



МИНИСТЕРСТВО ОБРАЗОВАНИЯ
РЕСПУБЛИКИ БЕЛАРУСЬ

Белорусский национальный
технический университет

Кафедра «Экономика и организация энергетики»

СЕТЕВЫЕ ТЕХНОЛОГИИ И БАЗЫ ДАННЫХ

Лабораторный практикум

Часть 2

Разработка HTML-страниц

Минск
БНТУ
2012

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ
Белорусский национальный технический университет

Кафедра «Экономика и организация энергетики»

СЕТЕВЫЕ ТЕХНОЛОГИИ И БАЗЫ ДАННЫХ

Лабораторный практикум

В 2 частях

Часть 2

РАЗРАБОТКА HTML-СТРАНИЦ

Минск
БНТУ
2012

УДК 004.65(076.5)(075.8)
ББК 39.973-018.2я7
С33

Составитель
А. В. Манюкевич

Рецензенты :
Т. Ф. Манцерава, А. И. Лимонов

Сетевые технологии и базы данных : лабораторный практикум :
С33 в 2 ч. / сост. А. В. Манюкевич. – Минск : БНТУ, 2010–2012. – Ч. 2 :
Разработка HTML-страниц. – 2012. – 61 с.
ISBN 978-985-525-950-4 (Ч. 2).

В издании собраны методические материалы и задания для выполнения лабораторных работ на языке HTML. Основной целью является получение базовых навыков по созданию простейших интернет-страниц.

Лабораторный практикум может быть использован для организации лабораторных работ у студентов дневной и заочной форм обучения

Часть 1 «СУБД Access» вышла в 2010 г.

УДК 004.65(076.5)(075.8)
ББК 39.973-018.2я7

ISBN 978-985-525-950-4 (Ч. 2)
ISBN 978-985-525-271-0

© Белорусский национальный
технический университет, 2012

Лабораторная работа № 1

Создание простейшего HTML-файла.

Краткие теоретические сведения

HTML-файл является простым текстовым файлом, содержащим инструкции программе-браузеру (Internet Explorer, Netscape, Opera, Mozilla и т. д.) для отображения информации на экране монитора. Эти инструкции записываются при помощи специальных команд – тегов. Тег представляет собой управляющую конструкцию, заключенную в скобки. Например:

* этот тег выведет на экран жирный текст *

В данном случае первая латинская буква «B», заключенная в скобки является открывающим тегом, а вторая со знаком прямой слэш – закрывающим. Такая конструкция называется парным тегом или тегом-контейнером и управляет внешним видом информации, заключенной между двумя тегами.

Кроме парных тегов существуют одиночные. Например:
. Данный тег вставляет в документ принудительный перенос на следующую строку.

В теги могут добавляться различные параметры, уточняющие его.

HTML-файл имеет строго определенную структуру. В обязательном порядке это должны быть три раздела:

1. HTML
2. HEAD
3. BODY.

Раздел HTML

Раздел HTML определяет специфику документа, содержание которого будет интерпретироваться браузером. Раздел описывается тегом-контейнером <HTML>... </HTML> и дает браузеру информацию о том, что документ разработан с помощью языка разметки HTML.

Раздел HEAD

Раздел HEAD выполняет функцию рабочего заголовка HTML-документа. Теги, указываемые внутри этого раздела, чрезвычайно важны и могут сильно влиять на внешний вид документа, но сами остаются незаметными глазу пользователя.

Данному разделу сопоставлен парный тег <HEAD>...</HEAD>.

Внутри этого тега могут использоваться следующие вложенные теги :

1. <TITLE> – это парный тег задает заголовок окна программы-браузера. Например:

```
<TITLE>Мой первый файл</TITLE>
```

В заголовке окна программы отобразится фраза «Мой первый файл». Какие-либо дополнительные параметры в данном теге не используются, поскольку внешний вид заголовка окна программы определяется настройками операционной системы компьютера.

2. <BASE> – непарный тег служит для указания полного базового URL-адреса документа. С его помощью относительная ссылка продолжает работать, если документ переносится в другой каталог или даже на другой компьютер. Тег <BASE> работает аналогично команде path MS-DOS.

```
<BASE HREF="//www.my_host.by/manukevich">
```

В данном примере использован тег <BASE> с параметром HREF, который определяет базовый адрес основного Интернет-документа.

3. Мета-определения <META>. Мета-определения электронного документа описываются целым рядом параметров, входящих в состав непарного тега <META> и предназначенных для описания внутренних свойств HTML-файла. Например:

```
а. <META NAME="keywords" CONTENT="HTML, Web, WWW, Web-страница, Интернет">
```

Данная строка в файле задает ключевые слова документа. Таких строк в документе может быть много.

```
б. <META NAME="description" CONTENT="HTML. Лабораторный практикум: Быстрое освоение популярного языка гипертекстовой разметки HTML">
```

Данная строка представляет краткое описание Интернет-документа.

```
в. <META NAME="Author" CONTENT="Александр Манюкевич">
```

Конструкция предназначена для указания автора (авторов) текущего документа.

Также могут использоваться иные параметры, приведенные в табл. 1.1.

Таблица 1.1

Параметры типа мета-определений NAME

Параметр	Функция	Подпараметры
keywords	Указание ключевых слов документа	
description	Указание краткого описания документа	
author	Информация об авторе (авторах) документа	
robots	Составление правил для индексирования документа поисковыми системами (роботами)	ALL NONE INDEX NOINDEX FOLLOW NOFOLLOW
rating	Установление возрастной категории содержания документа	
document-state	Определение частоты индексирования документа	
revisit	Определение промежутка времени, через которое должно производиться повторное индексирование документа	
distribution	Категория распространения документа (массовое или ограниченное)	
URL	Указание основного адреса документа для индексирования (в случае существования «зеркальных» копий)	
copyright	Установление авторского права и перечисление условий распространения документа	
generator	Информация о программном обеспечении, с помощью которого создавался документ	

Раздел Body описывается парным тегом <BODY>...</BODY>, внутри которого размещается большинство существующих тегов HTML. Тег <BODY> имеет ряд параметров (табл. 1.2), которые условно можно разделить на четыре основные группы (параметры фона, границ документа, текста и гиперссылки).

Параметры тега <BODY>

Параметр	Функция
BGCOLOR	Определение цвета фона
BACKGROUND	Указание фонового рисунка
BGPROPERTIES	Изменение свойств фона (например, фиксирование фонового рисунка)
TOPMARGIN, BOTTOMMARGIN, LEFTMARGIN, RIGHTMARGIN, MARGINWIDTH, MARGINHEIGHT	Определение размера отступов
TEXT	Определение цвета основного текста
LINK, ALINK, VLINK	Определение цвета гиперссылок

BGCOLOR устанавливает цвет фона, значение которого может быть введено в символьном эквиваленте, в шестнадцатеричном коде или в формате цветовой модели RGB.

На примере показано три варианта установки цвета фона (белого):

- 1) `<BODY BGCOLOR="white">`
- 2) `<BODY BGCOLOR="#FFFFFF">`
- 3) `<BODY BGCOLOR="255,255,255">`

Параметр BACKGROUND позволяет накладывать на фон документа графическое изображение:

```
<BODY BACKGROUND="images/bg.gif">
```

Из параметров текста документа реально применяется только один – TEXT. Он задает цвет основного текста на странице (значение параметра может быть введено аналогично цвету фона документа):

```
<BODY TEXT="black">
```

Параметры гиперссылок (связей с внутренними или внешними документами) определяют цвет активных (ALINK), не посещенных (LINK) и посещенных (VLINK) ссылок:

```
<BODY LINK="#0000FF" ALINK="#0000FF"
      VLINK="blue">
```


Форматирование текста

Область работы с текстом в HTML включает в себя большое количество всевозможных тегов, составляющих две основные группы: теги логического форматирования и теги физического форматирования.

Тег <ACRONYM> — используется для расшифровки аббревиатур. Реализуется через параметр TITLE и отображается в браузере при наведении курсора на слово-аббревиатуру. Пример:

<ACRONYM TITLE="HyperText Markup Language">HTML</ACRONYM> – один из наиболее популярных и распространенных на сегодня языков разметки текста, используемый для создания Web-сайтов.

Тег <CITE> — призван выделять различные цитаты и высказывания, названия библиографических источников и пр. Текст, помещенный между тегами <CITE> и </CITE>, браузером выделяется курсивом.

Тег <CODE> — предназначен для визуального выделения небольших фрагментов программного кода. Код, размещенный в этом парном теге, отображается моноширинным шрифтом.

Тег — используется для интонационного выделения определенного текстового фрагмента. Браузерами отображается курсивом (так же, как теги <CITE> и <I>).

Теги <H1>, <H2>... <H6> — тег <H> с указанием одной из цифр (от 1 до 6) задает определенный размер заголовка всего текста целиком или его конкретного фрагмента. Соответственно заголовок <H6> будет минимальным, а <H1> — самым большим. Особенностью тегов <H> является то, что они уже подразумевают отступ от текстовой части и разработчикам HTML-документов не приходится делать отступ самостоятельно.

Тег используется для выделения фрагментов текста, на которых необходимо акцентировать внимание пользователя. Браузерами результат такого форматирования отображается жирным начертанием. В визуальном плане аналогичен тегу .

Теги физического форматирования

Тег – функция этого тега аналогична действию тега , а именно – визуальное выделение фрагмента текста, на котором следует сделать акцент, отображается жирным начертанием.

Тег <BASEFONT> предназначен для определения типа шрифта, а также его цвета и размера, которые будут считаться принятыми по умолчанию для всего HTML-документа. Параметры тега <BASEFONT> аналогичны параметрам тега и могут быть изменены по ходу текста дополнительным форматированием с помощью тега .

Следует заметить, что тег <BASEFONT> используется в двух разделах документа: HEAD и BODY. Закрывающего тега <BASEFONT> не требует.

Тег – один из основных тегов физического форматирования текста отображающий свойства шрифтов. Для него могут использоваться следующие параметры:

FACE – параметр, посредством которого браузер отображает текст указанным в HTML-коде шрифтом. При этом значение параметра FACE должно соответствовать установленному на компьютере пользователя шрифту. В случае если такого шрифта нет, текст будет показан стандартным (по умолчанию) шрифтом. Часто разработчики HTML-документов добавляют сразу несколько значений параметра FACE. Таким образом, если первого шрифта не окажется на компьютере пользователя, то браузер отобразит текст вторым шрифтом. Не будет найдено второго шрифта, браузер приступит к поиску третьего и т. д.

Пример:

```
<FONT FACE="Tahoma", "Arial", "Verdana">Этот  
текст будет показан одним из 3-х указанных шриф-  
тов.</FONT>
```

COLOR — параметр, предназначенный для выбора цвета, которым будет написан текст. HTML-конструкция имеет следующий вид:

```
<FONT COLOR="red">Этот текст будет написан  
красным цветом.</FONT>
```

SIZE – функцией этого параметра является определение размера шрифта, которое происходит по условной шкале от 1 до 7. При этом указание размера может осуществляться как в виде относительного показателя (+1), так и в виде абсолютного (1). Средним значением размера шрифта принято считать показатель 3. Однако окончатель-

ный вид шрифта с учетом свойств его размеров может отличаться в зависимости от модели и версии браузера.

