

Для сохранения страницы выбирают пункты меню **File, Save**. Для сохранения всех открытых страниц выбирают пункт меню **File, Save All**.

Для закрытия текущей страницы выбирают пункты меню **File, Close**.

В верхней части окна находится панель управления текущим документом (рис. 5).

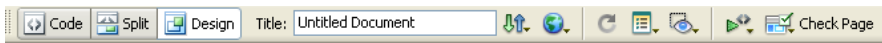




Рис.5. Панель управления текущим документом

Основными на данной панели являются следующие элементы:

- кнопки **Code**, **Split** и **Design**. Они отвечают за переключения вида отображения текущей web-страницы в окне редактора. Если щелкнуть по кнопке **Code**, то страница отобразится в режиме кода (т.е. будет показан текст HTML-кода web-страницы). Если щелкнуть по кнопке **Design** web-страница будет отображена в режиме дизайна (здесь можно набирать и форматировать текст страницы, а также добавлять визуальные элементы). В режиме **Split** страница показывается в обоих режимах одновременно.

- текстовое поле Title. В нем можно задать заголовок документа в окне браузера.

- кнопка . Позволяет просмотреть полученный документ в одном из браузеров. Для этого следует щелкнуть мышью по данной кнопке и в раскрывшемся списке выбрать нужный браузер (или нажать клавишу F12). Если нужного браузера в списке нет, то следует выбрать опцию Edit Browser List..., после чего откроется окно редактирования списка браузеров (Рис. 6).

В данном окне следует щелкнуть мышью по кнопке  и в открывшемся окне проводника указать путь к исполняемому файлу браузера. После того, как браузер выбран, можно сделать его запускающимся по умолчанию (т.е. по нажатию клавиши F12), включив опцию **Primary browser**. После окончания редактирования списка браузеров следует закрыть окно, щелкнув мышью по кнопке ОК.

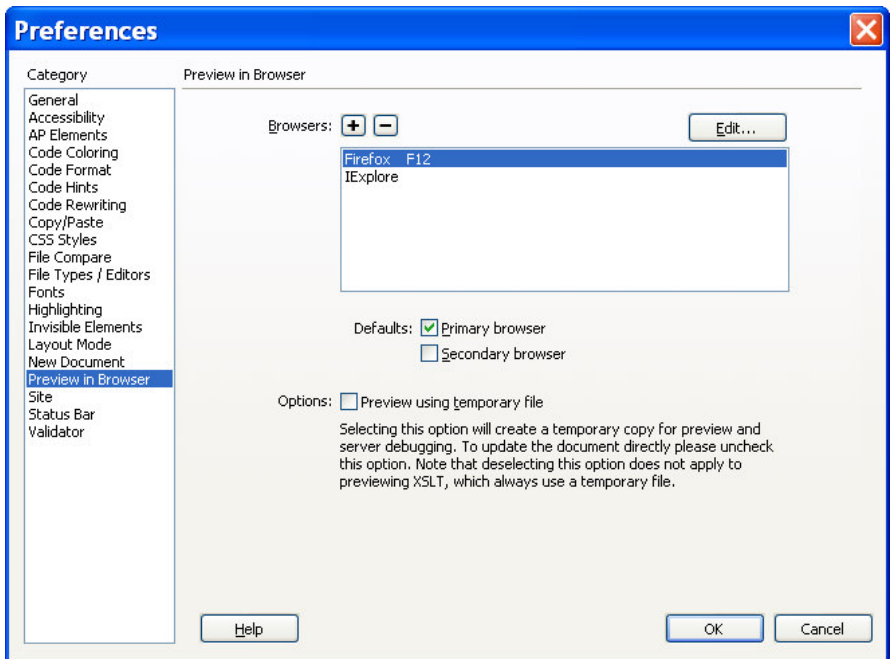


Рис.6. Окно редактирования списка браузеров

В верхней части окна также находится раскрывающаяся по щелчку мышью панель Insert, которая в основном содержит визуальные компоненты для размещения на странице. На рис. 7 приведен пример данной панели, открытой на вкладке Common.

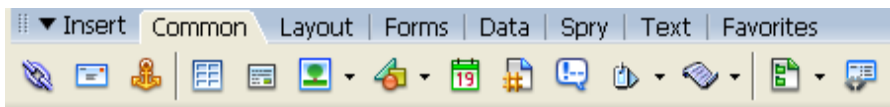


Рис.7. Панель компонентов Insert, открытая на вкладке Common.

В нижней части окна находится панель свойств элементов страницы (рис. 8).

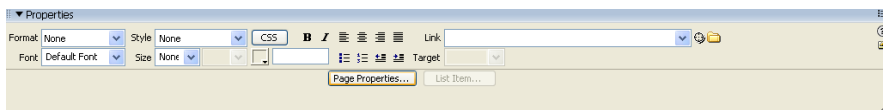


Рис.8. Панель свойств элементов страницы

Состав данной панели зависит от текущего выбранного элемента (текст, таблица, изображение и т.д.).

1.2. Настройка общих свойств страницы

К общим свойствам страницы следует отнести цвет текста и фона (или ссылка на файл с фоновым изображением), кодировку, заголовок, а также цвет гиперссылок.

Для того, чтобы обратиться к свойствам страницы следует щелкнуть мышью по кнопке **Page Properties...** на панели свойств в нижней части страницы. Откроется окно, представленное на рис. 9.

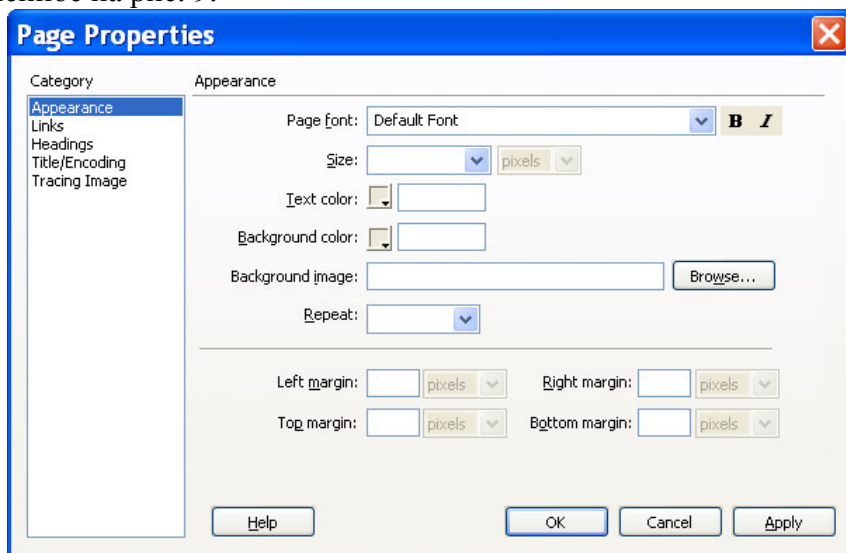


Рис.9. Панель свойств страницы, открытая на вкладке Appearance

Во вкладке **Appearance** можно указать параметры для текста и фона страницы:

- **Page Font** – стиль и гарнитура шрифта страницы;
- **Size** – размер шрифта;
- **Text color** – цвет текста;
- **Background color** – цвет фона;
- **Background image** – ссылка на файл с фоновым изображением.

Вкладка **Links** позволяет задать свойства гиперссылок (рис. 10).

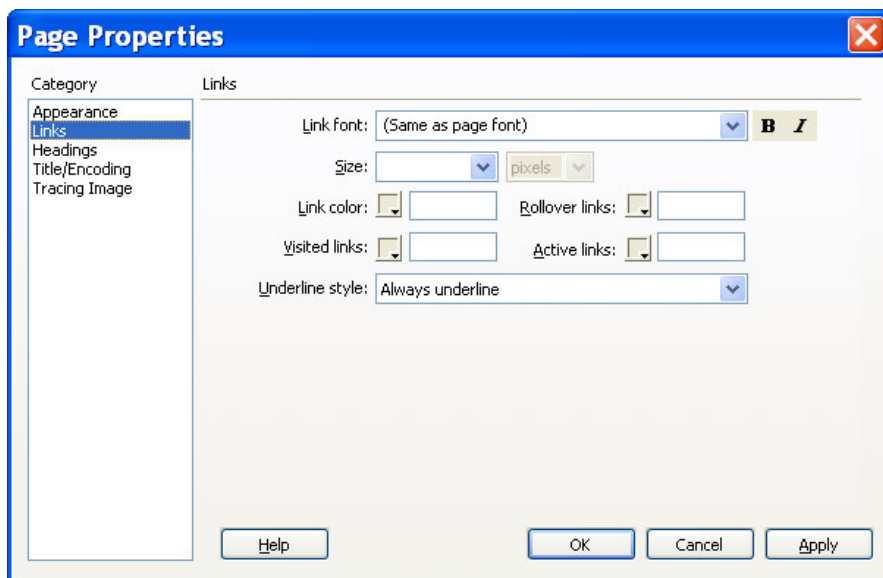


Рис.10. Панель свойств страницы, открытая на вкладке Links

В данной вкладке можно задать следующие параметры:

- **Link Font** – стиль и гарнитура шрифта ссылок;
- **Size** – размер шрифта гиперссылок;
- **Link color** – цвет обычных ссылок;

- **Visited links** – цвет посещенных ссылок;
- **Rollover links** – цвет ссылок при наведении на них курсора мыши;
- **Active links** – цвет активных ссылок.
- **Underline style** – стиль подчеркивания. Может принимать следующие значения: **Always underline** – ссылки подчеркиваются всегда; **Never underline** – ссылки никогда не подчеркиваются; **Show underline only on rollover** – ссылки подчеркиваются при наведении; **Hide underline on rollover** – подчеркивание ссылок исчезает при наведении на них курсора мыши.

Вкладка **Title/Encoding** позволяет задать заголовок, тип и кодировку документа (рис. 11).

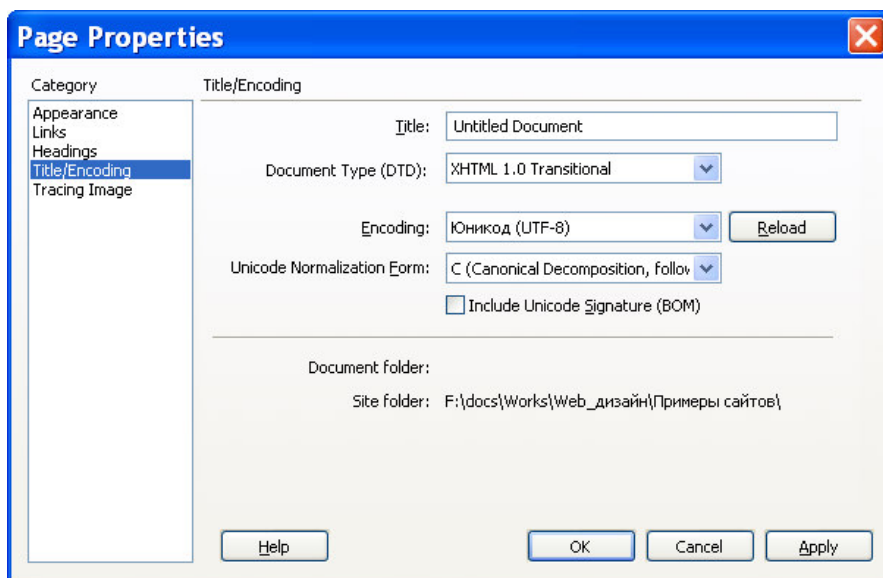


Рис. 11 Панель свойств страницы, открытая на вкладке Title/Encoding

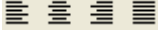
В данной вкладке можно задать следующие параметры:

- **Title** – заголовок документа в окне браузера;

- **Document Type (DTD)** – тип синтаксиса документа (HTML или XHTML);
- **Encoding** – кодировка документа (при изменении кодировки документа, в котором уже есть какое-то содержимое, следует щелкнуть мышью по кнопке **Reload** после выбора новой кодировки).

1.3. Оформление текста и гиперссылок

Чтобы отформатировать **текстовый абзац** используют панель свойств элемента страницы (**Properties** – рис. 8) и список **Format** на данной панели (или выбирают пункты меню **Text, Paragraph Format**, или пункт контекстного меню, **Paragraph Format**).


Для выравнивания абзаца на странице используют кнопки выравнивания на панели свойств элемента  или выбирают пункты меню **Text, Align**. Там можно выбрать следующие значения:

Left – выравнивание текста по левому краю;

Center – выравнивание текста по центру;

Right – выравнивание текста по правому краю;

Justify – выравнивание текста по ширине.


Для **создания списков** сначала выделяют строки, которые войдут в список. Затем щелкают на панели свойств элемента по кнопке маркированного списка (левая кнопка) или по кнопке нумерованного списка (правая кнопка) . В меню **Text** и в контекстном меню предусмотрено специальное подменю **List**, выполняющее те же действия.

Есть возможность изменить некоторые дополнительные параметры списка. Для этого ставят курсор куда-нибудь в список и нажимают кнопку **List Item** на панели редактора свойств.

Начертание шрифта можно изменить с помощью пунктов меню **Text, Style** (жирный – **Bold**, курсив – **Italic**, с подчер-

киванием текста линией - **Underline**). Два первых стиля можно задать и через панель свойств с помощью кнопок **B I**.

Выбор шрифта и размера осуществляется с помощью раскрывающихся списков **Font** и **Size** на панели свойств. Задать шрифт можно также с помощью пункта меню **Text**, и его подпунктов **Font** и **Size**.

Выбор цвета текста осуществляют с помощью кнопки селектора цвета  на панели свойств. При щелчке по данной кнопке открывается палитра цветов, где можно выбрать нужный. После выбора в текстовом поле рядом появляется код цвета шрифта в виде шестнадцатиричной константы (рис. 12) Для смены цвета шрифта с помощью меню следует выбрать пункты **Text, Color**.

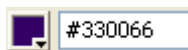


Рис. 12. Кнопка селектора цвета с выбранным цветом и текстовое поле с указанием кода цвета в виде шестнадцатиричной константы

Для **вставки горизонтальной линии** необходимо поставить курсор в начало абзаца, перед которым надо добавить линию и выбрать пункты меню **Insert, HTML, Horizontal Rule**.

Для вставки знака авторских прав (©) в место положения курсора на панели объектов **Insert** следует выбрать вкладку **Text** и щелкнуть по крайней справа во вкладке кнопке и из списка выбрать **Copyright** (Весь список специальных символов приведен на рис. 13). Другой вариант – выбрать пункты меню **Insert, HTML, Special Characters, Copyright**.

Для вставки разрыва строк можно использовать пункты меню **Insert, HTML, Special Characters, Line Break**, или выбрать нужный элемент из списка во вкладке **Text** (Рис. 13).

Для вставки неразрывного пробела можно использовать пункты меню **Insert, HTML, Special Characters, Non-Breaking**

Space, или выбрать нужный элемент из списка во вкладке Text (рис. 13).

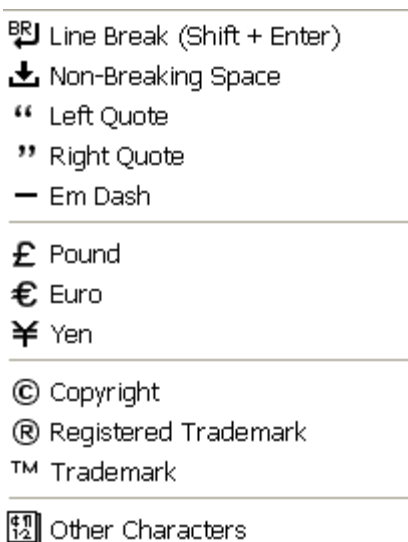




Рис. 13. Список специальных символов из вкладки Text панели компонентов Insert

Для преобразования текста в **гиперссылку** необходимо: выделить текст, который будет гиперссылкой; на панели свойств **Properties** в поле **Link** необходимо ввести web-адрес сайта или адрес другой страницы сайта (в конце поля **Link** находится кнопка , которая позволяет открыть диалоговое окно **Select File**. В данном окне необходимо выбрать файл web-страницы, на которую создается гиперссылка и нажать кнопку ОК) и нажать клавишу Enter.

Почтовая гиперссылка создается следующим образом: выделяют текст, который будет указывать на почтовый адрес (например, фамилия автора); в панели свойств в поле **Link** вводят почтовый адрес в формате `mailto:sergeev@somemail.ru` и нажимают клавишу Enter.

Для вставки **якоря** (метки или закладки, на которую можно сослаться с помощью гиперссылки) на панели объектов **Insert** следует выбрать вкладку **Common** и щелкнуть по кнопке  (или выбрать пункты меню **Insert, Named Anchor**). В результате этих действий на экране появится диалоговое окно **Named Anchor** (рис. 14).

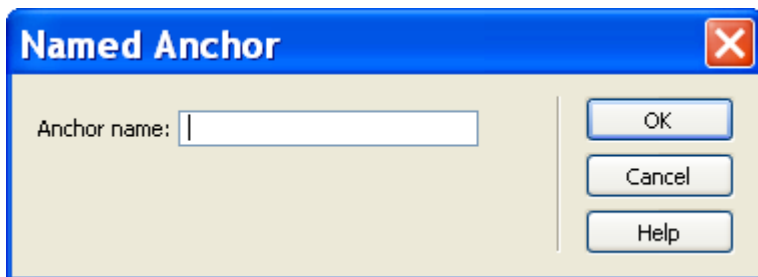



Рис. 14. Диалоговое окно ввода имени якоря

В поле ввода **Anchor Name** вводится имя вставляемого якоря, например, `par1`. Имена якорей должны состоять из латинских букв и цифр, причем первым символом должна быть буква.

Для создания гиперссылки на якорь нужно сделать следующее: выделить фрагмент текста, который будет гиперссылкой; в поле адреса **Link** редактора свойств **Properties** ввести имя якоря с символом «#» впереди, например, `#par1`. Значок «решетки» (#) означает, что эта гиперссылка переходит не на другую страницу, а на якорь.

1.4. Оформление таблиц

Для создания таблицы на панели объектов **Insert** следует выбрать вкладку **Common** и щелкнуть по кнопке  (или выбрать пункты меню **Insert, Table**). Откроется окно настройки параметров таблицы (рис. 15).

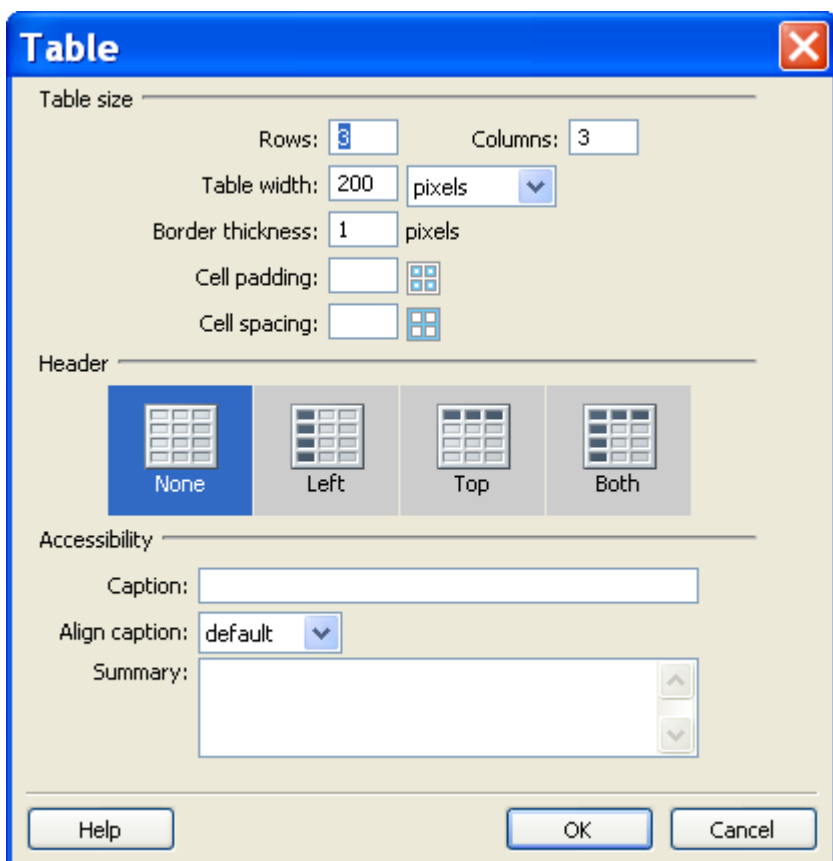



Рис. 15. Диалоговое окно настройки параметров таблицы

В полях ввода **Rows** и **Columns** вводится, соответственно, количество строк и столбцов создаваемой таблицы. В поле **Table Width** задается ширина таблицы (справа в раскрывающемся списке можно выбрать единицу измерения – пиксели или проценты). В поле ввода **Border thickness** задается толщина границ таблицы в пикселях. По умолчанию она равна 1; если ввести 0, то границы будут убраны совсем. В поле ввода **Cell padding** задается расстояние между границей ячейки таблицы и

ее содержимым в пикселях. Поле ввода **Cell spacing** служит для задания расстояния между границами отдельных ячеек.

Для добавления новой строки в уже имеющуюся таблицу помещают курсор в ячейку строки, над которой будет добавлена новая и выбирают пункты меню **Modify, Table, Insert Row**. Для добавления столбца ставят курсор в ячейку, слева от которой будет добавлен новый столбец. Затем выбирают пункты меню **Modify, Table, Insert Column**.

Для удаления текущей строки выбирают пункты меню **Modify, Table, Delete Row**. Для удаления текущего столбца выбирают пункты меню **Modify, Table, Delete Column**.

Для разделения ячейки таблицы: выбирают пункты меню **Modify, Table, Split Cell** или на панели свойств щелкают по кнопке разделения ячейки . В любом случае на экране появится диалоговое окно **Split Cell** (рис. 16).

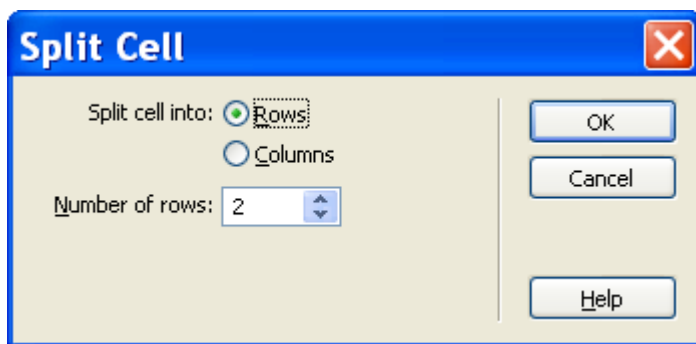



Рис. 16. Диалоговое окно разбиения ячейки таблицы

Группа переключателей **Split cell into** задает, как будет делиться ячейка. Переключатель **Rows** задает разделение текущей строки на несколько строк, количество которых устанавливается в поле счетчика **Number of rows**. Переключатель **Columns** задает, что текущий столбец будет делиться на несколько столбцов, количество которых задается в поле счетчика

Number of columns. После нажатия кнопки ОК текущая ячейка будет разделена.

Для слияния ячеек необходимо их выделить и щелкнуть по кнопке слияния ячеек , которая расположена слева внизу на панели свойств. Данный способ используют как для слияния ячеек по горизонтали, так и по вертикали. Вместо нажатия кнопки можно выбрать пункты меню **Modify, Table, Merge Cells**.

Для разъединения ячеек ставят курсор в объединенную ячейку и осуществляют следующее: если выполнялось слияние по горизонтали, выбирают пункты меню **Modify, Table, Decrease Column Span**; если выполнялось слияние по вертикали, выбирают пункты меню **Modify, Table, Decrease Row Span**.

Чтобы выделить всю таблицу, ставят курсор в любую ячейку таблицы и выбирают пункты меню **Modify, Table, Select Table**.

Чтобы изменить параметры ячейки, ставят курсор в ячейку и с помощью панели свойств **Properties** задают нужные параметры:

- раскрывающийся список **Horz** задает горизонтальное выравнивание содержимого ячейки. Можно выбрать значения **Left** (выравнивание влево), **Center** (выравнивание по центру), **Right** (выравнивание вправо);

- раскрывающийся список **Vert** задает вертикальное выравнивание содержимого ячейки. Можно выбрать значения **Top** (выравнивание по верхней границе ячейки), **Middle** (выравнивание по центру ячейки), **Bottom** (выравнивание по нижней границе ячейки) и **Baseline** (выравнивание по базовой линии ячейки);


- поля ввода **W** и **H** предназначены для ввода ширины и высоты ячейки соответственно;

- флажок **No Wrap** запрещает перенос строк внутри ячейки. В этом случае ячейка растягивается по ширине, чтобы вме-

стить весь имеющийся в ней текст, который в этом случае будет «вытянут» в одну строку;

- флажок **Header** позволит отформатировать ячейку, в которой находится текстовый курсор, как ячейку заголовка таблицы;

- поле ввода **Bg** служит для задания имени файла фонового изображения ячейки;

- селектор цвета **Bg**  (не путать с одноименным полем ввода) позволяет задать цвет фона ячейки таблицы;

- селектор цвета **Brdr** позволяет задать цвет границы ячейки.

Общий вид панели **Properties** со свойствами выделенной ячейки приведен на рис. 17.

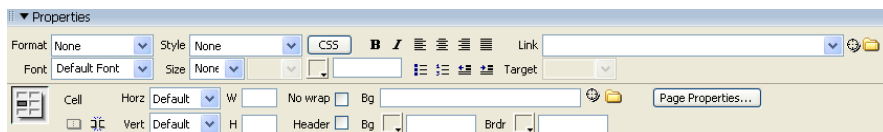


Рис. 17. Панель Properties со свойствами выделенной ячейки

Чтобы задать параметры таблицы, предварительно таблицу необходимо выделить. Параметры таблицы задаются с помощью панели редактора свойств:

- поля ввода **Rows** и **Cols** служат для быстрого изменения количества, соответственно, строк и столбцов таблицы;

- поле ввода **W** служит для задания ширины таблицы;

- поле ввода **Bg Image** служит для задания имени файла фонового изображения для всей таблицы;

- селектор цветов **Bg Color** задает цвет фона таблицы,

- селектор цветов **Brdr Color** задает цвет границ таблицы (если для ячейки заданы особые параметры для текста и фона, то они будут иметь преимущество);

- поле ввода **CellPad** служит для задания расстояния между границей ячейки и ее содержимым в пикселях;

- поле ввода **CellSpace** задает расстояние между границами находящихся рядом ячеек в пикселях;
- поле ввода **Border** служит для задания толщины границы ячеек в пикселях;
- раскрывающийся список **Align** служит для задания горизонтального выравнивания таблицы. Можно выбрать значения **Left** (выравнивание по левому краю), **Center** (выравнивание по центру) или **Right** (выравнивание по правому краю).

Общий вид панели **Properties** со свойствами выделенной таблицы приведен на рис. 18.

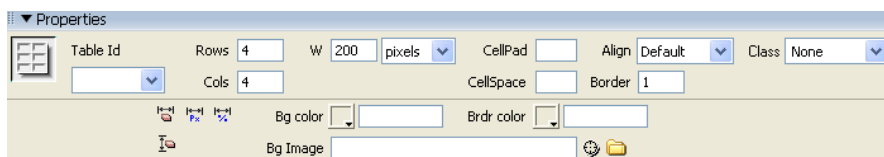



Рис. 18. Панель Properties со свойствами выделенной таблицы

1.5. Вставка изображений

Для вставки графического изображения следует поставить курсор в то место, где будет изображение, и на панели компонентов **Insert** во вкладке **Common** щелкнуть по кнопке . Откроется окно проводника, где можно выбрать файл с изображением. Можно также выбрать пункты меню **Insert, Image**.

Параметры изображения задаются с помощью панели редактора свойств (естественно, предварительно изображение нужно выделить):

- поля ввода **W** и **H** позволяют задать ширину и высоту изображения соответственно;
- поля ввода **V Space**, **H Space** задают соответственно вертикальное и горизонтальное расстояние от края изображения до обтекающего его текста. По умолчанию оба они равны нулю;

– поле ввода **Src** задает имя файла, где хранится графическое изображение;

– поле ввода **Low Src** аналогично полю **Src**, за тем исключением, что задает имя файла, где сохранено так называемое «черновое» изображение. «Черновое» изображение имеет меньший размер, как правило, за счет большего сжатия и низкого качества;

– поле ввода **Border** позволяет задать толщину рамки, отображаемой вокруг изображения – по умолчанию она равна нулю, т.е. рамки нет;

– поле ввода **Alt** задает так называемый «альтернативный текст» (текст, который будет появляться вместо изображения, если в свойствах браузера отключен показ изображений на веб-странице);

– раскрывающийся список **Align** позволяет задать выравнивание изображение и обтекание изображения текстом. Среди значений можно отметить следующие – **Left** (изображение смещается влево, текст обтекает его справа) и **Right** (изображение смещается вправо, текст обтекает его слева).

Общий вид панели **Properties** со свойствами выделенного изображения приведен на рис. 19.

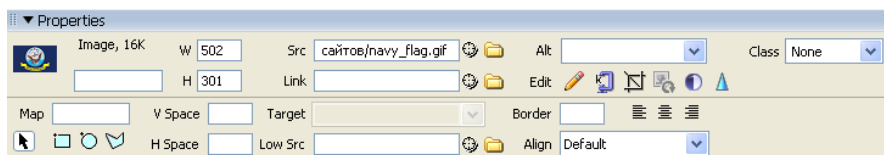




Рис. 19. Панель Properties со свойствами выделенного изображения

Для создания **изображения-гиперссылки** нужно выделить изображение и на панели свойств в поле ввода **Link** ввести нужный адрес (или с помощью кнопки  рядом с полем Link выбрать нужный файл).

Карта-изображение – это сложная графическая гиперссылка, представляющая собой изображение, разбитое на части, причем каждая часть является гиперссылкой и указывает на свой web-адрес.

Порядок создания:


– выделить изображение;

– с помощью группы из трех кнопок  на панели свойств и курсора мыши разместить на изображении прямоугольную, круглую и многоугольную область – будущую гиперссылку;

– на панели свойств в поле ввода **Map** появится строка «Map» (это имя изображения – гиперссылки; можно дописать к нему цифру, если областей будет несколько);

– в поле ввода **Link** ввести адрес ссылки.

– повторить все предыдущие этапы для каждой создаваемой области.

Для вставки **активного изображения** следует выбрать пункты меню **Insert, Image Objects, Rollover Image**. (или щелкнуть по кнопке  в панели компонентов **Insert** во вкладке **Common**. Откроется диалоговое окно создания активного изображения (рис. 20).

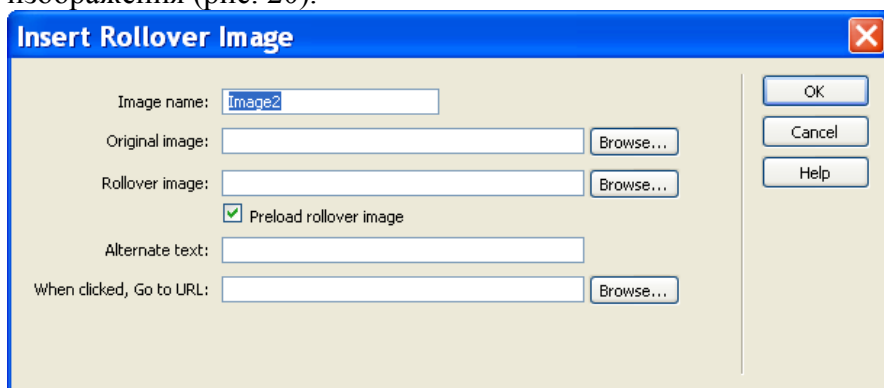


Рис. 20. Диалоговое окно создания активного изображения

В поле ввода **Image name** вводится уникальное имя вставляемого активного изображения (например, Image1). В поле ввода **Original image** вводится имя файла оригинального (первого) изображения. Имя файла вводится вручную или ищется с помощью кнопки **Browse** (она открывает окно проводника). В поле ввода **Rollover image** вводится имя файла перекрывающего изображения (т.е. которое будет появляться вместо оригинального изображения после наведения на него курсора мыши). Имя файла вводится вручную или ищется с помощью кнопки **Browse**. Флажок **Preload rollover image** заставляет Dreamweaver сгенерировать и вставить в HTML-код страницы небольшую программу-сценарий, которая заранее загрузит перекрывающее изображение. Этот флажок включен по умолчанию, и отключать его не стоит. В поле ввода **Alternate text** вводится альтернативный текст. Последнее поле ввода **When clicked, Go to URL** задает web-адрес, по которому произойдет переход, если пользователь щелкнет на изображении (или указывается адрес другой страницы сайта с помощью кнопки **Browse**).

1.6. Задание на лабораторную работу

1. Создать папку **Site** на сетевом диске **X:** (или каком-нибудь другом) в папке с Вашими фамилиями.
2. Запустить **Dreamweaver**.
3. Проверить настройки кодировки для создаваемых документов.
4. Создать четыре пустые web-страницы, сохранить их под разными именами и заполнить их содержимым.
5. Первая web-страница содержит следующий текст:

Турфирма «Воронежтур»
***Туристическая фирма «Воронежтур» приглашает за
путевками по России и зарубежью!***

Вставить нижеследующую таблицу и выровнять ее по центру страницы.

Вставить изображение	Туры по России Зарубежные туры О фирме
----------------------	---

На картинке сделать две ссылки: часть картинки – ссылка на страницу «Туры по России»; другая часть картинки – ссылка на страницу «Зарубежные туры».

Из текста «Туры по России» сделать гиперссылку на вторую страницу. Из текста «Зарубежные туры» сделать ссылку на третью страницу. Из текста «О фирме» сделать ссылку на четвертую страницу.

Добавить строку:

© *Сайт создан студентами группы ... перечислить фамилии и инициалы.*

6. Вторая web-страница содержит следующий текст:

Туры по России
Приглашаем в путешествия по России!

Вставить изображение и выровнять его по центру страницы, вставить всплывающую надпись при наведении мыши на изображение: «Студентам скидки при покупке турпутевок по России!».

Вставить следующую таблицу и выровнять ее по центру страницы.

Название тура, продолжительность и стоимость	Предлагаемые услуги	Ближайшие даты
Санкт-Петербург, 5 дней, 6550 руб.	<ul style="list-style-type: none">• проезд на автобусе,	<ul style="list-style-type: none">• 13.09.2010• 20.09.2010

	<ul style="list-style-type: none"> • экскурсии, • проживание в гостинице. 	<ul style="list-style-type: none"> • 27.09.2010
Москва , 2 дня, 3500 руб.	<ul style="list-style-type: none"> • проезд на автобусе; • экскурсии; • проживание в гостинице. 	<ul style="list-style-type: none"> • 04.08.2010 • 09.09.2010 • 10.09.2010
Золотое кольцо , 3 дня, 7500 руб.	<ul style="list-style-type: none"> • проезд на автобусе; • проживание в гостинице; • экскурсии. 	<ul style="list-style-type: none"> • 13.08.2010 • 17.09.2010 • 15.10.2010

Добавить гиперссылку на главную страницу – текст «**На главную**».

7. Третья web-страница содержит следующий текст:

Зарубежные туры

Вставить следующую таблицу с нулевой границей.

Активное изображение (ссылка на главную страницу)	<p><i>Фирма предлагает следующие туры:</i></p> <ul style="list-style-type: none"> • <i>Египет</i> • <i>Париж</i> • <i>Прага</i>
--	--

Из слов «Египет», «Париж», «Прага» сделать ссылки на информационные блоки о турах, которые пометить якорями. Например: слово «Египет» ссылается на якорь, стоящий перед фразой «Курорты Египта осенью» и т.д. Заголовок «Зарубеж-

ные туры» пометить якорем. На этот якорь ссылаются все фразы «Назад» в конце каждого информационного блока.

Курорты Египта осенью!

Изображение	Изображение
<p>Египет – одна из самых посещаемых туристами стран мира, влекущая своим загадочным прошлым, хранящая неповторимый дух культуры древнейшей цивилизации в истории человечества. Разные народы, населявшие эту землю на протяжении веков, оставили свой след в истории страны, оказав свое влияние на формирование ее культуры и традиций.</p>	
Изображение	Изображение

Название курорта, продолжительность поездки	Отель	Цена за номер	Цена за одного человека
Хургада, 4 дня	Minamark 3*	690 USD	345 USD
Шарм-Эль-Шейх, 4 дня	Regency Hotel 3*	776 USD	388 USD

Назад

Осенние каникулы во Франции!

