Министерство образования и науки Российской Федерации

НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Факультет инновационных технологий

С. Л. Миньков

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

Лабораторный практикум
Часть 1

Томск 2017

Миньков С.Л.

Информационные технологии: лабораторный практикум. Ч.1 – Томск: ТГУ, 2017.– 53 с.

ЛАБОРАТОРНАЯ РАБОТА
РАЗРАБОТКА ТЕМАТИЧЕСКОГО WEB-САЙТА С ИСПОЛЬЗОВАНИЕМ ТЕКСТОВОГО РЕДАКТОРА

Цель работы: освоить основные навыки работы с программами и сервисами Интернета, позволяющими осуществлять поиск и получение информации из Сети, а также освоить приемы разработки статических html-страниц тематического web-сайта и управления ими с использованием текстового редактора.

Введение

В конце XX века появилась и быстрыми темпами стала развиваться глобальная коммуникационная среда – Интернет. Это не только распределенная сеть, соединяющая сотни миллионов компьютеров по всему миру, но и новая форма общедоступного архива, основанная на гипертекстовом представлении мультимедийной информации.

Глобальные информационные сети становятся основой современной жизни во всех ее проявлениях: экономической, социальной, политической, культурной. Своей открытостью, доступностью, устойчивостью к сбоям Интернет способствует большей открытости, оперативной информированности, широкой гласности в обществе и бизнесе.

Одним из самых популярных сервисов Интернета наряду с электронной почтой является World Wide Web (WWW) – «всемирная паутина». Это название достаточно точно определяет суть сервиса. WWW – гигантская сеть страниц (документов), связанных между собой гипертекстовыми ссылками, что действительно очень напоминает своими колоссальными размерами и структурой паутину (web), протянувшуюся по всему миру.

Гипертекст – это легкая в использовании и чрезвычайно мощная система связанных слов и фраз, позволяющая легко перемещаться по особым образом организованным страницам. Она связывает фразу или слово одной страницы с любой другой страницей, абзацем, фразой или словом.

Если развить идею гипертекста и включить в него графику, видео и звук, мы получим гипермедиа. **Гипермедиа** – среда, основанная, как и гипертекст, на взаимосвязях, в которой в качестве гиперссылок могут также выступать визуальные и аудиокомпоненты. Гипертекст и гипермедиа являются фундаментальными для WWW технологиями, а HTML – средство для работы с этими технологиями.

World Wide Web можно представить и как большую библиотеку Internet. Каждый узел «паутины» – Web-сайт подобен книге из этой библиотеки, а Web-страницы подобны страницам этих книг. Обычно путешествие по WWW начинается с определенного узла.

Первая страница представляет собой исходную точку для узла – нечто подобное обложке или содержанию книги.

Каждая страница, включая начальную страницу узла, имеет уникальный адрес в формате URL (Uniform Resource Locator).

Страницы Web являются гипертекстовыми документами, написанными в формате HTML (HyperText Markup Language), и имеют расширение .html или .htm.

Для передачи гипертекстовых документов используется сетевой протокол HTTP (HyperText Transfer Protocol).

URL для HTTP выглядит так:

http://<host>

где <host> – доменное имя или IP-адрес сервера.

Например, http://192.252.19.1, http://www.fit.tsu.ru.

1. Структура html-документа

Web-страница – это текстовый документ, размеченный с помощью специальных элементов языка HTML, которые называются **теги**, или html-теги. Такие страницы часто называют html-страницами. Они имеют расширение .html или .htm. На экран html-теги не выводятся, они только указывают браузеру, как отображать содержимое документа.

Посмотреть html-код любой страницы в браузере можно, вызвав на этой странице контекстно-зависимое меню и щелкнув по строке «Исходный код страницы» (в браузере Mozilla Firefox), «Просмотр HTML-кода» (в браузере MS Internet Explorer)» или «Просмотр кода страницы» (в браузере Google Chrome).

HTML определяет логическую структуру документа и не является языком программирования. Это именно «язык разметки» документов (текста).

Теги html бывают двух типов – **контейнерные** и **одиночные** и заключаются в угловые скобки **<имя_тега>**. Регистр записи имени тега не имеет значения.

Контейнерные теги состоят из пары: открывающий тег <имя_тега> и закрывающий тег </имя_тега>. Закрывающий тег завершает действие открывающего, например теги **<html>** и **</html>** начинают и заканчивают html-страницу.

Одиночный тег не требует закрывающего тега, например тег **
** начинает новую строку.

Для работы рекомендуется использовать не стандартный Блокнот (Notepad) Windows, а текстовый редактор Notepad++.

Notepad++ (https://notepad-plus-plus.org) – это бесплатный редактор текстовых файлов с поддержкой синтаксиса большого количества языков программирования: C, C++, Java, C#, XML, HTML, PHP, Javascript, Pascal, Fortran, Perl, Python, VB и т.д. Он ориентирован для работы в операционной системе MS Windows, позволяет подсвечивать синтаксис в файлах, поддерживает поиск и замену по тексту, работу одновременно с несколькими файлами, а также поддержку функции drag & drop. В сентябре 2017 г. текущей версией была v.7.5.1.

Итак, откроем Notepad++ и наберем следующий текст с тегами (рис. 1).