Тег `<I>` – тег, аналогичный тегам логического форматирования `<CITE>` и `` (выделяет нужную часть текста курсивом). В ряде случаев использование `<I>` рекомендуется заменять соответствующими тегами логического форматирования.

Тег `<SMALL>` – предназначен для установки размера шрифта не-много меньшего, чем остальные слова (поместив нужный фрагмент между парным тегом `<SMALL>`, получим эффект, противоположный действию тега `<BIG>`).

Тег `` – применяется в случае, когда выделенному фрагменту текста необходимо присвоить определенные свойства, а к помощи никакого логического тега прибегнуть нельзя.

Тег `<SUB>` – позиционирует фрагмент текста относительно нижней линии строки. Весьма удобный инструмент для написания математических и химических формул.

Тег `<SUP>` – размещает текстовый фрагмент относительно верхней линии строки. Также подходит для включения в разнообразные формулы.

Тег `<TT>` – спецификация определяет его как тег для отображения телетайпа или текста, набранного моноширинным шрифтом. В ряде случаев может быть заменен тегом `<CODE>`.

Тег `<U>` делает текст подчеркнутым.

Разделение на абзацы

При отображении HTML-файла на компьютере необходимо производить структурирование документа при помощи специального тега `<P>`. Перед началом каждого абзаца текста следует поместить тег `<P>`. Закрывающий тег `</P >` не обязателен. Браузеры обычно отделяют абзацы друг от друга пустой строкой.

Тег `<P>` может задаваться с параметром горизонтального выравнивания `ALIGN`. Возможные значения параметра приведены в табл. 1.3. По умолчанию выполняется выравнивание по левому краю.

Тег `
` производит принудительный перенос строки текста.

Тег `<NOBR>` запрещает перевод строки.

Тег `<WBR>` дает возможность указать браузеру место возможного перевода строки, что будет выполнено только при необходимости.

Таблица 1.3

Значения параметра ALIGN

Параметр	Действие
LEFT	Выравнивание текста по левой границе окна браузера
CENTER	Выравнивание по центру окна браузера
RIGHT	Выравнивание по правой границе окна браузера
JUSTIFY	Выравнивание по ширине (по двум сторонам)

Горизонтальные линии

Тег <HR> позволяет провести рельефную горизонтальную линию. Этот тег не является контейнером, поэтому не требует закрывающего тега. До и после линии автоматически вставляется пустая строка. Параметры тега <HR> представлены в табл. 1.4.

Таблица 1.4

Параметры тега <HR>

Параметр	Назначение
ALIGN	Выравнивает по краю или центру; имеет значения LEFT, CENTER, RIGHT
WIDTH	Устанавливает длину линии в пикселях или процентах от ширины окна браузера
SIZE	Устанавливает толщину линии в пикселях
NOSHADOW	Отменяет рельефность линии
COLOR	Указывает цвет линии. Используется формат RGB или стандартное имя

Пример:

```
<HR ALIGN=CENTER WIDTH=50% NOSHADOW>
```

В этом примере задается горизонтальная линия, которая занимает половину ширины окна просмотра и располагается посередине окна.

Ход работы

1. Создать пустой текстовый файл с именем Proba_1.HTM на диске C:. Для этого можно использовать текстовый редактор Notepad, либо в любом файловом мониторе (FAR, TotalCommander) установить на одной из панелей диск C: и нажать сочетание клавиш ALT+F4. После этого ввести имя файла.

2. Набрать следующий текст:

```
<HTML>
<HEAD>
<TITLE>Мой первый HTML-документ</TITLE>
<META HTTP-EQUIV="Content-Type" CONTENT="text/html; charset=Windows-1251">
<META NAME="keywords" CONTENT="HTML, документ, первый">
<META NAME="description" CONTENT="Это мой первый HTML-документ">
</HEAD>
<BODY BGCOLOR="#FFFFFF" TOPMARGIN="30"
TEXT="black" LINK="#00FF00" ALINK="#00FF00"
VLINK="blue">
    Это мой первый HTML-документ!
</BODY>
</HTML>
```

3. Сохранить документ. Открыть при помощи доступного интернет-браузера. Внимательно изучить то, что появилось в окне просмотра.

4. Изменить цвет фона на желтый. Для этого задать значение BGCOLOR="#FFFF00".

5. Сохранить текстовый файл. Обновить отображаемый HTML-файл.

Дополнить существующий файл следующим фрагментом текста и изучить действие использованных тегов:

```
<BR> <ACRONYM TITLE=" Harry Potter and the Deathly Hallows">Гарри Поттер </ACRONYM> - фрагмент последнего романа Дж.Роулинг <BR>
```

<SITE> Чувство напряженного ожидания наполнило тёплый воздух шатра, гул голосов кое-где нарушался взрывами хохота. </SITE> <CODE> Мистер и миссис Уизли ходили по проходам, улыбались и приветствовали родственников. </CODE> Миссис Уизли была в мантии нежно-аметистового цвета и такого же цвета шляпе. Секундой позже Бил и Чарли встали перед шатром, <I> оба в праздничных робах с большими белыми розами, заколотыми в петлицах. </I> <SMALL> Фред свистнул по-волчьи, </SMALL> что вызвало <TT> всплеск хохота по залу</TT>, но затем в шатре наступила тишина<U> и только тихая музыка исходила из того, что выглядело как золотые надувные шары. </U>

6. Сохранить текстовый файл. Обновить отображаемый HTML-файл.

7. Набрать в существующем файле следующий текст:

```
<P Align= Right>  
Сжала руки под тёмной вуалью... <BR>  
"Отчего ты сегодня бледна?" <BR>  
-От того, что я терпкой печалью <BR>  
Напоила его допьяна.<BR>  
<SITE>Анна Ахматова<SITE><BR><BR> </P>
```

```
<P Align=CENTER>  
<FONT FACE="Verdana", "Arial", "Helvetica"  
SIZE=+3 COLOR=red>  
<b>Баллада о Любви </b> <br>  
Когда вода Всемирного потопа<br>  
Вернулась вновь в границы берегов, <br>  
Из пены уходящего потока<br>  
На берег тихо выбралась Любовь <br>  
-И растворилась в воздухе до срока, <br>  
А срока было - сорок сороков... <br>  
<I> Владимир Высоцкий</I>
```

8. Подвести под текстом черту при помощи следующего тега:

```
<HR ALIGN=CENTER WIDTH=50% NOSHADE COLOR=Blue>
```

9. Используя теги SUB и SUP написать формулы

$$(a + b)^2 = a^2 + 2ab + b^2;$$
$$c_{ij} = a_{ij} + b_{ij}$$

10. Выполнить самостоятельно следующее задание:

а) создать html-файл, в мета-описаниях которого фигурирует название дисциплины, университета, факультета и фамилии преподавателя и студента;

б) задать название, состоящее из фамилии и имени;

в) средствами html добиться, чтобы внешний вид документа был максимально приближен к следующему:

НЕСКАЗАННОЕ, СИНЕЕ, НЕЖНОЕ...

Несказанное, синее, нежное...

Тих мой край после бурь, после гроз,

И душа моя – поле безбрежное -

Дышит запахом меда и роз.

Я утих. Годы сделали дело,
Но того, что прошло, не кляню.
Словно тройка коней оголтелая
Прокатилась во всю страну.

Сергей Есенин

Лабораторная работа № 2

«Списки и ссылки»

Тестовые упражнения на закрепление пройденного материала

1. На диске С создайте HTML-файл, в котором:

- а) в мета-тегах будет описаны фамилия преподавателя, название предмета, учебное заведение и специальность;
- б) будет выводиться желтый текст на голубом фоне:

Над омраченным Петроградом
Дышал ноябрь осенним хладом.
Плеская шумною волной
В края своей ограды стройной,
Нева металась, как больной
В своей постеле беспокойной.
А. С. Пушкин. Медный всадник

в) выделить последнюю строку специальным тегом для определения авторства документа

г) используя совместно теги заголовков <H> и теги для установления параметров шрифта вывести на экран следующую информацию

Структура документа (серым цветом)

Раздел документа HEAD (красным цветом)

Название документа (черным цветом)

1. Покажите результат преподавателю

Краткие теоретические сведения

Списки

В языке HTML предусмотрены следующие основные типы списков: маркированный, нумерованный и список определений. Для реализации списков различных типов используются следующие теги: , , <DL>.

Маркированный список. В маркированном списке для выделения его элементов используются специальные символы, называемые маркерами списка. Вид маркеров списка определяется браузером.

Для создания маркированного списка необходимо использовать тег-контейнер `` ``, внутри которого располагаются все элементы списка. Открывающий и закрывающий теги списка обеспечивают перевод строки до и после списка, отделяя, таким образом, список от основного содержимого документа, поэтому здесь нет необходимости использовать теги абзаца `<P>` или принудительного перевода строки `
`.

Каждый элемент списка должен начинаться тегом `` (LI — List Item, элемент списка). Тег `` не нуждается в соответствующем закрывающем теге. Пример маркированного списка

```
<HTML>
<HEAD>
<TITLE>Пример маркированного списка</TITLE>
</HEAD>
<BODY>
<UL>
<B>Знаки зодиака:</B>
<LI>Овен
<LI>Телец
<LI>Близнецы
<LI>Рак
<LI>Лев
<LI>Дева
<LI>Весы
<LI>Скорпион
<LI>Стрелец
<LI>Козерог
<LI>Водолей
<LI>Рыбы
</UL>
</BODY>
</HTML>
```

В теге `` может быть указан параметр `TYPE`. Параметр `TYPE` может принимать следующие значения: `disc`, `circle` и `square`.

Этот параметр используется для принудительного задания вида маркеров списка. Конкретный вид маркера будет зависеть от используемого браузера.

Типичными вариантами отображения являются следующие:

TYPE = disc – маркеры отображаются закрашенными кружками;

TYPE = circle – маркеры отображаются незакрашенными кружками;

TYPE = square – маркеры отображаются закрашенными квадратами.

Пример записи: <UL TYPE = circle>

Параметр TYPE с теми же значениями может употребляться для указания вида маркеров отдельных элементов списка. Для этого параметр TYPE с соответствующим значением разрешено указывать в теге элемента списка .

Пример записи: <LI TYPE = circle>

В качестве маркеров списка можно использовать графические изображения, что широко применяется для создания привлекательных, красиво оформленных HTML-документов.

Пример списка с графическими маркерами

```
<HTML>
<HEAD>
<TITLE>Маркированный список</TITLE>
</HEAD>
<BODY>
<UL>
<B>Знаки зодиака : </B><BR>
<IMG SRC="Green_ball.gif">Овен<BR>
<IMG SRC="Green_ball.gif">Телец<BR>
<IMG SRC="Green_ball.gif">Близнецы<BR>
<IMG SRC="Green_ball.gif">Рак<BR>
<IMG SRC="Green_ball.gif">Лев<BR>
<IMG SRC="Green_ball.gif">Дева<BR>
<IMG SRC="Green_ball.gif">Весы<BR>
<IMG SRC="Green_ball.gif">Скорпион<BR>
<IMG SRC="Green_ball.gif">Козерог<BR>
<IMG SRC="Green_ball.gif">Водолей<BR>
<IMG SRC="Green_ball.gif">Рыбы
</UL>
</BODY>
</HTML>
```

В приведенном примере в качестве маркера элементов списка используется графический файл Green_ball.gif.