Изображение	Изображение
<p>Что такое Франция? Может быть, Франция - это страна, подарившая миру дружбу четырех неразлучных мушкетеров, преданность Жанны Д'Арк и величие Наполеона? Или Франция – это прежде всего Париж, парижане и парижанки? Возможно Франция – это изысканные ароматы духов и лучшие кутюрье? Со всем этим согласится каждый из Вас.</p>	
Изображение	Изображение

Название тура	Ближайшие даты	Продолжительность	Цена
Тур "Париж + Бургундия" Париж + Бургундия	01.10, 21.10, 05.11, 25.11	8 дней / 7 ночей	от 1380 евро
Тур "Страна замков" Париж + Долина Луары	24.09, 14.10, 29.10, 18.11	8 дней / 7 ночей	от 1390 евро
Тур "Прекрасная Франция" Париж + Долина Луары + Нормандия + Бретань	17.09, 07.10, 22.10, 11.11	8 дней / 7 ночей	от 1215 евро

Назад

Осенние туры в Прагу!

Изображение	Изображение
<p>Прага — столица Чехии. Один из красивейших городов не только Чехии, но и Европы. Удивительные памятники архитектуры и прекрасное месторасположение над рекой Влтавой наложили на Прагу отпечаток неповторимого очарования и атмосферы, которыми так восторгаются все ее посетители.</p>	
Изображение	Изображение

Название тура	Продолжительность, дни	Вылеты, дни недели	Стоимость
Романтическая Прага	8	1,6,7	от 409 евро

Классическая Прага	8	4,5,6	от 415 евро
Злата Прага	8	1,2,3,4	от 390 евро

Назад

8. Четвертая web-страница содержит следующий текст:

О фирме

«15 лет на рынке туристических услуг! Проверено временем!»

Вставить следующую таблицу и выровнять ее по центру страницы.

Вставить активное изображение – гиперссылку на первую страницу.	<u>Адреса фирмы:</u>
	1. Ул. Кирова, 3, тел. 53-99-53.
	2. Ул. Кропоткина, 17, тел. 56-90-56.
	3. Московский пр., 82, тел. 39-90-39.

Льготные цены! Скидки!

Вставить фразу «**На главную**» и сделать ее гиперссылкой на главную страницу.

9. На каждой странице:

- настроить цвет текста и фона страницы;
- настроить цвета обычных, посещенных и активных ссылок;
- выровнять таблицы по центру, по вкусу – задать таблицам и их границам свой цвет.

2. СОЗДАНИЕ WEB-САЙТА С ИСПОЛЬЗОВАНИЕМ ТАБЛИЧНОГО ДИЗАЙНА И ШАБЛОНОВ В DREAMWEAVER

2.1. Создание web-сайта и настройка его параметров

При грамотном подходе к разработке web-сайта следует не создавать разрозненный набор web-страниц, а сразу проектировать сайт как единое целое. Для этого в Dreamweaver предусмотрена возможность создавать web-сайт как некий объект и настраивать его параметры.

Для создания нового или открытия уже существующего сайта следует выбрать пункт меню **Site, Manage Sites....** На экране появится окно управления сайтами (рис. 21).

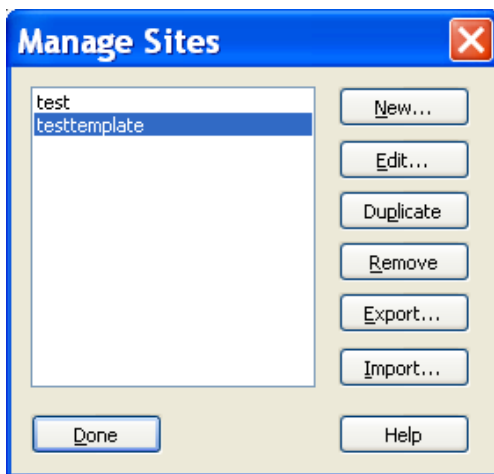


Рис 21. Окно управления сайтами

Для создания нового сайта следует щелкнуть мышью по кнопке **New...** и выбрать из списка опцию **Site**. Для открытия уже существующего следует выбрать из списка сайтов нужное имя и щелкнуть мышью по кнопке **Done**.

Также для создания нового сайта можно выбрать пункт меню **Site, New Site...** Вне зависимости от способа, на экране появится окно **Site Definition** настройки параметров нового сайт (рис. 22). В данном окне следует перейти во вкладку **Basic** для пошаговой настройки параметров с помощью мастера.

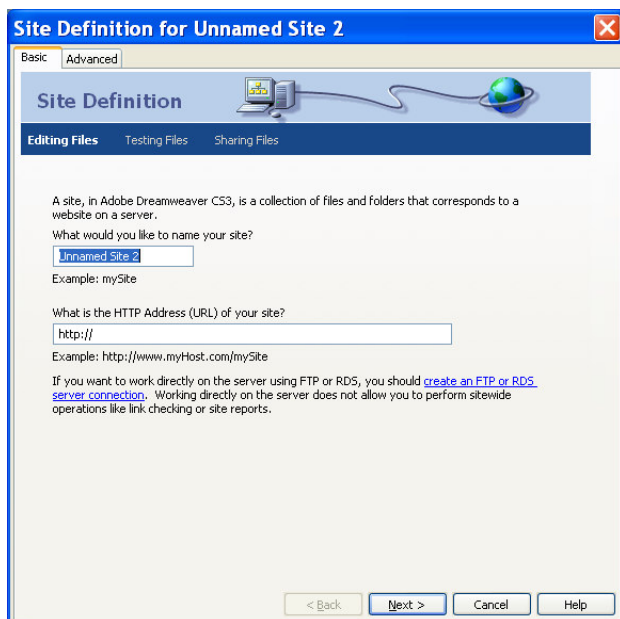


Рис. 22. Окно настройки параметров нового сайта Site Definition (вкладка Basic)

В нем необходимо осуществить следующие настройки:

- **первое окно** – задать имя сайта (например, **MySite**) и нажать кнопку **Next**;
- **второе окно** – выбрать опцию **No, I do not want to use a server technology** и нажать кнопку **Next**;
- **третье окно** – выбрать опцию **Edit local copies on my machine...** и указать каталог хранения файлов сайта (например, созданный специально каталог **MySite**) и нажать кнопку **Next**;

– **четвертое окно** – в поле со списком **How do you connect to your remote server?** выбрать **None** и нажать кнопку **Next**;

– **пятое окно** – посмотреть обзор настроенных свойств сайта и выбрать закладку **Advanced**, пункт **Local Info**, поле **Default images folder** и указать каталог, куда будут сохраняться изображения, помещенные на сайт (например, каталог **Pics**, созданный внутри корневого каталога сайта).

В итоге окно **Advanced** должно иметь примерно следующий вид (рис. 23).

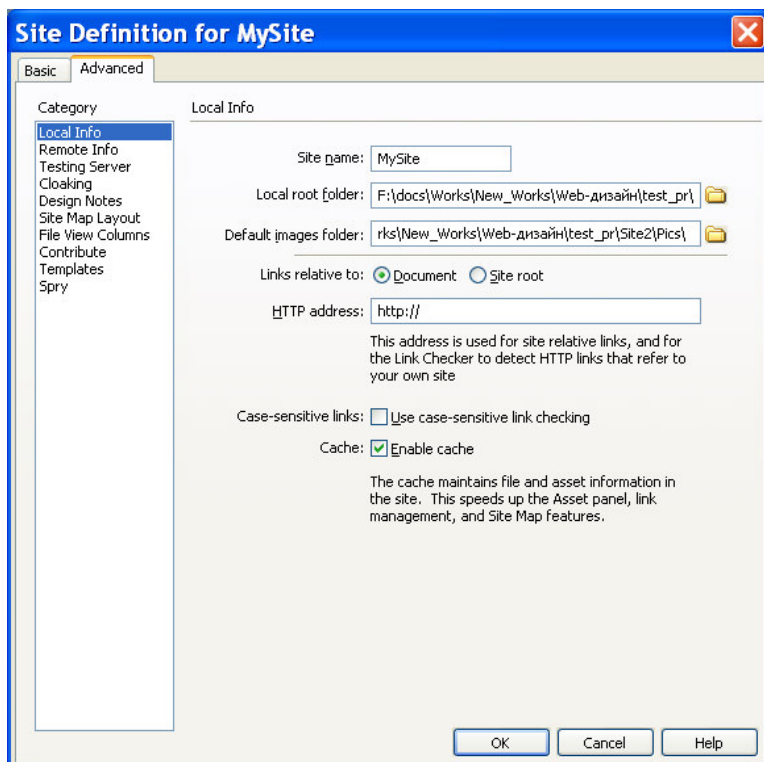


Рис. 23. Окно **Advanced** с итоговыми настройками сайта

В каталог, указанный в поле **Local root folder** по умолчанию будут сохраняться создаваемые web-страницы.

В каталог, указанный в поле **Default images folder** по умолчанию будут сохраняться изображения, размещаемые на web-страницах.

Теперь следует нажать кнопку **OK**. В окне **Manage Sites...** тоже необходимо нажать кнопку **Done**.

Теперь можно создавать новые страницы для сайта. Структура сайта со всеми элементами будет отображаться на правой панели **Local files** окна Dreamweaver.

2.2. Создание web-страницы с помощью табличной разметки и сохранение ее как шаблона

Страницы сайта должны иметь по возможности одинаковое стилистическое оформление. Для этого нужно грамотно спроектировать структуру типовой страницы сайта и потом использовать ее как шаблон для создания всех остальных web-страниц.

Для начала нужно создать пустую страницу и сохранить ее под каким-либо именем (например, default.html).

Для облегчения дальнейшей работы следует осуществить следующие шаги:

- включить линейку разметки страницы (пункт меню **View, Rulers, Show**);

- включить координатную сетку (пункт меню **View, Grid, Show Grid**).

- включить привязку к координатной сетке (пункт меню **View, Grid, Snap To Grid**).

- установить размер шага сетки 25 пикселей (пункт меню **View, Grid, Grid Settings**).

В итоге окно Dreamweaver должно принять примерно следующий вид (рис. 24).

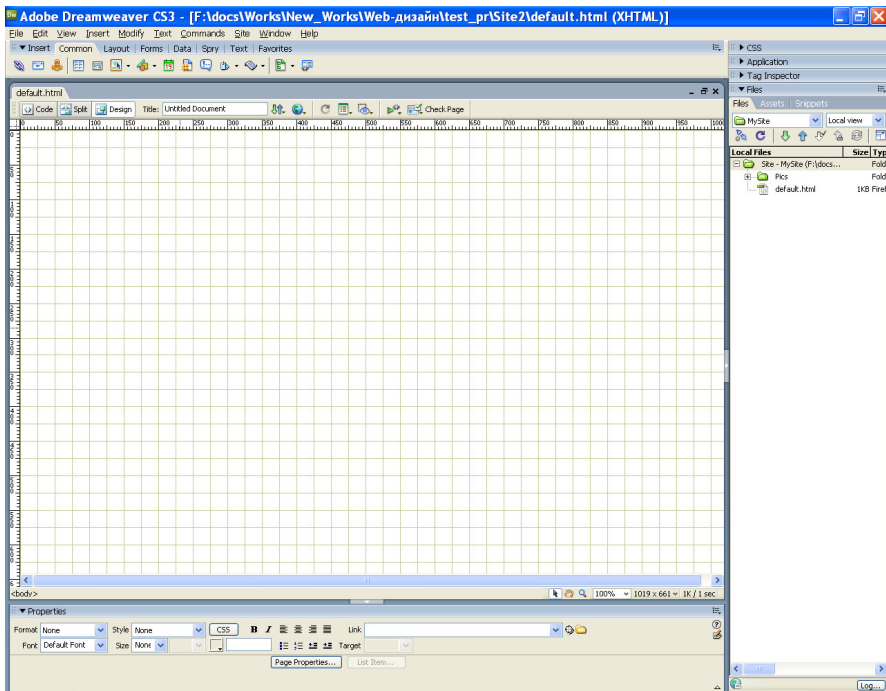



Рис. 24. Окно Dreamweaver со включенной линейкой и сеткой для разметки страницы


Теперь следует перейти в режим табличной разметки страницы. Для этого следует верхней части окна сменить вкладку **Common** на вкладку **Layout** на панели компонентов **Insert**. Также следует включить режим **Layout Mode**, выбрав пункт меню **View, Table Mode, Layout Mode**. Если будет выведено предупреждающее окно, следует нажать кнопку **Done**.

Теперь с помощью кнопки **Draw layout cell**  следует расчертить структуру страницы. В качестве базового, можно придерживаться варианта, представленного на рис. 25.

Заголовок	
Навигация	Основное содержание
Сведения об авторах	

Рис. 25. Типовая структура web-страницы

Следует нарисовать вышеуказанные ячейки (пока без надписей, но примерно соблюдая пропорции) на странице. Зазоров и пустых мест между ячейками не оставлять – иначе на странице могут появиться белые полосы между элементами.

Прорисовка осуществляется просто – щелчок по кнопке , после чего курсор принимает крестообразную форму. Теперь, зажав левую кнопку мыши, можно отчертить ячейку. Для прорисовки следующей ячейки следует повторить операцию.

Чтобы обратиться к свойствам ячейки, следует щелкнуть левой кнопкой мыши по границе ячейки. Для каждой ячейки с помощью селектора цвета Bg следует задать цвет фона ячейки.

Ячейку «Основное содержание» следует сделать авторастягивающейся. Для этого необходимо выделить данную ячейку и в панели свойств Properties выбрать опцию **Autostretch**. При этом на экран будет выведено диалоговое окно **Choose Spacer Image** [Выбор изображения-заполнителя], в котором следует выбрать переключатель **Create a spacer image file** и нажать **OK**. После этого сохранить предложенный файл **spacer.gif** в каталог, предназначенный для изображений, размещаемых на сайте).

В итоге текущая web-страница примет примерно следующий вид (рис. 26).

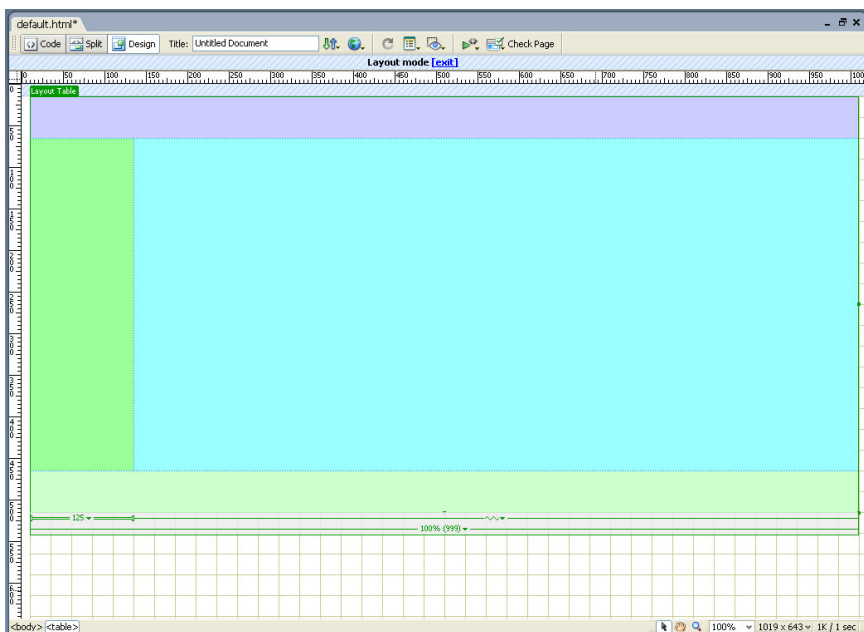


Рис.26. Web-страница с табличной разметкой

Далее необходимо заполнить те элементы страницы, которые будут одинаковы для всех web-страниц сайта (например, «Заголовок» и «Сведения об авторах»). Ячейку «Навигация» заполнять пока не следует, так как отсутствуют страницы, на которые будут ввести ссылки.

Теперь следует сохранить данную страницу как шаблон. Для этого необходимо выбрать пункт меню **File, Save as Template...** Откроется диалоговое окно сохранения шаблона (рис. 27).

В данном окне в раскрывающемся списке **Site** следует выбрать название своего сайта. В нижней части окна в текстовом поле **Save as** необходимо указать название шаблона.

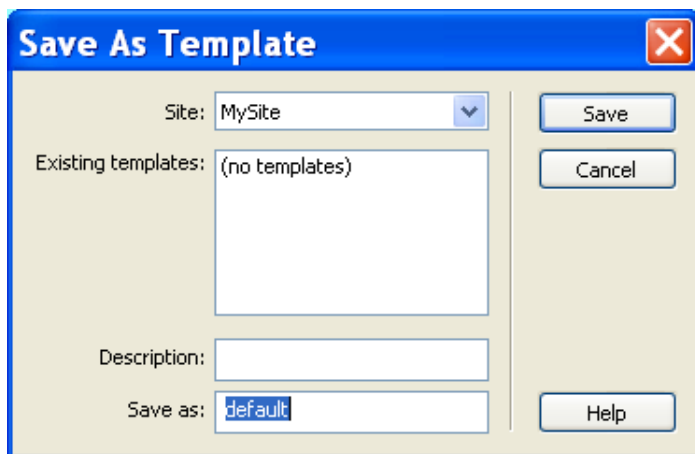


Рис. 27. Диалоговое окно сохранения шаблона

После ввода требуемых данных, можно щелкнуть мышью по кнопке **Save**. Если будет выведено небольшое окно-предупреждение, о том, что шаблон не содержит изменяемых областей, следует проигнорировать его, щелкнув мышью по кнопке ОК.

Теперь сохраненный файл получит расширение `.dwt`, которое указывает, что он может быть использован как шаблон Dreamweaver для создания web-страниц. Для него будет специально создана папка **Templates** внутри корневого каталога сайта, где будут размещаться все создаваемые шаблоны для web-страниц.

На данный момент шаблон не содержит важного элемента – изменяемых областей – специальных зон шаблона, в который может помещаться специфическое содержимое web-страниц, создаваемых на основе этого шаблона.

В данном конкретном случае нужно создать изменяемую область внутри ячейки «Основное содержание» шаблона. Для этого следует поместить курсор мыши внутри данной ячейки и

выбрать пункт меню меню **Insert, Template Objects, Editable region**. Откроется окно создания изменяемой области (рис. 28).

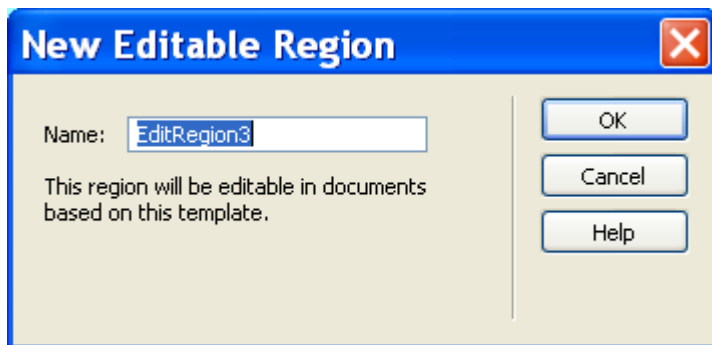


Рис. 28. Окно создания изменяемой области

В данном окне следует задать имя изменяемой области и щелкнуть мышью по кнопке ОК. Изменяемая область добавится в ячейку «Основное содержимое» и шаблон примет примерно следующий вид (рис. 29).

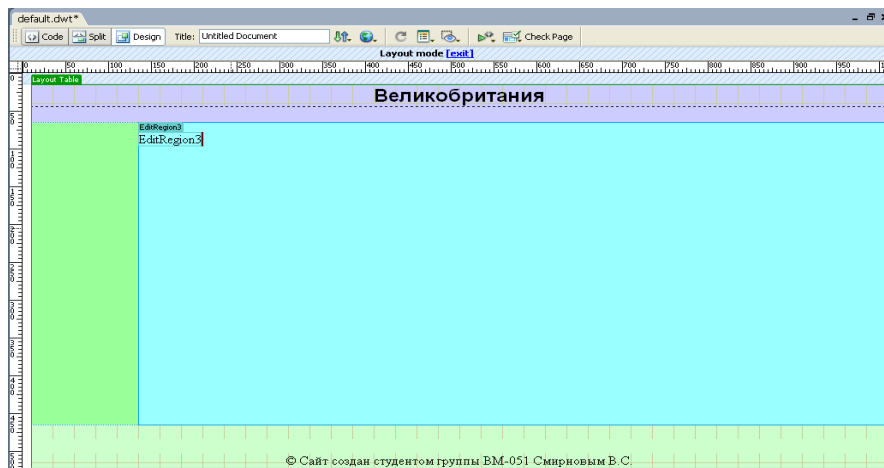


Рис. 29. Шаблон с добавленной изменяемой областью

Теперь по данному шаблону можно создать все web-страницы сайта.

Для создания страницы следует выбрать пункт меню **File, New....** Откроется окно создания нового документа в виде, представленном на рис. 3. В нем следует перейти на вкладку **Page from Template** (рис. 30).

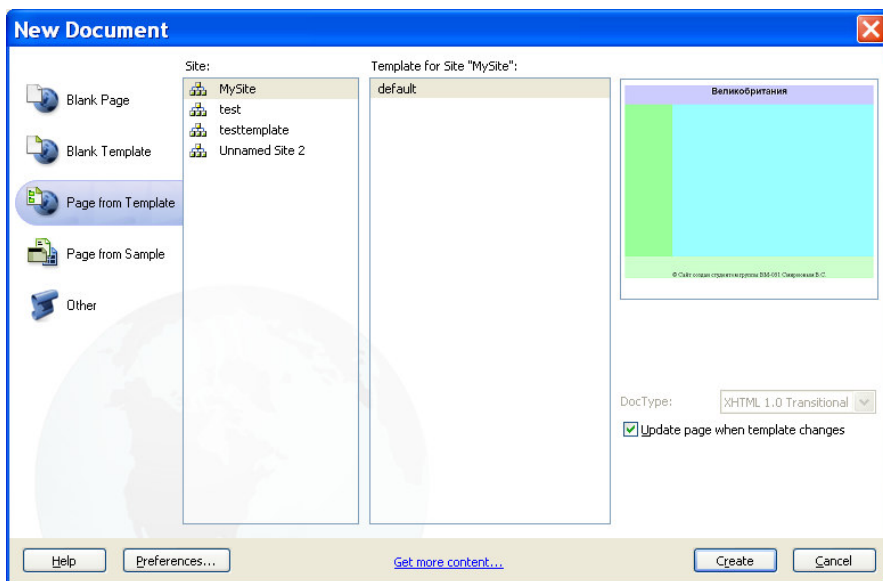


Рис. 30. Окно создание нового документа с активной вкладкой **Page from Template**

В данной вкладке в списке **Site** необходимо выбрать имя своего сайта, а в списке **Template for Site...** – имя своего шаблона. Обязательно следует оставить включенной опцию **Update page when template changes** – чтобы связанные с шаблоном web-страницы автоматически обновлялись при изменении шаблона. После выбора всех нужных параметров можно щелкнуть мышью по кнопке **Create** – и новая страница будет создана. Останется только сохранить ее под каким-либо именем.

Аналогичные операции следует проделать для всех остальных страниц сайта.

2.3. Разработка панели навигации

При большом количестве страниц сайта возникает проблема организации навигации между ними. Для осуществления связи между страницами одного уровня полезно создать панель навигации в соответствующем разделе страниц сайта.

Если речь идет о страницах, созданных на основе шаблона, то задача облегчается – требуется создать панель навигации в шаблоне, а потом сохранить его, параллельно автоматически обновив все страницы сайта.

Для создания навигационной панели следует перейти на шаблон, поставить курсор в ячейку «Навигация» и выбрать пункты меню **Insert, Image Objects, Navigation Bar**. Появится диалоговое окно вставки навигационной панели (рис. 31).

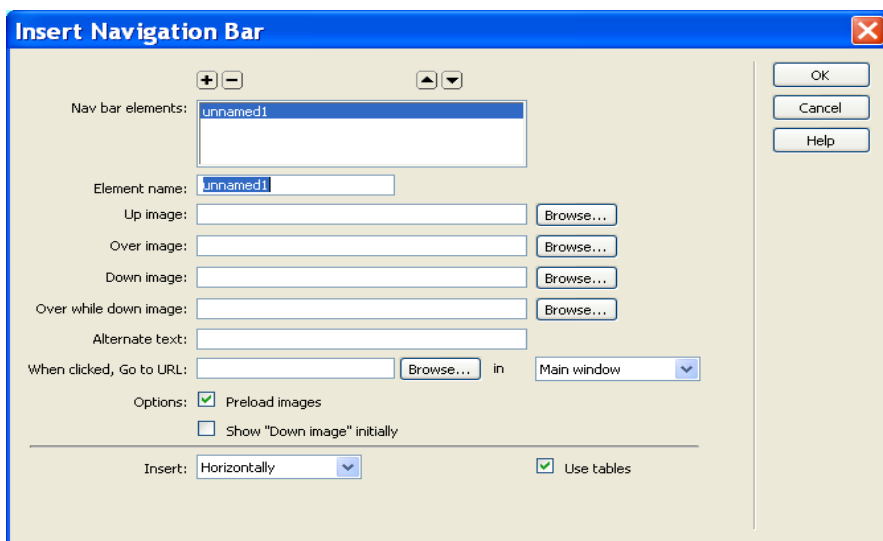


Рис. 31. Окно вставки навигационной панели

Для добавления новых элементов в панель используются кнопки **+** и **-**, для изменения порядка размещения элементов – кнопки **▲** и **▼**. Для каждого элемента следует задать следующие свойства:

- **Element name** – имя элемента (например page2, page3 или page4 – русские символы в названиях не допускаются);
- **Up image** – исходное изображение (с помощью кнопки **Browse...** следует указать нужный файл);
- **Over Image** – изображение, на которое меняется исходное при наведении курсора мыши (с помощью кнопки **Browse...** следует указать нужный файл);
- **When clicked, Go to URL** – указать куда следует перейти при щелчке на элементе.
- **Insert** – каким образом будут располагаться элементы во вставляемой панели: горизонтально или вертикально (рекомендуется выбрать **Vertically**, т.е. вертикально).

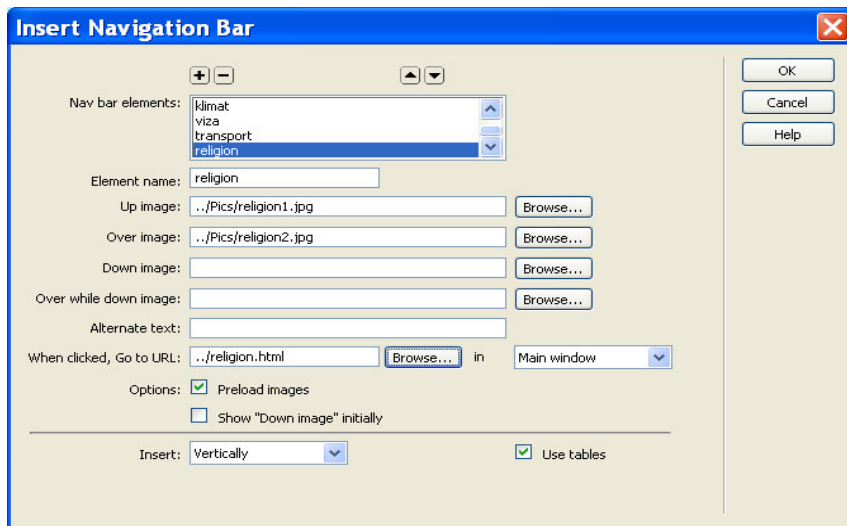


Рис.32. Окно вставки навигационной панели с заданными параметрами

Пример окна с введенными параметрами представлен на рис. 32.

После ввода всех данных следует щелкнуть мышью по кнопке ОК. Панель навигации добавится на страницу шаблона (рис. 33).

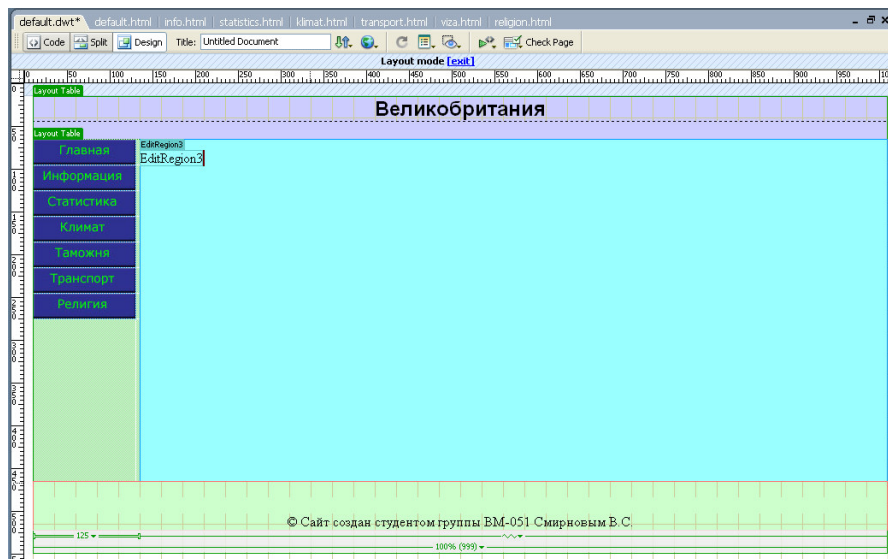


Рис. 33. Окно страницы шаблона со вставленной навигационной панелью

Если необходимо внести изменения в навигационную панель, то следует еще раз выбрать пункты меню **Insert, Image Objects, Navigation Bar**. Поскольку в шаблоне может быть только одна навигационная панель, будет предложено отредактировать уже существующую и откроется окно, приведенное на рис. 32, в котором можно менять настройки навигационной панели (кроме свойства **Insert** – его изменить уже нельзя).

После вставки навигационной панели следует сохранить страницу шаблона. При этом будет предложено обновить все страницы сайта, созданные на основе данного шаблона (рис 34).

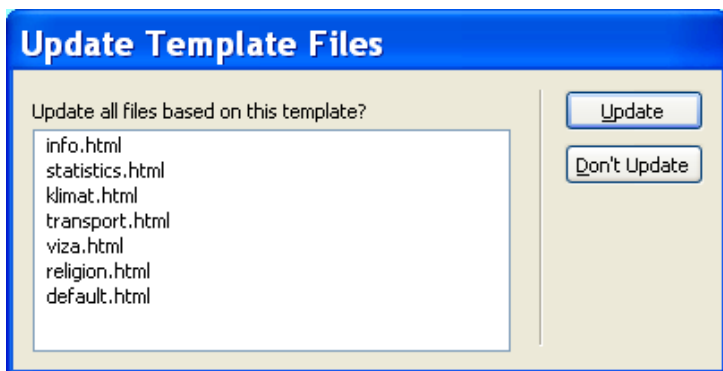


Рис. 34. Окно с предложением обновления связанных с шаблоном web-страниц

Для подтверждения обновления следует щелкнуть мышью по кнопке **Update**. На экране появится окно отчета об обновлении (рис. 35).

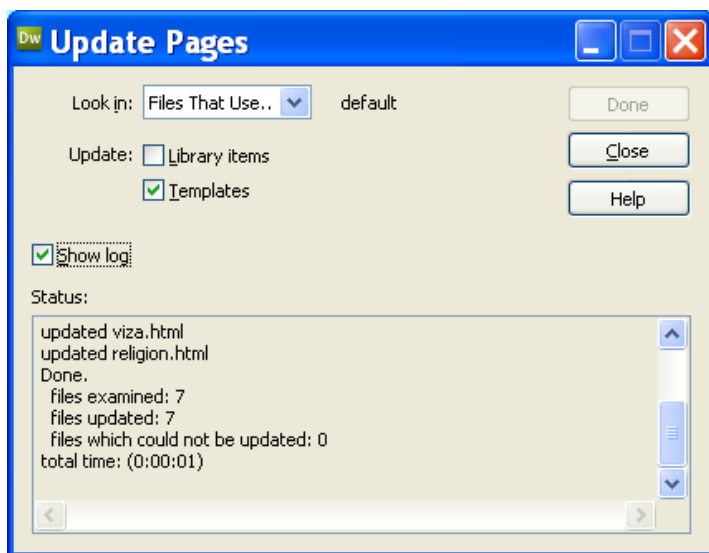


Рис. 35. Окно с отчетом об обновлении страниц

Если в отчете в строке **files which could not be updated** стоит цифра 0, то все страницы успешно обновлены. Чтобы сохранить все обновленные страницы, следует выбрать пункт меню **File, Save All**.

Теперь можно заполнять страницы сайта, добавляя в изменяемую область нужное содержимое. Пример одной из готовых страниц сайта приведен на рис. 36.



Рис. 36. Пример готовой web-страницы, разработанной на основе шаблона

2.4. Задание на лабораторную работу

1. На одном из локальных или сетевых дисков создать папку с Вашими фамилиями, затем в этой папке создать папку **MySite**. В папке **MySite** создать папку **Pictures**.
2. Запустить **Dreamweaver**.
3. Создать новый сайт и настроить его параметры (в качестве корневого каталога указать папку **MySite**, в качестве каталога для изображений указать папку **Pictures**).

4. Создать новую пустую страницу и сохранить ее под именем default.html.

5. Настроить отображение координатной сетки и линейки, а также перейти в режим табличной разметки.

6. Создать на странице несколько ячеек разметки: заголовок, основное содержание, навигацию и сведения об авторах (см. рис. 25). Настроить фон ячеек, ячейку «Основное содержание» сделать авторастягивающейся.

7. Заполнить ячейки заголовка и сведений об авторах согласно заданию (см. пункт 2.5).

8. Сохранить созданную страницу как шаблон.

9. В ячейке основного содержания создать изменяемую область.

10. На основе шаблона создать страницы со следующими именами:

- default.html;
- info.html;
- statistics.html;
- klimat.html;
- viza.html;
- valyuta.html;
- transport.html;
- religion.html.

11. Вернуться к шаблону и создать в ячейке навигации навигационную панель. Сохранить изменения в шаблоне и обновить все страницы.

12. Заполнить на всех страницах ячейки основного содержания согласно заданию (см. пункт 2.5).

13. Сохранить все созданные страницы.

14. Проверить работу сайта в каком-либо браузере.

Замечание:

Для каждого из вариантов работы существует каталог с материалами для заполнения страниц сайта. В нем находятся:

– файл «Информация.doc» - содержит текст для добавления на страницы сайта;

- файлы flag.gif, gerb.gif и karta.gif – изображения флага, герба и карты страны соответственно;
- каталог Pictures – содержит тематические изображения для размещения на страницах сайта.

2.5. Задание по заполнению страниц

1. Ячейка «Заголовок» имеет следующее содержание:

[Название страны – согласно варианту задания]

2. Ячейка «Сведения об авторах» имеет следующее содержание:

Сайт создан студентами группы ... [Группа, ФИО авторов]

3. Ячейка «Основное содержание» главной страницы (**default.html**) имеет следующий вид:

[Изображение]

На нашем сайте Вы найдете много полезной информации о [Название страны – согласно варианту задания], которая пригодится Вам в путешествии в эту прекрасную страну:

- Общая информация о стране, ее населении и достопримечательностях
- Статистическая информация о стране
- Климатические условия
- Информация по оформлению виз и перечень таможенных ограничений
- Валюта
- Виды транспорта и график его движения
- Обычаи религия населения страны

Все элементы списка (подчеркнутые) должны быть гиперссылками на соответствующие страницы.

4. Ячейка «Навигация» имеет следующую структуру – должна быть вставлена навигационная панель со следующими кнопками:

Главная
Информация
Статистика
Климат
Таможня
Валюта
Транспорт
Религия

Кнопка **Главная** - это ссылка на главную страницу **default.html**. Кнопка **Информация** – это ссылка на вторую страницу **info.html**. Кнопка **Статистика** – это ссылка на третью страницу **statistics.html**. Кнопка **Климат** – это ссылка на четвертую страницу **klimat.html**. Кнопка **Таможня**– это ссылка на пятую страницу **viza.html**. Кнопка **Валюта** - это ссылка на шестую страницу **valyuta.html**. Кнопка **Транспорт** - это ссылка на седьмую страницу **transport.html**. Кнопка **Религия** – это ссылка на восьмую страницу **religion.html**.

Замечания:

Картинки для кнопок находятся в папке **Buttons** (подкаталог **dark** – кнопки темного цвета, подкаталог **light** – светлые кнопки).

В свойствах навигационной панели **обязательно** указать вертикальное расположение панели (поле со списком **Insert** внизу окна – надо выбрать **Vertically**).

4. Ячейка основного содержания второй страницы «Общая информация» (**info.html**) имеет следующий вид:

Общая информация

[Изображение флага] [Изображение герба]

[Текст из файла Информация.doc – раздел Общая информация]

6. Ячейка основного содержания третьей страницы «Статистика» (**statistics.html**) имеет следующий вид:

Статистика

[Изображение карты страны]

[Текст из файла Информация.doc – раздел Статистика]

7. Ячейка основного содержания четвертой страницы «Климат» (**klimat.html**) имеет следующий вид:

Климат

[Изображение]

[Текст из файла Информация.doc – раздел Климат]

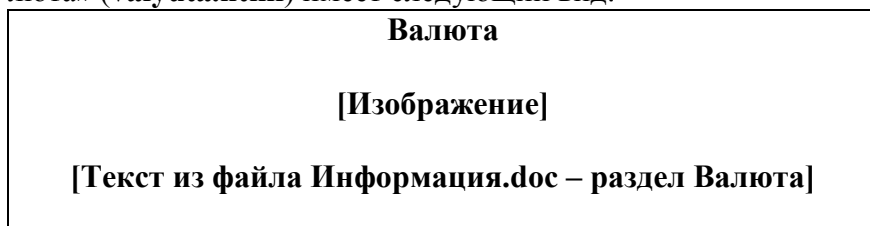
8. Ячейка основного содержания пятой страницы «Необходимые документы и таможенные ограничения» (**viza.html**) имеет следующий вид:

Необходимые документы и таможенные ограничения

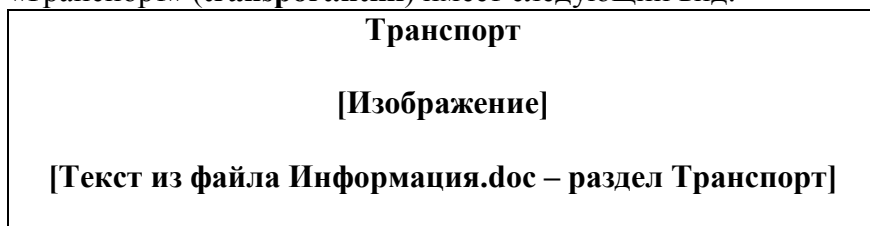
[Изображение]

[Текст из файла Информация.doc – раздел Необходимые документы и таможенные ограничения]

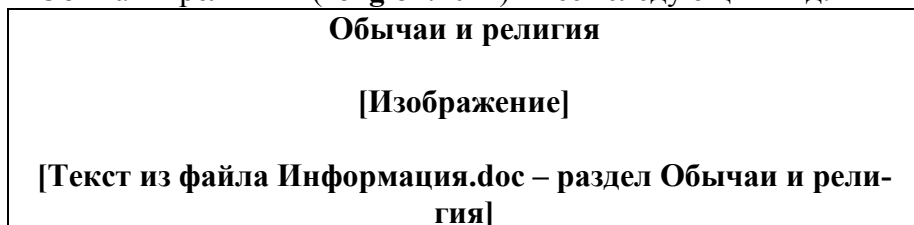
9. Ячейка основного содержания шестой страницы «Валюта» (**valyuta.html**) имеет следующий вид:



10. Ячейка основного содержания седьмой страницы «Транспорт» (**transport.html**) имеет следующий вид:



11. Ячейка основного содержания восьмой страницы «Обычаи и религия» (**religion.html**) имеет следующий вид:



3. СОЗДАНИЕ WEB-СТРАНИЦ С ПОМОЩЬЮ ТЕГОВ HTML

3.1. Структура HTML-документа

HTML –документ – это просто текстовый файл с расширением .html или .htm.

Язык HTML является описательным языком разметки документов, в нем используются указатели разметки (теги или метки). Теговая модель описывает документ как совокупность контейнеров, каждый из которых начинается и заканчивается (или не заканчивается) тегами. Например контейнер может начинаться с тега `<head>` и заканчиваться тегом `</head>`.

Чаще всего HTML-теги состоят из начального и конечного тега, между которыми размещаются текст и другие элементы документа. Имя конечного тега идентично имени начального, но перед именем конечного тега ставится косая черта (/). Например, для тега стиля шрифта – курсив начальный тег `<i>`, а конечный тег `</i>`.

Теги HTML –документов в большинстве своем просты и понятны, поскольку они образованы с помощью общеупотребительных слов английского языка. По своему значению теги близки к понятию скобок «begin/end» в универсальных языках программирования. Теги определяют область действия правил интерпретации текстовых документов.

Тег может иметь атрибуты с конкретными значениями, которые устанавливаются для изменения функции тега. Атрибуты тега следуют за начальным именем тега и отделяются друг от друга одним или несколькими знаками табуляции, пробелами или символами возврата к началу строки. Порядок записи атрибутов в теге не имеет значения. Значения атрибута, если таковое имеется, следует за знаком равенства, стоящим после имени атрибута. Если значение атрибута – одно слово или число, то его можно просто указать после знака равенства, не выделяя дополнительно. Все остальные значения необходимо заключать в одинарные или двойные кавычки, особенно если они содержат несколько слов, разделенных пробелами. Длина значения атрибута ограничена 1024 символами. Регистр символов в именах тегов и атрибутов не учитывается, чего нельзя сказать о значениях атрибутов.

Например, при описании таблицы открывающий тег с атрибутами может выглядеть так:

```
<table width=570 align=center cellpadding=10  
cellspacing=2 border=2>
```

Эта запись означает следующее: таблица шириной 570 пикселей, выровнена по центру, поле между рамкой и содержимым ячеек 10 пикселей, поле рамки 2 пикселя, ширина рамки 2 пикселя.

При использовании вложенных тегов в документе следует соблюдать аккуратность. Вложенные теги нужно закрывать, начиная с последнего тега.

Некоторые HTML-теги не имеют конечного компонента, поскольку являются автономными элементами. Например, тег изображения ``, который служит для вставки в документ графического изображения, конечного компонента не требует. К автономным тегам также относятся разрыв строки (`
`), горизонтальная линейка (`<hr>`) и теги, содержащие такую информацию о документе, которая не влияет на его отображаемое содержимое, например тег `<meta>`.

Общая схема построения контейнера в формате HTML может быть записана в следующем виде:

```
<имя тега список атрибутов>  
содержание контейнера  
</имя тега>
```

Кроме тегов, элементами HTML являются CER (Character Entity Reference), они предназначены для представления специальных символов в документе HTML.

Например, чтобы представить символ «<» в документе HTML, нужно заменить его на `<`, а символ «>» - `>`. Если указать в тексте HTML строку `<body>`, она будет выглядеть на экране как текст `<body>`.

В отличие от тегов HTML, наименования CER чувствительны к регистру символов.

Пример: задать авторский знак в тексте.

© Сайт создан студентом группы ВМ-041 Ивановым И.Ю.

Записать в тексте программы:

© Сайт создан студентом группы ВМ-041 Ивановым И.Ю.

Группы тегов HTML. Все теги HTML по их назначению и области действия можно разделить на следующие основные группы:

- определяющие структуру документа;
- оформление блоков гипертекста (параграфы, списки, таблицы, картинки);
- гипертекстовые ссылки и закладки;
- формы для организации диалога;
- вызов программ.

Вот самый простой HTML-документ:

```
<html>
```

```
<head>
```

```
<title>
```

```
    Пример 1
```

```
</title>
```

```
</head>
```

```
<body>
```

```
    <h1>
```

```
        Привет!
```

```
    </h1>
```

```
    <P>
```

```
        Это простейший пример HTML-документа.
```

```
    </P>
```

```
</body>
```

```
</html>
```

HTML-документ – это один большой контейнер, который начинается с тега `<html>` и заканчивается тегом `</html>`.

Контейнер HTML или гипертекстовый документ состоит из двух других вложенных контейнеров: заголовка документа (`head`) и тела документа (`body`).

3.2. Элементы разметки заголовка документа

Элемент разметки <head>. Элемент разметки head содержит заголовок HTML-документа. Данный элемент разметки не является обязательным. При наличии тега начала элемента разметки желательно использовать и тег конца элемента разметки. По умолчанию элемент head закрывается, если встречается либо тег начала контейнера body, либо тег начала контейнера frameset. Атрибутов у тега начала контейнера нет. Контейнер заголовка служит для размещения информации, относящейся ко всему документу в целом. В контейнер head вкладываются все остальные элементы разметки заголовка документа.

Элемент разметки <title>. Элемент разметки title служит для именованя документа в сети Интернет. Иными словами назначение элемента – именование окна браузера, в котором просматривается документ. Состоит контейнер из тега начала, содержания и тега конца. Наличие тега конца обязательно. Тег начала элемента не имеет специфических атрибутов.

При выборе текста для содержания контейнера title следует учитывать, что отображается он системным шрифтом, так как является заголовком окна браузера. В нелокализованных версиях операционных систем и графических оболочек русский текст содержания элемента title будет отображаться абракадаброй.

Синтаксис контейнера title в общем виде выглядит следующим образом:

```
<title> название документа </title>
```

Пример: задать заглавие документа в окне браузера – labour exchange.

```
<html>  
<head>  
<title> labour exchange </title>  
</head>  
...  
</html>
```

Заголовок не является обязательным контейнером документа. Его можно опустить. Однако многие поисковые системы используют содержание элемента title для создания поискового образа документа. Слова из title попадают в индекс поисковой системы. Из этих соображений элемент title всегда рекомендуется использовать на страницах web-узла.

Элемент разметки <meta>. meta содержит управляющую информацию, которую браузер использует для правильного отображения и обработки содержания тела документа.

В теге <meta> есть атрибуты практически совпадающие по назначению http-equiv и name, а также content. Для параметра name зарезервированы три наиболее используемых значения – author, description, keywords. После указания значения параметра name с помощью параметра content указывают данные для выбранной позиции. Например, для name =”author” в поле content указывают имя автора. description ответственен за описание страницы, за краткое содержание документа (многие поисковые машины проводят сортировку именно по этому тегу). keywords дает представление о темах, затрагиваемых в документе. Рекомендуется использовать и русские, и английские ключевые слова. Вот некоторые наиболее типичные meta-инструкции:

```
<meta name=”description” content=”Это руководство – учебник по Web-дизайну”>
```

```
<meta name=”keywords” content=”Интернет, HTML, WWW, руководство, публикация, гипертекст”>
```

Последняя инструкция определяет переменную keywords, содержащую набор ключевых слов, описывающих содержание документа.

Другая группа meta-инструкций определяет тип кодировки документа (ключевое слово – charset).

Следующая meta-инструкция дает браузеру указание интерпретировать загружаемый документ как содержащий HTML-текст в кодировке Windows-1251.

```
<meta http-equiv=”Content-type” content=”text/html;
```

```
charset=windows-1251">
```

Практика показывает, что при индексировании можно указывать и атрибут name, и атрибут http-equiv с одинаковыми значениями (но лучше в разных тегах meta). Это связано с тем, что одни роботы индексирования анализируют содержание meta-элемента по атрибуту name, а другие – по атрибуту http-equiv. Можно использовать эти атрибуты одновременно, но для разных целей. Например:

```
<head>  
<title> Основы Web-технологий</title>  
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">  
<meta name="keywords" content="HTML, meta, метатег, тег, поисковая система">  
</head>
```

Элемент разметки <style>. Данный тег предназначен для размещения описателей стилей.

В настоящее время контейнер используется только с одним атрибутом type, который задает тип описателя стиля. Это может быть либо text/css, либо text/javascript. В общем виде запись тега style выглядит так:

```
<style type=тип_описания_стилей>  
описание_стиля/стилей  
</style>
```

Элемент разметки <script>. Данный тег служит для размещения кода JavaScript, VBScript или Jscript. Если открыт тег начала, то нужно обязательно использовать тег конца контейнера.

В общем виде запись контейнера выглядит следующим образом:

```
<script [type=тип_языка_программирования][src=URL]>  
JavaScript/VBScript-код  
</script>
```

Существует несколько скриптовых языков: JavaScript, VBScript, Jscript. По умолчанию подразумевается JavaScript.

3.3. Определитель типа документа

Элемент `<!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>`.

Существует несколько видов `<!DOCTYPE>`, они различаются в зависимости от версии языка, на которого ориентированы. В таблице ниже приведены основные типы документов с их описанием.

Виды элемента `<!DOCTYPE>`

<code><!DOCTYPE></code>	Описание
<code><!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd"></code>	Строгий синтаксис HTML.
<code><!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd"></code>	Переходный синтаксис HTML.
<code><!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Frameset//EN" "http://www.w3.org/TR/html4/frameset.dtd"></code>	В HTML-документе применяются фреймы.
<code><!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd"></code>	Строгий синтаксис XHTML.
<code><!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"</code>	Переходный синтаксис

"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">	XHTML.
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Frameset//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-frameset.dtd">	Документ написан на XHTML и содержит фреймы.

Синтаксис

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

Параметры

Элемент верхнего уровня – указывает элемент верхнего уровня в документе, для HTML это тег <HTML>.

Публичность – объект является публичным (значение PUBLIC) или системным ресурсом (значение SYSTEM), 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: плюс (+) – разработчик зарегистрирован в ISO и - (минус) – разработчик не зарегистрирован. Для W3C значение ставится «-».

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

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

Имя – уникальное имя документа для описания DTD.

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

URL – адрес документа с DTD.

Закрывающий тег не требуется.

3.4. Теги для оформления тела документа

Теги тела документа предназначены для управления отображением информации в браузере пользователя. Тело документа состоит из:

- иерархических контейнеров и заставок;
- заголовков (от h1 до h6);
- блоков (параграфы, списки, формы, таблицы, картинки и т.п.);
- горизонтальных линий и адресов;
- текста, разбитого на области действия стилей (подчеркивание, выделение, курсив);
- математических описаний, графики и гипертекстовых ссылок.

Тело документа – контейнер body. Описание тегов тела документа следует начать с тега `body`, который имеет ряд атрибутов.

Атрибут `background` определяет фон, на котором отображается текст документа. Так, если источником для фона HTML-документа является графический файл `image.jpg`, то начальный тег будет выглядеть так:

```
<body background="image.jpg">
```

В качестве значения данного атрибута используется адрес в сокращенной форме URL. В данном случае это адрес локального файла. Существуют также следующие атрибуты тега `body`:

- `bgcolor` – определяет цвет фона документа;
- `text` – определяет цвет текста;
- `link` – определяет цвет непосещенных ссылок;
- `vlink` – определяет цвет посещенных ссылок;
- `alink` – определяет цвет активных ссылок.

В качестве значения данных атрибутов может использоваться английское обозначение цвета (`red`, `white`, `darkblue`) или константа вида `#XXXXXX`, определяющая цвет в терминах RGB (Red, Green, Blue) в шестнадцатеричной нотации.

Пример: установка цвета символов и фона для всего документа

```
<html>
<body bgcolor="#C0C0C0" text="#0000FF">
...
</body>
</html>
```

Пример: установка цвета символов и фона для всего документа, а также цвета ссылок и посещенных ссылок

```
<html>
<body bgcolor="#C0C0C0" text="#0000FF"
vlink="#FF0000" link="#FF0000">
...
</body>
</html>
```

Замечание: данные цветовые константы представляют собой цвета Silver, Blue, Red и Lime соответственно.

3.5 Теги для оформления текста документа

Заголовки. Заголовок обозначает начало раздела документа. В стандарте определено 6 уровней заголовков: от h1 до h6. Текст, окруженный тегами <h1> </h1>, является заголовком первого уровня (самый крупный). Пример написания:

```
<h1> Общие сведения</h1>
```

Заголовки очень «любят» поисковые системы, они повышают ценность текста на web-странице, который располагается внутри тегов <h1>...<h6>. По этой причине лучше всегда использовать эти теги, несмотря на то, что с помощью стилей любой текст можно установить большого размера и, тем самым, сделать его заголовком.

Параграфы. Тег <p> применяют для разделения текста на параграфы.

Атрибуты заголовков и параграфов. Теги заголовков и параграфов имеют одинаковые атрибуты.

Атрибут `align` позволяет выровнять текст по левому или правому краю, по центру или ширине. По умолчанию текст выравнивается по левому краю.

Возможные значения атрибута `align`:

– `justify` – выравнивание текста по ширине.

– `left` – выравнивание по левому краю. Задается автоматически, при этом правый край не выравнивается.

– `right` – выравнивание по правому краю. Левый край не выравнивается.

– `center` – выравнивание по центру.

Пример: задать заголовок текста как заголовок первого уровня, выровненный по центру.

```
<html>
```

```
...
```

```
<body>
```

```
...
```

```
<h1 align="center">
```

```
    ОБУЧЕНИЕ И ПЕРЕПОДГОТОВКА КАДРОВ
```

```
</h1
```

```
...
```

```
</body>
```

```
</html>
```

Пример: создать абзац, выровненный по ширине.

```
<p align="justify">
```

```
    Зарегистрироваться в отделе занятости может любой  
    безработный от 18 лет и до пенсионного возраста.
```

```
</p>
```

**Тег разрыва строки `
`.** Принудительный перевод строки используется для того, чтобы нарушить стандартный порядок отображения текста. При обычном режиме интерпретации программа интерфейса пользователя отображает текст в рабочем окне, автоматически разбивая его на строки. В этом режиме концы строк текста игнорируются. Иногда для большей выразительности требуется начать печать с новой строки. Для этого и нужен тег `br`.

Тег запрета переноса <nobr> . Тег <nobr> (no break, без обрыва) дает браузеру команду отображать весь текст в одной строке, не обрывая ее. Если текст, заключенный в теги <nobr> </nobr>, не поместится на экране, браузер добавит в нижней части окна документа горизонтальную полосу прокрутки.

Теги управления отображением символов. Курсив, усиление, подчеркивание, верхний индекс, нижний индекс, шрифт большой, маленький, красный, синий и т.д. – все это различные варианты формы отображения.

Теги <big> и <small> - управляют изменением размеров шрифтов. Текст, расположенный между тегами <big> </big> или <small> </small>, будет соответственно, больше или меньше стандартного.

С помощью тегов <sup> и <sub> можно задавать верхние и нижние индексы любого размера. Например:

```
<p> Встреча состоится в 19<sup> 00</sup> </p>
```

Тег задает шрифт текста в данной области.

Атрибут size тега позволяет задать размер текста в данной области. Некоторые браузеры тег не поддерживают, поэтому желательно употреблять его только внутри текстовой области. Например, следующий тег позволит отобразить текст между тегами на один уровень крупнее обычного:

```
<font size="+1"> сообщение </font>
```

В языке HTML для тега были определены семь основных размеров шрифта, измеряемых не в пунктах, а в некоторых условных единицах – от 1 до 7. Как правило, обычный шрифт имеет размер «3», если не определено иное с помощью тега <basefont> (например, так: <basefont size="6">).

Необходимо различать записи и . В первом случае указывается относительный размер шрифта, а во втором – абсолютный. В качестве цифр можно использовать 2, 3 и т.д.

Атрибут face тега позволяет задать название шрифта или список названий шрифтов, разделенный запятыми. Например:

```
<font face="Verdana">  
<font face="Arial, Times New Roman">
```

Можно указывать названия семейств:
serif – шрифты с засечками, типа Times;
sans-serif – рубленые шрифты (шрифты без засечек или гротески), типичный представитель Arial;
cursive – курсивные шрифты;
monospace – моноширинные шрифты, ширина каждого символа в таком семействе одинакова.

Атрибут color позволяет задать цвет символов. Например: тег `` сделает помещенный между тегами текст синим.

```
...  
<body>  
<meta http-equiv="Content-type" content="text/html;  
charset=windows-1251">  
<h1>  
Всем <font color="#0000FF">привет!</font>  
</h1>  
</body>
```

Пример: задать заголовок текста как заголовок первого уровня, выровненный по центру и задать для заголовка свой цвет, отличный от цвета остального текста:

```
<h1 align="center"> <font color="#FF0000">  
Муниципальные отделы занятости </font> </h1>
```

Существуют следующие теги для задания стиля отображения шрифта:

`<i>...</i>` – делает шрифт курсивным. Многие браузеры аналогично интерпретируют тег `<cite>`, который служит для отметки в тексте цитат;

`...` – делает шрифт жирным. Многие браузеры аналогично интерпретируют тег ``, который служит для выделения важного фрагмента текста;

`<s>...</s>` – перечеркивает текст. Также используется полная форма написания тега `<strike>...</strike>`;

`<u>...</u>` – подчеркивает текст.

Пример: задать для текста полужирный шрифт и шрифт с подчеркиванием.

```
<b> <u> Регистрация. </u> </b>
```

Создание списков. Списки являются важным средством структурирования текста. В HTML имеются следующие виды списков:

– нумерованный список (ul);

– нумерованный список (ol).

Маркированный список записывается в виде последовательности тегов:

```
<ul>
```

```
<li>первый элемент списка</li>
```

```
<li>второй элемент списка</li>
```

```
<li>третий элемент списка</li>
```

```
</ul>
```

Теги `` и `` - это теги начала и конца нумерованного списка, тег `` (list item) задает тег элемента списка.

Пример: создать абзац, имеющий маркированный список. Абзац имеет персональный цвет, отличный от цвета остального текста. Абзац выровнен по ширине. Например, следующий:

```
<p align="justify"> <font color="#800000">
```

Пособие по безработице начисляется в соответствии с законом "О занятости". При этом учитывается:

```
<ul type="disc">
```

```
<li> сколько времени гражданин находится без работы; </li>
```

```
<li> по каким причинам гражданин уволился; </li>  
<li> в какой раз гражданин становится на учет. </li>  
</ul>  
</font>  
</p>
```

Атрибут `type` задает вид маркеров для данного списка:

`<ul type="disc">` тег создает маркеры в виде закрашенных окружностей;

`<ul type="circle">` тег создает маркеры в виде не закрашенных окружностей.

`<ul type="square">` тег создает сплошные квадратные маркеры.

Нумерованные списки. Тег `` вместе с атрибутом `type` позволяет создавать нумерованные списки, используя в качестве номеров не только обычные числа, но и строчные и прописные буквы, а также строчные и прописные римские цифры.

Используются следующие значения атрибута `type`:

`<ol type="1">` тег создает список с нумерацией в формате 1., 2., и т.д.

`<ol type="A">` создает список с нумерацией в формате A., B., C. и т.д.

`<ol type="a">` тег создает список с нумерацией в формате a., b., c. и т.д.

`<ol type="I">` тег создает список с нумерацией в формате I., II., III. и т.д.

Для создания нумерации, начинающейся не с единицы, используют атрибут `start`. Например, запись `<ol start="10">` вызовет нумерацию элементов списка, начиная с числа 10. Далее пойдут элементы с номерами 11, 12 и т.д.

Списки определений. Список определений несколько отличается от других видов списков. Вместо меток `` в списках определений используются метки `<dt>` (от английского

definition term – определяемый термин) и <dd> (от английского definition definition – определение определения). Весь список определений заключается в теги <dl> и </dl>.

Например, необходимо создать следующий фрагмент текста:

Список определений:

Провайдер

Организация, предоставляющая услуги Интернет

Хост

Узел глобальной Сети

Зеркало

Сервер, являющейся копией другого популярного сервера

Код web-документа, создающий вышеуказанный текст, имеет следующий вид:

```
<html>
```

```
<body>
```

```
Список определений:
```

```
<dl>
```

```
<dt> Провайдер
```

```
<dd> Организация, предоставляющая услуги Интернет
```

```
<dt> Хост
```

```
<dd> Узел глобальной Сети
```

```
<dt> Зеркало
```

```
<dd> Сервер, являющейся копией другого популярного сервера
```

```
</dl>
```

```
</body>
```

```
</html>
```

Теги блоков текста. Кроме тегов параграфа, заголовков, шрифтов, существуют еще два тега позволяющие выделить в документе блок или фрагмент текста.

Тег `<div>` используется для выделения блока текста, который следует рассматривать как отдельную логическую единицу (или отдельный абзац). Данный тег имеет атрибут `align` (как параграф), а также атрибуты для привязки сценариев и стилей к блоку текста.

Тег `` используется для выделения слов или строк текста. Он весьма похож на тег `div`, но текст, выделяемый тегом `SPAN` не образует отдельный параграф или абзац.

Тег выравнивания по центру `<center>`. Для выравнивания текста или изображения по центру страницы или ячейки таблицы можно использовать тег `center`. Например:

```
<center>текст, выровненный по центру</center>
```

Тег бегущей строки. Тег `<marquee>` предназначен для создания текстовой строки, которая может перемещаться в выделенной для нее области (например справа налево по горизонтали).

Основные атрибуты тега:

`behavior` – управляет движением бегущей строки по области вывода. Имеет следующие значения:

– `alternate` – текст бегущей строки безостановочно движется то в одном, то в противоположном направлении, «отражаясь» от краев области вывода и меняя направление движения.

– `scroll` (значение по умолчанию) – текст постоянно движется в одну сторону до края области вывода, где исчезает, а потом появляется с другой стороны.

– `slide` – текст бегущей строки пересекает область вывода и останавливается, когда первый символ достигает противоположной стороны области вывода.

`direction` – задает направление бегущей строки. По умолчанию равен `left`. Также может иметь значения `right`, `up` и `down`.

`bgcolor` – цвет фона бегущей строки.

`height` и `width` – высота и ширина полосы вывода бегущей строки.

scrolldelay – устанавливает величину задержки в миллисекундах между последовательными выводами текста в бегущей строке. Большинство браузеров не делают задержку вывода менее 60 миллисекунд. Для того, чтобы это ограничение снять, следует добавить атрибут truespeed.

Комментарии. Комментарии начинаются с символа <!-- и заканчиваются символом -->. Комментарии могут состоять из нескольких строк текста.

3.6. Теги для оформления таблиц

Создание таблицы. Для описания таблиц используется тег <table>. Тег <table>, как и многие другие, автоматически переводит строку до и после таблицы.

Тег <tr> (Table Row, строка таблицы) создает строку таблицы. Весь текст, другие теги и атрибуты, которые требуется поместить в одну строку, должны размещаться между тегами <tr> и </tr>.

Внутри строки таблицы обычно размещаются ячейки с данными. Каждая ячейка, содержащая текст или изображение, должна быть окружена тегами <td> и </td>. Число тегов <td> </td> в строке определяет число ячеек.

Пример создания простейшей таблицы.

```
<html>
<head>
<title> Создание таблицы</title>
</head>
<h1>Простейшая таблица</h1>
<table border="2">      <!--Это начало таблицы-->
<caption>              <!--Это заголовок таблицы-->
  ВГУ
</caption>
<tr>                   <!--Это начало первой строки-->
<td>                   <!--Это начало первой ячейки-->
  Факультет
```

```

</td>                <!--Это конец первой ячейки-->
<td>                  <!--Это начало второй ячейки-->
  Специальности
</td>                <!--Это конец второй ячейки-->
</tr>                 <!--Это конец первой строки-->

<tr>                  <!--Это начало второй строки-->
<td>                  <!--Это начало первой ячейки-->
  ФАЭМ
</td>                <!--Это конец первой ячейки-->
<td>                  <!--Это начало второй ячейки-->
  ВМ
</td>                <!--Это конец второй ячейки-->
</tr>                 <!--Это конец второй строки-->
</table>              <!--Это конец таблицы-->
</body>
</html>

```

Данный HTML-код создаст следующую таблицу (рис. 37).

Простейшая таблица

ВГТУ	
Факультет	Специальности
ФАЭМ	ВМ

Рис. 37. Пример простейшей таблицы

Заголовки столбцов таблицы. Заголовки для столбцов и строк таблицы задаются с помощью тега заголовка `<th>` `</th>` (Table Header, заголовок таблицы). Эти теги подобны тегам `<td>` `</td>`. Отличие состоит в том, что текст, заключенный между тегами `<th>` `</th>`, автоматически записывается жирным шрифтом и по умолчанию располагается посередине ячейки. Если воспользоваться `<td>``</td>` с тегом `` и атрибутом

`<align="center">`, текст тоже будет выглядеть как заголовок. Однако следует иметь в виду, что не все браузеры поддерживают в таблицах жирный шрифт, поэтому лучше задавать заголовки таблиц с помощью `<th>`.

Пример. Создать таблицу, в которой задать заголовок первого столбца и общий заголовок для второго, третьего и четвертого столбцов.

```
<table border="4">
  <tr>
    <th> Шифр <br>группы</th>
    <th>Фамилия</th>
    <th>Имя</th>
    <th>Отчество</th>
  </tr>
  <tr>
    <td>ВМ-051 </td>
    <td> Антонов</td>
    <td> Юрий </td>
    <td> Михайлович</td>
  </tr>
</table>
```

В результате написания данного HTML-кода будет создана следующая таблица (рис. 38):

Шифр группы	Фамилия	Имя	Отчество
ВМ-051	Антонов	Юрий	Михайлович

Рис. 38. Пример таблицы с заголовками столбцов таблицы

Заголовок таблицы. Таблица может иметь заголовок (`<caption>...</caption>`), хотя заголовок не является обязательным. Тег `<caption>` может включать атрибут `align`. Допустимые значения: `<caption align="top">` (заголовок помещается над таб-

лицей) и `<caption align="bottom">` (заголовок помещается под таблицей). Заголовок может состоять из любого текста и изображений. Текст будет разбит на строки, соответствующие ширине таблицы. Иногда тег `<caption>` используется для подписи под рисунком. Для этого достаточно описать таблицу без границ.

Пример. Создать таблицу, имеющую заголовок над таблицей.

```
<table border="6">
  <caption align="top"> Факультеты </caption>
  <tr>
    <td> ФАЭМ</td>
    <td> РТФ</td>
    <td> ИЭФ</td>
  </tr>
</table>
```

Данный HTML-код создаст следующую таблицу (рис. 39).

Факультеты		
ФАЭМ	РТФ	ИЭФ

Рис. 39. Пример таблицы с заголовком

Атрибуты тегов таблиц. Теги, работающие с таблицами, имеют следующие атрибуты.

Атрибут `nowrap`. Обычно любой текст, не помещающийся в одну строку ячейки таблицы, переходит на следующую строку. Однако при использовании атрибута `nowrap` с тегами `<th>` или `<td>` длина ячейки расширяется настолько, чтобы заключенный в ней текст поместился в одну строку. Примеры записи атрибута:

```
<th nowrap> </th>
<td nowrap> </td>
```

Атрибут colspan. Теги `<td>` и `<th>` модифицируются с помощью атрибута `colspan` (Column Span, соединение столбцов). `colspan` – устанавливает «размах» ячейки по горизонтали. Например, `colspan="3"` означает, что ячейка простирается на три колонки.

Атрибут rowspan. Данный атрибут используется в тегах `<td>` и `<th>` и задает число строк, на которые растягивается ячейка. `rowspan` – устанавливает «размах» ячейки по вертикали. Например, `rowspan="2"` означает, что ячейка занимает две строки.

Атрибут width. Устанавливает ширину таблицы, группы ячеек или ячейки. Ширину таблицы можно задать в пикселях (например, `<table width="400">`) или в процентах от ширины страницы (например, `<table width="80%">`) или в процентах от ширины таблицы, если задается ширина ячейки (например, `<td width="50%">`). В первом случае, задают таблицу шириной 400 пикселей независимо от размера страницы на мониторе. Во втором случае таблица будет занимать 80 % ширины страницы при любом размере изображения на экране. В третьем случае задают ширину ячейки, которая будет занимать 50 % ширины таблицы.

Если ячейка таблицы пуста, вокруг нее не рисуется рамка. Если ячейка пуста, а рамка нужна, в ячейку можно ввести символный объект ` `; (`non-breaking space` – неразрывный пробел). Ячейка по-прежнему будет пустой, а рамка вокруг нее будет.

Атрибут cellpadding. Определяет ширину пустого пространства между содержимым ячейки и ее границами, то есть задает поля внутри ячейки. Например: `<table cellpadding="3">`.

Атрибут cellspacing. Определяет ширину промежутков между ячейками в пикселях. Если этот атрибут не указан, по умолчанию задается величина, равная двум пикселям. Например: `<table cellspacing="6">`.

Атрибуты align и valign. Теги `<tr>`, `<td>`, `<th>` имеют атрибуты `align` и `valign`. Атрибут `align` устанавливает выравнива-

ние текста и графики по горизонтали. Допустимые значения: left (выравнивание по левому краю), center (выравнивание по центру), right (выравнивание по правому краю). Атрибут valign устанавливает вертикальное выравнивание текста в ячейках строки. Допустимые значения: top (выравнивание по верхнему краю), middle (выравнивание по центру), bottom (выравнивание по нижнему краю).

Атрибут border. Устанавливает ширину внешней рамки таблицы и ячеек в пикселях (например, <table border="4">). Если атрибут не установлен, таблица показывается без рамки <table border="0">.

Атрибут bgcolor. Позволяет установить цвет фона. В зависимости от того, с каким тегом (table, tr, td) он применяется, цвет фона может быть установлен для всей таблицы (<table bgcolor="green">), для строки (<tr bgcolor="green">) или для отдельной ячейки (<td bgcolor="green">). Значением данного атрибута является RGB-код или стандартное название цвета.

Рамка таблицы не обязательно может быть серой. Для этого служит **атрибут bordercolor**, которому можно дать в качестве значения любой доступный цвет. Например, bordercolor="green".

Атрибут background задает фоновое изображение для таблиц. Применим к тегам table и td. Его значением является имя файла с фоновым изображением.

Любая ячейка таблицы может содержать в себе другую таблицу.

Пример. Создать таблицу, имеющую объединенные ячейки.

```
<html>
<body>
<table width="50%" border="3" cellspacing="6"
cellpadding="20" align="center" bordercolorlight="lime"
bgcolor="white">
<tr>
<td width="33%"> это первая ячейка</td>
```



```

    <td width="33%"> это вторая ячейка</td>
    <td rowspan="3"> это три ячейки третьего столбца объе-
динились в одну большую ячейку </td>
</tr>
<tr>
    <td colspan="2"> это единственная ячейка второй строки,
объединяющая оба столбца</td>
</tr>
<tr>
    <td> это первая ячейка третьей строки </td>
    <td> это вторая ячейка третьей строки</td>
</tr>
</table>
</body>
</html>

```

Данный HTML-код создаст следующую таблицу (рис. 40):

это первая ячейка	это вторая ячейка	это три ячейки третьего столбца объединились в одну большую ячейку
это единственная ячейка второй строки, объединяющая оба столбца		
это первая ячейка третьей строки	это вторая ячейка третьей строки	

Рис. 40. Пример таблицы с объединением ячеек по горизонтали и вертикали

Таблицы могут иметь такие элементы оформления, как заголовки и заключение. Можно все основное содержание таблицы обозначить тегом `<tbody>`. Тогда заголовок таблицы обозна-

чается тегом <thead>, а заключение – тегом <tfoot>. Все три тега употребляются с закрывающими тегами и могут иметь атрибуты bgcolor, align, valign.

Добавим в предыдущий пример заголовок и заключение.

```
<html>
```

```
<body>
```

```
<table width="50%" border="3" cellspacing="6" cellpadding="20" align="center" bgcolor="white">
```

```
<thead bgcolor="Aqua">
```

```
<tr>
```

```
<th colspan="3"> Учебная таблица</th>
```

```
</tr>
```

```
</thead>
```

```
<tbody>
```

```
<tr>
```

```
<td width="33%">
```

```
это первая ячейка
```

```
</td>
```

```
<td width="33%">
```

```
это вторая ячейка
```

```
</td>
```

```
<td rowspan="3">
```

это три ячейки третьего столбца объединились в одну большую ячейку

```
</td>
```

```
</tr>
```

```
<tr>
```

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

это единственная ячейка второй строки, объединяющая оба столбца

```
</td>
```

```
</tr>
```

```
<tr>
```

```
<td>
```

```

это первая ячейка третьей строки
</td>
<td>
это вторая ячейка третьей строки
</td>
</tr>
</tbody>
<tfoot bgcolor="yellow">
<tr>
<td colspan="3" align="center">
Итоговая строка
</td>
</tr>
</tfoot>
</table>
</body>
</html>

```

Данный HTML-код создаст следующую таблицу (рис. 41):

Учебная таблица		
это первая ячейка	это вторая ячейка	это три ячейки третьего столбца объединились в одну большую ячейку
это единственная ячейка второй строки, объединяющая оба столбца		
это первая ячейка третьей строки	это вторая ячейка третьей строки	
Итоговая строка		

Рис. 41. Пример таблицы с добавленным заголовком и заключением

3.7. Варианты для задания №1

Общее задание – создать web-страницу, содержащую текст и изображения. Оформить все элементы на странице соответствующими тегами HTML с добавлением надлежащих атрибутов.

Вариант 1

1. Задать заголовок документа в окне браузера – **golden section search**.