Другим типом списков, реализованных в языке HTML, является **нумерованный список**. Иначе списки такого типа называют упорядоченными. Последнее название часто используется как формальный перевод названия соответствующего тега , с помощью которого и организуются списки такого типа в HTML-документах (OL – Ordered List, упорядоченный список).

Отличием от маркированных списков является то, что в нумерованном списке перед каждым его элементом автоматически проставляется порядковый номер. Вид нумерации зависит от браузера и может задаваться параметрами тегов списка.

Для создания нумерованного списка следует использовать тег-контейнер , внутри которого располагаются все элементы списка.

Как и для маркированного списка, каждый элемент нумерованного списка должен начинаться тегом .

```
<HTML>
<HEAD>
<TITLE>Пример нумерованного списка</TITLE>
</HEAD>
<BODY>
<OL>
<B>Наиболее яркие звезды, видимые с Земли:</B>
<LI>Сириус
<LI>Жанопус
<LI>Арктур
<LI>Альфа Центавра
<LI>Вега
<LI>Капелла
<LI>Ригель
<LI>Процион
<LI>Ахернар
<LI>Бета Центавра
<LI>Бетельгейзе
<LI>Альдебаран
<BR>. . .
```

```
<LI value=58>Мицар  
<BR>. . .  
<LI value=75>Полярная  
</OL>  
</BODY>  
</HTML>
```

В теге могут быть указаны следующие параметры: TYPE и STAR. Параметр TYPE используется для задания вида нумерации списка. Может принимать следующие значения:

TYPE = A – задает маркеры в виде прописных латинских букв;

TYPE = a – задает маркеры в виде строчных латинских букв;

TYPE = I – задает маркеры в виде больших римских цифр;

TYPE = i – задает маркеры в виде маленьких римских цифр;

TYPE = 1 – задает маркеры в виде арабских цифр.

Параметр START тега позволяет начать нумерацию списка не с единицы. В качестве значения параметра START всегда должно указываться натуральное число вне зависимости от вида нумерации списка. Пример:

```
<OL TYPE = A START=5>
```

Списки определений, также называемые словарями определений специальных терминов, являются особым видом списков. В отличие от других типов списков, каждый элемент списка определений всегда состоит из двух частей. В первой части элемента списка записывается определяемый термин, а во второй – текст в форме словарной статьи, раскрывающий значение термина.

Списки определений задаются с помощью тега-контейнера <DL> (Definition List). Внутри контейнера тегом <DT> (Definition Term) помечается определяемый термин, а тегом <DD> (Definition Description) – абзац с его определением. Для тегов <DT> и <DD> можно не записывать соответствующие закрывающие теги. В общем виде список определений записывается следующим образом:

```
<DL>  
<DT>Термин  
<DD>Определение термина  
</DL>
```

Текст, содержащий определение термина, выводится, начиная со следующей строки (или через строку для некоторых браузеров) после определения термина с отступом вправо. В информации, помещенной после тега <DD>, могут располагаться элементы уровня блока. Отсюда следует, в частности, что списки определений могут быть вложенными.

Приведем пример HTML-документа, в котором использован список определений:

```
<HTML>
<HEAD>
<TITLE>Пример списка определений</TITLE>
</HEAD>
<BODY>
<DL>
<CENTER>
<H3>Классификация типичных темпераментов чело-
века,<BR>основанная
на воззрениях Гиппократ</H3>
</CENTER>
<DT>Флегматик
<DD>Пассивный, очень трудоспособный, медленно
приспосабливающийся; <BR>настроение устойчивое,
мало поддается внешнему влиянию;<BR>вялость эмо-
циональных реакций и медлительность в волевой
деятельности<BR><BR>
<DT>Сангвиник<DD>Активный, энергичный, легко
приспосабливающийся; <BR>живость и подвижность
эмоциональных реакций, быстрота и сила волевых
Проявлений<BR><BR>
<DT>Холерик
<DD>Активный, очень энергичный,
настойчивый;<BR>порывистость и сила
эмоциональных реакций, бурные волевые проявле-
ния<BR><BR>
<DT>Меланхолик
<DD>Пассивный, легко утомляющийся, тяжело при-
способливающийся; <BR>слабость волевых проявле-
```

ний и преобладание подавленного настроения, неуверенность в себе

```
</DL>  
</BODY>  
</HTML>
```

Бывают случаи, когда в элемент списка одного типа требуется включить целый список такого же типа или другого. При этом будут организованы многоуровневые или вложенные списки. В HTML допустимо произвольное вложение различных типов списков, однако при их организации следует проявлять аккуратность.

Ниже приводится HTML-код документа с вложенными списками. В этом примере в каждый элемент маркированного списка вложен свой нумерованный список.

```
<HTML>  
<HEAD>  
<TITLE>Пример вложенного списка</ TITLE >  
</HEAD>  
<BODY>  
<UL>  
<B>Спутники некоторых планет</B>  
<LI>Земля  
<OL>  
<LI>Луна  
</OL>  
<LI>Марс  
<OL>  
<LI>Фобос  
<LI>Деймос  
</OL>  
<LI>Уран  
<OL>  
<LI>Ариэль  
<LI>Умбриэль  
<LI>Титания  
<LI>Оберон  
<LI>Миранда  
</OL>
```

```
<LI>Нептун
<OL>
<LI>Тритон
<LI>Нереида
</OL>
</UL>
</BODY>
</HTML>
```

Ссылки на другие документы и файлы

Одним из важнейших понятий для HTML-документов являются ссылки. Само название – HTML, язык разметки гипертекста, указывает на принцип организации таких документов.

Гипертекстовый документ – это документ, содержащий ссылки на другие документы, позволяющие при помощи нажатия кнопки мыши быстро перемещаться от одного документа к другому.

Существует много типов мультимедийных объектов, которые могут быть размещены на Web-странице. В современных гипертекстовых документах в дополнение к самому тексту часто используют разнообразную графику, видео- и аудиообъекты, а в качестве ссылок часто применяют изображения.

Организация ссылок. Ссылка состоит из двух частей. Первая из них – это то, что вы видите на Web-странице; она называется указатель ссылки (anchor). Вторая часть, дающая инструкцию браузеру, называется адресной частью ссылки (URL- адрес). Когда вы щелкаете мышью по указателю ссылки, браузер загружает документ, адрес которого дается URL-адресом.

Правила построения отдельных элементов ссылок. Указателем ссылки может быть слово, группа слов или изображение. Внешний вид ссылки зависит от его типа, способов создания и установок программы просмотра читателя. Указатели бывают двух типов – текстовые и графические. Текстовые указатели обычно представляют собой слово или несколько слов выделенных на экране подчеркиванием. Цвет текстового указателя может регулироваться автором и установками программы просмотра.

Приведем пример записи для текстового указателя ссылки:

```
<A HREF="example.html">Этот текст является  
указателем ссылки</A>
```

В качестве ссылки можно использовать графическое изображение. По принципу действия графические ссылки ничем не отличаются от текстовых. Пример графического указателя ссылки:

```
<A HREF="example.html"><IMG SRC="picture.gif">
```

Второй частью ссылки является URL-адрес. Это не что иное, как адрес Web-страницы, которая будет загружена при щелчке мышью на указатель.

Указание адреса может быть относительным или абсолютным. Если в URL-адресе не указывается полный путь к файлу, то такая ссылка называется относительной. В этом случае определение местоположения файлов выполняется с учетом местоположения документа, в котором имеется такая ссылка. Например, если браузер загрузил страницу, находящуюся по адресу `http://www.mysite.com/page`, то относительный указатель `/picture` подразумевает адрес `http://www.mysite.com/page/picture`, т. е. подкаталог, расположенный на той же машине. Относительные указатели удобны в использовании. Намного проще вставить только имя файла, а не весь длинный URL-адрес. Они также позволяют вам перемещать файлы в пределах вашего сервера без больших изменений в межстраничной адресации.

URL-адрес, полностью определяющий компьютер, каталог и файл, называется абсолютным. В отличие от относительных, абсолютные указатели могут ссылаться на файлы, расположенные на других компьютерах.

Правила записи ссылок. Для организации ссылки необходимо сообщить браузеру, что является указателем ссылки, а также указать адрес документа, на который вы ссылаетесь. Оба действия выполняются при помощи тега `<A>`.

Тег `<A>` имеет единственный параметр `HREF`, значением которого является URL-адрес. Указатель может быть как относительным, так и абсолютным, например, `http://www.server.com/home/index.htm`.

Этот тег является контейнером, поэтому необходимо поставить закрывающий тег ``:

```
<A HREF=URL-адрес>Текстовый указатель ссылки</A>
```

Указатель ссылки может быть относительным или абсолютным. Для облегчения работы с относительными указателями ссылок введен тег `<BASE>`. Он располагается в начале документа в разделе `HEAD` и содержит URL-адрес, относительно которого в документе построена вся адресация. Это указание влияет на любой тег документа, в котором используется относительная адресация. Если тег `<BASE>` отсутствует, то адресация строится относительно адреса текущего документа.

Внутренние ссылки

Кроме ссылок на другие документы, часто бывает полезно включить ссылки на разные части текущего документа. Например, большой документ читается лучше, если он имеет оглавление со ссылками на соответствующие разделы. Для построения внутренней ссылки сначала нужно создать указатель, определяющий место назначения. Например, если вы хотите сделать ссылку на текст определенной главы документа, нужно разместить там указатель и дать ему имя при помощи параметра `NAME` тега `<A>`. При этом параметр `HREF` не используется и браузер не выделяет содержимое тега `<A>`. Например:

```
<A NAME=chapter_5> </A>
```

Обратите внимание, что в приведенном примере отсутствует содержимое тега `<A>`. Обычно именно так и делают, поскольку здесь нет необходимости как-то выделять текст, а требуется лишь указать местоположение. После того как место назначения определено, можно приступить к созданию ссылки на него. Для этого, вместо указания в параметре `HREF` адреса документа, как это делалось ранее, поместим туда имя ссылки с префиксом `#`, говорящим о том, что это внутренняя ссылка.

Глава 5

Теперь, если пользователь щелкнет кнопкой мыши на словах «глава 5», браузер выведет соответствующую часть документа в окне просмотра.

Ссылки на документы различных типов. Когда пользователь щелкает мышью на ссылке, указывающей на другую Web-страницу, она выводится непосредственно в окне браузера. Если же ссылка указывает на документ иного типа, программа просмотра принимает документ и затем решает, что с ним делать дальше. Следующими действиями браузера могут быть:

а) браузер знает этот тип документа и умеет с ним обращаться. Например, если вы создали ссылку на графический файл формата GIF и пользователь щелкнул мышью на этой ссылке, его программа просмотра очистит окно и загрузит указанное изображение. В некоторых случаях браузер может дополнительно использовать подключаемый программный модуль (plug-in), без которого задача не была бы решена.

б) браузер не распознает тип принятого документа и не знает, что с ним делать дальше. В этом случае он обратится к вспомогательным программам, имеющимся на машине пользователя. Если подходящая программа найдется, браузер запустит ее и передаст ей полученный документ для обработки. Например, если пользователь щелкнет на ссылке на видеофайл формата AVI, браузер загрузит файл, найдет программу для демонстрации AVI-файлов и запустит ее. Видеофайл будет показан в дополнительном небольшом окне.

Ссылки на другие ресурсы Интернета

Web-пространство является лишь частью сети Интернет. Другие ресурсы начали свое существование задолго до рождения WWW, поэтому накопили уже много достойной внимания информации и имеют достаточно большую аудиторию. Поэтому, разрабатывая свою персональную страницу или документ, вы, возможно, захотите включить ссылки и на другие ресурсы.

Ресурсы Интернета весьма разнообразны по форме и содержанию. Хотя HTML предполагает возможность создания своих собственных версий этих ресурсов с помощью механизма обработки

дан-ных форм, есть более простые пути к взаимодействию с системами UseNet, Telnet, FTP, e-mail и другими.

Например, вы можете создать документ с использованием различных тегов форм, текстовых элементов и кнопкой для отправки электронного письма на ваш адрес. Однако будет намного проще для связи указать лишь свой адрес электронной почты. В этом случае упрощается обновление страницы, которое не будет связано с изменениями форм. Кроме того, многие браузеры имеют встроенную поддержку некоторых ресурсов, что дает возможность сократить время на установление связи с ними.

Самой популярной деятельностью в Интернете является обмен электронными письмами. Пользователей этого ресурса намного больше, чем любого другого. Создание ссылки на электронную почту так же просто, как и на другую страницу. Для этого вместо URL-адреса следует указать адрес электронной почты, предварив его словом mailto:

```
<A  
  HREF="mailto:AMANUKEVICH@bntu.edu.by">Присылайте  
  ваши отзывы и предложения</A>.
```

Эта ссылка не будет ничем отличаться от остальных гипертекстовых ссылок вашего документа. В заключение приведем пример HTML-кода, в котором используется ряд тегов, описанных ранее:

```
<HTML>  
<HEAD>  
<TITLE>Личная страница группы 106715</ TITLE >  
</HEAD>  
<BODY>  
<H1 ALIGN=CENTER>Добро пожаловать!</H1>  
<HR>  
Основное содержание страницы<HR><BR>  
<font face="Arial" color="blue">  
<A HREF="Spisok.htm">1.Список группы </A><br>  
<A HREF="Foto.htm">2.Фотоальбом группы</A><br>  
<A HREF="Links.htm">3.Полезные ссылки</A><br>  
<ADDRESS>
```

Последнее обновление выполнено 01 сентября 2007 г.

```
</ADDRESS>
<A HREF="mailto: grupa715@bntu.edu.by ">
Присылайте ваши отзывы и предложения</A>
</BODY>
</HTML>
```

В приведенном примере есть ссылка на электронную почту и другие HTML-файлы. При наведении курсора мыши на указатель данной ссылки этот адрес будет виден в нижней части окна браузера. Для корректного функционирования данной странички необходимо создать и разместить в текущей папке три файла с названиями *Spisok.htm*, *Foto.htm*, *Links.htm*.

Ссылки на другие ресурсы Интернета записываются аналогично.

Ход работы

1. Создать HTML-файл на основе примеров, приведенных в теоретической части, и разобраться, как они работают.

2. Создать в текущей папке вложенную папку **files**, создать в ней три файла *Spisok.htm*, *Foto.htm*, *Links.htm* и изменить гиперссылки в основном файле таким образом, чтобы открывались файлы именно из этой папки.

3. В папке **files** создать папку **fotos**, скопировать в нее указанные преподавателем файлы и создать в файле *Foto.htm* гиперссылки на эти файлы с использованием списка определений, где в качестве термина будет выступать название фотографии, а в качестве определения – ее краткое описание.

4. В файле *Links.htm* создать 5–6 гиперссылок на Ваши наиболее популярные Интернет-ресурсы. Ссылки представить в виде маркированного списка.

5. В файле *Spisok.htm* привести список группы в виде упорядоченного списка.

6. Создать html-файл, состоящий из большого фрагмента текста, напечатанного крупным шрифтом. Например:

Дремотный покой узкой, залитой лунным светом улочки был нарушен появлением двух человек. Оба возникли прямо из воздуха и тут же застыли на

месте, направив друг на друга волшебные палочки. Несколько секунд прошло в напряжённом молчании. Затем, очевидно, узнав один другого, они спрятали палочки в складках мантий плечом к плечу зашагали вдоль улицы прочь.

- Какие новости? - спросил тот, что был повыше.

- Всё отлично, - ответил Северус Снэйп.

Слева вдоль дороги росли пышные кусты дикой ежевики, справа тянулась высокая, аккуратно подстриженная живая изгородь. Длинные полы мантий развевались у ног идущих.

- Я уж было решил, что опаздываю, - сказал Яксли. Грубо вылепленное лицо его то скрывалось в густой тени деревьев, то мертвенно белело в лучах ночного светила.

- Оказалось, что всё сложнее, чем мне представлялось раньше. Но уж теперь-то, надеюсь, он останется доволен. А ты, я смотрю, уверен, что нас примут хорошо?

Снэйп ограничился коротким кивком. Они свернули направо, на широкую подъездную аллею, под прямым углом подходившую к первой улочке. Аллею перегораживали массивные кованые ворота, за которыми снова тянулись вдаль, насколько хватало глаз и лунного света, кусты живой изгороди. Мужчины, не задержавшись перед воротами, дружно подняли левые руки в некоем подобии военного салюта и спокойно прошли прямо сквозь тяжёлую металлическую решётку, словно она была вылеплена из теней и дыма.

В нижней части текста поставить гиперссылку **НАВЕРХ**, которая возвращала бы читателя к началу документа.

Лабораторная работа № 3

Графика в интернет-документах

Тестовые упражнения для закрепления пройденного материала

1. Создайте файл на диске C:\, в котором сделайте нумерованный и маркированный списки студентов группы.
2. Обозначьте 2–3 фамилии студентов как гиперссылки, которые открывают файлы с информацией о студентах. В файлах предусмотрите кнопку возврата к первоначальному файлу.

Краткие теоретические сведения

Разработчики Web-страниц могут управлять цветом фона документа, а также указывать изображения, используемые в качестве фонового. Для задания цвета фона употребляется параметр BGCOLOR тега <BODY>, а фоновое изображение включается в документ при помощи параметра BACKGROUND. В качестве значения параметра BGCOLOR указывается название цвета или его составляющие в шестнадцатеричном коде. В качестве фонового изображения должен использоваться графический файл формата GIF или JPG. Приведем пример записи тега <BODY> с указанием фонового цвета и фонового изображения:

```
<BODY BACKGROUND=texture.gif BGCOLOR=gray>
```

Для встраивания изображений в HTML-документ следует использовать тег , имеющий единственный обязательный параметр SRC, определяющий URL-адрес файла с изображением. Простейший пример встраивания изображения:

```
<IMG SRC=picture.gif>
```

Данный тег может иметь ряд параметров.

Выравнивание изображений

При включении графического изображения в документ можно указывать его расположение относительно текста или других эле-

ментов страницы. Способ выравнивания изображения задается значением параметра ALIGN тега .

Возможные значения этого параметра приведены в табл. 3.1.

Таблица 3.1

Значения параметра ALIGN

Значение	Действие
TOP	Верхняя граница изображения выравнивается по самому высокому элементу текущей строки
TEXTTOP	Верхняя граница изображения выравнивается по самому высокому текстовому элементу текущей строки
MIDDLE	Выравнивание середины изображения по базовой линии текущей строки
ABSMIDDLE	Выравнивание середины изображения посередине текущей строки
BASELINE или BOTTOM	Выравнивание нижней границы изображения по базовой линии текущей строки
ABSBOTTOM	Выравнивание нижней границы изображения по нижней границе текущей строки
LEFT	Изображение прижимается к левому полю окна. Текст обтекает изображение с правой стороны
RIGHT	Изображение прижимается к правому полю окна. Текст обтекает изображение с левой стороны

Поясним действие параметров выравнивания, приведенных в таблице. Сразу же оговоримся, что все значения параметров выравнивания изображений можно условно разделить на две группы по принципу их действия. К одной группе относятся два значения параметра – LEFT и RIGHT. При использовании любого из этих параметров мы получаем так называемое «плавающее» изображение. В этом случае изображение прижимается к соответствующему краю окна просмотра браузера, а последующий текст (или другие элементы) «обтекают» изображение с противоположной стороны. Здесь текст, размещаемый рядом с изображением, может занимать несколько строчек.

К другой группе значений параметров относятся все остальные. При их использовании изображение как бы встраивается в строчку текста, а параметры выравнивания задают расположение изображения относительно строки текста. Таким образом, в отличие от плавающих изображений, здесь изображение является обычным элементом строки. Это легко понять, если представить, что изображение является просто одной буквой строки текста, правда, достаточно большой (типа буквицы).

Приведем пример HTML-кода, в котором используются изображения как элемент строки.

```
<HTML>
<TITLE>Выравнивание изображений</TITLE>
<BODY>
  Выравнивание<IMG SRC=eagle.gif ALIGN=top>no
  верхнему краю
  <P>
  Выравнивание no<IMG SRC=eagle.gif ALIGN
  =BASELINE>базовой линии
</BODY>
</HTML>
```

Приведем пример плавающего изображения. В примере изображение прижато к правому краю окна просмотра браузера, а следующий текст располагается с левой стороны от изображения. Количество строк, располагаемое рядом с изображением, может изменяться в зависимости от размеров шрифта текста, а также размеров окна просмотра. Текст, не поместившийся рядом с изображением, автоматически продолжается ниже. Этот пример построен на следующем исходном коде:

```
<HTML>
< TITLE >Выравнивание изображений</ TITLE >
<BODY>
<IMG SRC=spb.gif ALIGN=right>
<P ALIGN=JUSTIFY>
  Санкт-Петербург расположен в самой восточной
  оконечности Финского залива в устье реки Невы, на
  42-х островах ее дельты. Географические координаты
  города: 59"57' северной широты и 30"19' восточной
```


долготы от Гринвича. Из крупнейших городов мира (с населением свыше одного миллиона человек) Санкт-Петербург является ближайшим к Северному полюсу, он находится на одной широте с северной частью Камчатки и южной оконечностью Аляски.

<P ALIGN=JUSTIFY>

Высокоширотным положением города объясняется явление белых ночей. Они наступают 25-26 мая, когда солнце опускается за горизонт не более чем на 9', и вечерние сумерки практически сливаются с утренними. Наибольшая продолжительность дня приходится на 21-22 июня; заканчиваются белые ночи 16-17 июля, продолжаясь в общей сложности более 50 дней.

</BODY>

</HTML>

Базовая линия (BASELINE или BOTTOM) – это нижняя часть линии текста, которая проводится без учета нижней части (descender) некоторых символов, например, букв типа j, q, y. В отличие от выравнивания по базовой линии, при задании значения ABSBOTTOM выравнивание выполняется по нижней части самого низкого элемента в строке, т. е. по одному из символов строки, имеющему элементы, лежащие ниже базовой линии. Аналогично обстоит дело с различием между параметрами TOP и TEXTTOP.

Например, самым высоким элементом в строке может быть графическое изображение, в то время как самым высоким текстовым элементом строки является, как правило, заглавная буква.

<HTML>

<TITLE>Различие параметров выравнивания</TITLE >

<BODY>

Выравнивание ALIGN=TOP

Выравнивание ALIGN=TEXTTOP

</BODY>

</HTML>

Если в документе используются плавающие изображения, выровненные со значением RIGHT или LEFT, то имеется возможность принудительного прекращения обтекания в заданном месте текста. Это обеспечивается применением тега принудительного прерывания строки
 с параметром CLEAR .