2. Установить цвет фона и символов для всего текста, задать заголовок документа как заголовок первого уровня, задать для текста выравнивание по ширине. Для заголовка «Поиск минимума...» задать персональный цвет символов. Для абзаца «Метод золотого сечения» указать персональный цвет символов.

Поиск минимума с помощью метода «золотого сечения»

Когда невозможно применить классические методы поиска минимума функции (например, когда $f'(x)=0$) ее минимум ищется с помощью численных методов в некотором интервале $a < x < b$, в котором, как предполагается, лежит минимум, вычисляя значения функции в выбранных точках данного интервала (рисунок 1). Иногда это единственно возможная стратегия поиска.

(Вставить рисунок var01_01.JPG)

Метод золотого сечения – метод поиска значений *действительно-значной функции* на заданном отрезке. В основе метода лежит принцип деления в пропорциях золотого сечения. Золотое сечение (золотая пропорция, деление в крайнем и среднем отношении) — деление непрерывной величины на две части в таком отношении, при котором меньшая часть так относится к большей, как большая ко всей величине. Например, отрезав квадрат от прямоугольника, построенного по принципу золотого сечения, мы получаем новый, уменьшенный прямоугольник с тем же отношением сторон (рисунок 2).

(Вставить рисунок var01_02.JPG)

Формализация алгоритма по шагам:

1. **Шаг 1.** Задаются начальные границы отрезка a, b и точность ε , рассчитывают начальные точки деления: $x_1 = b - (b - a) / \varphi$, $x_2 = a + (b - a) / \varphi$ и значения в них целевой функции: $y_1 = f(x_1), y_2 = f(x_2)$.

2. **Шаг 2.**

Если $y_1 \leq y_2$, то

$$b = x_2, x_2 = x_1, x_1 = b - (b - a) / \varphi, y_2 = y_1, y_1 = f(x_1).$$

Иначе

$$a = x_1, x_1 = x_2, x_2 = a + (b - a) / \varphi, y_1 = y_2, y_2 = f(x_2).$$

3. **Шаг 3.**

- Если $|b - a| < \varepsilon$, то x_1 искомым минимум и останов.
- Иначе возврат к **Шагу 2.**

Вариант 2

1. Задать заголовок документа в окне браузера – **Square interpolation.**

2. Установить цвет фона и символов для всего текста, задать заголовок документа как заголовок первого уровня, задать для текста выравнивание по ширине. Для заголовка «Алгоритм» задать персональный цвет символов. Для второго абзаца указать персональный цвет символов.

Поиск минимума методом квадратичной интерполяции

Основой данного метода является использование нескольких значений функции в определенных точках для *аппроксимации функции обычным полиномом в небольшой области значений*. Положение минимума функции аппроксимируется положением минимума полинома, так как последний вычислить проще.

Если известны значения функции $f(x)$ в трех различных точках α, β, γ , равные соответственно $f_\alpha, f_\beta, f_\gamma$, то функция $f(x)$ может быть аппроксимирована квадратичной функцией

$$\varphi(x) = A \cdot x^2 + B \cdot x + C,$$

где A, B и C определяются из уравнений

$$A \cdot \alpha^2 + B \cdot \alpha + C = f_\alpha,$$

$$A \cdot \beta^2 + B \cdot \beta + C = f_\beta,$$

$$A \cdot \gamma^2 + B \cdot \beta + C = f_\gamma.$$

Считается, что функция $\varphi(x)$ будет иметь минимум в точке $x = -B/2 \cdot A$, если $A > 0$.

Алгоритм

1. Вычислить $f(A)$ и $f(A + H)$.
2. Если $f(A) < f(A + H)$, то взять в качестве третьей точки $A - H$ и вычислить $f(A - H)$ (рис. 1, а). В противном случае в качестве третьей точки взять $A + 2 \cdot H$ и найти $f(A + 2 \cdot H)$ (рис. 1, б).

(Вставить рисунок var02_01.JPG)

3. Используя эти три точки, найти минимум аппроксимирующей параболы δ и вычислить $f(\delta)$.
4. Если разница между наименьшим значением функции и следующим наименьшим значением функции меньше заданной точности, то процедура заканчивается.
5. Если процедура не завершилась на шаге 4, то точка с наибольшим значением обычно отбрасывается, и происходит возврат на шаг 3. Но если точка с наибольшим значением функции определяет конечные границы интервала, в котором лежит минимум, то следует оставить это значение и вернуться на шаг 3.

Вариант 3

1. Задать заголовок документа в окне браузера – **Hook Jeeves 1.**

2. Установить цвет фона и символов для всего текста, задать заголовок документа как заголовок первого уровня, задать для текста выравнивание по ширине. Для заголовка «Алгоритм (часть 1)» задать персональный цвет символов. Для первого абзаца указать персональный цвет символов.

РЕАЛИЗАЦИЯ МЕТОДА ПОИСКА МИНИМУМА ФУНКЦИИ НЕСКОЛЬКИХ ПЕРЕМЕННЫХ С ИСПОЛЬЗОВАНИЕМ МЕТОДА ХУКА-ДЖИВСА

Данный метод относится к методам прямого поиска для определения минимума функции n переменных. Поиск состоит из последовательности *шагов* исследующего поиска вокруг *базисной точки*, за которой в случае успеха следует *поиск по образцу*.

Алгоритм (часть 1)

А. Выбрать начальную базисную точку b_1 и шаг длиной h_j для каждой переменной x_j , $j = 1, 2, \dots, n$. В приведенной далее программе для каждой переменной используется шаг h , однако указанная выше модификация тоже может оказаться полезной.

В. Вычислить $f(\mathbf{x})$ в базисной точке b_1 , с целью получения сведений о локальном поведении функции $f(\mathbf{x})$. Эти сведения будут использоваться для нахождения подходящего направления поиска по образцу, с помощью которого можно надеяться достичь большего убывания значения функции. Функция $f(\mathbf{x})$ в базисной точке b_1 находится следующим образом:

1. Вычисляется значение функции $f(b_1)$ в базисной точке b_1 .

2. Каждая переменная по очереди изменяется прибавлением длины шага. Таким образом, мы вычисляем значение функции $f(b_1 + h_1 e_1)$, где e_1 - единичный вектор в направле-

нии оси x_1 . Если это приводит к уменьшению значения функции, то b_1 заменяется на $b_1 + h_1 e_1$. В противном случае вычисляется значение функции $f(b_1 - h_1 e_1)$, и если ее значение уменьшилось, то b_1 заменяем на $b_1 - h_1 e_1$. Если ни один из проделанных шагов не приводит к уменьшению значения функции, то точка b_1 остается неизменной и рассматриваются изменения в направлении оси x_2 , т. е. находится значение функции $f(b_1 + h_2 e_2)$ и т. д. Когда будут рассмотрены все и переменные, мы будем иметь новую базисную точку b_2 .

3. Если $b_2 = b_1$, т. е. уменьшение функции не было достигнуто, то исследование повторяется вокруг той же базисной точки b_1 , но с уменьшенной длиной шага. На практике удовлетворительным является уменьшение шага (шагов) в десять раз от начальной длины.

4. Если $b_2 \neq b_1$, то производится *поиск по образцу*.

Блок-схема данного метода приведена на рисунке 1.

(Вставить рисунок var03_01.JPG)

Рисунок 1 – Блок схема метода Хука – Дживса

Вариант 4

1. Задать заголовок документа в окне браузера – **Hook Jeeves 2**.

2. Установить цвет фона и символов для всего текста, задать заголовок документа как заголовок первого уровня, задать для текста выравнивание по ширине. Для заголовка «Алгоритм (часть 2)» задать персональный цвет символов. Для первого абзаца указать персональный цвет символов.

РЕАЛИЗАЦИЯ МЕТОДА ПОИСКА МИНИМУМА ФУНКЦИИ
НЕСКОЛЬКИХ ПЕРЕМЕННЫХ С ИСПОЛЬЗОВАНИЕМ
МЕТОДА ХУКА-ДЖИВСА

Данный метод относится к методам прямого поиска для определения минимума функции n переменных. Поиск состоит из последовательности *шагов* исследующего поиска вокруг *базисной точки*, за которой в случае успеха следует *поиск по образцу*.

Алгоритм (часть 2)

С. При поиске по образцу используется информация, полученная в процессе исследования, и минимизация функции завершается поиском в направлении, заданном образцом. Эта процедура производится следующим образом:

1. Разумно двигаться из базисной точки b_2 в направлении $b_2 - b_1$, поскольку поиск в этом направлении уже привел к уменьшению значения функции. Поэтому вычислим функцию в точке образца

$$P_1 = b_1 + 2 \cdot (b_2 - b_1).$$

В общем случае

$$P_1 = b_i + 2 \cdot (b_{i+1} - b_i).$$

2. Затем исследование следует продолжать вокруг точки $P_1 (P_i)$.

3. Если наименьшее значение на шаге В, 2 меньше значения в базисной точке b_2 (в общем случае b_{i+1}), то получают новую базисную точку $b_3 (b_{i+2})$, после чего следует повторить шаг В, 1. В противном случае не производить поиск по образцу из точки $b_2 (b_{i+1})$, а продолжить исследования в точке $b_2 (b_{i+1})$.

Д. Завершить этот процесс, когда длина шага (длины шагов) будет уменьшена до заданного малого значения.

На рисунке 1 приведена блок-схема процедуры исследования вокруг базисной точки.

(Вставить рисунок var04_01.JPG)

Рисунок 1 – Блок-схема процедуры исследования вокруг базисной точки

Задание.

Используя метод Хука – Дживса вычислить минимум функции

$$F(x_1, x_2, x_3) = (x_1 - 2)^2 + (x_2 - 5)^2 + (x_3 + 2)^4.$$

Начальная точка (4, -2, 3). Величина шага равна 1.

Вариант 5

1. Задать заглавие документа в окне браузера – **Nelder Mead**.

2. Установить цвет фона и символов для всего текста, задать заголовок документа как заголовок первого уровня, задать для текста выравнивание по ширине. Для заголовка «Описание» задать персональный цвет символов. Для первого абзаца указать персональный цвет символов.

РЕАЛИЗАЦИЯ МЕТОДА ПОИСКА МИНИМУМА ФУНКЦИИ НЕСКОЛЬКИХ ПЕРЕМЕННЫХ С ИСПОЛЬЗОВАНИЕМ МЕТОДА НЕЛДЕРА-МИДА

Метод Нелдера – Мида, также известный как метод деформируемого многогранника и симплекс-метод, – метод безусловной оптимизации функции от нескольких переменных, не использующий производной функции.

Суть метода заключается в последовательном перемещении и деформировании симплекса вокруг точки экстремума.

Пример реализации метода приведен на рисунке ниже

(вставить рисунок var05_01.gif)

Рисунок – реализация метода на основе функции Химмельблау

Симплекс или **n-симплекс** (от лат. *simplex* — простой) — геометрическая фигура, являющаяся *n*-мерным обобщением треугольника. Определяется как выпуклая оболочка *n*+1 точек,

не лежащих в одной n -мерной гиперплоскости. Эти точки называются **вершинами** симплекса. В частности:

- 0-симплекс это точка;
- 1-симплекс это отрезок;
- 2-симплекс это треугольник;
- 3-симплекс это тетраэдр.

Метод находит как глобальные, так и локальные экстремумы и может «застрять» в одном из них. Если всё же требуется найти глобальный экстремум, можно попробовать выбирать другой начальный симплекс.

Описание

Пусть требуется найти безусловный минимум функции n переменных $f(x^{(1)}, x^{(2)}, \dots, x^{(n)})$. Предполагается, что серьёзных ограничений на область определения функции нет, то есть функция определена во всех встречающихся точках.

Параметрами метода являются:

1. Коэффициент отражения $\alpha > 0$, обычно выбирается равным 1.
2. Коэффициент сжатия $\beta > 0$, обычно выбирается равным 0,5.
3. Коэффициент растяжения $\gamma > 0$, обычно выбирается равным 2.
4. Шаг алгоритма.

Подготовка. Вначале выбирается $n + 1$ точка $x_i = (x_i^{(1)}, x_i^{(2)}, \dots, x_i^{(n)})$, образующие симплекс n -мерного пространства. В этих точках вычисляются значения функции: $f_1 = f(x_1), f_2 = f(x_2), \dots, f_{n+1} = f(x_{n+1})$.

Вариант 6

1. Задать заглавие документа в окне браузера – **Nelder Mead 2**.

2. Установить цвет фона и символов для всего текста, задать заголовок документа как заголовок первого уровня, задать

для текста выравнивание по ширине. Для заголовка «Описание» задать персональный цвет символов. Для первого абзаца указать персональный цвет символов.

РЕАЛИЗАЦИЯ МЕТОДА ПОИСКА МИНИМУМА ФУНКЦИИ НЕСКОЛЬКИХ ПЕРЕМЕННЫХ С ИСПОЛЬЗОВАНИЕМ МЕТОДА НЕЛДЕРА-МИДА

Метод Нелдера – Мида, также известный как метод деформируемого многогранника и симплекс-метод, — метод безусловной оптимизации функции от нескольких переменных, не использующий производной функции.

Суть метода заключается в последовательном перемещении и деформировании симплекса вокруг точки экстремума.

Пример реализации метода приведен на рисунке ниже

(вставить рисунок var06_01.gif)

Рисунок – реализация метода на основе функции Химмельблау

1. Сортировка. Из вершин симплекса выбираем три точки: x_h с наибольшим (из выбранных) значением функции f_h , x_g со следующим по величине значением f_g и x_l с наименьшим значением функции f_l . Целью дальнейших манипуляций будет уменьшение по крайней мере f_h .

2. Найдём центр тяжести всех точек, за исключением x_h :

Вычислить $f_c = f(x_c)$.

3. Отражение. Отразим точку x_h относительно x_c с коэффициентом α (при $\alpha = 1$ это будет центральная симметрия), получим точку x_r и вычислим в ней функцию: $f_r = f(x_r)$. Координаты новой точки вычисляются по формуле:

$$x_r = (1 + \alpha)x_c - \alpha x_h$$

4. Далее смотрим, насколько нам удалось уменьшить функцию, ищем место f_r в ряду f_h, f_g, f_l :

а. Если $f_r < f_l$, то направление выбрано удачное и можно попробовать увеличить шаг — производим растяжение. Новая точка $x_e = (1 - \gamma)x_c + \gamma x_r$ и значение функции $f_e = f(x_e)$.

б. Если $f_l < f_r < f_g$, то выбор точки уже неплохой (новая лучше двух прежних). Заменяем точку x_h на x_r и переходим на шаг 8.

с. Если $f_h > f_r > f_g$, то меняем обозначения x_r, x_h (и соответствующие значения функции) местами и идём на шаг 5.

д. Если $f_r > f_h$, то просто идём на следующий шаг 5.

5. Сжатие. Строим точку $x_s = \beta x_h + (1 - \beta)x_c$ и вычисляем в ней значение f_s .

6. Если $f_s < f_h$, то заменяем точку x_h на x_s и идём на шаг 8.

7. Если $f_s > f_h$, то первоначальные точки оказались самыми маленькими. Делаем глобальное сжатие симплекса —: $x_i \rightarrow x_l + (x_i - x_l)/2$ для всех требуемых точек x_i .

8. Последний шаг — проверка сходимости. Суть проверки заключается в том, чтобы проверить взаимную близость полученных вершин симплекса, что предполагает и близость их к искомому минимуму.

Вариант 7

1. Задать заглавие документа в окне браузера – **Control theory**.

2. Установить цвет фона и символов для всего текста, задать заголовок документа как заголовок первого уровня, задать для текста выравнивание по ширине. Для заголовка «Основные понятия» задать персональный цвет символов. Для первого раздела указать персональный цвет символов.

Введение в теорию управления

Теория управления — наука о принципах и методах управления различными системами, процессами и объектами.

Основами теории управления являются кибернетика и теория информации.

Суть теории управления: на основе системного анализа составляется математическая модель объекта управления (ОУ), после чего синтезируется алгоритм управления (АУ) для получения желаемых характеристик протекания процесса или целей управления.

Основные понятия

А. **Система** – соединение элементов и устройств в единую структуру определенного функционального назначения.

В. Под **управляемой системой** обычно понимается любая совокупность материальных объектов, на поведение которой во времени можно влиять выбором целенаправленных внешних воздействий.

С. Совокупность управляющих операций образует **процесс управления**.

Д. Совокупность средств управления и объекта управления образует **систему управления**.

Е. Систему, в которой все рабочие и управляющие операции выполняют автоматические устройства, называют **автоматической системой**. Систему, в которой автоматизирована только часть рабочих и управляющих операций, называют **автоматизированной системой**.

Объект управления

Обычно процесс или объект, подлежащий управлению, представляют в виде прямоугольника (рис. 1).

(Вставить рисунок var07_01.JPG)

Рисунок 1 – Объект управления

Пусть $x = \{x_1, x_2, \dots, x_n\}$ – совокупность управляемых координат процесса; $u = \{u_1, u_2, \dots, u_m\}$ – управляющие воздействия, прикладываемые к управляющему органу объекта управления, с помощью которого можно изменять координаты x ; $z = \{z_1, z_2, \dots, z_l\}$ – возмущающие воздействия (рис. 2).

(Вставить рисунок var07_02.JPG)

Рисунок 2 – Объект управления и возмущающие и управляющие воздействия

Вариант 8

1. Задать заглавие документа в окне браузера – Automatic control.

2. Установить цвет фона и символов для всего текста, задать заголовок документа как заголовок первого уровня, задать для текста выравнивание по ширине. Для заголовка «Основные понятия» задать персональный цвет символов. Для первого раздела указать персональный цвет символов.

Основные виды автоматического управления

На первом этапе развития техники управления использовался практически лишь один вид автоматического управления – поддержание заданного постоянного значения регулируемой величины.

Долгое время под системами автоматического регулирования понимался именно этот вид. Впоследствии число видов увеличивалось, и вполне вероятно, что приведенные ниже шесть основных их видов не исчерпывают не только возможные виды в будущем, но и существующие сегодня.

Перечень основных видов управления

1. Стабилизация.
2. Программное управление.
3. Следящие системы.
4. Системы с поиском экстремума показателя качества.
5. Оптимальное управление.
6. Адаптивные системы.

Стабилизация

Системы поддержания постоянства управляемой величины называют также *системами стабилизации*. Желаемый закон в них имеет вид

$$x_0(t) = \text{const}$$

Рассмотрим схему с простейшими линейными преобразовательными звеньями.

(вставить рисунок var08_01.JPG)

Уравнения статики для такой схемы будут

$$x = k_0 u - k_z z; \quad u = k_p \Delta x = k_p (x_0 - x),$$

где k_0 , k_p и k_z — постоянные коэффициенты, называемые соответственно коэффициентами передачи объекта, регулятора и нагрузки.

Статический регулятор поддерживает постоянное значение регулируемой величины с ошибкой. Статизм — это величина относительной статической ошибки при изменении нагрузки от холостого хода до номинальной. В некоторых системах статическая ошибка нежелательна. Тогда переходят к регулированию, в котором она в силу структуры системы равна нулю, т. е. к астатическому регулированию.

3.8. Варианты для задания №2

Общее задание

Создать документ HTML, в котором должна быть размещена таблица, соответствующая варианту задания.

Для таблицы:

- указать цвет фона и символов,
- выровнять по центру страницы
- сделать ее ширину величиной 80% от размера окна.
- указать цвет фона для ячеек заголовков и подзаголовков;
- добавить изображения, где указано в задании.

– внизу после таблицы добавить сведения об авторах (фразу «© Информация оформлена студентом (кой) группы ВМ-0.... Фамилия и инициалы»)

Вариант 1

Истребитель Су-27 СМ (Российская Федерация)	
Изображение 1	Разработчик – ОКБ Сухого
	Главный конструктор – М.П. Симонов
	Начало эксплуатации - 1984
Тактико-технические характеристики	
Экипаж	1 человек
Длина	21,9 м
Размах крыла	14,7 м
Максимальная взлетная масса	33000 кг
Двигатели	2 × ТРДДФ АЛ-31Ф
Максимальная скорость на высоте 11 км	2 500 км/ч (2,35М)
Вооружение	Пушечное - 1 × 30 мм пушка ГШ-30-1
	Ракеты «воздух-воздух» (8 × Р-27 или 8 × Р-77 и 4-6 × Р-73)
	Ракеты «воздух-земля» (6 × Х-29 или 6 × Х-31 или 2 × Х-59)
	Неуправляемые авиационные ракеты 80 × С-8 или 20 × С-13 или 4 × С-25, авиабомбы
Изображение 2	

Вариант 2

Фронтовой бомбардировщик Су-24М (Российская Федерация)	
Изображение 1	Разработчик – ОКБ Сухого
	Главный конструктор – Е. С. Фельснер
	Начало эксплуатации - 1975
Тактико-технические характеристики	
Экипаж	2 человека
Длина	24,6 м
Размах крыла	<i>при угле стреловидности 16°: 17,638 м</i> <i>при угле стреловидности 69°: 10,366 м</i>
Макс. взлетная масса	39700 кг
Двигатели	2 × ТРДДФ АЛ-21Ф-3
Максимальная скорость на большой высоте	1 700 км/ч (1,6М)
Вооружение	<i>Пушечное</i> - 1 х шестиствольная 23 мм пушка ГШ-6-23
	<i>Ракеты «воздух-воздух»</i> (2 × Р-60)
	<i>Ракеты «воздух-земля»</i> 4 × Х-25МЛ/МР или Х-23 3 × Х-29Л/Т или Х-59 6 × С-25Л 2 × Х-58
	<i>Неуправляемые авиационные ракеты</i> 120 × С-8 или 30 × С-13 или 6 × С-25, <i>авиабомбы</i>
Изображение 2	

Вариант 3

Ударный вертолет Ми-28 (Российская Федерация)	
Изображение 1	Разработчик – Московский вертолётный завод имени М. Л. Миля
	Главный конструктор – М. В. Вайнберг
	Начало эксплуатации - 2007
Тактико-технические характеристики	
Экипаж	2 человека
Длина	16,85 м
Длина несущего винта	17,2 м
Максимальная взлетная масса	12100 кг
Двигатели	2 х ТВ3-117ВМА
Максимальная скорость	382 км/ч
Вооружение	<i>Пушечное</i> 1 х 30-мм пушка 2А42
	<i>Управляемые ракеты</i> 16 ПТУР «Атака-В», Р-73
	<i>Неуправляемые авиационные ракеты</i> С-8 – 4 х 20 шт.; С-13 – 2 х 5 шт.; С-24 – 2 шт.
Изображение 2	

Вариант 4

Палубный истребитель-бомбардировщик F/A-18F Hornet (США)	
Изображение 1	Разработчик – корпорация Макдоннел-Дуглас
	Всего построено – свыше 1500 машин
	Начало эксплуатации - 1983
Тактико-технические характеристики	
Экипаж	1 человек
Длина	18,5 м
Размах крыла	13,7 м
Максимальная взлетная масса	30000 кг
Двигатели	2 × F414-GE-400 ТВД
Максимальная скорость на большой высоте	1915 км/ч
Вооружение	Пушечное - 1 x 20 мм пушка M61A1/A2 Vulcan
	Ракеты: AIM-9 Sidewinder, AIM-7 Sparrow, AIM-120 AMRAAM, Harpoon, Harm, Shrike, SLAM, SLAM-ER, Walleye, Maverick; (9 точек подвески)
	Различные варианты обычных и корректируемых авиабомб
Изображение 2	

Вариант 5

Тактический истребитель F-15C Eagle (США)	
Изображение 1	Разработчик – корпорация Макдоннел-Дуглас
	Всего построено – свыше 1200 машин всех модификаций
	Начало эксплуатации - 1976
Тактико-технические характеристики	
Экипаж	1 человек
Длина	19,44м
Размах крыла	13 м
Максимальная взлетная масса	30845 кг
Двигатели	2× двухконтурных турбореактивных с форсажной камерой (ТРДДФ) Pratt & Whitney F100-PW-100
Максимальная скорость на большой высоте	2650 км/ч (2,5+М)
Вооружение	<i>Пушечное</i> - Шестиствольная 20-мм пушка Джeneral Электрик М-61А1 «Вулкан»
	<i>Ракеты «воздух-воздух»:</i> <ul style="list-style-type: none"> • УР AIM-9L/AIM-9M Sidewinder: 4 ракеты • УР AIM-7F/AIM-7M Sparrow: 4 ракеты
	Различные варианты обычных и корректируемых авиабомб
Изображение 2	

Вариант 6

Ударный вертолет AH-64 Apache (США)	
Изображение 1	Разработчик – корпорация Макдоннел-Дуглас (фирма «Hughes Helicopters»)
	Всего построено – 827 машин
	Начало эксплуатации - 1984
Тактико-технические характеристики	
Экипаж	2 человека
Длина	17,73 м
Диаметр несущего винта	14,63 м
Максимальная взлетная масса	9500 кг
Двигатели	Дженерал Электрик Т700-GE-701 (2x1695 л.с.)
Максимальная скорость	293 км/ч
Вооружение	Пушечное 30-мм пушка M230 Chain Gun
	Управляемые ракеты: Ракеты «воздух-воздух» AIM-92, AIM-9 Ракеты «воздух-земля» AGM-114
	Неуправляемые ракеты: Hydra 70
Изображение 2	

Вариант 7

Стратегический бомбардировщик Ту-160 (Российская Федерация)	
Изображение 1	Разработчик – КБ Туполева
	Главный конструктор – В.И. Близнюк
	Начало эксплуатации - 1987
Тактико-технические характеристики	
Экипаж	4 человека
Длина	54,1 м
Размах крыла	55,7 / 35,6 м
Максимальная взлетная масса	275000 кг
Двигатели	4 × ТРДДФ НК-32
Максимальная скорость на большой высоте	2230 км/ч
Вооружение	2 барабанные ПУ с 6 СКР Х-55 и Х-55М,
	2 барабанные ПУ с 6 СКР Х-55 и Х-55М, 2 барабанные ПУ с 12 УР малой дальности Х-15
	КАБ различных типов до КАБ-1500, термоядерные и обычные бомбы, мины
Изображение 2	

Вариант 8

Дальний бомбардировщик-ракетоносец Ту-22М3 (Российская Федерация)	
Изображение 1	Разработчик – КБ Туполева
	Главный конструктор – Д.С. Марков
	Начало эксплуатации - 1983
Тактико-технические характеристики	
Экипаж	4 человека
Длина	42,46 м
Размах крыла	34,28 / 22,3 м
Максимальная взлетная масса	124000 кг
Двигатели	2 × ТРДДФ НК-25
Максимальная скорость на большой высоте	2300 км/ч
Вооружение	три ракеты Х-22 или четыре ракеты Х-15П на пилонах под корневой частью крыла, шесть ракет Х-15П в барабанной пусковой установке в оружейном отсеке
	до 12 т обычных/ядерных бомб
	<i>Оборонительное вооружение</i> – одна двухствольная пушка ГШ-23Л калибра 23 мм в хвостовой турели
Изображение 2	

4. СОЗДАНИЕ МНОГОСТРАНИЧНОГО САЙТА С ПОМОЩЬЮ ТЕГОВ HTML

4.1. Создание и оформление гиперссылок

В HTML переход от одного фрагмента текста к другому задается с помощью метки вида:

`` фрагмент текста или какой-либо другой объект ``

В качестве параметра [адрес перехода] может использоваться несколько типов аргументов. Самое простое – это задать имя другого HTML-документа, к которому нужно перейти. Например, из текущего файла необходимо перейти к файлу с именем p1.html:

```
<a href="p1.htm"> На первую </a>
```

Такой фрагмент HTML-текста приведет к появлению в документе выделенного фрагмента «На первую», при нажатии на который в текущее окно будет загружен документ p1.html.

Если в адресе перехода не указан каталог, переход будет выполнен внутри текущего каталога. Если в адресе перехода не указан сервер, переход будет выполнен на текущем сервере.

Из этого следует одно очень важное практическое сообщение. Если группа HTML-документов ссылается друг на друга только по имени файла и находится в одном каталоге, то можно переместить ее в другой каталог и на другой компьютер, где она будет также работать.

Гиперссылка может ссылаться на документ, хранимый в Интернете. Например:

```
<a href="http://www.intuit.ru/index.htm">Справка</a>
```

Содержание контейнера гипертекстовой ссылки, заключенное между тегом начала и тегом конца, выделяется в тексте цветом, определенным для контекстных гипертекстовых ссылок в атрибутах тега `<body>`.

Другой формой использования тега `<a>` является определение точек внутри текста (анкеров, якорей), на которые можно ссылаться.

Для создания **точки (якоря)** задают оператор:

```
<a name="имя точки">Текст, который помечается точкой</a>
```

Например:

```
<a name="AAA">ВГТУ</a>
```

Слова «ВГТУ» при этом никак не будут выделены в тексте документа.

Для задания перехода на точку (якорь) в месте перехода создают гиперссылку следующим образом:

```
<a href="#имя точки">гиперссылка</a>
```

Например, переход на точку с именем AAA:

```
<a href="AAA">На начало</a>
```

Пример фрагмента программы, в которой слово «На начало» ссылается на слово «ВГТУ».

```
<html>
<!-- Простейшая программа-->
<head>
<meta http-equiv="Content-type" content="text/html;
charset=windows-1251">
<title>
```

Пример ссылок на фрагменты текста

```
</title>
</head>
<body>
<h1>
<a name="aaa">ВГТУ</a>
</h1>
<p> Факультеты ВГТУ</p>
<ol>
<li>ФАЭМ</li>
<li>ЕГФ</li>
```

```
<li>РТФ</li>
</ol>
<a href="#aaa">На начало</a>
</body>
</html>
```

Если необходимо перейти к определенному слову в другом файле, то гиперссылку оформляют следующим образом:

```
<a href="2.html#AAA">Переход на точку</a>
```

Во избежание недоразумений рекомендуется задавать имена якорей латинскими буквами. Большинство браузеров отличают прописные и строчные буквы. Если имя якоря определено как “AAA”, ссылка на якорь “aaa” или “AaA” не переведет на нужный анкер.

Возможны ссылки не только на HTML-документы, но и на другие виды ресурсов:

```
<a href="ftp://server/directory/file.ext"> Загрузка файла </a>
```

Такая ссылка, если ей воспользоваться, запустит протокол передачи файлов и начнет загрузку файла file.ext, находящегося в каталоге directory на сервере server, на локальный диск пользователя.

```
<a href="mailto:user@mail.box"> Послать письмо </a>
```

Если пользователь совершит переход по такой ссылке, у него на экране откроется окно ввода исходящего сообщения его почтовой программы. В строке To: (“Куда») окна почтовой программы будет указано user@mail.box.

Чтобы не загружать ссылочную страницу в том же окне браузера, в которое была загружена текущая страница, а открыть ее в новом окне, нужно в теге <a> установить атрибут target= со значением «_blank». Значение «_self» - загружает в то же окно, в котором был сделан щелчок на гиперссылке. Значение “_self” определено по умолчанию. Например:

```
<a href="1.htm" target="_blank"> На первую </a>
```

4.2. Добавление и оформление изображений

Для того, чтобы вставить в web-страницу изображение, необходимо иметь изображение в формате gif (файл с расширением *.gif) или jpeg (файл с расширением *.jpeg или *.jpg). Может также использоваться изображение формата PNG (*.png).

Вставка изображения осуществляется с помощью тега , который не имеет закрывающего тега. Пример включения в документ изображения из файла pic.gif, находящегося в одном каталоге с HTML-документом:

```

```

Атрибут src определяет имя файла. Кроме того, может быть атрибут alt="текст", содержащий альтернативный текст (Еще можно добавить атрибут title для появления всплывающей подсказки). Например:

```

```

```

```

Изображения на Web-странице могут использоваться как гиперссылки. Для обозначения изображения как гиперссылки используют тег <a>, где между <a> и вставляется тег изображения :

```
<a href="адрес файла или изображения">
```

```

```

```
</a>
```

При этом изображение, используемое в качестве гипертекстовой ссылки, обводится дополнительной рамкой. Например:

```
<a href="1.htm">
```

```

```

```
</a>
```

Пример: задать заголовок текста как заголовок первого уровня и рядом с ним картинку, являющуюся ссылкой на первую страницу

```
<h1 align=center> УЧЕБНЫЙ ЦЕНТР "КАДРЫ"  
  <a href="str1.html">  
      
  </a>  
</h1>
```

Тег изображения имеет один обязательный атрибут `src` и необязательные атрибуты.

`src` = "адрес файла". Указывает файл изображения и путь к нему; изображение должно быть загружено в браузер и размещено в том месте документа, где расположен тег изображения.

`alt` = "альтернативный текст". Позволяет указать текст, который будет выводиться вместо изображения браузерами, в настройках которых отключен показ изображений. В некоторых случаях при недостаточной пропускной способности линий связи пользователи отключают отображение графики. Наличие названий вместо картинок облегчает восприятие web-страниц в таком режиме.

Атрибут `align` определяет положение изображения относительно окружающего его текста. Основные значения атрибута:

`left` – определяет огибаемое текстом изображение. Изображение располагается вдоль левой границы документа, а последующие строки текста огибают его справа;

`right` – определяет огибаемое текстом изображение. Изображение располагается вдоль правой границы документа, а последующие строки текста огибают его слева;

`top` – выравнивает верх изображения по верхнему краю самого высокого элемента в строке окружающего текста.

`middle` – выравнивает центр изображения по базовой линии строки окружающего текста.

`bottom` – выравнивает нижний край изображения по базовой линии строки окружающего текста.

Кроме основных значений атрибута существует еще ряд аргументов, которые расширяют возможности взаимного размещения графики и текста.

Атрибут border определяет толщину рамки вокруг изображения. Если значение равно нулю, рамка отсутствует. Чтобы не вводить пользователей в заблуждение, не стоит задействовать атрибут border в изображениях, которые представляют собой часть ссылки, поскольку рисунки, применяемые в качестве гиперссылок, обычно выделяются цветной рамкой.

Атрибут hspace задает горизонтальное расстояние между вертикальной границей страницы и изображением, а также между изображением и огибающим его текстом.

Атрибут vspace задает вертикальное расстояние между строками текста и изображением.

Атрибут width и height задают значения размеров изображения по горизонтали и по вертикали соответственно.

Пример применения аргументов.

```

```

Создание карты изображения. Карта изображения – это изображение с так называемыми активными областями, которые ссылаются на URL других страниц или узлов.

Порядок создания активного изображения следующий:

1. Вставить изображение на страницу с атрибутом usemap, который должен содержать имя карты графических ссылок с предварительным знаком #. Когда карта еще не создана, ей можно дать любое имя. Например:

```

```

2. Создать карту графических ссылок, которая начинается с тега <map> и завершается закрывающим тегом </map>. В теге <map> следует установить единственный возможный атрибут name. Его значением должно быть имя карты. Например:

<map name="map1">

3. Между тегами <map> и </map> должна находиться основная часть карты. Она состоит из нескольких тегов <area>, каждый из которых определяет какую-то активную область рисунка. При щелчке мышью на «активной области» происходит переход по гиперссылке.

4. «Активные области» могут быть прямоугольной формы, либо круглой, либо многоугольной. Это определяется установкой значения атрибута shape в теге <area> (он используется без закрывающего тега). Если значение этого атрибута – "rect", область будет прямоугольной, если "circle" – круглой, а если "poly" – многоугольной.

5. Для каждой создаваемой области нужно определить координаты области и записать их как значения атрибута coords. Если «активная область» прямоугольная, то следует указать координаты левого верхнего ее угла и правого нижнего. Местоположение любой точки на рисунке может быть определено с помощью двух чисел – ее горизонтальной и вертикальной координат. Самая левая верхняя точка на рисунке имеет координаты «0,0». Чем больше горизонтальная координата, тем правее расположена точка, и, соответственно, чем больше вертикальная координата, тем точка ниже. Например, если рисунок имеет размер 400x300 точек и нужно определить прямоугольную активную область, занимающую его левый верхний угол и имеющую половину ширины и высоты рисунка, то необходимо написать следующее:

```
<area shape="rect" coords = "0,0,200,150">
```

Если «активная область» круглая, то нужно задать координаты центра окружности, а также указать ее радиус, например:

```
<area shape="circle" coords = "200,150,40">
```

Такая запись определяет круглую область, расположенную в центре рисунка и имеющую окружность радиусом 40.

Если «активная область» - многоугольная, то необходимо указать координаты всех углов многоугольника.

6. В теге <area> необходимо установить атрибут href, определяющий гиперссылку.

7. Полезно определить также атрибут alt для появления всплывающего текста, поясняющего, куда будет происходить переход.