В качестве значений параметра CLEAR можно использовать следующие: LEFT, RIGHT или ALL. Так, для приведенного выше примера в нужном месте текста можно разместить строку:

```
<BR CLEAR=right>.
```

Текст, следующий далее, будет размещаться ниже изображения с новой строки.

Задание размеров выводимого изображения

Тег встраивания изображений имеет два необязательных параметра, указывающих размеры изображения при отображении, – WIDTH и HEIGHT. Значения параметров могут указываться как в пикселах, так и в процентах от размеров окна просмотра. Значения параметров ширины и высоты изображения могут не совпадать с истинными размерами изображения. В этом случае браузеры автоматически при загрузке изображений выполняют его перемасштабирование. Заметим, что неаккуратное задание параметров может привести к изменению пропорций рисунка и, как следствие, к его искажению.

```
<HTML>  
< TITLE >Задание размеров изображений</ TITLE >  
<BODY>  
<IMG SRC=spb.gif WIDTH=300>  
<IMG SRC=spb.gif>  
<IMG SRC=spb.gif WIDTH=75>  
</BODY>  
</HTML>
```

Отделение изображения от текста

Для тега можно задавать параметры HSPACE и VSPACE, значения которых определяют отступы от изображения, оставляемые пустыми соответственно по горизонтали и вертикали. Это га-

рантирует, что между текстом и изображением останется пространство, необходимое для нормального восприятия. В приведенном ниже HTML-коде со всех четырех сторон изображения задан отступ, равный 20 пикселям.

```
<HTML>
<TITLE>Использование параметров HSPACE и VSPACE</
TITLE>
<BODY>
<IMG SRC=spb.gif ALIGN=left HSPACE=20 VSPACE=20>
(Текст абзаца)
<p>
(Текст абзаца)
</BODY>
</HTML>
```

Рамки вокруг изображений

Изображение, встраиваемое на страницу, можно поместить в рамку различной ширины. Для этого служит параметр BORDER тега . В качестве значения параметра используется число, означающее толщину рамки в пикселях. По умолчанию рамка вокруг изображений не рисуется. Исключением из этого правила является случай, когда изображение является ссылкой.

```
<HTML>
<TITLE>Помещение изображения в рамку</ TITLE >
<BODY>
<IMG SRC=map.gif>
<IMG SRC=map.gif BORDER=1>
<IMG SRC=map.gif BORDER=5>
<IMG SRC=map.gif BORDER=10>
</BODY>
</HTML>
```

Альтернативный текст

Одним из параметров тега является параметр ALT, определяющий альтернативный текст. Его указание дает возможность пользователям неграфических браузеров или пользователям, рабо-

тающим в режиме отключения загрузки изображений, получить некоторую текстовую информацию о встроенных изображениях.

При отключенном изображении вместо них на экране появится альтернативный текст, определенный значением параметра ALT. Значение этого параметра имеет смысл и для случаев, когда загрузка изображений будет выполняться. Поскольку загрузка изображений выполняется на втором проходе после отображения текстовой информации, то изначально на экране на месте изображения появится альтернативный текст, который по мере загрузки будет сменяться изображением.

Современные браузеры будут также отображать альтернативный текст в качестве подсказки (tooltip) при помещении курсора мыши в область изображения.

```
<IMG SRC=spb.gif ALT="Герб Санкт-Петербурга" WIDTH=150  
HEIGHT=174>
```

```
<IMG SRC=moscow.gif ALT="Герб Москвы">
```

Использование изображения в качестве ссылки

Графические изображения могут использоваться не только в качестве иллюстраций, но и выполнять роль указателей гипертекстовых связей. Для обеспечения работы изображения в качестве ссылки на другие ресурсы достаточно включить изображение внутри тега-контейнера <A>. Пример:

```
<A HREF=My_doc.html> <IMG SRC=map.gif> </A>
```

Любая часть такого изображения будет работать как указатель ссылки на документ My_doc.html. Существуют возможности задания изображений, отдельные фрагменты которых будут указывать на различные ресурсы.

Параметр LOWSRC

Еще одним параметром, который можно задать в теге , является LOWSRC. Его значение определяет файл с альтернативным изображением, которое должно появляться при первом проходе выдачи на экран загружаемого документа. Смысл задания двух загружаемых на одно и то же место изображений заключается в

следующем. В качестве изображения, указываемого значением параметра LOWSRC, рекомендуется выбирать картинку того же содержания, но более низкого разрешения, возможно с меньшей глубиной цвета или даже монохромное. Изображения такого рода занимают значительно меньше места и, как следствие, быстро загружаются. Это позволяет уже на первом проходе формирования документа увидеть общие черты изображений и не ожидать их полной загрузки. Пример записи:

```
<IMG SRC=main.gif LOWSRC=low.gif>
```

Использованию параметра LOWSRC присущ ряд особенностей. Если была загружена картинка, адрес которой был задан параметром LOWSRC, то основная картинка будет иметь точно такие же размеры, что и предыдущая. Если изображения исходно имели различные размеры, то основное изображение будет приводиться к размерам первого. При этом могут быть нарушены пропорции основного изображения. Во избежание этого разумно указывать настоящие размеры основного изображения (параметрами WIDTH и HEIGHT тега). Тогда на первом проходе изображение низкого разрешения будет приводиться к размерам настоящего изображения, а на втором проходе - будет заменено им.

Использование миниатюрных версий изображений

На Web-страницах часто используются миниатюрные версии изображений (thumbnail) в качестве графических указателей ссылок на полноразмерные изображения. Изображения такого рода представляют собой уменьшенные копии оригинальных изображений, имеющие иногда также меньшую глубину цвета или представленные в оттенках серого цвета. Файлы с такими изображениями занимают значительно меньше места по сравнению с полноразмерными и поэтому гораздо быстрее грузятся. С помощью миниатюрных версий можно быстро просмотреть набор изображений, доступных для загрузки, и выбрать понравившееся.

В Интернете можно найти целый ряд специализированных сайтов, содержащих разного рода изображения и обеспечивающих предпросмотр (preview) с помощью миниатюрных копий. Большое число сайтов не предназначено для детских глаз, но есть и ряд ин-

тересных серверов, предлагающих изображения различной тематики. Можно отметить сайт <http://www.snap-shot.com>.

Ход работы

1. Подготовьте интернет-страницу с фотографиями и описаниями. Используйте различные способы обтекания текстом графических изображений. Задайте фоновый рисунок. Фотографии получить у преподавателя.

2. Подготовить интернет-страницу в которой, маленькие фотографии являются кнопками для гиперссылок на файлы с полноразмерными фотографиями. Предусмотреть отступы, рамки, альтернативный текст.

Лабораторная работа №4

Создание графического меню с использованием навигационных областей HTML

Краткие теоретические сведения

В последнее время многие Web-страницы для организации ссылок используют так называемые карты-изображения. Реализация этой возможности предусмотрена языком HTML и позволяет привязывать гипертекстовые ссылки к различным областям изображения. Такой подход нагляднее, чем применение обыкновенных текстовых связей, поскольку пользователь может не читать словесное описание связи, а сразу понять ее смысл по графическому образу.

Карта-изображение фактически представляет собой обычное встроенное графическое изображение на Web-странице. Эти изображения могут иметь любой допустимый формат (GIF или JPG). При этом в формате GIF может использоваться прозрачный цвет, а также режим чередования строк. Для того чтобы изображение могло использоваться в качестве опорного для карты-изображения, формально не накладывается никаких дополнительных ограничений.

Конфигурация карты-изображения записывается в виде обычного текста, который в зависимости от используемого формата может

быть сохранен в отдельном файле или являться частью HTML-документа. Описание конфигурации содержит координаты для каждой из активных областей изображения, а также URL-адреса, связанные с каждой из этих областей. Активные области могут иметь форму прямоугольников, кругов и многоугольников.

Допускается любая комбинация этих фигур. Также может задаваться одно значение URL-адреса, определенное для случая, когда пользователь выполняет щелчок в пределах изображения, но вне любой из заданных активных областей. Конкретные правила записи конфигурации области зависят от выбранного варианта реализации.

Обычно используются карты-изображения серверные и клиентские. В данной лабораторной работе мы рассмотрим построение клиентских карт, поскольку серверная реализация данной возможности HTML выходит за рамки курса.

Для начала необходимо определить, какие именно области будут выделены. Зайдите в любой графический редактор и определите для каждой области свои координаты.

Создание карты рисунка начинается с описания

```
<map name="name">
```

Существуют три вида навигационных областей, для которых необходимо определить координаты их размещения на рисунке:

rect – прямоугольник, для него необходимо определить координаты левого верхнего и нижнего правого угла « X_1, Y_1, X_2, Y_2 »;

poly – полигон, область, состоящая из большого количества разнообразных кривых, для этой области необходимо определить координату каждой точки, соединяющей кривые « $X_1, Y_1, X_2, Y_2, \dots, X_n, Y_n$ »;

circle – окружность, определяется координата центра окружности и ее радиус « X, Y, R ».

Каждая область описывается отдельным тегом *area*:

```
<area shape="rect" COORDS="0, 0, 10, 20" href="http://google.com">
```

Для того чтобы связать созданную карту с рисунком, необходимо в описании рисунка *img* вставить свойство *usemap*:

```

```

Обратите внимание, что к имени карты в начале добавляется символ "#".

Рассмотрим небольшой пример создания навигационных областей.

Пример 1.

```
<map name=figure>
  <area shape="circle" coords="104, 100, 42"
  href="http://www.google.com">
  <area shape="poly" coords="153, 135, 186, 62, 228, 135"
  href="http://www.microsoft.com">
  <area shape="rect" coords="241, 60, 320, 135"
  href="http://www.open.by">
</map>

```

Каждая фигура выделена в отдельную область (рис. 4.1).



Рисунок 1 – Пример использования геометрических фигур в качестве указателей гиперссылок

Ход работы

1. Скопировать рисунок 1 в графический редактор и сохранить в рабочем каталоге с именем *geometry.jpg*.

Создать текстовый файл и внести в него фрагмент кода *Пример 1*, дополнив его необходимыми тегами начала и завершения файла.

Проверить, как работает файл, и разобраться с описанием геометрических фигур.

2. Создать HTML-файл с графическим меню следующего вида



Первые две ссылки должны открывать отдельные HTML-файлы, третья ссылка должна содержать указание на почтовый адрес студента. Первая ссылка голубая, вторая – зеленая, третья – желтая.

Графическое меню по желанию можно оформить более творчески.

Лабораторная работа № 5

Таблицы в интернет-документах

Тестовые упражнения для закрепления пройденного материала

1. Создать файл, в котором фоном является рисунок, указанный преподавателем.

2. Создать файл, в который внедрены рисунки с использованием выравнивания различными способами.

Краткие теоретические сведения

Описание таблиц должно располагаться внутри раздела документа `<BODY>`. Документ может содержать произвольное число таблиц, причем допускается вложенность таблиц друг в друга. Каждая таблица должна начинаться тегом `<TABLE>` и завершаться тегом `</TABLE>`. Внутри этой пары тегов располагается описание содержимого таблицы. Любая таблица состоит из одной или нескольких строк, в каждой из которых задаются данные для отдельных ячеек.

Каждая строка начинается тегом `<TR>` (Table Row) и завершается тегом `</TR>`. Отдельная ячейка в строке обрамляется парой тегов `<TD>` и `</TD>` (Table Data) или `<TH>` и `</TH>` (Table Header). Тег `<TH>` используется обычно для ячеек-заголовков таблицы, а `<TD>` – для ячеек-данных. Различие в использовании заключается лишь в типе шрифта, используемого по умолчанию для отображения со-

держимого ячеек, а также в расположении данных внутри ячейки. Содержимое ячеек типа <TH> отображается полужирным (Bold) шрифтом и располагается по центру (ALIGN=CENTER, VALIGN=MIDDLE). Ячейки, определенные тегом <TD>, по умолчанию отображают данные, выровненные влево (ALIGN=LEFT) и посередине (VALIGN=MIDDLE) в вертикальном направлении.

Теги <TD> и <TH> не могут появляться вне описания строки таблицы <TR>. Завершающие теги </TR>, </TD> и </TH> могут быть опущены. В этом случае концом описания строки или ячейки является начало следующей строки или ячейки или конец таблицы. Завершающий тег таблицы </TABLE> не может быть опущен. Количество строк в таблице определяется числом открывающих тегов <TR>, а количество столбцов – *максимальным количеством* <TD> или <TH> среди *всех строк*. Часть ячеек может не содержать никаких данных, такие ячейки описываются парой следующих подряд тегов: <TD>, </TD>. Если одна или несколько ячеек, располагающихся в конце какой-либо строки, не содержат данных, то их описание может быть опущено, а браузер автоматически добавит требуемое количество пустых ячеек. Отсюда следует, что построение таблиц, в которых в разных строчках располагается различное количество столбцов одного и того же размера, не разрешается.

Таблица может иметь заголовок, который заключается в пару тегов <CAPTION> и </CAPTION>. Описание заголовка таблицы должно располагаться внутри тегов <TABLE> и </TABLE> в любом месте, однако вне области описания любого из тегов <TD>, <TH> или <TR>. Согласно спецификации языка HTML расположение описания заголовка регламентировано более строго: оно должно располагаться сразу же после тега <TABLE> и до первого <TR>. По умолчанию текст заголовка таблицы располагается над ней (ALIGN=TOP) и центрируется в горизонтальном направлении.

Приведем пример простейшей таблицы, состоящей из двух строк и двух столбцов:

Ячейка 1 строки 1	Ячейка 2 строки 1
Ячейка 1 строки 2	Ячейка 2 строки 2

```

<HTML>
<HEAD>
<TITLE>Пример простейшей таблицы</TITLE>
</HEAD>
<BODY>
<Table border>
<TR>
<TD>Ячейка 1 строки 1</TD>
<TD>Ячейка 2 строки 1</TD>
</TR>
<TR>
<TD>Ячейка 1 строки 2</TD>
<TD>Ячейка 2 строки 2</TD>
</TR>
</TABLE>
</BODY>
</HTML>

```

Заголовок таблицы <CAPTION>

Тег заголовка таблицы <CAPTION> имеет единственный допустимый параметр ALIGN, принимающий значения TOP (заголовок над таблицей) или BOTTOM (заголовок под таблицей). Параметр ALIGN может быть опущен, что соответствует значению ALIGN=TOP. В горизонтальном направлении заголовок таблицы всегда располагается по ее центру. Таблица может не иметь заголовка.

Приведем пример записи заголовка таблицы:

```

<CAPTION ALIGN=БОТТОМ>Заголовок, располагаемый
внизу таблицы</CAPTION >

```

Параметры тега <TABLE>

Основным тегом, применяемым при создании таблиц, является тег <TABLE>. Он может использоваться с рядом параметров, каждый из которых допустимо опускать. Набор допустимых параметров зависит от браузера. Согласно спецификации HTML в теге <TABLE> могут использоваться следующие параметры:

BORDER
CELLSPACING
CELLPADDING
WIDTH
HEIGHT
ALIGN
BGCOLOR.

Параметр BORDER управляет изображением рамки вокруг каждой ячейки, совокупность которых, по сути, дает линии сетки таблицы, и вокруг всей таблицы. По умолчанию рамки не рисуются, и на экране пользователь увидит лишь ровно расположенный текст ячеек таблицы.

Для добавления в таблицу рамок необходимо включить в код <TABLE> параметр BORDER, который может иметь численное значение. Например, <TABLE BORDER> или <TABLE BORDER=10>.

Численное значение параметра определяет толщину рамки в пикселах, рисуемую вокруг всей таблицы, однако на толщину рамок вокруг каждой ячейки это значение не влияет. При отсутствии численного значения обычно оно принимается равным минимальному значению (1), хотя для различных браузеров стиль показа рамок может отличаться. Возможность независимого управления отображением рамки вокруг всей таблицы и рамками вокруг ячеек отсутствует.

Параметр CELLSPACING. Форма записи параметра: CELLSPACING = num, где num – численное значение параметра в пикселах, которое не может быть опущено. Величина num определяет расстояние между смежными ячейками (точнее между рамками ячеек) как по горизонтали, так и по вертикали. По умолчанию значение принимается равным двум. При задании CELLSPACING=0 рамки смежных ячеек сольются и создадут впечатление единой сетки таблицы.

Параметр CELLPADDING. Форма записи параметра CELLPADDING = num. Величина num определяет размер свободного пространства (отступа) между рамкой ячейки и данными внутри ячейки. По умолчанию значение принимается равным единице. Установка параметра CELLPADDING равным нулю может привести к тому, что некоторые части текста ячейки могут касаться ее рамки, что выглядит не очень эстетично.

Действие параметров `CELLPADDING` и `CELLSPACING` очень похоже друг на друга. Для таблицы без рамок изменение того или другого параметра приводит к одному и тому же результату. Оба параметра влияют на соответствующие отступы одновременно по горизонтали и по вертикали. К сожалению, раздельного управления горизонтальными и вертикальными отступами не предусмотрено.

Все три параметра – `BORDER`, `CELLPADDING` и `CELLSPACING` – действуют независимо друг от друга, если какой-нибудь из них опущен, то берется его значение, принятое по умолчанию.

Параметры `WIDTH` и `HEIGHT`. При отображении таблиц их ширина и высота автоматически вычисляется браузером и зависит от многих факторов: значений параметров, заданных в описании всего документа, данной таблицы, отдельных ее строк и ячеек, содержимого ячеек, а также параметров, задаваемых при просмотре документа в том или ином браузере, например, типа и размеров шрифта, размеров окна просмотра и др. Однако бывает необходимо принудительно указывать ширину или высоту таблицы. Для этой цели используют параметры `WIDTH` (ширина таблицы) и `HEIGHT` (высота таблицы) тега `<TABLE>`. Форма записи: `WIDTH = 120` или `WIDTH = 40 %`. Численное значение ширины всей таблицы задается в пикселах или в процентах от всего размера окна. Заметим, что допустимо задавать значения, большие 100 %, хотя трудно представить себе случаи, где это необходимо.

Аналогичные параметры могут задаваться и для отдельных ячеек. Заметим, что задание конкретного значения параметра не означает, что таблица в любом случае будет иметь указанную ширину, а лишь определяет рекомендуемую ширину, которая будет выдержана по возможности.

Параметр `ALIGN`. Данный параметр тега `<TABLE>` определяет горизонтальное расположение таблицы в области просмотра. Допустимые значения: `LEFT` (выравнивание влево) и `RIGHT` (выравнивание вправо). По умолчанию таблицы выровнены по левому краю. Заметим, что среди допустимых значений нет типичного значения для параметра выравнивания `CENTER`. В некоторых источниках по языку HTML значение `CENTER` (по центру) приводится в качестве допустимого в данном случае. Это соответствует спецификации HTML, но на практике и Netscape Navigator, и Microsoft Internet Explorer реализуют только два значения. Дело в том, что присутствие

параметра ALIGN в теге <TABLE> не только определяет месторасположение таблицы, но и разрешает выполнить обтекание таблицы текстом с противоположной стороны аналогично обтеканию картинок. Обтекание таблицы текстом с двух сторон не предусматривается ни в каких случаях. Для более точного управления обтеканием следует использовать тег
 с параметром CLEAR так же, как это выполняется для . Если параметр ALIGN опущен, то место справа и/или слева от таблицы всегда будет пустым независимо от ее ширины. Если таблица не требует обтекания текстом, то можно добиться ее расположения по центру окна просмотра. Для этого, например, можно все описание таблицы поместить внутри пары тегов <CENTER> и </CENTER>.

Приведем пример таблицы с обтекающим текстом..

```
<HTML>
<HEAD>
<TITLE>Таблица с обтекающим ее текстом</TITLE>
</HEAD>
<BODY>
<TABLE ALIGN=LEFT WIDTH=70%>
<CAPTION><H3>Наиболее употребительные мужские
имена<BR> взрослого
населения </H3>
</CAPTION>
<UL>
<TR><TD VALIGN=TOP>
<LI>Абрам <LI>Александр <LI>Алексей
<LI>Альберт <LI>Анатолий <LI>Андрей
<LI>Аркадий <LI>Борис <LI>Вадим <LI>Валентин
<LI>Валерий <LI>Василий
<LI>Виктор <LI>Виталий <LI>Владимир
<LI>Владислав <LI>Вячеслав
<LI>Геннадий <LI>Георгий <LI>Герман
<LI>Григорий <LI>Дмитрий
</TD>
<TD VALIGN=TOP>
<LI>Евгений <LI>Ефим <LI>Иван <LI>Игорь
<LI>Илья<LI>Иосиф <LI>Константин
```

```

<LI>Лев <LI>Леонид <LI>Михаил <LI>Николай
<LI>Олег <LI>Павел <LI>Петр
<LI>Роман <LI>Семен <LI>Сергей <LI>Станислав
<LI>Эдуард <LI>Юрий <LI>Яков
</TD>
</UL>
</TR>
</TABLE>
<BR><BR><BR><BR><BR><BR>

```

Приведенные данные получены на основе анализа репрезентативной выборки, содержащей сведения о 5000 мужчин в возрасте старше 18 лет.
 Указанные 43 наиболее часто встречаемых имени охватывают 92% выборки.
 Частота встречаемости каждого из остальных имен не превосходит 0.3%

```

</BODY>
</HTML>

```

Форматирование данных внутри таблицы

Каждую отдельную ячейку внутри таблицы можно рассматривать как область для независимого форматирования. Все правила, которые действуют для управления отображением текста, могут быть использованы для форматирования текста внутри ячейки. Внутри ячейки допустимо использование практически всех элементов HTML, которые могут появляться внутри тела документа <BODY>, в том числе теги, управляющие расположением текста – <P>,
, <HR>, коды заголовков – от <H1> до <H6>, теги форматирования символов – , <I>, , <BIG>, , , , теги вставки графических изображений , гипертекстовых ссылок <A> и т. д.

Сразу же подчеркнем, что область действия тегов, заданных внутри отдельной ячейки, ограничивается пределами этой ячейки независимо от наличия завершающего тега. Например, если внутри ячейки определен цвет текста - , то даже при отсутствии завершающего кода или расположения его через несколько ячеек или строк таблицы текст следующей ячейки будет отражен цветом по умолчанию.