Пример создания двух «активных областей» на изображении размером 77 на 109.

```
<html>
<body>
  <h1>
    Создание карты-изображения
  </h1>
  <map name="мумар">
    <area alt="файл p1" shape="rect" coords="0,0,35,50"
href="p1.html">
    <area alt="файл p2" shape="rect" coords="35,0,77,109"
href="p2.html">
  </map>
  
</body>
</html>
```

4.3. Варианты заданий

Общее задание:

Требуется создать сайт состоящий из четырех взаимосвязанных страниц, выдержанных в едином стиле. На страницах должны быть размещены ссылки-изображения, позволяющие осуществить переходы между страницами, иллюстрации, текст и таблицы.

Вариант 1

1. Создать для сайта отдельную папку, а внутри нее подкаталоги Pictures и Buttons. (В данные каталоги в дальнейшем сохранять вставляемые изображения и кнопки).

2. Создать web-страницу со структурой, приведенной ниже. Структура создается путем вставки таблицы с атрибутами width=100% и height=100% с невидимыми границами.

Изображение – Логотип KIA Motors	АВТОСАЛОН «АВТОСТИЛЬ»			
	Главная	О компа- нии	Модельный ряд	Прайс- лист

3. Страницу назвать следующим образом:

– index.htm.

4. Создать еще 3 пустые web-страницы:

– comp.htm;

– model.htm;

– price.htm.

5. В ячейки таблицы, вместо надписей «Главная», «О компании», «Модельный ряд», «Прайс-лист» вставить изображения соответствующих кнопок. Сделать изображения ссылками на вышеуказанные страницы.

6. Установить цвет фона и символов самостоятельно.

7. Перекопировать созданное содержимое страницы **index.htm** в остальные, чтобы все страницы были одинаковой структуры, различаясь лишь ячейкой основного содержания.

8. Заполнить ячейки основного содержания на всех страницах.

8.1. Страница **index.htm** должна содержать следующее:

ДОБРО ПОЖАЛОВАТЬ НА САЙТ ООО «АВТОСТИЛЬ»!

Вставить изображение (сделать карту-изображение со ссылками на 2-ю, 3-ю и 4-ю страницы)	О компании
	Модельный ряд
	Прайс-лист

В конце страницы вставить бегущую строку со сведениями об авторских правах.

8.2. Страница **comp.htm** должна содержать следующее:

О КОМПАНИИ Изображение

На сегодняшний день ООО "АвтоСтиль" является официальным дилером по продаже корейских автомобилей марки **KIA Motors (Киа Моторс)**, которые в настоящее время являются оптимальными в соотношении цена/качество.

В просторном и красивом автосалоне, оформленном в современном стиле, площадью более 300 квадратных метров, представлен практически весь модельный ряд автомобилей KIA Motors, производимых этой фирмой в Корее.

Наличие различных моделей, разнообразных цветов и комплектаций, дает возможность покупателям осмотреть несколько автомобилей и выбрать для себя именно тот, который наиболее полно удовлетворяет его запросам.

Нижеследующий текст вставить в таблицу с невидимыми границами:

Адрес	г. Воронеж, пр. Патриотов, 47
Телефон	(4732) 204-000, 399-599
Адрес в Интернете	Kia-voronezh.ru

8.3. Страница model.htm должна содержать следующее:

МОДЕЛЬНЫЙ РЯД

KIA Picanto Двигатель 4 цилиндра, 12 клапанов Объем (см3) 999 Макс. мощность (л.с./об.мин.) 62/5500 Макс. скорость (км/ч) 150	Изображение
KIA Rio Двигатель 4 цилиндра, 16 клапанов Объем (см3) 1339 Макс. мощность (л.с./об.мин.) 97/6000 Макс. скорость (км/ч) 177	Изображение
KIA Cerato Двигатель 4 цилиндра, ДОНС 16 клапанов	Изображение

<p>Объем (см3) 1599</p> <p>Макс. мощность (л.с./об.мин.) 105/5800</p> <p>Макс. скорость (км/ч) 186</p>	
<p>KIA Magentis Двигатель 4 цилиндра, ДОНС 16 клапанов</p> <p>Объем (см3) 1998</p> <p>Макс. мощность (л.с./об.мин.) 144/1600</p> <p>Макс. скорость (км/ч) 208</p>	Изображение
<p>KIA Sorento Двигатель V-образный, 4 цилиндра, ДОНС 24 клапана</p> <p>Объем (см3) 3294</p> <p>Макс. мощность (л.с./об.мин.) 238/6000</p> <p>Макс. скорость (км/ч) 190</p>	Изображение

Назад

(ссылка на верхнюю часть страницы – сделать с помощью якоря)

8.4. Страница price.htm должна содержать следующее:

ПРАЙС-ЛИСТ

Изображение

108

Название	Минимальная цена, р.	Максимальная це- на, р.
KIA Picanto	329000	422000
KIA Rio	439000	499000
KIA Cerato	559000	639000
KIA Magentis	777000	873000
KIA Sorento	839990	1149990

Вариант 2

1. Создать для сайта отдельную папку, а внутри нее под-каталоги Pictures и Buttons.

2. Создать web-страницу следующей структуры. Структура создается путем вставки таблицы с атрибутами width=100% и height=100% с невидимыми границами.

Изображение – Логотип AUDI	КОМПАНИЯ «АУДИ-СЕРВИС ВОРОНЕЖ»			
	Главная	О компа- нии	Модельный ряд	Прайс- лист

3. Страницу назвать следующим образом:

– index.htm.

4. Создать еще 3 пустые web-страницы:

– comp.htm;

- model.htm;
- price.htm.

5. В ячейки таблицы, вместо надписей «Главная», «О компании», «Модельный ряд», «Прайс-лист» вставить изображения соответствующих кнопок. Сделать изображения ссылками на вышеуказанные страницы.

6. Установить цвет фона и символов самостоятельно.

7. **Перекопировать созданное содержимое страницы index.htm в остальные, чтобы все страницы были одинаковой структуры, различаясь лишь ячейкой основного содержания.**

8. Заполнить ячейки основного содержания на всех страницах.

8.1. Страница index.htm должна содержать следующее:

**ДОБРО ПОЖАЛОВАТЬ НА САЙТ КОМПАНИИ
«АУДИ-СЕРВИС ВОРОНЕЖ»**

Вставить изображение (сделать карту-изображение со ссылками на 2-ю, 3-ю и 4-ю страницы)	О компании
	Модельный ряд
	Прайс-лист

В конце страницы вставить бегущую строку со сведениями об авторских правах.

8.2. Страница comp.htm должна содержать следующее:

**О КОМПАНИИ
Изображение**

На сегодняшний день Ауди Сервис Воронеж является единственным в Воронеже и Центрально-Черноземном регионе специализированным центром по продаже и сервисному обслуживанию автомобилей Audi.

Удобное географическое месторасположение, наличие выставочного автосалона, технического центра и всех необходимых для обслуживания запасных частей, а также грамотный

персонал привлекает внимание не только воронежцев, но и жителей других регионов - Липецка, Курска, Белгорода и Тамбова.

Мы стараемся максимально быстро и качественно реагировать на пожелания клиентов, поскольку считаем, что бизнес - это прежде всего люди, для которых мы работаем.

Нижеследующий текст вставить в таблицу с невидимыми границами:

Адрес	г. Воронеж, ул. Остужева, 64
Телефон	(4732) 21-77-77, 21-77-76
Адрес в Интернете	http://www.audi-voronezh.ru

8.3. Страница model.htm должна содержать следующее:

МОДЕЛЬНЫЙ РЯД

<p>Audi A4 Двигатель Рядный 4-цилиндровый двигатель с газотурбонагнетателем и интеркулером DONC Объем (см3) 1798 Макс. мощность (л.с./об.мин.) 118 / 4500 – 6200 Макс. скорость (км/ч) 225</p>	Изображение
<p>Audi A6 Двигатель V6, бензиновый Объем (см3) 2393 Макс. мощность (л.с./об.мин.) 177/6000 Макс. скорость (км/ч) 231</p>	Изображение

<p>Audi A8 Двигатель 8-цилиндровый, V-образный, бензиновый Объем (см3) 3697 Макс. мощность (л.с./об.мин.) 280/6000 Макс. скорость (км/ч) 250</p>	<p>Изображение</p>
<p>Audi Q7 Двигатель V6, дизельный, турбонаддув, технология common rail Объем (см3) 2967 Макс. мощность (л.с./об.мин.) 233/4000 Макс. скорость (км/ч) 210</p>	<p>Изображение</p>
<p>Audi TT Двигатель 4-цилиндровый, рядный, бензи- новый с непосредственным впрыском топлива, турбонаддув с охлаждением нагнетаемого воздуха, DOHC Объем (см3) 1984 Макс. мощность (л.с./об.мин.) 200/5100-6000 Макс. скорость (км/ч) 240</p>	<p>Изображение</p>

(ссылка на верхнюю часть страницы – сделать с помощью якоря)

8.4. Страница price.htm должна содержать следующее:

ПРАЙС-ЛИСТ Изображение

Название	Минимальная цена, р.	Максимальная цена, р.
Audi A4	1113200	1921700
Audi A6	1587200	2223700
Audi A8	3281190	4945270
Audi Q7	2634000	3584900
Audi TT	1580400	2365200

Вариант 3

1. Создать для сайта отдельную папку, а внутри нее подкаталоги Pictures и Buttons.

2. Создать web-страницу следующей структуры. Структура создается путем вставки таблицы с атрибутами width=100% и height=100% с невидимыми границами.

Изображение – Логотип TOYOTA	КОМПАНИЯ «ТОУОТА ЦЕНТР ВОРОНЕЖ»			
	Главная	О компа- нии	Модельный ряд	Прайс- лист

3. Страницу назвать следующим образом:

– index.htm.

4. Создать еще 3 пустые web-страницы:

– comp.htm;

– model.htm;

– price.htm.

5. В ячейки таблицы, вместо надписей «Главная», «О компании», «Модельный ряд», «Прайс-лист» вставить изображения соответствующих кнопок. Сделать изображения ссылками на вышеуказанные страницы.

6. Установить цвет фона и символов самостоятельно.

7. **Перекопировать созданное содержимое страницы index.htm в остальные, чтобы все страницы были одинаковой структуры, различаясь лишь ячейкой основного содержания.**

8. Заполнить ячейки основного содержания на всех страницах.

8.1. Страница index.htm должна содержать следующее:

**ДОБРО ПОЖАЛОВАТЬ НА САЙТ КОМПАНИИ
«ТОУОТА ЦЕНТР ВОРОНЕЖ»!**

Вставить изображение (сделать карту-изображение со ссылками на 2-ю, 3-ю и 4-ю страницы)	О компании
	Модельный ряд
	Прайс-лист

В конце страницы вставить бегущую строку со сведениями об авторских правах.

8.2. Страница comp.htm должна содержать следующее:

**О КОМПАНИИ
Изображение**

ООО «Бизнес Кар Воронеж» - владелец Торгово-Технического Центра Тойота Центр Воронеж по продаже и сервисному обслуживанию автомобилей марки Toyota, продажи запчастей, аксессуаров и лакокрасочных материалов.

Наш дилерский центр, общая площадь которого составляет 3000 м², включает в себя демонстрационный зал, склад запасных частей и аксессуаров и сервисную зону.

Тойота Центр Воронеж выполнен в соответствии с единой концепцией оформления шоу-рума Toyota, в основе которого лежит принцип «Customer First» (Клиент - на первом месте). В соответствии с данной концепцией, основной задачей дилерского центра является не только представление всего модельного ряда автомобилей Toyota, продаваемого в России, но и обеспечение каждому гостю максимального комфорта и удобства.

Нижеследующий текст вставить в таблицу с невидимыми границами:

Адрес	г. Воронеж, ул. Остужева, 64
Телефон	(4732) 44-60-60
Адрес в Интернете	http://www.toyota-vrn.ru/

8.3. Страница model.htm должна содержать следующее:

МОДЕЛЬНЫЙ РЯД

Toyota Corolla	Изображение
Двигатель 4-цилиндровый, рядный, 16 клапанов, два верхних распределительных вала (DOHC) Объем (см³) 1329 Макс. мощность (л.с./об.мин.) 101/6000 Макс. скорость (км/ч) 180	

<p>Toyota Avensis Двигатель 4-цилиндровый, рядный, 16 клапанов, VVT-i, DOHC с цепным приводом Объем (см3) 1798 Макс. мощность (л.с./об.мин.) 129/6000 Макс. скорость (км/ч) 200</p>	<p>Изображение</p>
<p>Toyota Camry Двигатель 4-цилиндровый, рядный, Два верхних распределительных вала (DOHC), электронная система изменения фаз газораспределения (VVT-i), 16 клапанов Объем (см3) 2362 Макс. мощность (л.с./об.мин.) 167/6000 Макс. скорость (км/ч) 210</p>	<p>Изображение</p>
<p>Toyota Land Cruiser Prado Двигатель 6-цилиндровый, V-образный, С двумя распределительными валами (DOHC), 24 клапана, VVT-i Объем (см3) 3956 Макс. мощность (л.с./об.мин.) 249/5200</p>	<p>Изображение</p>

Макс. скорость (км/ч) 180	
Toyota Auris Двигатель 4-цилиндровый, рядный, 16-клапанный механизм с двумя верхними распредвалами, с системой VVT-i Объем (см3) 1329 Макс. мощность (л.с./об.мин.) 101/6000 Макс. скорость (км/ч) 175	Изображение

Назад

(ссылка на верхнюю часть страницы – сделать с помощью якоря)

8.4. Страница price.htm должна содержать следующее:

ПРАЙС-ЛИСТ Изображение

Название	Минимальная цена, р.	Максимальная цена, р.
Toyota Corolla	533000	691000
Toyota Avensis	860000	1387000
Toyota Camry	853000	1273000
Toyota Land Cruiser Prado	1578200	1762000
Toyota Auris	710000	896000

Вариант 4

1. Создать для сайта отдельную папку, а внутри нее подкаталоги Pictures и Buttons.

2. Создать web-страницу следующей структуры. Структура создается путем вставки таблицы с атрибутами width=100% и height=100% с невидимыми границами.

Изображение – Логотип MITSUBISHI	КОМПАНИЯ «МАРТЕН АВТО»			
	Главная	О компа- нии	Модельный ряд	Прайс -лист

3. Страницу назвать следующим образом:

– index.htm.

4. Создать еще 3 пустые web-страницы:

– comp.htm;

– model.htm;

– price.htm.

5. В ячейки таблицы, вместо надписей «Главная», «О компании», «Модельный ряд», «Прайс-лист» вставить изображения соответствующих кнопок. Сделать изображения ссылками на вышеуказанные страницы.

6. Установить цвет фона и символов самостоятельно.

7. Перекопировать созданное содержимое страницы index.htm в остальные, чтобы все страницы были одинаковой структуры, различаясь лишь ячейкой основного содержания.

8. Заполнить ячейки основного содержания на всех страницах.

8.1. Страница index.htm должна содержать следующее:

**ДОБРО ПОЖАЛОВАТЬ НА САЙТ КОМПАНИИ
«МАРТЕН АВТО»!**

Вставить изображение (сделать карту-изображение со ссылками на 2-ю, 3-ю и 4-ю страницы)	О компании
	Модельный ряд
	Прайс-лист

В конце страницы вставить бегущую строку со сведениями об авторских правах.

8.2. Страница comp.htm должна содержать следующее:

**О КОМПАНИИ
Изображение**

Компания **Мартен-Авто** — официальный дилер Mitsubishi Motors Corporation в Вологодском и Воронежском регионах.

Высокое качество обслуживания, доступные цены, разумные сроки исполнения заказов, а также широкий спектр сопутствующих услуг — вот те преимущества, которыми пользуются наши клиенты. Для реализации этой задачи компания располагает обученным персоналом, как в области продаж, так и в области обслуживания и ремонта автомобилей.

В «Мартен-Авто» внимательно следят за тенденциями в автобизнесе, стремясь использовать в работе всё лучшее. Цель компании Мартен-Авто — обеспечивать растущие потребности наших сограждан в качественных автомобилях, а также квалифицированное обслуживание этих автомобилей в течение всего срока службы.

Нижеследующий текст вставить в таблицу с невидимыми границами:

Адрес	г. Воронеж, ул. 9 января, 221Д
Телефон	(4732) 62-90-90

Адрес в Интернете	http://www.marten-mitsubishi.ru
-------------------	---

8.3. Страница model.htm должна содержать следующее:

МОДЕЛЬНЫЙ РЯД

<p>Mitsubishi Colt Двигатель Mitsubishi 4A90: бензиновый, 4-цилиндровый, 16-клапанный, DOHC Объем (см3) 1332 Макс. мощность (л.с./об.мин.) 95/6000 Макс. скорость (км/ч) 180</p>	Изображение
<p>Mitsubishi Lancer X Двигатель бензиновый, рядный, 4- цилиндровый, 16-клапанный, MIVEC, Евро-4 Объем (см3) 1499 Макс. мощность (л.с./об.мин.) 109/6000 Макс. скорость (км/ч) 191</p>	Изображение
<p>Mitsubishi Galant Двигатель бензиновый, рядный, 4- цилиндровый, 16-клапанный, MIVEC Объем (см3) 2378</p>	Изображение

<p>Макс. мощность (л.с./об.мин.) 158/5500</p> <p>Макс. скорость (км/ч) 200</p>	
<p>Mitsubishi Outlander Двигатель Бензиновый, 4-цилиндровый, рядный, 16-клапанный, с двумя верхними распредвалами и системой впрыска ECI- MULTI Объем (см3) 1997 Макс. мощность (л.с./об.мин.) 136/6000 Макс. скорость (км/ч) 192</p>	Изображение
<p>Mitsubishi Pajero Sport Двигатель Дизельный, 4-цилиндровый, с турбокомпрессором и проме- жуточным охладителем Объем (см3) 2477 Макс. мощность (л.с./об.мин.) 116/4000 Макс. скорость (км/ч) 150</p>	Изображение

Назад

(ссылка на верхнюю часть страницы – сделать с помощью
якоря)

8.4. Страница price.htm должна содержать следующее:

ПРАЙС-ЛИСТ Изображение

Название	Минимальная цена, р.	Максимальная це- на, р.
Mitsubishi Colt	449000	569000
Mitsubishi Lancer X	539000	834000
Mitsubishi Galant	759000	908000
Mitsubishi Out- lander	849000	1190900
Mitsubishi Pajero Sport	941000	1079000

5. СОЗДАНИЕ WEB-САЙТА С ИСПОЛЬЗОВАНИЕМ ФОРМ

5.1. Создание элементов форм

Формы создают и используют для получения отклика пользователя на предоставленную информацию и для обеспечения диалога с пользователем.

Форма – это инструмент, с помощью которого HTML-документ может послать некоторую информацию в некоторую заранее определенную точку внешнего мира, где информация будет некоторым образом обработана.

Формы передают информацию программам-обработчикам в виде пар:

[имя переменной]=[значение переменной].

Имена переменных следует задавать латинскими буквами. Значения переменных воспринимаются обработчиками как строки, даже если они содержат только цифры.

Задание формы осуществляется с помощью тега `<form>`. Форма определяется последовательностью тегов `input`, размещенных внутри пары `<form>` и `</form>`. Например:

```
<form>
  Введите фамилию <input type=«text» name=NameF>
  Введите имя <input type=«text» name=NameI>
</form>
```

HTML-документ может содержать в себе несколько форм, однако формы не должны находиться одна внутри другой.

Тег `form` может содержать три атрибута.

`action` – атрибут определяет, где находится обработчик формы (указывает URL того приложения, которое будет взаимодействовать с формой). Например:

```
<form action=»pro1.html»>
```

`method` – атрибут определяет, каким образом (иначе говоря, с помощью какого метода протокола передачи гипертекстов) данные из формы будут переданы обработчику. Допустимые значения: `post` и `get`. Если значение атрибута не установлено, по умолчанию предполагается `method=get`. Разница между ними состоит в том, что при значении `get` вся информация помещается в специальную переменную среды с именем `query_string`, которая добавляется к адресу URL, указанному в атрибуте `action`, а при значении `post` данные передаются с помощью специальной HTTP-транзакции `post`.

`enctype` – атрибут определяет, каким образом данные из формы будут закодированы для передачи обработчику. Если значение атрибута не установлено, по умолчанию предполагается `enctype=application/x-www-form-unrecorded`. При использовании данной кодировки пробелы в данных заменяются на символы «+», а все другие непечатаемые символы – на комбинацию символа «%» и шестнадцатеричного кода, соответствующего данному символу в ASCII-таблице.

Тег **<input>** определяет область внутри формы, куда вводят данные. Каждый тег **<input>** должен включать атрибут **name** = [имя], определяющий имя переменной, через которую будет передаваться обработчику. Имя должно задаваться только латинскими буквами.

Большинство тегов **<input>** должно включать атрибут **value**="значение", определяющий значение, которое будет передано обработчику под этим именем. Для тегов **<input type=text>** и **<input type=password>**, этот атрибут не обязателен, поскольку значение соответствующей переменной может вводиться пользователем с клавиатуры.

Тег **<input>** формирует поле для ввода информации пользователем. Это может быть текстовое поле; поле-флажок, группа переключателей, изображение или кнопка. Вид поля ввода определяется значением атрибута **type**.

Атрибут type=text. Когда необходимо ввести небольшое количество текста (одну или несколько строк) используется тег **input** с атрибутом **type**, имеющим значение **text**. Это значение принято по умолчанию и указывать его необязательно.

Также указывается атрибут **name** для определения наименования переменной поля. Например:

Введите название **<input type=«text» name=«nazv»>**

или

Введите название **<input name=«nazv»>**

Существуют дополнительные атрибуты:

атрибут **maxlength** ограничивает число символов, вводимых пользователем в текущее поле. По умолчанию данное число не ограничено;

атрибут **size** определяет размер видимой на экране области, занимаемой текущим полем. Значение по умолчанию определяется типом браузера. Если значение **maxlength** больше, чем **size**, браузер будет прокручивать данные в окне;

атрибут **value** определяет начальное значение поля ввода.

Например, формирование двух полей для ввода символьных данных осуществляется следующим образом:

```
<form>
  Ваша фамилия <input name=«namef» type=«text»
size=«35»>
  <br>Ваше имя <input name=«namen» type=«text»
size=«20»>
</form>
```

Разновидностью текстового поля можно считать поле для ввода пароля, которое создается при значении атрибута TYPE="password". При наборе текст в данном поле скрыт символами «*».

Атрибут type=checkbox позволяет создать поле-флажок, позволяющее получить ответ на вопрос типа «Да/Нет». Форма может содержать несколько полей-флажков для ввода данных. Когда форма использует тег <input> с атрибутом checkbox, в ней должны присутствовать и атрибуты name и value. Атрибут name указывает на наименование данного поля (флага) ввода. В атрибуте VALUE будет содержаться значение поля. Пример использования:

```
Россия <input name="strana1" type="checkbox"
value="Россия">
```

```
Страны СНГ <input name="strana2" type="checkbox" value="СНГ">
```

В некоторых случаях необходимо инициализировать данный флаг, как уже отмеченный. В таких случаях тег <input> должен содержать атрибут checked. Пример использования:

```
Россия<input name="strana" type="checkbox"
value="Россия" checked>
```

Атрибут type=radio позволяет организовать выбор одного из нескольких возможных значений. Для создания формы ввода при выборе пользователем одного значения из нескольких возможных необходимо использовать тег <input> с атрибутом type="radio". Когда в форме применяется данный атрибут, в теге <input> должны быть указаны атрибуты name и value. Атрибут name указывает наименование соответствующего поля

(кнопки). Атрибут `value` содержит значение поля. Пример использования:

```
Пол мужской<input name="pol" type="radio" value="мужской">
```

```
Пол женский<input name="pol" type="radio" value="женский">
```

Атрибут `type=image` позволяет создать в форме изображение (например, пользователь может щелкнуть мышью на изображении, чтобы завершить работу с формой). Когда пользователь щелкает мышью по изображению, браузер сохраняет координаты соответствующей точки экрана. Далее он «обрабатывает» введенную в форму информацию. Когда форма использует атрибут `image`, тег `<input>` должен содержать также атрибуты `name` и `src`. `name` указывает наименование поля ввода формы. Атрибут `src` содержит имя файла – источник изображения. Пример использования:

```
<br>Для выхода нажмите на кнопку<input type="image" name=point src="ris1.bmp">
```

Атрибут `type=button` позволяет создать на форме простую кнопку. Надпись на кнопку добавляется с помощью атрибута `VALUE`.

Атрибут `type=file` позволяет создать поле для ввода имени файла, который пересылается на сервер. Рядом с полем автоматически добавляется кнопка с именем «Обзор...», которая позволяет выбрать имя файла через диалог.

Атрибут `type=reset` используют для возможности начать ввод информации сначала и очистить от данных уже введенные поля. Если в форме используется атрибут `reset`, тег `<input>` может дополнительно содержать атрибут `value`. Данный атрибут определяет надпись на изображении кнопки, нажатие на которую будет очищать содержимое полей формы. Пример использования:

```
<input type=reset value="Очистить форму">
```

Атрибут `type=submit` обеспечивает процесс передачи данных из формы обработчику. Браузер выводит данный эле-

мент как кнопку, по которой пользователь может щелкнуть, чтобы все имеющиеся в форме данные будут переданы обработчику, определенному в теге `<form>`. Когда в форме используется тег `<input>` с атрибутом `submit`, данный элемент может содержать два дополнительных атрибута: `name` и `value`. Атрибут `value` указывает наименование кнопки `Submit`. Пример использования:

```
<br><input type=submit value="Закончить ввод">
```

Надпись, нанесенную на кнопку, можно при необходимости передать обработчику путем введения в определение кнопки атрибута `name=[имя]`. Например:

```
<input type="submit" name="button" value="Ввод">
```

При нажатии на такую кнопку обработчик вместе со всеми остальными данными получит и переменную `button` со значением `Ввод`.

В форме может быть несколько кнопок типа `submit` с различными именами и/или значениями. Обработчик, таким образом, может действовать по-разному в зависимости от того, какую кнопку `submit` нажал пользователь.

Для того, чтобы создать простую кнопку следует в атрибуте `type` указать значение **button**.

Тег `<select>` используется для создания поля со списком. Для этого применяют тег `select` с атрибутом `type=select`. Для определения списка используют тег `<option>`. Тег `<select>` поддерживает три необязательных атрибута: `multiple`, `name`, `size`.

Атрибут `multiple` позволяет выбрать более чем одно наименование. Атрибут `name` определяет наименование объекта. Атрибут `size` определяет число видимых пользователю пунктов списка. Если в форме установлено значение атрибута `size=1`, то браузер выводит на экране список в виде выпадающего меню. В случае `size > 1` браузер представляет на экране обычный список.

В форме может использоваться тег `<option>` только внутри тега `<select>`. Эти теги поддерживают два дополнительных атрибута: `selected` и `value`.

Атрибут `selected` используется для первоначального выбора значения элемента по умолчанию. Атрибут `value` указывает на значение, возвращаемое формой после выбора пользователем данного пункта. По умолчанию значение поля равно значению тега `<option>`. Пример использования:

```
<br>
ВУЗ<select name="vuz">
  <option value="ВГУ">ВГУ</option>
  <option value="ВГТУ" selected>ВГТУ</option>
  <option value="ВГАУ">ВГАУ</option>
  <option value="ВГТА">ВГТА</option>
  <option value="ВГПУ">ВГПУ</option>
</select>
```

В примере формируется поле со списком, которое имеет подпись «ВУЗ», результат выбора сохраняется в переменной `vuz`. По умолчанию переменная имеет значение «ВГТУ».

Пример текста программы для формирования формы для ввода данных:

```
<h1> Форма для ввода данных</h1>
<form>
  Ваша фамилия <input name="namef" type="text"
  size="35">
  <br>Ваше имя <input name="namen" type="text"
  size="20">
  <br>Россия <input name="strana" type="checkbox"
  value="Россия" checked>
  Страны СНГ <input name="strana" type="checkbox"
  value="СНГ">
  <br>Пол мужской<input name="pol" type="radio"
  value="мужской">
  Пол женский<input name="pol" type="radio"
  value="женский">
  <br>
  ВУЗ<select name="vuz">
  <option value="ВГУ">ВГУ</option>
```



```

<option value="ВГТУ" selected>ВГТУ</option>
<option value="ВГАУ">ВГАУ</option>
<option value="ВГТА">ВГТА</option>
<option value="ВГПУ">ВГПУ</option>
</select>
<br>Для выхода нажмите на кнопку<input type="image"
name="point" src="ris1.bmp">
<p>
<input type="reset" value="Очистить форму">
</p>
<br><input type="submit" value="Закончить ввод или редак-
тирование">
</form>

```

В результате выполнения данной программы формируется форма, представленная на рис. 42.

Форма для ввода данных

Ваша фамилия

Ваше имя

Россия Страны СНГ

Пол мужской Пол женский

ВУЗ


Для выхода нажмите на кнопку 

Рис. 42. Пример формы для ввода данных

5.2. Варианты заданий

Общее задание

Разработать сайт, состоящий из нескольких взаимосвязанных страниц, выдержанных в едином стиле. На всех страницах сайта (кроме главной) разместить формы для ввода данных, содержащие различные элементы согласно варианту задания.

Вариант 1

1. Первая страница имеет следующий вид (**структуру страницы задать в виде таблицы – остальные страницы должны быть сделаны по такому же принципу, только в ячейке основного содержания должны находиться элементы формы для ввода данных**):

ВОРОНЕЖКРЕДИТБАНК	
Главная Личные данные Контактная информация Образование и работа	На данном сайте Вы можете оформить заявление на получение кредита в ООО «ВоронежКредитБанк». Вам необходимо заполнить следующие формы: – Личные данные заявителя; – Контактная информация заявителя; – Образование и работа заявителя.
Сайт создан студентом группы... Фамилия, инициалы	

Содержание ячейки навигации:

ссылка **Главная** – это ссылка на страницу main.html;

ссылка **Личные данные** – ссылка на страницу persondata.html;

ссылка **Контактная информация** – ссылка на страницу contact.html;

ссылка **Образование и работа** – ссылка на страницу rabota.html.

2. Основное содержание второй страницы (persondata.html) имеет следующий вид (элементы формы рекомендуется разместить в таблице с нулевой границей)

Вид кредита ▼

Сумма кредита

Срок кредита, месяцев Валюта кредита

ЛИЧНЫЕ ДАННЫЕ ЗАЯВИТЕЛЯ

Фамилия

Имя

Отчество

Дата рождения Пол М Ж

Место рождения

Гражданство

Паспорт: Серия Номер

Дата выдачи Код подразделения

Кем выдан

Водительское удостоверение Серия

Номер Дата выдачи

ИНН

Поле **Вид кредита** сделать полем со списком (автокредит, ипотечный кредит, краткосрочный кредит, долгосрочный кредит).

Поле **Пол** сделать переключателем.

В нижней части страницы разместить две кнопки – **Очистить** и **Добавить**.

3. Основное содержание третьей страницы (contact.html) имеет следующий вид (элементы формы рекомендуется разместить в таблице с нулевой границей):

АДРЕС И КОНТАКТНАЯ ИНФОРМАЦИЯ

Адрес регистрации

Индекс

Регион

Район

Населенный пункт

Улица

Дом

Квартира

Контактная информация

e-mail

Телефон по месту
проживания

Служебный телефон

Мобильный телефон

В нижней части страницы разместить две кнопки – **Очистить** и **Добавить**.

4. Основное содержание четвертой страницы (rabota.html) имеет следующий вид (элементы формы рекомендуется разместить в таблице с нулевой границей):

ОБРАЗОВАНИЕ И РАБОТА

Образование

Незаконченное среднее	<input type="checkbox"/>	Среднее	<input type="checkbox"/>
Среднее специальное	<input type="checkbox"/>	Неполное высшее	<input type="checkbox"/>
Высшее	<input type="checkbox"/>	Несколько высших	<input type="checkbox"/>
Ученая степень	<input type="checkbox"/>		

Отношение к воинской службе

Отслужил/запас	<input type="radio"/>	Имею отсрочку	<input type="radio"/>
Призывник	<input type="radio"/>	Невоеннообязанный	<input type="radio"/>

Информация о работе заявителя

Временная работа	<input type="radio"/>	Постоянная работа	<input type="radio"/>
------------------	-----------------------	-------------------	-----------------------

Форма организации			
ЗАО, ОАО	<input type="radio"/>	ООО	<input type="radio"/>
		Иное	<input type="radio"/>

Тип организации		Собственный бизнес	<input type="radio"/>
-----------------	--	--------------------	-----------------------

Государственное предприятие	<input type="radio"/>	Коммерческое предприятие	<input type="radio"/>
-----------------------------	-----------------------	--------------------------	-----------------------

Индивидуальный предприниматель Иное

Наименование организации

ИНН

Адрес организации

Индекс Населенный пункт

Улица Дом Строение

ФИО руководителя организации

Служебный телефон

Web-сайт

Название должности заявителя

Стаж работы на последнем месте ▼

Группу полей **Образование** оформить как поля-флажки (CheckBox).

Группы полей **Отношение к воинской службе, Информация о работе заявителя, Тип организации** оформить как группы переключателей (Radio).

Поле **Стаж работы** оформить как поле со списком, состоящее из следующих пунктов:

- до 0,5 года;
- от 0,5 до 1 года;
- от 1 до 3-х лет;

– свыше 3-х лет.

В нижней части страницы разместить две кнопки – **Очистить** и **Добавить**.

Вариант 2

1. Первая страница имеет следующий вид (структуру страницы задать в виде таблицы – остальные страницы должны быть сделаны по такому же принципу, только в ячейке основного содержания должны находиться элементы формы для ввода данных):

ИНСТИТУТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ	
Главная Личные данные Образование Родители	На данном сайте Вы можете анкету абитуриента для подачи заявления на поступление в институт информационных технологий. Вам необходимо заполнить следующие формы: – Личные данные абитуриента; – Сведения об образовании и желаемом факультете и специальности для поступления; – Сведения о родителях заявителя.
Сайт создан студентом группы... Фамилия, инициалы	

Содержание ячейки навигации:

ссылка **Главная** – это ссылка на страницу main.html;

ссылка **Личные данные** – ссылка на страницу persondata.html;

ссылка **Образование** – ссылка на страницу education.html;

ссылка **Родители** – ссылка на страницу parents.html;

2. Основное содержание второй страницы (persondata.html) имеет следующий вид(элементы формы рекомендуется разместить в таблице с нулевой границей):

ЛИЧНЫЕ ДАННЫЕ АБИТУРИЕНТА

Фамилия

Имя

Отчество

Дата рождения Пол М Ж

Место рождения

Гражданство РФ Иное

Паспорт: Серия Номер

Дата выдачи Код подразделения

Кем выдан

Адрес регистрации

Индекс

Регион

Район

Населенный пункт

Улица

Дом Квартира

Контактные телефоны

Группу полей **Пол** оформить как группу переключателей.

Группу полей **Гражданство** оформить как поля-флажки (CheckBox).

В нижней части страницы разместить две кнопки – **Очистить** и **Добавить**.

3. Основное содержание третьей страницы (education.html) имеет следующий вид (элементы формы рекомендуется разместить в таблице с нулевой границей):

ОБРАЗОВАНИЕ АБИТУРИЕНТА

Факультет куда хотите поступить

Специальность

Вид оконченого
учебного заведения

Название учебного заведения

Местонахождение учебного заведения

Дата окончания

Вид документа об окончании

Аттестат

Диплом

Другое

Имеется (отметить галочкой)

Золотая (серебряная) медаль об окончании школы

Диплом с отличием об окончании среднего специального или профессионально технического учебного заведения

Какой иностранный язык изучали

Оценки по предметам

Алгебра ▼ Геометрия ▼

Русский язык ▼ Физика ▼

Информатика ▼

Группу полей **Имеется...** оформить как поля-флажки (CheckBox).

Поле **Факультет** оформить как поле со списком – пункты придумать самостоятельно (3-4 пункта).

Поле **Вид учебного заведения** оформить как поле со списком, состоящее из следующих пунктов:

- средняя школа;
- лицей;
- гимназия;
- колледж;
- школа-интернат;
- другое.

Поле **Документ об окончании** оформить как группу переключателей.

Каждое поле из группы **Оценки по предметам** оформить как поле со списком, элементами которого являются целые значения от 3 до 5.

В нижней части страницы разместить две кнопки – **Очистить** и **Добавить**.