Параметры для форматирования данных внутри ячеек таблицы

Параметры выравнивания содержимого ячеек – ALIGN и VALIGN. Могут применяться в кодах <TR>, <TD> и <TH>. Параметр горизонтального выравнивания ALIGN может принимать значения LEFT, RIGHT и CENTER (по умолчанию LEFT для <TD> и CENTER для <TH>). Параметр вертикального выравнивания VALIGN может принимать значения TOP (по верхнему краю), BOTTOM (по нижнему краю), MIDDLE (посередине), BASELINE (по базовой линии). По умолчанию – MIDDLE. Выравнивание по базовой линии обеспечивает привязку текста отдельной строки во всех ячейках к единой линии. Задание параметров выравнивания на уровне кода <TR> определяет выравнивание для всех ячеек данной строки, при этом в каждой отдельной ячейке строки может быть определены свои параметры, переопределяющие действие параметров, заданных в <TR>.

Приведем пример таблицы, в которой данные в ячейках первого столбца выровнены вправо, второго столбца – по центру, а третьего – влево (значение по умолчанию):

```
<HTML>
<HEAD>
< TITLE >Выравнивание элементов ТАБЛИЦЫ</TITLE>
</HEAD>
<BODY>
<TABLE BORDER WIDTH=100%>
<TR>
<TD ALIGN=RIGHT>Ячейка 1 <TD>
<TD ALIGN=CENTER> Ячейка 2</TD>
<TD>Ячейка 3</TD>
</TR>
<TR>
<TD ALIGN=RIGHT> Ячейка 4</TD>
<TD ALIGN=CENTER> Ячейка 5</TD>
<TD>Ячейка 6</TD>
</TR>
</TABLE>
</BODY>
</HTML>
```


Ячейка 1	Ячейка 2	Ячейка 3
Ячейка 4	Ячейка 5	Ячейка 6

Для сложных таблиц характерна потребность в объединении нескольких смежных ячеек по горизонтали или по вертикали в одну. Данная возможность реализуется с помощью параметров COLSPAN (COLunm SPANning) и ROWSPAN (ROW SPANning), задаваемых в кодах <TD> или <TH>. Форма записи: COLSPAN=num, где num – числовое значение, определяющее, на сколько столбцов следует расширить текущую ячейку по горизонтали. Применение параметра ROWSPAN аналогично, только здесь указывается количество строк, которые должна захватить текущая ячейка по вертикали. По умолчанию для этих параметров устанавливается значение, равное единице. Допустимо одновременное задание значений обоих параметров для одной ячейки. Правильная установка значений этих параметров может оказаться не очень простой задачей, тем более, что большинство HTML-редакторов позволяют визуально конструировать с последующей генерацией HTML-кодов лишь простейшие таблицы.

Далее показан пример отображения таблицы, полученный по следующему HTML-коду:

```
<HTML>
<HEAD>
<TITLE>Использование параметров COLSPAN и
ROWSPAN</TITLE>
</HEAD>
<BODY>
<TABLE BORDER>
<TR>
<TD ROWSPAN=2> Ячейка, захватывающая 2 строки
</TD>
<TD COLSPAN=2> Ячейка, захватывающая 2 столбца
</TD>
</TR>
<TR>
<TD>Ячейка 3</TD>
<TD>Ячейка 4</TD>
```

```

</TR>
<TR>
<TD>Ячейка 5</TD>
<TD>Ячейка 6</TD>
<TD>Ячейка 7</TD>
</TR>
</TABLE>
</BODY>
</HTML>

```

Ячейка, захватывающая 2 строки	Ячейка захватывающая 2 столбца	
	Ячейка 3	Ячейка 4
Ячейка 5	Ячейка 6	Ячейка 7

Ход работы

1. Создать HTML-файл на основе примеров приведенных в теоретической части и разобраться, как они работают.
2. Создать HTML-файл, в котором создать таблицы следующего вида:

Таблица 1

Код	Фамилия	Имя	Отчество	Год рождения	Факультет	Группа
1	Иванникова	Анна	Ивановна	1984	ЭФ	106713
2	Баранова	Ирина	Алексеевна	1983	ФИТР	107614

Таблица 2

№ п/п	Марка стали	A ₀	B ₁	B ₂	B ₃	t ₀	А при температуре t													
							1000	1010	1020	1030	1040	1050	1060	1070	1080	1090	1100			
1	Малоуглеродистая	54	0	32	0,24	975														

Лабораторная работа № 6

Фреймы.

Краткие теоретические сведения

Фреймы позволяют разбить окно просмотра браузера на несколько прямоугольных подобластей, располагающихся рядом друг с другом. В каждую из подобластей можно загрузить отдельный HTML-документ, просмотр которого осуществляется независимо от других. Между фреймами, также как и между отдельными окнами браузера, при необходимости можно организовать взаимодействие, которое заключается в том, что выбор ссылки в одном из фреймов может привести к загрузке нужного документа в другой фрейм или окно браузера.

Выбор фреймовой структуры отображения информации на WWW оправдан в следующих случаях:

- при необходимости организовать управление загрузкой документов в одну из подобластей окна просмотра браузера при работе в другой подобласти;
- для расположения в определенном месте окна просмотра информации, которая должна постоянно находиться на экране вне зависимости от содержания других подобластей экрана;
- для представления информации, которую удобно расположить в нескольких смежных подобластях окна, каждая из которых может просматриваться независимо.

Приведенный список не исчерпывает все возможные случаи, где можно применить фреймы, а носит рекомендательный характер. Фреймы очень похожи на таблицы — и те и другие осуществляют разбиение окна просмотра браузера на прямоугольные области, в которых располагается некоторая информация. Однако при помощи фреймов можно решить не только задачу форматирования страниц документа, но организовать взаимодействие между ними. Принципиальная разница между фреймами и таблицами состоит в том, что каждому фрейму должен соответствовать отдельный HTML-документ, а содержимое всех ячеек таблицы всегда является частью одного документа. Кроме того, отображаемая во фрейме страница

может прокручиваться при просмотре независимо от других. Каждый фрейм по существу является отдельным «мини-браузером». В отличие от фреймов, вся структура которых всегда представлена на экране, таблицы могут полностью не помещаться в окне и быть просмотрены только по частям. Отсюда следует вывод, что если в HTML-таблицах общее число ячеек практически не ограничено и может достигать нескольких сотен, то число фреймов в документе обычно не превосходит нескольких единиц.

Правила описания фреймов

Перейдем теперь к рассмотрению правил записи тегов, используемых для документов с фреймовыми структурами.

Для начала рассмотрим полный HTML-код, создающий документ с фреймами средней сложности.

Пример 1:

```
<HTML>
<HEAD>
<TITLE> </TITLE>
</HEAD>
<FRAMESET ROWS="25%, 50%, 25 %">
<FRAME SRC="header.htm">
<FRAMESET COLS="25%, 75 %">
<FRAME SRC="list.htm">
<FRAME SRC="info.htm">
</FRAMESET>
<FRAME SRC="footer.htm">
</FRAMESET>
<NOFRAMES>
Ваш браузер не может отображать фреймы
</NOFRAMES>
</HTML>
```

Этот пример создает страницу с фреймами, показанную на рис. 6.1. Как вы видите, этот HTML-код определяет четыре фрейма. Верхний фрейм занимает всю ширину страницы и содержит заголовок. Далее идут два центральных фрейма, один из которых расположен с левой

стороны и занимает 25 % от ширины экрана, а второй занимает оставшееся место. Последний, четвертый фрейм занимает нижнюю четверть экрана. В каждый из фреймов загружается отдельный HTML-документ, имя которого определяется параметром SRC.

Как видно из примера, для описания структуры фреймов применяются теги <FRAMESET>, <FRAME> и <NOFRAMES>. Рассмотрим назначение этих тегов.

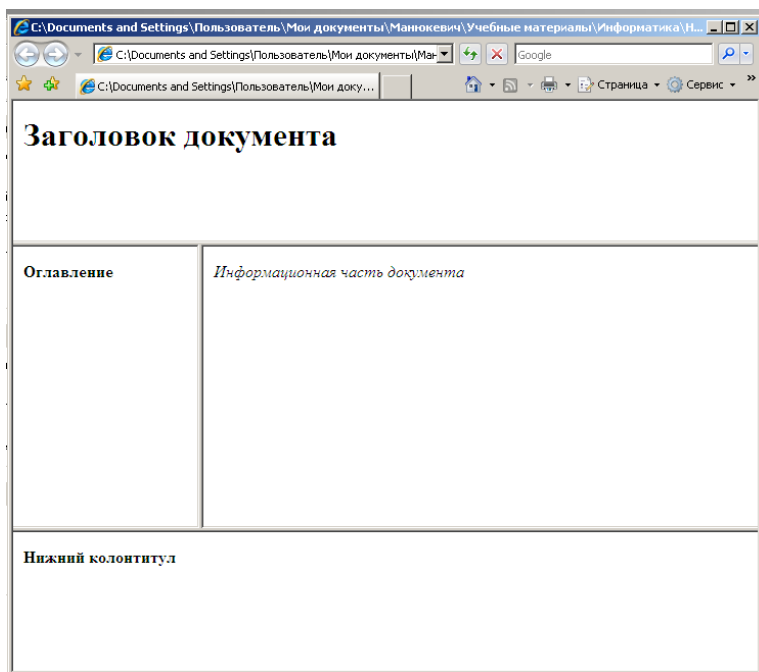


Рисунок 6.1- Документ с четырьмя фреймами

Тег <FRAMESET>

Фреймы определяются в структуре, называемой FRAMESET, которая используется для страниц, содержащих фреймы, вместо раздела BODY обычного документа. Web-страницы, составленные из фреймов, не могут содержать раздел BODY в своем HTML-коде. В свою очередь, страницы с разделом BODY не могут использовать фреймы. Контейнер из тегов <FRAMESET> и </FRAMESET> об-

рамляет каждый блок определений фрейма. Внутри контейнера <FRAMESET> могут содержаться только ТЕГИ <FRAME> И вложенные теги <FRAMESET>.

Тег <FRAMESET> имеет два параметра: ROWS (строки) и COLS (столбцы) и записывается в следующем виде:

```
<FRAMESET ROWS="Список значений" COLS=" Список значений">.
```

Можно определить значения для ROWS или COLS, или обоих вместе. Необходимо определить по меньшей мере два значения хотя бы одного из этих параметров. Если другой параметр опущен, то его значение принимается равным 100 %.Список значений параметров ROWS и COLS тега <FRAMESET> представляет собой разделенный запятыми список значений, которые могут задаваться в пикселах, в процентах или в относительных единицах. Число строк или столбцов определяется числом значений в соответствующем списке. Например, запись <FRAMESET ROWS="100,240,140"> определяет набор трех фреймов. Эти значения представляют собой абсолютные значения в пикселах. Другими словами, первый фрейм (первая строка) имеет высоту 100 пикселей, второй – 240 и последний – 140 пикселей.

Задание значений размеров фреймов в пикселах не очень удобно. Здесь не учитывается тот факт, что браузеры запускаются в различных операционных системах и с различными разрешениями дисплеев. В то же время можно определить абсолютные значения размеров для некоторых случаев, например для отображения небольшого изображения с известными размерами. Лучшим вариантом будет задание значений в процентах или в относительных единицах, например:

```
<FRAMESET ROWS="25%, 50%, 25%">.
```

В этом примере создаются три фрейма, размещаемые как строки во всю ширину экрана. Верхняя строка займет 25 процентов от доступной высоты экрана, средняя строка – 50 % и нижняя – 25%. Если сумма заданных процентов не равна 100 %, то значения будут пропорционально отмасштабированы, чтобы в итоге получилось ровно 100 %. Значения в относительных единицах выглядят следующим образом:

`<FRAMESET COLS="*, 2*, 3*">.`

Звездочка (*) используется для пропорционального деления пространства. Каждая звездочка представляет собой одну часть целого. Складывая все значения чисел, стоящих у звездочек (если число опущено, то подразумевается единица), получим знаменатель дроби. В этом примере первый столбец займет 1/6 часть от общей ширины окна, второй столбец — 2/6 (или 1/3), а последний — 3/6 (или 1/2).

Помните, что числовое значение без каких-либо символов определяет абсолютное число пикселей для строки или колонки. Значение со знаком процента (%) определяет долю от общей ширины (для COLS) или высоты (для ROWS) от окна просмотра, а значение со звездочкой (*) задает пропорциональное распределение оставшегося пространства.

Приведем пример, использующий все три варианта задания значений:

`<FRAMESET COLS="100,25%,*,2*">.`

В этом примере первый столбец будет иметь ширину 100 пикселей. Второй столбец займет 25 процентов от всей ширины окна просмотра, третий столбец — 1/3 оставшегося пространства и, наконец, последний столбец — 2/3. Абсолютные значения рекомендуется назначать первыми по порядку слева направо. За ними следуют процентные значения от общего размера пространства. В заключение записываются значения, определяющие пропорциональное разбиение оставшегося пространства. Если используется тег `<FRAMESET>`, в котором заданы значения и COLS, и ROWS, то будет создана сетка из фреймов.

Например:

`<FRAMESET ROWS="*,2*,*" COLS="2*,*">`

Эта строка HTML-кода создает сетку фреймов с тремя строками и двумя столбцами. Первая и последняя строки занимают 1/4 высоты каждая, а средняя строка — половину. Первый столбец занимает 2/3 ширины, а второй — 1/3.

Контейнер `<FRAMESET>` `</FRAMESET>` может быть вложен внутрь другого та-

кого же контейнера, как это было показано в начальном примере. Рассмотрим далее использование тега `<FRAME>`.

Тег <FRAME>

Тег `<FRAME>` определяет одиночный фрейм. Он должен располагаться внутри пары тегов `<FRAMESET>` и `</FRAMESET>`. Например:

```
<FRAMESET ROWS="*,2*">  
<FRAME>  
<FRAME>  
</FRAMESET>
```

Обратите внимание, что тег `<FRAME>` не является контейнером и в отличие от `<FRAMESET>` не имеет завершающего тега. Все определение одиночного фрейма выполняется одной строчкой HTML-кода. Необходимо записать столько тегов `<FRAME>`, сколько отдельных фреймов определено при задании тега `<FRAMESET>`. В предыдущем примере тегом `<FRAMESET>` задано две строки, поэтому потребовалось записать два тега `<FRAME>`. Однако этот пример, по существу, бесполезен, так как ни один из фреймов не имеет какого-либо содержания!

Тег `<FRAME>` имеет шесть параметров: SRC, NAME, MARGINWIDTH, MARGINHEIGHT, SCROLLING и NORESIZE.

Приведем запись тега `<FRAME>` со всеми параметрами:

```
<FRAME SRC="url" NAME="window_name" SCROLLING=YES  
|NO| AUTO MARGINWIDTH="value" MARGINHEIGHT="value"  
NORESIZE>
```

На практике в теге `<FRAME>` редко используются одновременно все параметры. Наиболее важный параметр — SRC (сокращение от слова source). Довольно часто в теге `<FRAME>` задается единственный параметр SRC. Например:

```
<FRAME SRC="url">
```

Значение параметра SRC определяет URL-адрес документа, который будет загружен изначально в данный фрейм. Обычно в качестве такого адреса записывается имя HTML-файла, расположенного

в том же самом каталоге, что и основной документ. Тогда строка определения фрейма будет выглядеть, например, так:

```
<FRAME SRC="sample.htm">.
```

Обратите внимание, что любой HTML-файл, заданный в описании фрейма, должен быть полным HTML-документом, а не фрагментом. Это означает, что документ должен иметь теги HTML, HEAD, BODY и т. д. Конечно, в качестве значения SRC может быть задан любой допустимый URL-адрес. Если, например, фрейм используется для отображения изображения в формате GIF, которое располагается на сервере, то следует записать:

```
<FRAME SRC="http://www.BNTU.BY/example.gif ">.
```

Параметр NAME определяет имя фрейма, которое может использоваться для ссылки к данному фрейму. Обычно ссылка задается из другого фрейма, располагающегося на той же самой странице. Например:

```
<FRAME SRC="sample.htm" NAME="Frame_1" > .
```

Такая запись создает фрейм с именем "Frame_1", на который может быть выполнена ссылка. Например:

```
<A HREF="other.htm" TARGET="Pgate_1">Щелкните здесь для загрузки документа other.htm во фрейм с именем Frame_1</A>.
```

Обратите внимание на параметр TARGET, который ссылается на имя фрейма. Если для фрейма не задано имя, то будет создан фрейм без имени, и не будет возможности использовать ссылки на него из другого фрейма. Имена фреймов должны начинаться с алфавитно-цифрового символа.

Параметры MARGINWIDTH и MARGINHEIGHT дают возможность устанавливать ширину полей фрейма. Записывается это следующим образом:

```
MARGINWIDTH="value",
```

где "value" — абсолютное значение в пикселах.

Например:

<FRAME MARGINHEIGHT= "5" MARGINWIDTH= "7">.

Данный фрейм имеет поля сверху и снизу по 5 пикселей, а слева и справа по 7 пикселей. Не забудьте, что здесь идет речь о полях, а не о рамках. Параметры MARGINWIDTH и MARGINHEIGHT определяют пространство внутри фрейма, в пределах которого не будет располагаться никакая информация. Минимально допустимое значение этих параметров равно единице.

Для фреймов будут автоматически создаваться и отображаться полосы прокрутки, если содержимое фрейма не помещается полностью в отведенном пространстве. Иногда это нарушает дизайн страницы, поэтому было бы удобно иметь возможность управлять отображением полос прокрутки. Для этих целей используется параметр SCROLLING. Формат записи:

<FRAME SCROLLING="YES/NO/AUTO">.

Параметр SCROLLING может принимать три значения: YES, NO или AUTO. Значение AUTO действует так же, как и в случае отсутствия параметра SCROLLING. Значение YES вызывает появление полос прокрутки вне зависимости от необходимости этого, а NO — запрещает их появление. Например:

<FRAME SCROLLING=YES>.

Обычно пользователь может изменять размер фреймов при просмотре страницы. Если установить курсор мыши на рамке фрейма, то курсор примет форму, указывающую на возможность изменения размеров, и позволит выполнить перемещение рамки в нужное место. Это иногда нарушает структуру красиво спроектированных фреймов. Для предотвращения возможности изменения пользователем размера фреймов следует воспользоваться параметром NORESIZE:

<FRAME NORESIZE>.

Этот параметр не требует никаких значений. Естественно, когда задан параметр NORESIZE для одного из фреймов, то размер любо-

го из смежных фреймов также не может быть изменен. Иногда, в зависимости от расположения фреймов, использования параметра NORESIZE в одном из фреймов будет достаточно, чтобы предотвратить возможность изменения размеров любого из них на экране.

Тег <NOFRAMES>

Возможность работы с фреймами не предполагалась ни в стандарте HTML 3.0, ни в HTML 3.2. Здесь до последнего времени складывалась достаточно типичная ситуация, когда реально используемые возможности активно применяются на многих WWW-страницах, однако не являются частью стандарта. Это означало, что браузеры вполне законно могли игнорировать фреймы.

С появлением стандарта HTML 4.0 ситуация изменилась — теперь поддержка фреймовых структур закреплена стандартом. Заметим, что большинство современных браузеров распознавали фреймы и до появления HTML 4.0. Тем не менее, необходимо предоставлять информацию пользователям, применяющим браузеры без поддержки фреймов. Для таких браузеров можно предусмотреть альтернативную информацию, которая записывается между парой тегов <NOFRAMES> и </NOFRAMES>. Это выглядит следующим образом:

```
<NOFRAMES>  
весь HTML-документ  
</NOFRAMES>
```

Все, что размещено между тегами <NOFRAMES> и </NOFRAMES>, будет отображаться браузерами, не имеющими возможностей поддержки фреймов. Браузеры с поддержкой фреймов проигнорируют всю информацию между этими тегами.

Заметим, что в реальной жизни разработчики HTML-страниц часто не используют возможности тега <NOFRAMES> для создания страниц без фреймовых структур, а просто создают две версии своих HTML-документов. Для такого варианта на стартовой странице обычно предлагается выбор загрузки документа с фреймовой структурой или без нее. Далее в зависимости от выбора пользователя загружается только один вариант документа.

Ход работы

1. Создать документ на основании текста программы приведенного в теоретической части (пример 1).

2. Создать фреймовую структуру следующего вида:

ФРЕЙМ 1: Заголовок сайта	
ФРЕЙМ 2: Список ссылок на фотографии студенческой группы. Фотографии открываются во ФРЕЙМЕ 3.	ФРЕЙМ 3: В этом фрейме открываются фотографии по ссылкам из ФРЕЙМА 2.
ФРЕЙМ 4: В этом фрейме открываются подписи к фотографиям, открываемым во ФРЕЙМЕ 3.	

Размеры фреймов задать самостоятельно.

Список использованных источников

1. Матросов, А. В. HTML 4.0 / А. В. Матросов, А. О. Сергеев, М. П. Чаунин – СПб. : БХВ-Петербург, 2004. – 671 с. : ил. – (В подлиннике).
2. ДеБольш, Вирджиния HTML и CSS. Совместное использование / ДеБольш Вирджиния. – М. : NT Press, 2006. – 499 с. : ил. – (Школа Web-мастерства).
3. Комолова, Н. В. HTML : учебный курс / Н. В. Комолова. – СПб : Питер, 2007. – 267 с. : ил. – (Учебный курс).
4. Квинт И. HTML, XHTML и CSS на 100% / Квинт И. . - Санкт-Петербург [и др.] : Питер, 2011. - - 383 с. : ил., табл.. - (На 100%)

Учебное издание

**СЕТЕВЫЕ ТЕХНОЛОГИИ
И БАЗЫ ДАННЫХ**

Лабораторный практикум

В 2 частях

Часть 2

РАЗРАБОТКА HTML-СТРАНИЦ

Составитель
МАНЮКЕВИЧ Александр Владимирович

Редактор *В. О. Кутас*

Подписано в печать 03.09.2012. Формат 60×84 ¹/₁₆. Бумага офсетная. Ризография.

Усл. печ. л. 3,55. Уч.-изд. л. 2,77. Тираж 100. Заказ 1110.

Издатель и полиграфическое исполнение: Белорусский национальный технический университет. ЛИ № 02330/0494349 от 16.03.2009. Пр. Независимости, 65. 220013, г. Минск.