4. Основное содержание четвертой страницы (parents.html) имеет следующий вид(элементы формы рекомендуется разместить в таблице с нулевой границей):

СЕМЬЯ АБИТУРИЕНТА

ФИО отца

Дата рождения

Место работы (название организации)

Тип организации

Должность

Контактные телефоны

ФИО матери

Дата рождения

Место работы (название организации)

Тип организации

Должность

Контактные телефоны

Поле **Тип организации** оформить как поле со списком, состоящее из следующих пунктов:

- государственное предприятие;
- коммерческое предприятие;
- индивидуальный предприниматель;
- иное.

В нижней части страницы разместить две кнопки – **Очистить** и **Добавить**.

Вариант 3

1. Первая страница имеет следующий вид (структуру страницы задать в виде таблицы – остальные страницы должны быть сделаны по такому же принципу, только в ячейке основного содержания должны находиться элементы формы для ввода данных):

ВОРОНЕЖБИЗНЕСКОНСАЛТИНГ	
Главная Личные данные Образование и навыки Характеристики	На данном сайте Вы можете оформить анкету соискателя при приеме на работу в фирму «ВоронежБизнесКонсалтинг». Вам необходимо заполнить следующие формы: – Личные данные и контактная информация о соискателе; – Образование и навыки соискателя; – Характеристики соискателя.
Сайт создан студентом группы... Фамилия, инициалы	

Содержание ячейки навигации:

ссылка **Главная** – это ссылка на страницу main.html;

ссылка **Личные данные** – ссылка на страницу persondata.html;

ссылка **Образование и навыки** – ссылка на страницу education.html;

ссылка **Характеристики** – ссылка на страницу charact.html;

2. Основное содержание второй страницы (contact.html) имеет следующий вид (элементы формы рекомендуется разместить в таблице с нулевой границей):

**ЛИЧНЫЕ ДАННЫЕ И КОНТАКТНАЯ ИНФОРМАЦИЯ
О СОИСКАТЕЛЕ**

Должность, на которую Вы претендуете

Фамилия

Имя

Отчество

Дата рождения

Пол

М

Ж

Место рождения

Гражданство

РФ

Иное

Паспорт: Серия

Номер

Дата выдачи

Код подразделения

Кем выдан

Адрес регистрации

Район

Населенный пункт

Улица

Дом Квартира

Контактная информация

Контактные телефоны

E-mail

Группу полей **Пол** оформить как группу переключателей.

Группу полей **Гражданство** оформить как поля-флажки (CheckBox).

В нижней части страницы разместить две кнопки – **Очистить** и **Добавить**.

3. Основное содержание третьей страницы (education.html) имеет следующий вид (элементы формы рекомендуется разместить в таблице с нулевой границей):

ОБРАЗОВАНИЕ И НАВЫКИ СОИСКАТЕЛЯ

Навыки работы на компьютере

Компьютерные программы, которыми Вы владеете

Водительское удостоверение Серия

Номер Дата выдачи

ИНН

Образование

Незаконченное среднее Среднее

Среднее специальное Неполное высшее

Высшее Несколько высших

Ученая степень

Название оконченного учебного заведения

Отношение к воинской службе

Отслужил/запас Имею отсрочку

Призывник Невоеннообязанный

Последнее место работы

Тип организации ▼

Название организации

Должность

Период работы

с по

Поле **Навыки работы** на компьютере сделать полем со списком, состоящим из следующих значений:

- программирование;
- пользователь;
- знание основ;
- нет.

Группу полей **Образование** сделать полями-флажками (CheckBox).

Группу полей **Отношение к воинской службе** оформить как группу переключателей.

Поле **Тип организации** оформить как поле со списком, состоящим из следующих значений:

- государственное предприятие;
- коммерческое предприятие;
- собственный бизнес;
- иное.

В нижней части страницы разместить две кнопки – **Очистить** и **Добавить**.

4. Основное содержание четвертой страницы (charact.html) имеет следующий вид (элементы формы рекомендуется разместить в таблице с нулевой границей):

ХАРАКТЕРИСТИКИ СОИСКАТЕЛЯ

Семейное положение

Состав семьи

Религиозные убеждения

Положительные черты Вашего характера

Отрицательные черты Вашего характера

Вредные привычки

Желаемая зарплата

Когда готовы приступить к работе

Поле **Навыки работы** на компьютере сделать полем со списком, состоящим из следующих значений:

- Женат (замужем);
- Разведен (-а);
- Холост (незамужем);
- Иное.

В нижней части страницы разместить две кнопки – **Очистить** и **Добавить**.

6. РАЗРАБОТКА WEB-САЙТА С ИСПОЛЬЗОВАНИЕМ КАСКАДНЫХ ТАБЛИЦ СТИЛЕЙ CSS

6.1. Способы применения стилей

Каскадные таблицы стилей CSS (Cascading Style Sheets) позволяют разработать и применить стиль отображения страницы или сайта в целом.

CSS позволяют остаться в рамках декларативного характера разметки страницы и полностью контролировать форму представления элементов HTML-разметки.

Таблицы стилей также позволяют определить цвет и начертание текстового фрагмента, изменять эти параметры внутри текстового блока, выполнять выравнивание текстового блока относительно других блоков и компонентов страницы.

Наличие подобных возможностей позволяет говорить о CSS как о средстве разделения логической структуры докумен-

та и формы его представления. Логическая структура документа определяется элементами HTML-разметки, в то время как форма представления каждого из этих элементов задается CSS-описателем элемента.

Различают четыре способа применения стилей:

– переопределение стиля в элементе разметки с помощью атрибута style;

– размещение описания стилей в заголовке документа в теге <style> (формирование внутренней таблицы стилей);

– размещение ссылки на внешнее описание через элемент Link (формирование внешней таблицы стилей);

– импорт описания стиля в документ.

Переопределение стиля с помощью атрибута style.

Самый простой способ использования CSS – это ввести в код HTML атрибут Style с соответствующим значением. Его можно добавить практически ко всем тегам (кроме таких, как, например, <head> и <html>).

Например:

```
<h1 style="color: red; font-size: 24px;">
```

Но данное объявление стиля неудобно и загромождает код программы. Поэтому лучше создавать таблицы стилей, размещаемые в теге style документа или в отдельном файле.

Внутренние таблицы стилей

Внутренние таблицы стилей оформляются с помощью тега <style>, который помечает начало и конец таблицы стилей (в конце ставится закрывающий тег </style>). Обычно таблица стилей размещается в заголовке документа, обрамленном тегом head.

В таблице стилей может быть один стиль или множество разных стилей. Каждый стиль содержит несколько элементов. Структуру стиля можно представить следующим образом:

```
Селектор { свойство1: значение1; свойство2: значение2;  
...; свойствоN: значениеN}
```

Назначение элементов стиля:

– селектор – сообщает браузеру, к какому элементу или элементам web-страницы применяется стиль;

– блок объявлений стиля – содержит все форматирующие команды, которые можно применить к селектору. Блок обрамляется фигурными скобками;

– объявление свойства – в блок объявлений можно добавить одно или несколько определений или форматирующих команд. Каждое объявление имеет две части – свойство и значение свойства, разделяемые двоеточием, и заканчивается точкой с запятой;

– свойство – представляет собой слово или несколько записанных через дефис слов, определяющих конкретный стиль или стилевой эффект. Обычно эти слова доступны для понимания (font-size, background-color, margin-top);

– значение – различные CSS-свойства требуют определенных типов значений. Это может быть цвет (red или #FF000), длина (18px или 2in), URL(images/background/gif) или ключевое слово (top, center, bottom).

Описание стиля может размещаться не в одной, а в нескольких строках для облегчения восприятия, например:

```
p {  
  color: red;  
  font-size: 1em;  
}
```

Внутренняя таблица стилей может выглядеть следующим образом:

```
<style type="text/css">  
h1 {  
  color: #FF7643;  
  font-size: 20pt;  
}  
p {  
  color: red;  
  font-size: 1em;
```

```
}  
</style>
```

Внутренние таблицы стилей – довольно удачный способ добавления стилей в код web-страницы, но имеет определенные недостатки. Во-первых, объемные таблицы усложняют код web-страницы, что приводит к увеличению времени ее загрузки. Во-вторых, при наличии большого количества web-страниц одинаковой структуры и оформления, требуется добавлять такую таблицы на каждую web-страницу, что не очень удобно. Поэтому существует возможность описания таблиц стилей в отдельном файле.

Внешние таблицы стилей

Внешняя таблица стилей – не что иное, как текстовый файл, содержащий набор стилей CSS. Он не должен содержать HTML-кода, поэтому тег style в файле не используется. Такой файл всегда должен иметь расширение .css (и относительно наглядное название).

После создания таблицы стилей ее необходимо подключить к web-странице, которую нужно отформатировать. Это можно сделать двумя способами:

1. Связывание таблицы стилей с HTML-кодом.

Для это надо использовать тег link.

Код HTML:

```
<link rel="stylesheet" type="text/css" href="css/global.css">
```

Код XHTML:

```
<link rel="stylesheet" type="text/css" href="css/global.css" />
```

Атрибут rel указывает тип ссылки – в данном случае это ссылка на таблицу стилей; атрибут type сообщает браузеру, данные какого типа ему следует ожидать (в данном случае – текстовый файл, содержащий CSS-код); href – указывает местонахождение внешнего CSS-файла.

2. Импорт таблиц стилей в HTML-документ.

CSS имеет встроенный способ привязки внешних таблиц стилей к коду HTML – правило `@import`. Его нужно добавить в содержимое тега `style`. Например:

```
<style type="text/css">  
  @import url(css/global.css);  
</style>
```

В скобках после `url` следует указать ссылку на файл CSS. Кавычки использовать не обязательно.

Можно использовать несколько правил `@import` внутри одного тега `style`. После правила `@import` можно добавлять внутренние описания стилей.

6.2. Управление селекторами стилей

Селекторы типов

Селекторы типов представляют весьма эффективное средство проектирования дизайна web-страниц, поскольку определяют стиль всех экземпляров конкретного HTML-элемента.

В качестве селекторов можно использовать название форматизируемых элементов, например `h1`, `h2`, `p`, `table` и т.д.

Например, такой тег установить гарнитуру шрифта и его цвет для всех тегов `h1`:

```
h1 {  
  font-family: "Arial";  
  color: #000000;  
}
```

Но данный метод имеет свои недостатки – невозможно элементам одного типа присвоить разные стили. Поэтому в многих случаях следует использовать более точное указание с помощью селекторов классов.

Селекторы классов позволяют указать конкретный элемент web-страницы, независимо от тегов. Названия классов должны удовлетворять следующим правилам:

- все названия селекторов классов должны начинаться с точки;

– при именовании стилевых классов разрешается использовать только буквы алфавита, цифры, дефисы, знаки подчеркивания;

– название после точки всегда должно начинаться с символа – буквы алфавита;

– имена стилевых классов чувствительны к регистру.

Пример объявления класса:

```
.special {  
  color: #F0F122;  
  font-family: Verdana;  
}
```

Чтобы применить этот стиль к конкретному абзацу, следует добавить к тегу `p` данного абзаца атрибут `class` с указанием имени класса.

```
<p class="special">
```

Название класса в значении атрибута пишется без точки.

В качестве альтернативного варианта точного управления селекторами можно воспользоваться идентификаторами элементов страницы – создавать `id`-селекторы.

У практически каждого тега может присутствовать атрибут `id` с некоторым уникальным значением, которое позволяет однозначно идентифицировать данный тег. Например:

```
<h1 id="header">
```

Данный тег можно связать с `id`-селектором. `id`-селектор состоит из значения атрибута `id` с добавленным в начало символом `#`.

```
#header {  
  font-size: 36px;  
  text-align: center;  
}
```

Чтобы точно выбрать, что именно использовать – `id`-селекторы или селекторы классов, можно воспользоваться следующими правилами:

– для стиля используемого неоднократно, применяют классы;

– для определения однократно используемого стиля лучше применять id-селекторы;

–при конфликте селекторов классов и id-селекторов более высокий приоритет имеют последние.

Селекторы потомков, групповые селекторы и псевдо-классы

Использование селекторов потомков позволяет отформатировать теги, находящиеся внутри тегов более высокого уровня. Например, если необходимо изменить цвет текста выделенного жирным шрифтом внутри абзаца, можно создать стиль следующего вида:

```
p strong {  
  color: red;  
}
```

В данном случае, красным цветом будет выделен только тот текст, помеченный тегом STRONG, который находится внутри тега P.

В данной конструкции тег более высокого уровня пишется слева, более низкого – справа. Еще один пример:

```
li a {  
  font-family: Arial;  
}
```

В данном случае стиль будет применен к гиперссылкам, расположенным в элементах списка.

Допускается несколько уровней иерархии в селекторе потомков. Также можно указывать не только имена тегов элементов, но идентификаторы или имена классов. Например:

```
p.intro a {  
  color: yellow;  
}
```

Данный стиль будет применен к гиперссылкам, находящимся внутри абзацев p, относящихся к классу .intro. Между p и .intro пробел не ставится, иначе селектор будет интерпретироваться по-другому.

Можно создавать групповые селекторы применяя один и тот же стиль к разным типам элементов. Например:

```
h1, h2, h3, h4, h5, h6 {  
  color: #F1CD33;  
}
```

Друг от друга селекторы отделяются запятыми. Также можно использовать разные виды селекторов:

```
h1, p, .copyright, #header {  
  color: #F1CD33;  
}
```

Существует понятие универсального селектора, который можно применить ко всем элементам страницы. В данном селекторе в качестве названия используется символ *.

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

Такой селектор обнулит значения отступов и полей у всех элементов страницы.

Для более гибкого управления содержимым тегов страницы используются так называемые псевдоклассы элементов.

Четыре псевдокласса предназначены для управления отображение гиперссылки:

a:link – обозначает обычную непосещенную ссылку;

a:visited – обозначает посещенную ссылку;

a:hover – обозначает ссылку, на которую наведен указатель мыши;

a:active – обозначает ссылку в момент нажатия на нее указателем мыши.

Например:

```
a:hover {  
  color: yellow;  
}
```

Замечание: если разные псевдоклассы для работы со ссылками используются одновременно, то они должны быть

описаны в таблице стиля в определенном порядке: **link**, **visited**, **hover**, **active**. Иначе возможны сбои в работе данных стилей.

Существует псевдокласс **:focus**, который используется для управления отображением элементов, получивших фокус ввода. Обычно он используется вместе с полями для ввода информации пользователем (тег `input`). Например:

```
input:focus {  
    background-color: #FFFFAA;  
}
```

С тегом абзаца `P` могут быть использованы также псевдоклассы **:first-letter** и **:first-line**, которые могут быть использованы для форматирования первой буквы и первой строки абзаца соответственно.

Также можно использовать селекторы атрибутов, применяемые к тегам, у которых либо есть определенные атрибуты или атрибуты имеют определенные значения. Например:

```
p[align] {  
    color: red;  
}
```

```
a[href="http://vorstu.ru"] {  
    color: red;  
    font-weight: bold;  
}
```

Каскадность стилей

Каскадность – набор правил, определяющих, какие именно стилевые свойства необходимы элементам web-страницы, то есть задающих последовательность применения многократно определенных стилей. Каскадность определяет, каким образом браузер должен обработать сложную структуру, относящуюся к одному и тому же тегу, если возникает конфликт свойств.

Если стили оказываются вложенным друг в друга, то стили более низкого уровня унаследуют свойства от стилей более высоких уровней. Например:

```
body {font-family: Verdana;}
```

```
p {color: #999944;}  
strong {font-size: 24px;}
```

Если тег `strong` будет указан внутри тега `P`, то он унаследует цвет текста от стиля тега `P` и гарнитуру шрифта от стиля тега `body`.

Если между стилями возникает конфликт свойств, то преимущество будет у стиля тега более близкого элемента-предка. Например:

```
body {font-family: Verdana;}  
p {color: #999944; font-family: Arial}  
strong {font-size: 24px;}
```

Если тег `strong` будет указан внутри тега `p`, то он унаследует гарнитуру шрифта от стиля тега `p`, а не от стиля тега `body`. Однако, если у тега `strong` указать значение гарнитуры шрифта, то именно оно будет применено к его содержимому. Таким образом, непосредственно примененный стиль имеет более высокий приоритет, по сравнению со стилем предком.

В зависимости от типа селекторов стили имеют разные приоритеты.

По возрастанию приоритета стили можно выстроить следующим образом:

- селектор тегов;
- селектор классов;
- id-селектор;
- встроенный стиль.

Если конфликтуют стили, имеющие одинаковый приоритет, то побеждает стиль, описанный последним. Например, если сначала в раздел стилей добавлена внешняя таблица, а потом описаны внутренние стили, то последние будут иметь преимущество.

Можно разрешить конфликт стилей, путем указания свойства **!important**. Например, в случае пересечения стилей для тега документа приоритет будет иметь первый вариант:

```
body {background-color: red !important;}  
body {background-color: black}
```

6.3. Стилизация текста и списков

Большинство сайтов в Интернете построены на текстовом содержимом. С помощью CSS можно придать текстовому содержанию привлекательный дизайн.

CSS предлагает обширный набор команд форматирования, которые позволяют назначать шрифты, цвет, размеры, межстрочный интервал и другие свойства и атрибуты, которые оказывают влияние на визуальное восприятие как отдельных элементов веб-страницы (заголовков, маркированных списков, обычных абзацев текста), так и всей веб-страницы.

Существуют следующие свойства CSS, предназначенные для форматирования текста.

font-family

Используется для выбора типа шрифта. Например:

```
p {  
  font-family: Arial;  
}
```

В данном случае ко всем абзацам будет применен шрифт Arial.

Рекомендуется выбирать шрифты, которые делают текст привлекательным для глаза (если это заголовок) и читабельным (если это основной текст). Существуют два основных вида шрифтов – с засечками и без засечек (гротесковые). Первые используются для больших текстовых блоков, поскольку засечки – небольшие черточки на концах основных штрихов литер – смягчают переходы между символами, делая текст более читабельным. Примеры – Times, Times New Roman, Georgia. Гротесковые шрифты часто используются для прорисовки заголовков благодаря их четким и простым начертаниям. Примеры: Arial, Helvetica, Verdana.

К сожалению, нельзя пользоваться какими угодно шрифтами, так как текст web-страницы не будет отображен адекватно в браузере пользователя, если у него не установлен аналогичный шрифт (будет использован шрифт, установленный в браузере по умолчанию).

Одно из решений данной проблемы – определение в стиле необходимого шрифта, а также нескольких резервных. Например:

```
p {  
  font-family: Arial, Helvetica, Verdana;  
}
```

В данном случае браузер будет использовать шрифт Arial, если он установлен, в противном случае будет использоваться шрифт Helvetica и т.д.

Примеры часто используемых сочетаний шрифтов:

Arial, Helvetica, sans-serif;

“Times New Roman”, Times, serif;

“Courier New”, Courier, monospace;

Georgia, “Times New Roman”, Times, serif;

Verdana, Arial, Helvetica, sans-serif;

Geneva, Arial, Helvetica, sans-serif;

Как видно из примеров, если название состоит из двух или более слов, разделенных пробелами, название шрифта пишется в кавычках.

color

Данное свойство задает цвет шрифта. Наиболее распространенный вариант – задание цвета с помощью шестнадцатиричной константы, задающей уровень интенсивности RGB. Например:

```
h1 {  
  color: #3E8988;  
}
```

Можно также использовать метод кодирования RGB (Red, Blue, Green). Значение цвета кодируется тремя числами, пред-

ставленными в диапазоне 0-255 или процентами соотношениями 0 – 100%, показывающими интенсивности цветов.

Например, для белого цвета следует установить
color: rgb(100%, 100%, 100%);

или

color: rgb(255, 255, 255);

Как вариант можно использовать названия цветов. Например:

color: red;

или

color: yellow;

font-size

Данное свойство отвечает за установку размера шрифта. Существует ряд единиц измерения, в которых может выставляться данная величина:

– px – пиксели – наиболее простая и понятная единица измерения (но обладает одним недостатком - браузер Internet Explorer 6 не увеличивает размеры элементов, заданные в пикселях).

– em – стандартная единица измерения, пришедшая из типографской системы. Она равна ширине буквы m базового шрифта браузера (обычно его размер 16 пикселей).

– % – проценты от базового шрифта. Могут быть как больше, так и меньше 100%. Например, 50% - это половина размера базового шрифта, а 200% - двойной размер базового шрифта. (Но при наследовании стилей эти свойства надо использовать осторожно – так как при наложении свойств размеров, выраженных в процентах, могут получиться непредсказуемые результаты).

– ключевые слова – CSS предлагает семь ключевых слов, определяющих размер шрифта относительно базового: xx-small, x-small, small, medium, large, x-large, xx-large. Среднее значение medium – базовый размер шрифта браузера. Остальные значения уменьшают или увеличивают размер шрифта с коэффициентом 1,2.

– pt - пункты – еще один типографский вариант размера шрифта. Он часто используется в текстовых редакторах (например, в MS Word).

Примеры:

font-size: 12px;

font-size: 1.5em;

font-size: large;

font-size: 14pt;

Как видно из примеров, единица измерения указывается после величины шрифта без пробела, а в дробных значениях в качестве разделителя используется точка.

font-style

Данное свойство управляет стилем шрифта. Имеет три значения:

italic – делает шрифт курсивным;

normal – устанавливает обычный шрифт;

oblique – принципиально не отличается от *italic*.

font-weight

Свойство устанавливает жирность шрифта. В качестве значения могут использоваться числа в диапазоне 100, 200...900. Также можно использовать два символьных значения:

bold – устанавливается жирный шрифт;

normal – устанавливается обычный шрифт.

text-transform

Свойство управляет регистром отображения символов. Имеет следующие значения:

uppercase – преобразовывает символы текста в верхний регистр;

lowercase – преобразовывает символы текста в нижний регистр;

capitalize – превращает первые буквы слов в прописные;

none – отменяет изменения регистра текста (удобно при управлении наследуемыми свойствами).

font-variant

Свойство, позволяющее преобразовывать текст в малые прописные буквы. Имеет значения:

`small-caps` – преобразует буквы текста в малые прописные;
`normal` – преобразует буквы текст в обычный вид.

text-decoration

Свойство используется для улучшения внешнего вида текста путем добавления дополнительных элементов. Имеет значения:

`underline` – подчеркнутый снизу текст;

`overline` – подчеркнутый сверху текст;

`line-through` – перечеркнутый текст;

`blink` – мерцающий («мигающий») текст;

`none` – отменяет декорирование текста (можно, например, удалить подчеркивание гиперссылки).

В данном свойстве можно комбинировать значения, например:

`text-decoration: underline overline;`

letter-spacing

Задает межсимвольный интервал. Выставляется в почти любых единицах измерения, применимых к тексту: пикселях, em, процентах. Для разреженного интервала следует использовать положительные значения, для сжатия – отрицательные, например:

`letter-spacing: -1px;`

`letter-spacing: 10px;`

word-spacing

Задает межсловный интервал. Принцип применения аналогичен предыдущему тегу.

line-height

Задает межстрочный интервал – промежуток между базовыми линиями двух соседних строк текстов. Стандартный интервал равен 120% - следовательно, меньшее значение сжимает данный интервал, а большее растягивает.

Можно использовать в качестве единиц измерения пиксели, проценты или em (последние два более желательны, так как изменяются пропорционально свойству font-size).

Чтобы определить размер межстрочного интервала браузер вычитает высоту шрифта из высоты строки. В результате получается размер межстрочного интервала между двумя соседними строками текста абзаца. Допустим, размер шрифта составляет 12 пикселей. Межстрочный интервал, установленный в размере 150%, в итоге равняется 18 пикселям. В итоге, браузер добавляет пустой промежуток размером 6 пикселям между двумя строками текста.

Межстрочный интервал можно также указать в виде числового значения. Например:

```
line-height: 1.5;
```

После этого значения не нужно указывать единицу измерения. Чтобы определить межстрочный интервал или высоту строки, браузер умножит указанное число на размер шрифта.

text-align

Данное свойство предназначено для выравнивания текста в блоке. Имеет следующие значения:

left – выравнивание по левому краю;

right – выравнивание по правому краю;

justify – выравнивание по ширине;

center – выравнивание по центру.

text-indent

Данное свойство предназначено для указания отступа красной строки абзаца. Для указания величины отступа лучше всего использовать величины em или пиксели. Например:

```
p {  
  text-indent: 5em;  
}
```

Управлять отображение первой строки абзаца также можно с помощью псевдоклассов first-line и first-letter, которые описаны в предыдущей лекции.

Стилизация списков

Теги `` и `` создают маркированные и нумерованные списки соответственно. Списки состоят из последовательности элементов помеченных тегом ``. Для управления их отображением можно использовать ряд свойств.

list-style-type

Используется для задания типа маркеров (тег ``) или типа нумерации (тег ``).

Для задания маркеров можно использовать следующие значения свойства:

`square` – маркер в виде квадрата, залитого цветом;

`disc` – маркер в виде круга, залитого цветом;

`circle` – маркер в виде полой окружности.

Для задания типа нумерации можно использовать следующие значения свойства:

`decimal` – список вида 1, 2, 3 и т.д.;

`decimal-leading-zero` – список вида 01, 02, 03 и т.д. (некоторые браузеры воспринимают это свойство как `decimal`);

`upper-alpha` – список вида A, B, C и т.д.;

`lower-alpha` – список вида a, b, c и т.д.;

`upper-roman` – список вида I, II, III и т.д.;

`lower-roman` – список вида i, ii, iii и т.д.

Можно совсем убрать маркеры, указав в свойстве значение `none`.

Например:

```
ul {  
  list-style-type: none;  
}
```

Данное свойство можно применять не только к тегам `` и ``, но и к тегам ``, смешивая в одном списке разные маркеры. Например, можно создать такой класс стиля:

```
.circle {  
  list-style-type: circle;  
}
```

и применить его следующим образом:

```
<ul>
  <li>Item 1 </li>
  <li class="circle">Item 2 </li>
  <li>Item 3 </li>
  <li class="circle">Item 4 </li>
</ul>
```

В данном списке элементы 2 и 4 будут иметь маркер в виде полой окружности, а остальные два – в виде залитого круга (он устанавливается по умолчанию).

list-style-position

Определяет, каким образом будут размещаться маркеры относительно текста. Имеет два значения:

`outside` – маркеры размещаются обособленно от элементов списка слева (стандартный вариант);

`inside` – маркеры размещаются внутри текстовых блоков элемента списка слева.

list-style-image

Позволяет указать путь к графическому файлу, который используется в качестве маркера. Синтаксис свойства следующий:

```
list-style-image: url(images/bullet.gif);
```

Термин `url` и круглые скобки обязательны. Часть, заключенная в скобки, указывает путь к графическому файлу, который будет использован в качестве маркера.

6.4. Управление полями, отступами и границами

Понятие блочной модели

Браузер обрабатывает все теги как небольшие блоки. Для него любой тег является контейнером с содержимым: текстом, изображением и т.д. У каждого блока может быть набор следующих свойств:

`padding` – *отступ*, «заполнение» - промежуток между содержимым и его границей;

`border` – *граница, рамка* – линия, черта или контур с любой стороной элемента. Граница может быть со всех четырех сторон или с одной стороны в любой их комбинации;

`background-color` – *цвет фона* – заполняет пространство внутри границы, включая область отступа `padding`;

`margin` – *поле* – пустое пространство, отделяющее один тег от другого. Поле может являться промежутком отделяющий один абзац от другого.

Управление размерами полей и отступов

Для управления размерами отступов используются четыре свойства:

`margin-top` – отступ сверху;
`margin-bottom` – отступ снизу;
`margin-left` – отступ слева;
`margin-right` – отступ справа.

Для управления размерами полей используются аналогичные свойства:

`padding-top` – поле сверху;
`padding-bottom` – поле снизу;
`padding-left` – поле слева;
`padding-right` – поле справа.

В качестве единицы измерения можно использовать пиксели, `em`, проценты.

```
margin-left: 10px;  
padding-bottom: 2em;  
margin-top: 10%;
```

Величина, указанная в процентах, измеряется относительно ширины элемента-контейнера. Например, если ширина элемента-контейнера 500 пикселей, то отступ слева, равный 10%, будет шириной 50 пикселей. Если ширина контейнера изменится, размер отступа тоже будет пересчитан.

Можно использовать сокращенное обозначение групп свойств `margin` и `padding`. Например:

```
margin: 5px 0 10px 20px;  
padding: 10px 10px 5px 0;
```

Порядок значений важен – они должны идти в следующей последовательности:

top
right
bottom
left

Если значение свойства равно 0, то единицу измерения можно не указывать.

Если есть необходимость использовать одинаковое значение для всех отступов, то можно указать единственное значение:

padding: 10px;

Замечание: при использовании полей может произойти конфликт между соседними элементами. Например, если первый абзац имеет поле снизу в 20 пикселей, а второй – поле сверху 30 пикселей, то большинство браузеров не будут складывать эти значения, а выберут *большее* из них. Таким образом, получится промежуток между абзацами, равные не 50, а 30 пикселям.

Добавление границ

Граница представляет собой линию, которая очерчивает стилизуемый элемент. Она располагается между отступом и полем элемента. Можно добавить границу ко всем сторонам блока, а можно – только к некоторым из них.

У любой границы есть три различных свойства:

color – цвет;
width – ширина;
style – стиль.

Цвет выставляется стандартными способами, описанными ранее.

Ширина – толщина линии границы. Можно использовать любые единицы измерения CSS кроме процентов (лучше всего – пиксели) или ключевые слова thin (тонкая линия), medium (средняя) или thick (толстая).

Свойство `style` отвечает за стиль прорисовки границы и имеет следующие значения: `solid`, `dotted`, `dashed`, `double`, `groove`, `ridge`, `inset`, `outset`, `none` и `hidden`. Например, `solid` рисует сплошную линию, `dashed` – пунктирную, `dotted` – линию из точек, `none` – убирает прорисовку линии и т.д.

С помощью свойства `border` можно указать параметры сразу всех границ.

```
border: 4px solid #FF00AC;
```

Здесь указаны сразу все три свойства границы (толщина, стиль и цвет). Порядок указания не имеет значения.

Можно добавить только одну линию, например снизу:

```
border-bottom: 2px dashed red;
```

Можно добавить две линии справа и слева:

```
border-left: 2px solid red;
```

```
border-right: 2px solid green;
```

Можно указать свойство для всех границ, а потом переопределить для одной из границ:

```
border: 2px solid red;
```

```
border-bottom: 2px dashed black;
```

Подобным образом можно убрать одну из границ:

```
border: 2px inset #BBA09;
```

```
border-top: none;
```

Можно более тонко настраивать разные свойства границ. Например:

```
border-width: 4px;
```

```
border-left-color: black;
```

Можно также переопределять конкретное свойство конкретной границы:

```
border: 4px solid red;
```

```
border-bottom-style: dashed;
```

Во всех случаях переопределения порядок имеет значение. Сначала идет общее свойство для всех границ, а потом – свойство, которое нужно переопределить.

6.5. Стилизация фона элементов

Цвет фона

Цвет фона элемента устанавливается с помощью свойства **background-color**. В качестве значения указывается цветовая константа в одном из ранее указанных форматов. Данное свойство может быть применено ко всему документу (как стиль тега `body`), так и к отдельному блоку (как стиль тегов `h1`, `p`, `div` и т.д.). Например:

```
body {
    background-color: aqua;
}
h1 {
    background-color: #AADEFB;
}
```

Работа с фоновыми изображениями.

Для работы с фоновыми изображениями используется свойство **background-image**. Данное свойство добавляет рисунок в качестве фона элемента. Чтобы поместить его на заднем плане всей web-страницы, можно создать стиль для тега `body`:

```
body {
    background-image: url(images/bg.jpg);
}
```

В качестве значения указывается слово `url`, за которым следует путь к файлу рисунка, заключенный в круглые скобки. Можно использовать абсолютный URL, например:

```
url(http://www.mysite.ru/images/bg.jpg)
```

Можно использовать относительный путь от документа или корневого каталога сайта:

```
url(../images/bg.jpg)
url(/images/bg.jpg)
```

Если файл с таблицами стилей лежит в том же каталоге, что и документ (или таблица стилей находится в самом документе), то лучше использовать второй вариант. Если файл страницы лежит в корневом каталоге, файл со стилями – в подката-

loge styles, а файл с изображением – в подкаталоге images, то лучше воспользоваться первым вариантом, т.е.

```
body {  
    background-image: url(../images/bg.jpg);  
}
```

Этот путь будет интерпретироваться следующим образом:

../ – переход на уровень выше (к каталогу, содержащему папку style);

images/ – переход из корневого каталога к папке images;

bg.jpg – название файла изображения.

Кавычки в названии могут использоваться, но они не обязательны.

С помощью свойства **background-repeat** можно управлять повторением изображения. Данное свойство имеет четыре значения:

– repeat – параметр по умолчанию, обеспечивает повторение изображения по всей площади страницы;

– no-repeat – отображает фоновый рисунок один раз, без повторения и перекрытия;

– repeat-x – повторяет фоновое изображение горизонтально вдоль оси X, т.е. по всей ширине web-страницы (можно использовать для добавления баннера или заголовка);

– repeat-y – повторяет фоновое изображение вертикально вдоль оси Y, т.е. по всей длине web-страницы (можно использовать при добавлении вертикальных панелей на страницу).

Позиционирование фоновых изображений можно осуществлять с помощью свойства **background-position**, в качестве значения которого можно указать пару ключевых слов. Первое слово отвечает за горизонтальное позиционирование, второе – за вертикальное.

Горизонтальное позиционирование может быть трех видов:

left

center

right

Вертикальное позиционирование может быть трех видов:

top

center

bottom

Например, чтобы переместить фоновое изображение в правый верхний угол рисунка, следует указать свойство:

```
background-position: right top;
```

Для помещения изображения по центру страницы:

```
background-position: center center;
```

Следует заметить, что в Firefox не всегда правильно отображается фоновое изображение со значением вертикальной позиции bottom. Если высота содержимого страницы меньше высоты окна браузера, то нижний край последнего элемента считается нижним краем страницы (и изображение выравнивается по нему). Данная ситуация решается добавлением к странице стиля

```
html {  
  height: 100%;  
}
```

Можно использовать в качестве значений данного свойства точные значения: одно будет определять расстояние между левой стороной изображения и левым краем элемента-контейнера, а второе – расстояние между верхней стороной изображения и верхним краем элемента контейнера. например:

```
background-position: 10px 20px;
```

Можно также указывать отрицательные значения отступов, чтобы «подрезать» часть изображения относительно краев элемента.

Можно также управлять фиксацией изображения на странице, т.е. при прокрутке страницы перемещать ее содержимое, при этом не меняя позицию фонового изображения (например, чтобы на фоне страницы все время находился логотип сайта или что-нибудь еще).

Для этого используется свойство **background-attachment**, которое может принимать два значения:

`scroll` – фоновое изображение прокручивается вместе с текстом и другим содержимым;

`fixed` – фоновое изображение остается на месте, а содержимое прокручивается на фоне изображения.

6.6. Варианты заданий

Общее задание

Разработать сайт, состоящий из нескольких взаимосвязанных страниц, выдержанных в едином стиле. На всех страницах сайта следует стилизовать элементы содержания, т.е. создать стили для заголовков, текста, изображений, гиперссылок, таблиц и т.д.

В качестве основы для разработки страниц сайта рекомендуется воспользоваться файлами `shablon.html` и `styles.css`, находящимися в каталоге с материалами к лабораторной работе (Также там находятся текстовые файлы и иллюстрации).

Файл `shablon.html` предназначен для создания на его основе остальных web-страниц. Он содержит следующий HTML-код:

```
<!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; char-
set=windows-1251">
<!-- Раздел описания стилей-->
<style>
@import url("styles.css"); <!-- Импорт таблицы стилей-->
</style>
</head>
<body>
```

```

<!-- "Обертка" содержимого web-страницы-->
<div id="wrapper">
  <!-- Блок заголовка-->
  <div id="banner">
    BANNER
  </div>
  <!-- Блок панели навигации-->
  <div id="sidebar">
    NAV
  </div>
  <!-- Блок основного содержания-->
  <div id="main">
    MAIN
  </div>
  <!-- Блок подножия-->
  <div id="footer">
    FOOTER
  </div>
</div>
</body>
</html>

```

Данный файл импортирует таблицу стилей styles.css, которая реализует блочную разметку страницы сайта. Подробнее его содержимое будет разобрано в следующей главе, посвященной разработке web-страниц на основе блочной разметки.

В данном файле содержатся описания стилей, помеченные подробными комментариями.

```
/*
```

Стиль для тега документа. Устанавливает:

– цвет фона (background-color: aqua);

– поля тела документа (margin:0;)

– отступы содержимого от края тела документа (padding:0;).

```
*/
```

```

body {
    background-color: aqua;
    margin:0;
    padding:0;
}
/*

```

Стиль для «обертки» содержимого страницы. Устанавливает:

- ширину «обертки» (width:900px;)
 - автовыравнивание блока по центру (margin-left:auto; margin-right:auto;)
 - границы по левому и правому краю обертки (border-left: 1px solid black; border-right: 1px solid black;)
 -
- ```

*/

```

```

#wrapper {
 width:900px;
 margin-left:auto;
 margin-right:auto;
 border-left: 1px solid black;
 border-right: 1px solid black;
}
/*

```

Стиль для блока заголовка страницы. Устанавливает:

- цвет фона (background-color: green;)
  - цвет текста (color: white;)
- ```

*/

```

```

#banner {
    background-color: green;
    color: white;
}
/*

```

Стиль для блока навигации страницы. Устанавливает:

- выравнивание блока по левому краю (float: left;) - НЕ

ТРОГАТЬ

- ширину блока (width: 150px;)
- цвет фона (background-color: green;)
- цвет текста (color: white;)

```
*/
#sidebar {
  float: left;
  width: 150px;
  background-color: red;
  color: white;
}
/*
```

Стиль для блока основного содержания страницы. Устанавливает:

- поле слева (margin-left: 155px;) - чтобы поместилась панель навигации

- цвет фона (background-color: green;)
- цвет текста (color: white;)

```
*/
#main {
  margin-left: 155px;
  background-color: blue;
  color: white;
}
/*
```

Стиль для блока подножия страницы. Устанавливает:

- цвет фона (background-color: green;)
- цвет текста (color: white;)
- предотвращает "обертывание" данного элемента с обеих сторон (clear: both;) - НЕ ТРОГАТЬ

```
*/
#footer {
  background-color: yellow;
  color: black;
  clear: both;
}
/*
```

Рекомендуется создаваемые стили добавлять в данный файл. Также можно изменять уже добавленные в файл свойства стилей отвечающие за цвет текста и фона блоков страницы.

Вариант 1

1. Создать в своем каталоге на диске X отдельную папку для сайта, а в ней подкаталоги Pics и Buttons.

2. Главная страница имеет следующий вид (по такой же схеме организованы и следующие страницы):

Основной боевой танк Т-90	
<p>Главная История Модификации ТТХ Численность Галерея</p>	<p>Общие сведения [Изображение]</p> <p>Текст из файла «Информация.doc» (раздел «Общие сведения»)</p> <p>Страна-разработчик Российская Федерация [Изображение флага страны]</p>
<p>Сайт создан студентом группы Фамилия, инициалы</p>	

Для правильного оформления требуется воспользоваться файлами **shablon.html** (шаблон, на основе которого делаются все страницы) и **styles.css** (таблица стилей, отвечающая за разметку страниц).

В блок навигации должны быть вставлены кнопки, каждая из которых является гиперссылкой.

Кнопка «Главная» - это ссылка на страницу `index.html`. Кнопка «История» - это ссылка на вторую страницу `history.html`. Кнопка «Модификации» - это ссылка на третью страницу `modif.html`. Кнопка «ТТХ» - это ссылка на четвертую страницу `tth.html`. Кнопка «Численность» - это ссылка на пятую страницу `chisl.html`. Кнопка «Галерея» - ссылка на шестую страницу `gallery.html`.

3. Создать на каждой странице следующие стили:

- стили для заголовков;
- стили для абзацев текста;
- стили для кнопок навигационной панели;
- стили для изображений (если это необходимо).
- произвольные стили для различных элементов страницы

по своему усмотрению.

4. Основное содержание второй страницы (history.html) имеет следующий вид:

История создания и применения

[Изображение]

Текст из файла «Информация.doc»
(раздел «История создания и производства»)

5. Основное содержание третьей страницы (modif.html) имеет следующий вид:

Модификации

[Изображение]

Текст из файла «Информация.doc»
(раздел «Модификации»)

6. Основное содержание четвертой страницы (tth.html) имеет следующий вид:

Тактико-технические характеристики танка Т-90А

[Изображение]

Текст из файла «Информация.doc»
(раздел «Тактико-технические характеристики»),
сведенный в таблицу

7. Основное содержание пятой страницы (chisl.html) имеет следующий вид:

На вооружении

[Изображение]

Текст из файла «Информация.doc»
(раздел «На вооружении»)

8. Основное содержание шестой страницы (gallery.html) имеет следующий вид (каждое изображение в таблице сделать ссылкой само на себя, но указать в ссылке в атрибуте target значение _blank):

Галерея

[Изображение]	[Изображение]
[Изображение]	[Изображение]
[Изображение]	[Изображение]

Вариант 2

1. Создать в своем каталоге на диске X отдельную папку для сайта, а в ней подкаталоги Pics и Buttons.

2. Главная страница имеет следующий вид (по такой же схеме организованы и следующие страницы):

Основной боевой танк М1А2 «Абрамс»	
Главная История Модификации ТТХ Численность Галерея	Общие сведения [Изображение]
	Текст из файла «Информация.doc» (раздел «Общие сведения»)
	Страна-разработчик
	Соединенные Штаты Америки [Изображение флага страны]
	Сайт создан студентом группы Фамилия, инициалы

В блок навигации должны быть вставлены кнопки, каждая из которых является гиперссылкой.

Кнопка «Главная» - это ссылка на страницу index.html. Кнопка «История» – это ссылка на вторую страницу history.html. Кнопка «Модификации» – это ссылка на третью страницу modif.html. Кнопка «ГТХ» – это ссылка на четвертую страницу tth.html. Кнопка «Численность» – это ссылка на пятую страницу chisl.html. Кнопка «Галерея» – ссылка на шестую страницу gallery.html.

3. Создать на каждой странице следующие стили:

- стили для заголовков;
- стили для абзацев текста;
- стили для кнопок навигационной панели;
- стили для изображений (если это необходимо).
- произвольные стили для различных элементов страницы

по своему усмотрению.

4. Основное содержание второй страницы (history.html) имеет следующий вид:

История создания и применения

[Изображение]

Текст из файла «Информация.doc»
(раздел «История создания и производства»)

5. Основное содержание третьей страницы (modif.html) имеет следующий вид:

Модификации

[Изображение]

Текст из файла «Информация.doc»
(раздел «Модификации»)

6. Основное содержание четвертой страницы (tth.html) имеет следующий вид:

Тактико-технические характеристики танка М1А2

[Изображение]

Текст из файла «Информация.doc»
(раздел «Тактико-технические характеристики»),
сведенный в таблицу

7. Основное содержание пятой страницы (chisl.html) имеет следующий вид:

На вооружении

[Изображение]

Текст из файла «Информация.doc»
(раздел «На вооружении»)

8. Основное содержание шестой страницы (gallery.html) имеет следующий вид (каждое изображение в таблице сделать ссылкой само на себя, но указать в ссылке в атрибуте Target значение _blank):

Галерея

[Изображение]	[Изображение]
[Изображение]	[Изображение]
[Изображение]	[Изображение]

Вариант 3

1. Создать в своем каталоге на диске X отдельную папку для сайта, а в ней подкаталоги Pics и Buttons.

2. Главная страница имеет следующий вид (по такой же схеме организованы и следующие страницы):

Основной боевой танк «Челленджер 2»	
Главная История Вооружение ТТХ Численность Галерея	Общие сведения [Изображение] Текст из файла «Информация.doc» (раздел «Общие сведения») Страна-разработчик Великобритания [Изображение флага страны]
Сайт создан студентом группы Фамилия, инициалы	

Для правильного оформления требуется воспользоваться файлами `shablon.html` (шаблон, на основе которого делаются все страницы) и `styles.css` (таблица стилей, отвечающая за разметку страниц).

В блок навигации должны быть вставлены кнопки, каждая из которых является гиперссылкой.

Кнопка «Главная» - это ссылка на страницу `index.html`. Кнопка «История» - это ссылка на вторую страницу `history.html`. Кнопка «Вооружение» - это ссылка на третью страницу `weapon.html`. Кнопка «ТТХ» - это ссылка на четвертую страницу `tth.html`. Кнопка «Численность» - это ссылка на пятую страницу `chisl.html`. Кнопка «Галерея» - ссылка на шестую страницу `gallery.html`.

3. Создать на каждой странице следующие стили:

- стили для заголовков;
- стили для абзацев текста;
- стили для кнопок навигационной панели;
- стили для изображений (если это необходимо).
- произвольные стили для различных элементов страницы по своему усмотрению.

4. Основное содержание второй страницы (`history.html`) имеет следующий вид:

История создания и применения

[Изображение]

Текст из файла «Информация.doc»
(раздел «История создания и производства»)

5. Основное содержание третьей страницы (weapon.html) имеет следующий вид:

Вооружение

[Изображение]

Текст из файла «Информация.doc»
(раздел «Вооружение»)

6. Основное содержание четвертой страницы (tth.html) имеет следующий вид:

Тактико-технические характеристики танка Challenger 2

[Изображение]

Текст из файла «Информация.doc»
(раздел «Тактико-технические характеристики»),
сведенный в таблицу

7. Основное содержание пятой страницы (chisl.html) имеет следующий вид:

На вооружении

[Изображение]

Текст из файла «Информация.doc»
(раздел «На вооружении»)

8. Основное содержание шестой страницы (gallery.html) имеет следующий вид (каждое изображение в таблице сделать ссылкой само на себя, но указать в ссылке в атрибуте Target значение _blank):

Галерея

[Изображение]	[Изображение]
[Изображение]	[Изображение]
[Изображение]	[Изображение]

Вариант 4

1. Создать в своем каталоге на диске X отдельную папку для сайта, а в ней подкаталоги Pics и Buttons.

2. Главная страница имеет следующий вид (по такой же схеме организованы и следующие страницы):

Основной боевой танк «Леопард 2»	
Главная История Модификации ТТХ Численность Галерея	Общие сведения [Изображение]
	Текст из файла «Информация.doc» (раздел «Общие сведения»)
	Страна-разработчик Федеративная Республика Германия
	[Изображение флага страны]
	Сайт создан студентом группы Фамилия, инициалы

Для правильного оформления требуется воспользоваться файлами shablon.html (шаблон, на основе которого делаются все страницы) и styles.css (таблица стилей, отвечающая за разметку страниц).

В блок навигации должны быть вставлены кнопки, каждая из которых является гиперссылкой.

Кнопка «Главная» - это ссылка на страницу index.html. Кнопка «История» – это ссылка на вторую страницу history.html. Кнопка «Модификации» – это ссылка на третью страницу modif.html. Кнопка «ТТХ» – это ссылка на четвертую страницу tth.html. Кнопка «Численность» – это ссылка на пятую страницу chisl.html. Кнопка «Галерея» – ссылка на шестую страницу gallery.html.

3. Создать на каждой странице следующие стили:

- стили для заголовков;
- стили для абзацев текста;
- стили для кнопок навигационной панели;
- стили для изображений (если это необходимо).
- произвольные стили для различных элементов страницы

по своему усмотрению.

4. Основное содержание второй страницы (history.html) имеет следующий вид:

История создания и применения

[Изображение]

Текст из файла «Информация.doc»
(раздел «История создания и производства»)

5. Основное содержание третьей страницы (modif.html) имеет следующий вид:

Модификации

[Изображение]

Текст из файла «Информация.doc»
(раздел «Модификации»)

6. Основное содержание четвертой страницы (tth.html) имеет следующий вид:

Тактико-технические характеристики танка Leopard 2

[Изображение]

Текст из файла «Информация.doc»
(раздел «Тактико-технические характеристики»),
сведенный в таблицу

7. Основное содержание пятой страницы (chisl.html) имеет следующий вид:

На вооружении

[Изображение]

Текст из файла «Информация.doc»
(раздел «На вооружении»)

8. Основное содержание шестой страницы (gallery.html) имеет следующий вид (каждое изображение в таблице сделать ссылкой само на себя, но указать в ссылке в атрибуте target значение _blank):

Галерея

[Изображение]	[Изображение]
[Изображение]	[Изображение]
[Изображение]	[Изображение]

Вариант 5

1. Создать в своем каталоге на диске X отдельную папку для сайта, а в ней подкаталоги Pics и Buttons.

2. Главная страница имеет следующий вид (по такой же схеме организованы и следующие страницы):

Основной боевой танк «Леклерк»	
Главная История Вооружение ТТХ Численность Галерея	Общие сведения [Изображение] Текст из файла «Информация.doc» (раздел «Общие сведения») Страна-разработчик Франция [Изображение флага страны]
Сайт создан студентом группы Фамилия, инициалы	

Для правильного оформления требуется воспользоваться файлами `shablon.html` (шаблон, на основе которого делаются все страницы) и `styles.css` (таблица стилей, отвечающая за разметку страниц).

В блок навигации должны быть вставлены кнопки, каждая из которых является гиперссылкой.

Кнопка «Главная» - это ссылка на страницу `index.html`. Кнопка «История» - это ссылка на вторую страницу `history.html`. Кнопка «Вооружение» - это ссылка на третью страницу `weapon.html`. Кнопка «ТТХ» - это ссылка на четвертую страницу `tth.html`. Кнопка «Численность» - это ссылка на пятую страницу `chisl.html`. Кнопка «Галерея» - ссылка на шестую страницу `gallery.html`.

3. Создать на каждой странице следующие стили:

- стили для заголовков;
- стили для абзацев текста;
- стили для кнопок навигационной панели;
- стили для изображений (если это необходимо).
- произвольные стили для различных элементов страницы

по своему усмотрению.

4. Основное содержание второй страницы (`history.html`) имеет следующий вид:

История создания и применения

[Изображение]

Текст из файла «Информация.doc»
(раздел «История создания и производства»)

5. Основное содержание третьей страницы (weapon.html) имеет следующий вид:

Вооружение

[Изображение]

Текст из файла «Информация.doc»
(раздел «Вооружение»)

6. Основное содержание четвертой страницы (tth.html) имеет следующий вид:

Тактико-технические характеристики танка Leclerc

[Изображение]

Текст из файла «Информация.doc»
(раздел «Тактико-технические характеристики»),
сведенный в таблицу

7. Основное содержание пятой страницы (chisl.html) имеет следующий вид:

На вооружении

[Изображение]

Текст из файла «Информация.doc»
(раздел «На вооружении»)

8. Основное содержание шестой страницы (gallery.html) имеет следующий вид (каждое изображение в таблице сделать

ссылкой само на себя, но указать в ссылке в атрибуте Target значение `_blank`):

Галерея

[Изображение]	[Изображение]
[Изображение]	[Изображение]
[Изображение]	[Изображение]

Вариант 6

1. Создать в своем каталоге на диске X отдельную папку для сайта, а в ней подкаталоги `Pics` и `Buttons`.

2. Главная страница имеет следующий вид (по такой же схеме организованы и следующие страницы):

Основной боевой танк «Меркава»	
Главная История Модификации ТТХ Численность Галерея	Общие сведения [Изображение]
	Текст из файла «Информация.doc» (раздел «Общие сведения»)
	Страна-разработчик Израиль
	[Изображение флага страны]
	Сайт создан студентом группы Фамилия, инициалы

Для правильного оформления требуется воспользоваться файлами `shablon.html` (шаблон, на основе которого делаются все страницы) и `styles.css` (таблица стилей, отвечающая за разметку страниц).

В блок навигации должны быть вставлены кнопки, каждая из которых является гиперссылкой.

Кнопка «Главная» - это ссылка на страницу `index.html`. Кнопка «История» - это ссылка на вторую страницу `history.html`. Кнопка «Модификации» - это ссылка на третью

страницу `modif.html`. Кнопка «ГТХ» – это ссылка на четвертую страницу `tth.html`. Кнопка «Численность» – это ссылка на пятую страницу `chisl.html`. Кнопка «Галерея» – ссылка на шестую страницу `gallery.html`.

3. Создать на каждой странице следующие стили:

- стили для заголовков;
- стили для абзацев текста;
- стили для кнопок навигационной панели;
- стили для изображений (если это необходимо).
- произвольные стили для различных элементов страницы

по своему усмотрению.

4. Основное содержание второй страницы (`history.html`) имеет следующий вид:

История создания и применения

[Изображение]

Текст из файла «Информация.doc»
(раздел «История создания и производства»)

5. Основное содержание третьей страницы (`modif.html`) имеет следующий вид:

Модификации

[Изображение]

Текст из файла «Информация.doc»
(раздел «Модификации»)

5. Основное содержание четвертой страницы (`tth.html`) имеет следующий вид:

Тактико-технические характеристики танка Merkava Mk.4

[Изображение]

Текст из файла «Информация.doc»
(раздел «Тактико-технические характеристики»),
сведенный в таблицу

7. Основное содержание пятой страницы (chisl.html) имеет следующий вид:

На вооружении

[Изображение]

Текст из файла «Информация.doc»
(раздел «На вооружении»)

8. Основное содержание шестой страницы (gallery.html) имеет следующий вид (каждое изображение в таблице сделать ссылкой само на себя, но указать в ссылке в атрибуте Target значение _blank):

Галерея

[Изображение]	[Изображение]
[Изображение]	[Изображение]
[Изображение]	[Изображение]

7. СОЗДАНИЕ WEB-СТРАНИЦ НА ОСНОВЕ БЛОЧНОЙ РАЗМЕТКИ

7.1. Управление размерами элементов

Для установки размеров объектов в CSS используются два свойства:

width (ширина);

height (высота).

Данные свойства можно применять для разработки базового дизайна: назначения ширины таблиц, изображений, создания заголовков, бокового меню или подножий страниц.

Для обозначения величины ширины или высоты объекта можно пользоваться стандартными единицами измерения:

– пиксели – наиболее простой и понятный вариант, обеспечивающий точное значение высоты или ширины;

– em – примерно то же самое, что и размер шрифта но в условных единицах (величина имеет относительных характер и зависит от базового шрифта браузера);

– проценты – в данном случае величина объекта рассчитывается на основании ширины элемента-контейнера. Например, если ширина заголовка установлена в 75% и заголовок не вложен ни в какие другие элементы веб-страницы с явно определенной шириной, то ширина текста заголовка составит $\frac{3}{4}$ от ширины окна браузера. Если данный заголовок размещен в теге `<div>` шириной 200 пикселей, то ширина заголовка будет равна 150 пикселям. Если речь идет о высоте (height), то расчет базируется на высоте элемента-контейнера.

Есть определенные нюансы при комбинировании свойств width и height со свойствами, задающими поля и отступы.

Например:

```
div {  
  margin :10px;  
  border-width: 5px;  
  padding: 15px;
```

```
width: 100px;  
}
```

В данном случае фактическая ширина тега `div` будет равна:

$100 + 2 * 10 + 2 * 5 + 2 * 15 = 160$ пикселей.

Это справедливо для браузеров Firefox, IE 6 и выше.

Для браузера IE 5 вместо 160 будет 120 пикселей (т.к. там учитываются только свойства `margin` и `width`).

Свойство `overflow`

Если содержимое тега, в стиле которого заданы точные размеры, вылезает за рамки элемента, то может произойти следующее:

– при отображении в IE (по крайней мере, в версиях до 6.0 включительно) размеры элемента будут растянуты по величине его содержимого;

– при отображении в других браузерах содержимое вылезет за пределы элемента.

Для решения данной проблемы можно использовать свойство **`overflow`**.

Оно имеет четыре значения, с помощью которых можно указать, как должно отображаться содержимое, выходящее за пределы блочного элемента.

– `visible` – это значение, принимаемое браузером по умолчанию. Имеет тот же эффект, что и отсутствие установки свойства `overflow`;

– `scroll` – позволяет добавить в элемент полосы прокрутки. Параметр создает своего рода окно мини-браузера, подобное HTML-рамкам (фреймам). Недостаток – полосы прокрутки отображаются всегда, даже если содержимое полностью помещается внутри блока;

– `auto` – аналогично предыдущему свойству, но добавляет полосы прокрутки только при необходимости;

– **hidden** – скрывает любое содержимое, выходящее за пределы блочного элемента (т.е. не вмещающаяся часть содержимого будет просто не видна).

Управление плавающими элементами

Элементы HTML по умолчанию отображаются в обычном порядке начиная от верхнего края браузера к нижнему один под другим. Есть возможность изменить данный порядок компоновки элементов, воспользовавшись свойством **float**.

Свойство **float** перемещает любой элемент в нужное место, выравнивая его по левому или правому краю страницы. В процессе перемещения содержимое, находящееся ниже позиционируемого плавающего элемента, смещается вверх и плавно обтекает сам плавающий элемент. Таким образом, можно управлять размещением изображений в тексте или создавать боковые панели меню на страницах.

В качестве значения может указываться одно из трех ключевых слов:

– **left** – перемещает стилизуемый элемент влево, при этом содержимое, находящееся ниже его обтекает правый край элемента;

– **right** – перемещает элемент вправо;

– **none** – отменяет обтекание и возвращает элемент в его обычную позицию.

Данное свойство можно применять не только к тегу ``, но и к тегу `<div>`, содержащему другие HTML – теги. Если добавить к данному тегу свойство **width**, то можно создать боковое меню, примечание или отдельный блок на веб-странице.

Для отмены правил обтекания можно воспользоваться свойством **clear**, которое позволяет запретить обтекание элемента с одной или с двух сторон.

Данное свойство указывает браузеру, что стилизуемый элемент не должен обтекать плавающий элемент. Это приводит к тому, что он смещается вниз по тексту веб-страницы, располагаясь ниже плавающего элемента.

Данное свойство допускает следующие значения:

– left – стилизуемый элемент смещается вниз относительно плавающего с установленным левым обтеканием, но правое обтекание остается в силе;

– right – стилизуемый элемент смещается вниз относительно плавающего с установленным правым обтеканием, параметр левого обтекания остается в силе;

– both – вызывает перемещение стилизуемого элемента относительно плавающего с установленным правым или левым обтеканием.

Например, если необходимо поместить блок информации об авторском праве, который должен всегда располагаться внизу страницы и не должен обтекать другое содержимое, то для него следует создать следующий стиль:

```
#copyright {  
  clear: both;  
}
```

Встроенные и блочные элементы

В CSS все элементы делятся на два вида: встроенные и блочные.

В блочных элементах (block) создается разрыв строки перед тегом и после него. Например это теги <div>, <p>, <table> и т.д.

Встроенные элементы (inline) не создают разрывов строк, а размещаются на одной строке с содержимым соседних тегов. Например , , <cite> и т.д.

В большинстве случаев CSS работает с данными элементами одинаково. Можно устанавливать шрифты, цвет, фоновые параметры, границы к обоим типам элементов. Но к встроенным элементам нельзя применять свойства, устанавливающие отступы и поля сверху или снизу (можно обойти данное ограничение путем установки межстрочного интервала).

Если необходимо, чтобы блочный элемент вел себя как встроенный (или наоборот) можно воспользоваться свойством **display**. Это свойство может принимать следующие значения:

- inline – заставляет браузер интерпретировать стилизуемый элемент как встроенный;
- block – заставляет браузер интерпретировать стилизуемый элемент как блочный;
- none – заставляет браузер вообще не отображать данный элемент.

7.2. Разметка страницы на основе плавающих элементов

Изначально для разметки страницы (разбиения ее на области типа баннера, навигации, основного содержимого и сведений об авторах) использовался тег `table`. Это было простое, но неэффективное решение – код страницы становился громоздким и неудобочитаемым.

CSS дал возможность осуществлять разметку страницы с помощью плавающих элементов. Каждая часть web-страницы оформлялась как тег `<div>`, к которому был применен стиль, позиционирующий элемент в нужном месте с помощью свойств `float`, `clear`, `width` и `height`.

Осуществлять разметку web-страницы можно несколькими способами:

- с фиксированной шириной элементов;
- с непостоянной шириной элементов;
- с использованием гибкого дизайна.

Каждый вариант имеет свои достоинства и недостатки. Первый вариант достаточно прост в реализации и неплохо смотрится, но на мониторах с разным разрешением будет выглядеть весьма по-разному.

Второй вариант может соответствовать любой ширине браузера, но на больших мониторах может смотреться несколько размашисто.

Третий вариант основывается на указании постоянного размера, но не в пикселях, а в `em`. Таким образом, размер элементов будет зависеть от базового шрифта браузера (т.е. при его

изменении будет и меняться размер элементов, что удобно для людей с плохим зрением).

Разметка, основанная на плавающих элементах, использует возможности свойства float, для позиционирования элементов на странице относительно друг друга.

Рассмотрим наиболее простой вариант разметки страницы (рис. 43).

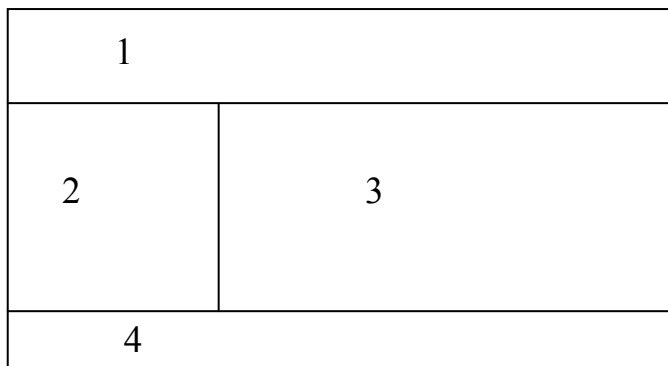


Рис. 43. Типовой вариант разметки страницы из четырех блоков

Здесь 1 – баннер, 2 – навигационная панель, 3 – основное содержание, 4 – подножие страницы.

Элементы страницы следует описывать в порядке их нумерации.

Последовательность действий будет примерно следующей:

1. **ЗаклЮчить каждый элемент страницы в тег <div>**. Каждому элементу присвоить свой уникальный идентификатор. Например, баннеру – “banner”; навигации – “sidebar”; основному содержанию – “main”; подножию – “footer”.

2. **Создать стиль для баннера** (настроить цвет фона, фоновое изображение, поля и отступы и т.д.).

3. **Создать стиль для бокового меню.** Обязательно указать свойство `float` со значением `left`. Установить ширину для бокового меню в пикселях или в `em`. В итоге должно получиться, например:

```
#sidebar {  
    width: 140px;  
    float: left;  
}
```

4. **Создать стиль для основного содержания.** Обязательно установить свойство поля слева (`margin-left`) в значение, равное (или несколько большее) ширине бокового меню. Например:

```
#main {  
    margin-left: 145px;  
}
```

5. **Создать стиль для подножия.** Чтобы панель со сведениями об авторских правах всегда находилась под основным содержанием и главным меню следует добавить в стиль свойство `clear` со значением `both`. Например:

```
#footer {  
    clear: both;  
}
```

6. **При необходимости – создать «обертку» вокруг элементов страницы,** для задания фиксированной ширины. Для этого следует разместить все элементы страницы внутри тега `<div>`, помеченного идентификатором (например, `<div id="wrapper">`), и создать новый стиль. Например:

```
#wrapper {  
    margin-left: auto;  
    margin-right: auto;  
    width: 1000px;  
}
```

В данном случае вся страница помещается в «обертку» шириной `1000px` и выравнивается по середине экрана (с помо-

щью указания значения auto в свойствах margin-left и margin-right).

Управление перемещением столбцов.

С помощью CSS можно разбивать содержимое страницы не только на два, но и на большее количество столбцов.

Например, требуется реализовать следующую схему (рис. 44).

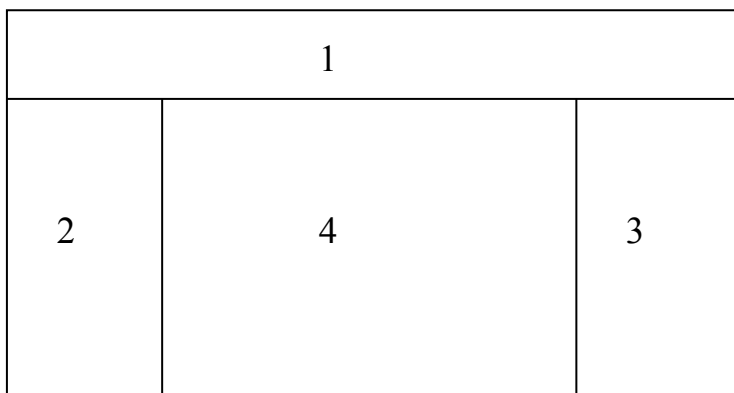


Рис. 44. Типовой вариант разметки страницы с двумя боковыми столбцами

Цифрами обозначены:

- 1 – баннер;
- 2 – левое боковое меню;
- 3 – правое боковое меню;
- 4 – основное содержимое.

Элементы должны описываться именно в том порядке, в котором указаны.

Для добавления правого бокового меню, надо добавить тег div, его описывающий, и присвоить ему отдельный идентификатор или класс. В стиле данного меню следует задать ему фиксированную ширину и сместить его вправо. Например:

```
#rightmenu {
  float: right;
  width: 100px;
}
```

Для стиля, описывающего основное содержимое, следует добавить соответствующий отступы справа и слева, чтобы он поместился между левым и правым меню. Например,

```
#main {
  margin-left: 105px;
  margin-right: 105px;
}
```

Иногда требуется отображать элементы страницы не в том порядке, который приведен выше. Например, для более удобной и корректной обработки сайта поисковыми машинами, следует размещать основное содержимое как можно выше (часто для поисковых машин, текст размещенный в HTML выше имеет более высокий приоритет, чем остальной текст). Есть два варианта решения данной проблемы:

Первый вариант заключается в размещении элементов страницы в следующем порядке:

- 1 – баннер;
- 2 – левое боковое меню;
- 3 – основное содержимое;
- 4 – правое боковое меню.

Для правильного отображения основного содержимого, следует добавить в стиль тега основного содержимого свойство **float: left** и задать ему фиксированную ширину (так, чтобы поместились все столбцы). Например:

```
#main {
  margin-left: 5px;
  margin-right: 5px;
  width: 590px;
  background-color: blue;
  float: left;
}
```

Также можно указать небольшие отступы, чтобы было небольшое свободное пространство между блоками.

Второй вариант заключается в размещении элементов в следующем порядке:

- 1 – баннер;
- 2 – основное содержимое;
- 3 – левое боковое меню;
- 4 – правое боковое меню.

Для данного размещения следует использовать методику перемещения элемента внутри другого плавающего элемента. Для этого следует добавить еще один тег `div`, «обертывающий» основное содержимое и левое боковое меню. В его стиле должно быть свойство **float** со значением **left**.

Внутри данного тега надо сместить левое боковое меню влево, основное содержимое – вправо. Стили для данных компонентов могут иметь следующий вид.

```
#main {
    margin-left:5px;
    margin-right: 5px;
    width: 590px;
    background-color: blue;
    float: right;
}

#leftmenu {
    width: 200px;
    float: left;
    background-color: yellow;
}
```

Содержимое тела документа может быть представлено следующим образом:

```
<div id="wrapper">
  <div id="banner">
    BANNER
```

```
</div>
<div id="wrapper1">
  <div id="main">
    MAIN
  </div>
  <div id="leftmenu">
    NAV1
  </div>
  </div>
  <div id="rightmenu">
    NAV2
  </div>
</div>
```

Работа с отрицательными полями

Как известно, значения полей и отступов могут быть как положительными, так и отрицательными. Используя отрицательные поля, можно поместить теги `div` в любом порядке в HTML-коде, а затем разместить их в другом порядке на экране.

В данном методе есть особенность – необходимо тщательно высчитывать отступы, поля и размеры элементов, чтобы достичь корректного результата.

Рассмотрим порядок действий при разметке страницы с использованием отрицательных полей.

1. Добавить тег `<div>`, который определяет все содержимое страницы («обертка» - `wrapper`). Этот шаг предусматривает создание контейнера для установки фиксированной ширины всего содержимого страницы и предоставляет легкий способ сделать баннер, столбцы и нижний колонтитул одинаковой ширины.

2. Установить ширину для созданного тега `<div>`.

3. Описать каждый столбец в теге `<div>` с атрибутом `ID` или `class`. Эта часть процесса – практически такая же, как при создании любой блочной разметки.

Порядок описания – обертка, баннер, основное содержимое, левое боковое меню, правое боковое меню.

4. Переместить разделы `div` для каждого столбца с помощью свойства `float`. Наиболее распространенный вариант – левое боковое меню и основное содержимое влево, а правое боковое меню – вправо.

5. Установить ширину всех столбцов. Определять ширину плавающего элемента надо практически всегда, но при этом следует учитывать, что полная ширина элемента в окне браузера – это сумма значений CSS-свойства `width`, левых и правых полей, отступов и границ для каждого столбца. Если неправильно рассчитать ширину элементов может возникнуть перепад в размещении элементов.

6. Добавить левое поле к главному столбцу. Именно здесь методика отрицательных полей отличается от ранее указанных методов разметки. Левое поле должно занимать то же пространство, что и левое боковое меню. Если ширина бокового левого поля – 200 пикселей, то следует установить левое поле основного содержимого (`margin-left: 200px;`). Если у бокового меню есть поля и отступы, то их следует учесть.

7. Применить отрицательное поле к тегу `<div>` левого бокового меню. На данный момент левое боковое меню перемещено влево, но поскольку оно идет после главного столбца оно появляется справа от основного содержимого.

Для перемещения левого меню в правильную позицию, следует добавить к нему отрицательный отступ слева, который рассчитывается по следующей формуле:

Отступ= $(-1) * (\text{ширина основного столбца} + \text{левые и правые отступы и поля} + \text{левые и правые границы})$.

Например, если ширина основного столбца равна 598 пикселей, левое поле 200 пикселей, а правая и левая граница имеют ширину по 1 пикселю, то отступ будет равен

$$(-1) * (598 + 200 + 1 + 1) = -800 \text{ пикселей}$$

Таким образом, следует добавить в стиль левого меню свойство

```
left-margin: -800px;
```

8. Устранить ошибку двойного поля старых версий Internet Explorer. При использовании отрицательного поля, в IE версий 6 и ниже возникает ошибка двойного поля, приводящая к удвоению левого поля основного столбца. Устранение данной проблемы состоит в добавлении к стилю основного столбца свойства `display: inline`.

При правильном расчете данный метод разметки является самым гибким. Например, если необходимо поменять местами боковые меню, то можно просто поменять местами содержимое их стилей.

Текст файла со стилями может быть примерно следующим:

```
#banner {
    background-color: aqua;
}
#main {
    margin-left:200px;
    width: 598px;
    border-left: 1px solid black;
    border-right: 1px solid black;
    background-color: blue;
    float: left;
}

#leftmenu {
    width: 200px;
    float: left;
    background-color: yellow;
    margin-left:-800px;
}
#rightmenu {
```



```

        width: 200px;
        float: right;
        background-color: green;
    }
    #wrapper {
        width: 1000px;
        margin-left: auto;
        margin-right: auto;
    }

```

Текст файла основного содержимого будет примерно следующим:

```

<html>
<head>
<style>
    @import url("styles.css");
</style>
</head>
<body>
<div id="wrapper">
    <div id="banner">
        BANNER
    </div>
    <div id="main">
        MAIN
    </div>
    <div id="leftmenu">
        NAV1
    </div>
    <div id="rightmenu">
        NAV2
    </div>
</div>
</body>
</html>

```

Создание столбцов на всю высоту.

Табличная разметка по сравнению с блочной является сложной и громоздкой, но имеет одно преимущество – возможность создавать столбцы равной высоты. Их применение позволяет добавлять фоновый цвет или графику к отдельной ячейке и делать так, чтобы таблица или строка заполняла всю высоту страницы. В CSS выравнивание столбцов по высоте следует воспользоваться обходными путями.

Например, существует страница следующей структуры (рис. 45).

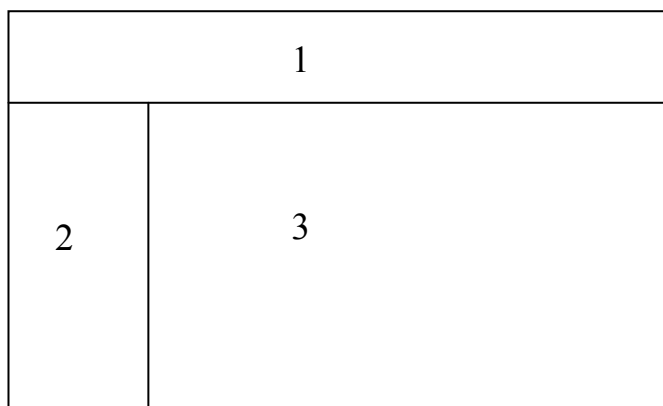


Рис. 45. Типовая структура страницы с одним боковым столбцом

Необходимо сделать так, чтобы левое меню (2) и основное содержимое (3) были одинаковы по высоте. Для этого можно сделать следующее:

- «обернуть» блоки 2 и 3 тегом `div`, присвоив ему особый идентификатор (например, «`wrapper`») – в итоге он будет иметь высоту, такую же как самый длинный элемент;

- подобрать изображение, цвет и ширина которого совпадают с цветом и шириной левого меню;

– сделать данное изображение фоном для тега обертки, прописав ему следующие свойства:

```
#wrapper { background: url (images/back.jpg) repeat-y left top; }
```

Браузер разместит данное изображение прямо под боковым меню, создавая иллюзию того, что у меню есть фоновый цвет.

Если необходимо применить данный метод для правого столбца, то следует прописать:

```
#wrapper { background: url (images/back.jpg) repeat-y right top; }
```

Если необходимо использовать данный метод для двух столбцов, то можно сделать две обертки вокруг боковых меню и основного содержимого и прописать для них два стиля:

```
#wrapper1 { background: url (images/back.jpg) repeat-y left top; }
```

```
#wrapper2 { background: url (images/back.jpg) repeat-y right top; }
```

7.3. Варианты заданий

Общее задание

Разработать сайт, состоящий из нескольких взаимосвязанных страниц, выдержанных в едином стиле. На всех страницах сайта следует стилизовать элементы содержания, т.е. создать стили для заголовков, текста, изображений, гиперссылок, таблиц и т.д.

В качестве основы для разработки страниц сайта рекомендуется воспользоваться файлами `shablon.html` и `styles.css`, находящихся в каталоге с материалами к лабораторной работе (Также там находятся текстовые файлы и иллюстрации).

Файл `shablon.html` предназначен для создания на его основе остальных web-страниц. Он содержит следующий HTML-код:

```
<!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; char-
set=windows-1251">
<!-- Раздел описания стилей-->
<style>
@import url("styles.css"); <!-- Импорт таблицы стилей-->
</style>
</head>
<body>

<!-- "Обертка" содержимого web-страницы-->
<div id="wrapper">

<!-- Блок заголовка-->
<div id="banner">
BANNER
</div>

<!-- Блок основного содержания-->
<div id="main">
MAIN
</div>

<!-- Блок панели навигации-->
<div id="sidebar">
NAV
</div>

<!-- Блок подножия-->
<div id="footer">
FOOTER
</div>
```

```
</div>
```

```
</body>
```

```
</html>
```

Данный файл импортирует таблицу стилей styles.css, которая реализует блочную разметку страницы сайта.

В файле styles.css содержатся описания стилей, помеченные подробными комментариями.

```
/*
```

Стиль для тега документа. Устанавливает:

– цвет фона (background-color: aqua);

– поля тела документа (margin:0);

– отступы содержимого от края тела документа (padding:0);

```
*/
```

```
body {
```

```
background-color: aqua;
```

```
margin:0;
```

```
padding:0;
```

```
}
```

```
/*
```

Стиль для «обертки» содержимого страницы. Устанавливает:

– ширину «обертки» (width:900px;)

– автовыравнивание блока по центру (margin: 0 auto;)

– границы по левому и правому краю обертки (border-left: 1px solid black; border-right: 1px solid black;)

```
*/
```

```
#wrapper {
```

```
width:900px;
```

```
margin: 0 auto;
```

```
border-left: 1px solid black;
```

```
border-right: 1px solid black;
```

```
}
```

```
/*
```

Стиль для блока заголовка страницы. Устанавливает:

- цвет фона (background-color: green;)

- цвет текста (color: white;)

```
*/
```

```
#banner {
```

```
background-color: green;
```

```
color: white;
```

```
}
```

```
/*
```

Стиль для блока навигации страницы. Устанавливает:

- выравнивание блока по левому краю (float: left;) - НЕ

ТРОГАТЬ

- ширину блока (width: 150px;)

- цвет фона (background-color: green;)

- цвет текста (color: white;)

```
*/
```

```
#sidebar {
```

```
float: left;
```

```
width: 150px;
```

```
background-color: red;
```

```
color: white;
```

```
}
```

```
/*
```

Стиль для блока основного содержания страницы. Устанавливает:

- выравнивание блока по правому краю (float: right;) - НЕ

ТРОГАТЬ

- ширину блока (width: 750px;)

- цвет фона (background-color: green;)

- цвет текста (color: white;)

```

*/
#main {
    float: right;
    width: 750px;
    background-color: blue;
    color: white;
}

```

```

/*
Стиль для блока подножия страницы. Устанавливает:
- цвет фона (background-color: green;)
- цвет текста (color: white;)
- предотвращает "обертывание" данного элемента с обеих
сторон (clear: both;) - НЕ ТРОГАТЬ
*/

```

```

#footer {
    background-color: yellow;
    color: black;
    clear: both;
}

```

Рекомендуется создаваемые стили добавлять в данный файл. Также можно изменять уже добавленные в файл свойства стилей отвечающие за цвет текста и фона блоков страницы.

Вариант 1

1. Первая страница имеет следующий вид (блочная структура на основе файла shablon.html и таблицы стилей styles.css – по такой же схеме организованы и остальные страницы):

Военно-морские силы США	
Главная	Данный сайт посвящен описанию военно-морских сил Соединенных Штатов Америки. Со времен Второй мировой войны и по настоящий момент по численности и боевой мощи они занимают первое место в
Общие сведения	
История	
Организация	
Состав	

Корабли	<p>мире. На сайте Вы найдете:</p> <ul style="list-style-type: none"> – общие сведения о ВМС США; – краткую историю возникновения и развития флота США; – сведения об организации и боевом составе; – фотографии и характеристики основных типов боевых кораблей и подводных лодок.
Сайт создан студентом группы... Фамилия, инициалы	

Содержание блока **Навигация**:

Кнопка **Главная** – это ссылка на страницу main.htm.

Кнопка **Общие сведения** – ссылка на страницу common.htm.

Кнопка **История** – ссылка на страницу history.htm.

Кнопка **Организация** – ссылка на страницу org.htm.

Кнопка **Состав** – на страницу fleet.htm.

Кнопка **Корабли** – на страницу ships.htm.

Создать на каждой странице следующие стили:

- стили для заголовков;
- стили для абзацев текста;
- стили для кнопок навигационной панели;
- стили для изображений (если это необходимо).
- стили для таблиц и их элементов;
- другие стили, если это необходимо.

2. Основное содержание второй страницы (common.htm) имеет следующий вид:

ОБЩИЕ СВЕДЕНИЯ

[Изображение флага ВМС США]

Текст из файла «Общие сведения.doc»

3. Основное содержание третьей страницы (history.htm) имеет следующий вид:

ИСТОРИЯ

Таблица с нулевой границей:

Изображение	Список разделов (берется из файла «История.doc») (Каждый элемент списка должен быть ссылкой на соответствующий текстовый блок)
-------------	--

Далее идут текстовые блоки по следующему принципу:

Название блока

[Изображение]

Текст блока из файла «История.doc»

Назад (ссылка на начало страницы)

4. Основное содержание четвертой страницы (org.htm) имеет следующий вид:

ОРГАНИЗАЦИЯ

[Изображение – Эмблема министерства ВМС США]

Административная организация Оперативная организация
(ссылки на соответствующие текстовые блоки)

Текст из файла «Организация.doc»

5. Основное содержание пятой страницы (fleet.htm) имеет следующий вид:

БОЕВОЙ СОСТАВ

[Изображение]

Таблица из файла «Боевой состав.doc»

6. Основное содержание шестой страницы (ships.htm) имеет следующий вид:

КОРАБЛИ И ПОДВОДНЫЕ ЛОДКИ

Список типов кораблей из файла «Корабли и ПЛ.doc»
(каждый элемент списка – ссылка на соответствующее описание корабля)

Далее идут блоки следующей структуры (таблица с ненулевыми границами):

Название (из файла «Корабли и ПЛ.doc»)	
Описание (из файла «Корабли и ПЛ.doc»)	
[Изображение]	[Изображение]
Характеристики (из файла «Корабли и ПЛ.doc»)	

Вариант 2

1. Первая страница имеет следующий вид (блочная структура на основе файла shablon.html и таблицы стилей styles.css – по такой же схеме организованы и остальные страницы):

Военно-морской флот Российской Федерации
--

<p>Главная Общие сведения История Организация Состав Корабли</p>	<p>Данный сайт посвящен описанию военно-морского флота Российской Федерации.</p> <p>На сайте Вы найдете:</p> <ul style="list-style-type: none"> – общие сведения о ВМФ РФ; – краткую историю возникновения и развития флота России; – сведения об организации и боевом составе; – фотографии и характеристики основных типов боевых кораблей и подводных лодок.
<p>Сайт создан студентом группы... Фамилия, инициалы</p>	

Содержание блока **Навигация**:

Кнопка **Главная** – это ссылка на страницу main.htm.

Кнопка **Общие сведения** – ссылка на страницу common.htm.

Кнопка **История** – ссылка на страницу history.htm.

Кнопка **Организация** – ссылка на страницу org.htm.

Кнопка **Состав** – на страницу fleet.htm.

Кнопка **Корабли** – на страницу ships.htm.

Создать на каждой странице следующие стили:

- стили для заголовков;
- стили для абзацев текста;
- стили для кнопок навигационной панели;
- стили для изображений (если это необходимо).
- стили для таблиц и их элементов;
- другие стили, если это необходимо.

2. Основное содержание второй страницы (common.htm) имеет следующий вид:

ОБЩИЕ СВЕДЕНИЯ

[Изображение флага ВМФ России]

Текст из файла «Общие сведения.doc»

3. Основное содержание третьей страницы (history.htm) имеет следующий вид:

ИСТОРИЯ

Таблица с ненулевой границей:

Изображение	Список разделов (берется из файла «История.doc») (Каждый элемент списка должен быть ссылкой на соответствующий текстовый блок)
-------------	--

Далее идут текстовые блоки по следующему принципу:

Название блока

[Изображение]

Текст блока из файла «История.doc»

Назад (ссылка на начало страницы)

4. Основное содержание четвертой страницы (org.htm) имеет следующий вид:

ОРГАНИЗАЦИЯ

[Изображение – Эмблема ВМФ России]

Структурная организация

Пункты базирования

(ссылки на соответствующие текстовые блоки)

Текст из файла «Организация.doc»

5. Основное содержание пятой страницы (fleet.htm) имеет следующий вид:

БОЕВОЙ СОСТАВ

[Изображение]

Таблица из файла «Боевой состав.doc»

6. Основное содержание шестой страницы (ships.htm) имеет следующий вид:

КОРАБЛИ И ПОДВОДНЫЕ ЛОДКИ

Список типов кораблей из файла «Корабли и ПЛ.doc»
(каждый элемент списка – ссылка на соответствующее описание корабля)

Далее идут блоки следующей структуры (таблица с ненулевыми границами):

Название (из файла «Корабли и ПЛ.doc»)	
Описание (из файла «Корабли и ПЛ.doc»)	
[Изображение]	[Изображение]
Характеристики (из файла «Корабли и ПЛ.doc»)	

8. СОЗДАНИЕ МНОГОСТРАНИЧНОГО САЙТА СО СТИЛИЗОВАННОЙ НАВИГАЦИЕЙ

8.1. Стилизация ссылок

Большинство браузеров распознает четыре основных состояния ссылок:

- непосещенная ссылка;
- посещенная (visited) ссылка;
- ненажатая (rollover) ссылка, над которой находится указатель мыши;

– нажатая (active) ссылка, по которой еще не осуществлен переход.

Как рассказывалось ранее, CSS предоставляет четыре псевдокласса селекторов, соответствующих этим состояниям:

```
:link  
:visited  
:hover  
:active
```

Используя псевдоклассы, можно применить различное форматирование для каждого состояния ссылки, обеспечивая соответствующие подсказки и однозначно давая понять посетителям сайта, какие ссылки он уже посещал, а какие – нет.

Предположим, необходимо изменить цвет текста непосещенной ссылки с синего цвета на оранжевый. Для этого следует добавить следующий стиль:

```
a:link {  
  color: orange;  
}
```

При щелчке кнопкой мыши на этой ссылке, ее состояние изменяется на посещенное, а цвет в большинстве браузеров становится фиолетовым. Чтобы изменить его на красный, след:

```
a:visited {  
  color: red;  
}
```

С помощью псевдокласса :hover можно полностью изменить вид гиперссылки при наведении на нее указателя мыши. Например, можно поменять ее цвет и стиль шрифта на курсив:

```
a:hover {  
  color: aqua;  
  font-style: italic;  
}
```

Для полной стилизации гиперссылки можно использовать псевдокласс :active, который меняет вид гиперссылки на те доли секунды, когда пользователь щелкает кнопкой мыши на ссылке (впрочем, он может полюбоваться эффектом подольше, нажав и

удержав кнопку мыши в тот момент, когда она находится на ссылке).

Например, можно выделить ссылку зеленым цветом:

```
a: active {  
  color: green;  
}
```

Если для стилизации ссылок используются все псевдо-классы, то они должны располагаться в определенной последовательности:

```
:link  
:visited  
:hover  
:active
```

Для запоминания данного порядка можно использовать аббревиатуру по буквам соответствующих стилям английских слов **LoVe/HaTe** (в переводе «любовь/ненависть»).

Выборка отдельных ссылок

Стили, которые создавались выше, представляют собой простейшие стили ссылки `<a>`. Они применяются к ссылкам в определенном состоянии, но при этом стилизуются абсолютно все ссылки, независимо от их расположения на web-странице. Что же делать, если нужно отформатировать одни ссылки одним способом, а другие – другим? Существует простое решение – применить стилевые классы к нужным тегам.

Допустим, на веб-странице имеется набор ссылок, одни из которых указывают на другие страницы этого же сайта, другие – на внешние источники. Последние можно отформатировать особым образом, чтобы посетители заранее знали, что при щелчке кнопкой мыши на ссылке они перейдут с текущего сайта на какой-либо другой. В данном случае рекомендуется применить стилевой класс:

```
<a href="http://google.ru" class="external">Google</a>
```

Для стилизации данной ссылки можно создать ряд стилей:

```
a.external:link {
```

```
font-family: Arial;
font-size: 12pt;
color: red;
}
a.external:visited { color: blue; }
a.external:hover { color: orange; }
a.external:active { color: green; }
```

Имя тега <a> указывать необязательно.

Если набор ссылок располагается в одной области web-страницы, то можно применить селекторы потомков, позволяющие адресовать стиль к тегу, вложенному в другой тег. Предположим, имеется пять ссылок, которые ведут на основные разделы сайта.

Они представляют собой главную панель навигации, и необходимо придать им характерный вид. Для этого следует заключить эти ссылки в тег <div> и применить к нему стилевой класс или стиль с идентификатором:

```
<div id="nav">.
```

Теперь появилась возможность произвести выборку и отформатировать только эти ссылки:

```
#nav a:link {color: red;}
#nav a:visited {color: blue;}
#nav a:hover {color: black;}
#nav a:active {color: yellow;}
```

Если на web-странице тег <a> используется не только для гиперссылок, но и для якорей, то часто возникает необходимость стилизовать их отдельно. Одним из вариантов является использование селекторов атрибутов - учитывая, что у гиперссылки есть атрибут href, а у якоря – name.

Например:

```
a[href] {color: white;}
a[name] {color: aqua;}
```


Подчеркивание ссылок

Еще с самого начала развития Интернет ярко-синий подчеркнутый текст сигнализировал: «Нажмите здесь, чтобы перейти туда-то». Однако стандартные атрибуты подчеркивания и цветового оформления ссылок – это то, что в первую очередь изменяет любой web-дизайнер.

Полное удаление подчеркивания. Чтобы убрать стандартное подчеркивание, следует использовать свойство `text-decoration` со значением `none`:

```
a:link {text-decoration: none;}
```

Естественно, полное удаление подчеркивания может смутить посетителей сайта. Поэтому надо каким-то образом выделять гиперссылки в тексте, чтобы посетитель догадывался о значении данного элемента

Добавление подчеркивания только при наведении на ссылку указателя мыши. Некоторые web-дизайнеры убирают подчеркивание для всех ссылок, выделяют их каким-то другим способом, а затем включают атрибут подчеркивания только при наведении указателя мыши. Чтобы обеспечить такой эффект, следует удалить подчеркивание ссылок, а затем повторно ввести его в псевдокласс `: hover`:

```
a:link, a:visited {  
text-decoration: none;  
background-color: lightblue;  
}
```

```
a:hover {  
background-color: transparent;  
text-decoration: underline;  
}
```

Использование свойства нижней границы. Существует возможность управлять цветом, толщиной или стилем стандартной линии подчеркивания ссылок, которое всегда отобра-

жается в виде линии толщиной 1 пиксель того же цвета, что и текст.

Большого разнообразия можно добиться путем использования вместо подчеркивания свойства `border-bottom`. Скрыть обычное подчеркивание и добавить черту в виде пунктирной линии-границы можно следующим образом:

```
a {
  text-decoration: none;
  border-bottom: dashed 2px red;
}
```

Создание ссылок-кнопок.

Можно придать ссылкам вид кнопок, присутствующих в окнах и на панелях инструментов компьютерных программ. Для этого следует воспользоваться свойствами `border`, `background-color` и `padding`.

Предположим, существует ссылка, которой присвоен класс `<button>`:

```
<a href="price.html" class="button">Прайс-лист</a>
```

Для добавления рамки-контура требуется добавить следующий стиль:

```
a.button {
  text-decoration: none;
  border: solid 1px red;
}
```

Можно окрасить кнопку путем установки цвета фона:

```
a.button {
  text-decoration: none;
  border: solid 1px red;
  background-color: gold;
}
```

Также можно добавить отступы, чтобы текст ссылки не примыкал к рамке:

```
a.button {
```

```
text-decoration: none;
border: solid 1px red;
background-color: gold;
padding: 3px 5px 3px 5px;
}
```

Можно добиться, чтобы кнопка выглядела объемной путем использования разных стилей для линий границ.

Широко применяемая дизайнерская методика превращения плоской кнопки в объемную предполагает использование четырех границ разного цвета. Придание кнопке объемности не представляет сложности – главное следует помнить, что освещение заставляет прел выглядеть таким. Необходимо представить, что свет падает на одну из четырех сторон; эта обращенная к источнику света сторона – самая светлая, а противоположная — самая темная. Остальные две стороны должны иметь цвета промежуточных оттенков между «светлыми» и «темными».

Например, можно попробовать такой вариант:

```
a.button {
background-color: gold;
color: green;
font-weight: bold;
border-width: 3px;
border-style: solid;
border-top-color: lightblue;
border-left-color: deepskyblue;
border-right-color: blue;
border-bottom-color: darkblue;
text-decoration: none;
padding: 5px;
}
```

Также можно создать стиль для состояния `:hover`. Кнопки будут реагировать на наведение посетителем на ссылку указателя мыши, обеспечивая интерактивность и обратную связь. В случае с объемной кнопкой при ее нажатии очень эффективен

метод инверсной смены цветов: темный фон кнопки дол стать светлым, светлая граница — темной и т. д.

Например, чтобы инвертировать цвета рамок можно создать следующий стиль:

```
a.button:hover {  
    border-top-color: darkblue;  
    border-left-color: blue;  
    border-right-color: deepskyblue;  
    border-bottom-color: lightblue;  
}
```

Если необходимо создать кнопки фиксированного размера, можно преобразовать ссылку в блочный элемент, после чего задать фиксированную ширину и выравнивание надписи на кнопке посередине. Для этого следует добавить в стиль `a.buttons` следующие свойства:

```
display: block;  
width: 120px;  
text-align: center;
```

При преобразовании ссылки из из встроенного элемента в блочный в браузере Internet Explorer 6 возникает ошибка – по-прежнему можно щелкать только на тексте ссылки, остальная часть кнопки является неактивной. Чтобы исправить ситуацию, следует добавить стиль, явно определяющий высоту для тега `<a>` (причем высота может быть любой, вплоть до 1 пикселя). Поскольку он уменьшает ссылки до нечитаемых размеров в в более поздних версиях Internet Explorer и остальных браузерах, необходимо скрыть от них стиль. Один из способов использовать конструкцию `*html`:

```
*html a.buttons {height: 1px;}
```

Использование изображений. Добавление изображений – достаточно легкий и эффективный способ улучшения навигации сайта. Существует множество методики вариантов дизайна, но следует заметить, что в хороших грамотных методов не используется тег `image`.

Вместо этого используется прием с CSS-свойством `background-image`, с помощью которого можно добавить фоновое изображение к ссылке. При этом следует руководствоваться следующими принципами:

- грамотно использовать значение свойства `no-repeat`;
- управлять позиционированием изображения с помощью свойства `background-position`;
- добавлять отступы с помощью свойства `padding` (если используется фоновое изображение как значок слева или справа от надписи на кнопке);
- использовать псевдоклассы (например, для смены фонового изображения для посещенной ссылки или при наведении на нее).

8.2. Создание панелей навигации

Каждому сайту требуется хорошая управляющая панель с элементами навигации: во-первых, чтобы привести посетителей к нужной им информации, а во-вторых, чтобы им понравилось управление сайтом и они вернулись сюда вновь. Содержимое большинства сайтов организовано в виде разделов. Например, продукция, контактная информация, журнал отзывов и т. д. Такая организация позволяет посетителям хорошо ориентироваться в содержимом сайта, и они точно знают, что и где можно найти. В большинстве случаев ссылки на основные разделы сайта можно найти на панели навигации.

В качестве основы для панели навигации можно воспользоваться маркированным списком, который можно стилизовать и преобразовать в вертикальную или горизонтальную панель навигации.

Например, существует такой маркированный список гиперссылок, помеченный соответствующим классом:

```
<ul class="nav">  
  <li><a href="main.html">Главная</a></li>  
  <li><a href="comp.html">О компании</a></li>
```

```
<li><a href="prod.html">Продукция</a></li>
<li><a href="price.html">Прайс-лист</a></li>
<li><a href="contacts.html">Контакты</a></li>
</ul>
```

Для начала необходимо провести определенную подготовительную работу:

1) Удалить маркеры. Для этого можно создать стиль с соответствующим свойством:

```
ul.nav {
  list-style-type:none;
}
```

2) Убрать отступы и поля. В стиль надо добавить еще пару свойств:

```
ul.nav {
  list-style-type:none;
  margin-left: 0;
  padding-left: 0;
}
```

Теперь можно начинать создавать навигационную панель.

Вертикальная панель навигации

Для создания вертикальной панели навигации необходимо осуществить следующие действия:

1) Отобразить ссылки в виде блочных элементов. Чтобы выровнять элементы панели навигации по ширине, их следует преобразовать в блочные, создав следующий стиль:

```
ul.nav a {
  display: block;
}
```

2) Ограничить ширину кнопок. В данный момент ширина кнопок панели навигации равна ширине родительского элемента. Для ограничения ширины следует добавить в созданный стиль свойство `width`:

```
ul.nav a {
  display: block;
  width: 8em;
```

```
}
```

Если текст кнопки занимает всего одну строку, то можно его центрировать вертикально, чтобы над и под ним были одинаковые промежутки. Для этого следует установить высоту ссылки и назначить такое же значение свойства межстрочного интервала:

```
ul.nav a {  
    display: block;  
    width: 8em;  
    height: 1.25em;  
    line-height: 1.25em;  
}
```

В принципе не обязательно явно назначать свойство width, если панель навигации вложена в элемент разметки веб-страницы, для которого уже явно установлена ширина – например, боковое меню.

3) Устранить ошибки отображения панели в старых версиях Internet Explorer. Во-первых, убрать лишние промежутки между элементами панели навигации, добавив стиль:

```
ul.nav li {  
    display: inline;  
}
```

Во-вторых, как уже говорилось ранее, исправить неверное отображение ссылки при преобразовании ее в блочный элемент, добавив:

```
*html ul.nav a {height: 1px;}
```

Далее можно заняться стилизацией кнопок панели навигации - добавить отступы, поля, изображения, изменить фоновый цвет и т.д.

Горизонтальные панели навигации

Существует два основных метода создания навигационной панели – принципиальная разница заключается в том, что в одном случае получатся кнопки разных размеров (зависит от надписи на кнопке), во втором случае можно получить кнопки одинакового размера.

Вне зависимости от метода, изначально следует удалить маркеры, поля и отступы слева от элементов списка.

Первый способ

1) Преобразовать пункты списка во встроенные элементы. Для этого следует добавить следующий стиль:

```
ul.nav li {  
    display: inline;  
}
```

Желательно, чтобы панель навигации была не слишком большой – иначе элементы в ней могут разместиться на двух или нескольких строках.

2) Стилизовать ссылки. Можно добавить границу кнопок, убрать подчеркивание ссылок, добавить цвет фона, добавить отступы, а также назначить поле справа, если надо отдалить кнопки друг от друга. Например:

```
ul.nav a {  
    border: 1px dashed red;  
    text-decoration: none;  
    background-color: aqua;  
    color: darkgreen;  
    padding: 5px 10px 5px 10px;  
    margin-right: 10px;  
}
```

3) Добавить для тега верхний и нижний отступы. Поскольку теги <a> – встроенные элементы, добавление для них верхнего и нижнего отступов фактически не увеличит высоты ссылок. Вместо этого границы и фон ссылок наложатся на соседние вышестоящие и нижестоящие элементы.

Чтобы решить проблему, следит добавить отступ для тега , который создает соответствующее пустое пространство, предотвращающее наложение границ и фона ссылок. Для нижнего отступа следует использовать такое же значение, как размер нижнего отступа ссылки.

Для определения верхнего отступа тега следует добавить 1 пиксель к значению верхнего отступа ссылки (при ис-

пользовании `em` просто надо удостовериться в том, что отступ тега `` больше).

Например, стиль тега ``, соответствующий ссылкам в шаге 2, должен выглядеть следующим образом:

```
ul.nav {
margin-left: 0;
list-style: none;
padding-left: 0;
padding-top: 6px;
padding-bottom: 5px;
}
```

Второй способ

Хотя методика с использованием `display: inline` для создания горизонтальной панели навигации проста, у нее есть один существенный недостаток: нет способа выровнять ширину кнопок. Установка ширины тегов `` или `<a>` не приведет к требуемому результату, поскольку это встроенные элементы. Чтобы решить проблему, нужно прибегнуть к более сложному и хитрому решению — применению плавающих элементов.

Последовательность действий следующая:

1) Преобразовать пункты списка в плавающие элементы. Для этого следует добавить стиль:

```
ul.nav li {
float: left;
}
```

2) Преобразовать ссылки в блочные элементы. Для этого следует добавить стиль:

```
ul.nav a {
display: block;
}
```

3) Стилизовать ссылки. Аналогично шагу 2 предыдущего способа.

4) Определить ширину кнопок. В стиль, описывающий тег `<a>` надо добавить свойство `width` (для него рекомендуется

использовать параметр `em`, так как он наиболее хорошо масштабируем).

5) Назначить для тега `` свойство `float`. Если для тега `` заданы границы, цвет фона или изображение, придется установить для него свойство `float` с выравниванием по левому краю. Если этого не сделать, то высота элементов списка выйдет за пределы размеров ``.

6) Очистить параметр `float` для последующего содержания. Самый просод – создать стилевой класс `.clear`, например:

```
.clear {  
  clear: both;  
}
```

Созданный класс следует применить к первому тегу, следующему за списком.

В итоге может получиться примерно следующее:

```
ul.nav {  
  list-style-type:none;  
  margin-left: 0;  
  padding-left: 0;  
  float: left;  
}  
ul.nav li {  
  float: left;  
}  
ul.nav a {  
  display: block;  
  border: 1px dashed red;  
  text-decoration: none;  
  background-color: aqua;  
  color: darkgreen;  
  margin-right: 2px;  
  width: 7em;  
  text-align:center;  
}
```

```
.clear {
  clear: both;
}
```

8.3. Задание на лабораторную работу

1. Создать в своем каталоге на диске X отдельную папку для сайта, а в ней подкаталоги Pics, Styles и Buttons.

2. Все страницы должны быть организованы по следующей схеме (**на этот раз файл шаблона и внешней таблицы стилей следует сделать самостоятельно**), построенной с использованием блочной разметки (каждый блок поместить в тег <div>, помеченный идентификатором, и оформить отдельным стилем):

Заголовок	
Навигация	Основное содержимое
Сведения об авторах	

3. Должны быть созданы следующие страницы:

default.html

biography.html

group.html

discography.html

gallery.html

4. Блок «Заголовок» имеет следующее содержание (эмблему группы сделать фоновым изображением и разместить в левой части ячейки):

Эмблема группы – файл Logo.jpg	Название группы (согласно варианту задания)
--------------------------------	---

5. Блок «Сведения об авторах» имеет следующее содержание (разместить ее внизу, добавить в стиль свойство clear: both) :

Сайт создан студентами группы ... [Группа, ФИО авторов]

6. Блок «Основное содержание» на странице Default.htm имеет следующий вид:

[Изображение]
(изображение в конце работы разбить на четыре части, которые должны ссылаться на страницы сайта Биография, Состав, Дискография, Галерея)
Текст из файла «О группе.doc»

7. Блок «Навигация» имеет следующую структуру (для стилизации ссылок можно использовать изображения-кнопки из каталога Buttons):

Главная
Биография
Состав
Дискография
Галерея

Ссылка Главная - это ссылка на главную страницу default.html. Ссылка Биография – это ссылка на вторую страницу biography.html. Ссылка Состав– это ссылка на третью страницу group.html. Ссылка Дискография – это ссылка на четвертую страницу discography.html. Ссылка Галерея - это ссылка на пятую страницу gallery.html.

8. Блок «Основное содержание» на странице «Биография» (biography.html) имеет следующий вид:

Сначала таблица с ненулевой границей:

Биография группы	
Изображение	Список разделов биографии (берется из начала файла «Биография.doc») (Каждый элемент списка должен быть ссылкой на соответствующий текстовый блок)

Далее идут текстовые блоки – каждый состоит из следующих элементов:

Название блока
Изображение
Текст блока из файла «Биография.doc» Назад (ссылка на начало страницы)

9. Блок «Основное содержание» на странице «Состав» (group.html) имеет следующий вид:

Сначала таблица с ненулевой границей:

СОСТАВ
Имена участников группы (каждый в отдельной строчке таблицы). Каждое имя – ссылка на соответствующий блок (Информацию взять из файла «Состав.doc»)

Далее идут текстовые блоки следующей структуры:

Имя участника группы	
Изображение (взять из папки Group из каталога с именем участника)	Изображение (взять из папки Group из каталога с именем участника)
Текст блока из файла «Состав.doc» Назад (ссылка на начало страницы)	

10. Блок «Основное содержание» на странице «Дискография» (discography.html) имеет следующий вид (текст должен

быть вставлен в таблицу с ненулевой границей, выровненную по центру):

Официальная дискография	
Изображение обложки альбома (Все изображения обложек хранятся в папке Discography – в названии файла указаны и год выпуска и имя альбома)	Название альбома Год выпуска (Информацию взять из файла «Дискография.doc»)
...	...
Изображение обложки альбома	Название альбома Год выпуска

11. Блок «Основное содержание» на странице «Галерея» (gallery.html) имеет следующий вид:

Все картинки вставлены в таблицу из 2 строк и 5 столбцов (всего 10 изображений – изображения взять из папки Gallery и вставить в таблицу в уменьшенном в несколько раз виде)

Изображение 1	Изображение 2	Изображение 3	Изображение 4	Изображение 5
Изображение 6	Изображение 7	Изображение 8	Изображение 9	Изображение 10

После таблицы поместить комментарий:

Для просмотра картинки в полном размере щелкните по ней левой кнопкой мыши

Каждое изображение сделать ссылкой на самого себя, добавив атрибут Target со значением `_blank`, чтобы изображение открывалось в отдельном окне.

ЗАКЛЮЧЕНИЕ

Учебное пособие рассматривает основы разработки web-сайтов с использованием визуальных web-редакторов, языка гипертекстовой разметки HTML, каскадных таблиц стилей CSS.

Освоение базовых принципов разработки web-сайтов, web-страниц и их отдельных элементов позволит разрабатывать качественное статичное содержимое web-сайтов различной тематической направленности и назначения.

Особое внимание в пособии уделено вопросам применения каскадных таблиц стилей CSS, изучение которых позволит освоить особенности оформления различных элементов web-страниц для создания визуально эффектных и удобных для использования web-сайтов.

В пособии также рассматриваются принципы разработки web-страниц и web-сайтов в специальных визуальных редакторах.

Современные технологии web-дизайна базируются на разработке структуры web-страниц на основе блочной разметки (верстки). Особенности построения типичных структур страниц с использованием блочной верстки нашли свое отражение в учебном пособии.

Таким образом, учебное пособие содержит описание основных технологий HTML и CSS, применяемых при создании современных web-сайтов.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Дронов В.А. Macromedia Dreamweaver М. / В.А. Дронов. СПб.: БХВ-Петербург, 2003. 736 с.
2. Дронов В.А. Adobe Dreamweaver CS4 / В.А. Дронов. СПб.: БХВ-Петербург, 2009. 832 с.
3. Основы Web-технологий / П.Б. Храмцов, С.А. Брик, А.М. Русак, А.И. Сурин. М.: ИНТУИТ.РУ «Интернет-Университет Информационных Технологий», 2003. 512 с.
4. Прохоренок Н.А. HTML, JavaScript, PHP и MySQL. Джентельменский набор Web-мастера / Н.А. Прохоренок. СПб.: БХВ-Петербург, 2010. 912 с.
5. Макфарланд Д. Большая книга CSS / Д. Макфарланд. СПб.: Питер, 2010. 512 с.
6. Сергеева Т.И. Web-дизайн: учеб. пособие / Т.И. Сергеева, М.Ю. Сергеев. Воронеж: ВГТУ, 2006. 185 с.
7. Сергеева Т.И. Введение в Web-дизайн/ Т.И. Сергеева, М.Ю. Сергеев. Воронеж: ГОУВПО «ВГТУ», 2007. 175 с.
8. Основы Web-дизайна / Т.И. Сергеева, М.Ю. Сергеев, Валеев С.С., Строкина Ю.Г. Уфа: УГАТУ, 2009. 160 с.

ОГЛАВЛЕНИЕ

Введение	3
1. Создание многостраничного сайта средствами Adobe Dreamweaver	4
1.1. Запуск Adobe Dreamweaver и начальная настройка параметров	4
1.2. Настройка общих свойств страницы	10
1.3. Оформление текста и гиперссылок	13
1.4. Оформление таблиц	16
1.5. Вставка изображений	21
1.6. Задание на лабораторную работу	24
2. Создание web-сайта с использованием табличного дизайна и шаблонов в Dreamweaver	30
2.1. Создание web-сайта и настройка его параметров	30
2.2. Создание web-страницы с помощью табличной разметки и сохранение ее как шаблона	33
2.3. Разработка панели навигации	40
2.4. Задание на лабораторную работу	44
2.5. Задание по заполнению страниц	46
3. Создание web-страниц с помощью тегов HTML	49
3.1. Структура HTML-документа	49
3.2. Элементы разметки заголовка документа	53
3.3. Определитель типа документа	56
3.4. Теги для оформления тела документа	58
3.5. Теги для оформления текста документа	59
3.6. Теги для оформления таблиц	67
3.7. Варианты для задания №1	76
3.8. Варианты для задания №2	88
4. Создание многостраничного сайта с помощью тегов HTML	97
4.1. Создание и оформление гиперссылок	97
4.2. Добавление и оформление изображений	100
4.3. Варианты заданий	104
5. Создание web-сайта с использованием форм	122

5.1. Создание элементов форм	122
5.2. Варианты заданий	130
6. Разработка web-сайта с использованием каскадных таблиц стилей css	145
6.1. Способы применения стилей	145
6.2. Управление селекторами стилей	149
6.3. Стилизация текста и списков	155
6.4. Управление полями, отступами и границами	162
6.5. Стилизация фона элементов	166
6.6. Варианты заданий	169
7. Создание web-страниц на основе блочной разметки	188
7.1. Управление размерами элементов	188
7.2. Разметка страницы на основе плавающих элементов	192
7.3. Варианты заданий	203
8. Создание многостраничного сайта со стилизованной навигацией	213
8.1. Стилизация ссылок	213
8.2. Создание панелей навигации	221
8.3. Задание на лабораторную работу	227
Заключение	231
Библиографический список	232

Учебное издание

Сергеев Михаил Юрьевич

РАЗРАБОТКА WEB-САЙТОВ:
ПРАКТИКУМ

В авторской редакции

Подписано в печать 15.12.2010.

Формат 60×84/16. Бумага для множительных аппаратов.

Усл. печ. л. 14,9. Уч.-изд. л. 12,5. Тираж 250 экз.

Заказ. №

ГОУВПО «Воронежский государственный
технический университет»
394026 Воронеж, Московский просп., 14