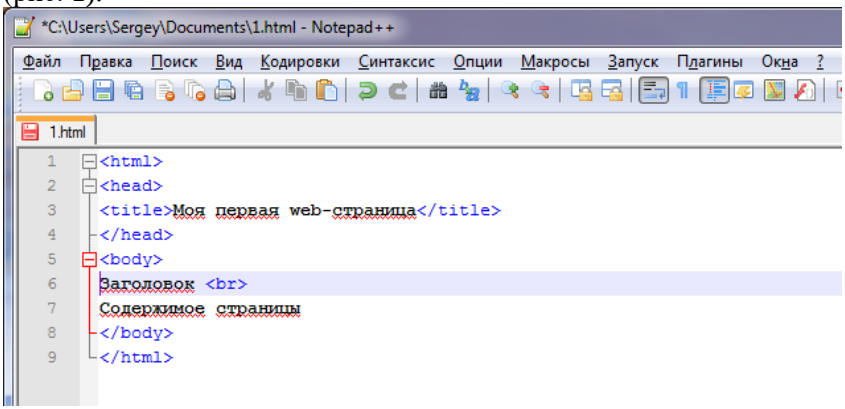


Рис. 1. Код html в Notepad++

Сохраним этот текст в файле с именем 1.html в папке Мой сайт, выбрав Тип файлов All types (".") (рис. 2).

Закроем Блокнот и перейдем в папку Мой сайт. Сохраненный файл как html-документ по умолчанию будет открываться браузером (рис.3).

Теперь вся дальнейшая работа будет заключаться в добавлении все новой и новой информации в этот пока унылый информационный пейзаж по следующему алгоритму:

1) щелкнуть правой кнопкой мыши по полю web-страницы и вызвать «Просмотр HTML-кода» (в IE) или «Edit with Notepad++» (в Chrome или в IE) ;

2) внести изменения в html-код;

3) сохранить изменения и закрыть (желательно) Блокнот;

4) обновить web-страницу (кнопка F5 на клавиатуре или кнопка Обновить в Панели инструментов браузера) и проанализировать изменения;

5) повторить с позиции 1.

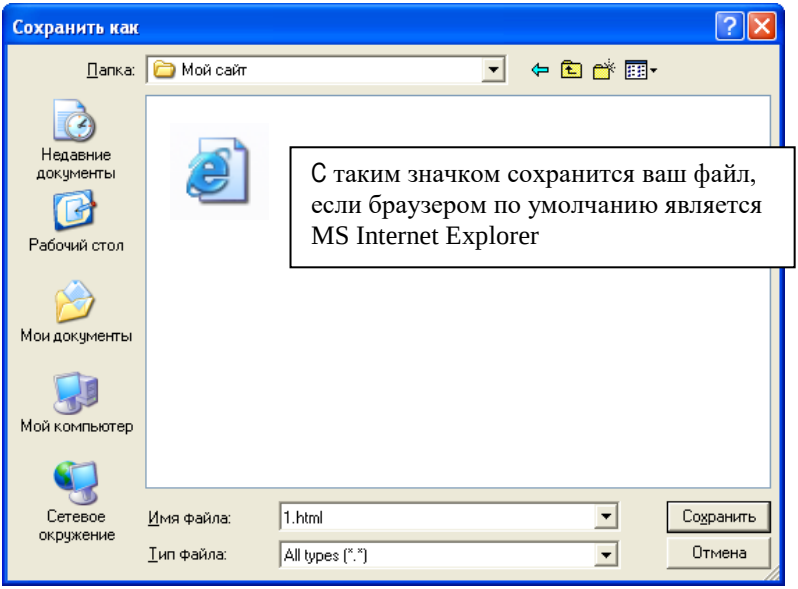


Рис. 2. Первое сохранение html-страницы

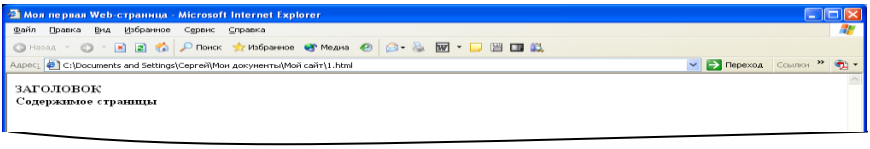


Рис. 3. Отображение html-страницы в браузере

Рассмотрим структуру html-документа.

Документ состоит из двух частей: заголовок, выделяемый тегами <**head**> и </**head**>, и тело, выделяемое тегами <**body**> и </**body**>. В заголовке содержится информация о документе – название, мета-информация, ссылки на внешние и внутренние таблицы стилей, сценарии JavaScript и т. д. В теле находится само содержимое документа – то, что выводится в окне браузера: текст (здесь: ЗАГОЛОВОК Содержимое страницы), картинки, таблицы и т. д.

Как любой язык, HTML позволяет вставлять в тело документа **комментарии**, которые сохраняются при передаче документа по сети, но не отображаются браузером.

Синтаксис комментария:

<!-- Это комментарий -->

Комментарии могут встречаться в документе где угодно и в любом количестве.

Название документа (Моя первая Web-страница) выделяется тегами <**title**> и </**title**>. Это название отображается в заголовке окна браузера, открывающего документ, и в файле закладок «Избранное».

Для расширения или модификации действия тега в него добавляют *атрибуты*, которые могут принимать различные значения. Наборы допустимых атрибутов для конкретного тега описаны в спецификации языка HTML Некоторые из них приведены в Приложении 1.

Правила записи атрибутов и значений:

- атрибуты следуют после имени тега через пробел;
- атрибуты отделяются друг от друга пробелами;
- порядок следования атрибутов произволен;

– атрибуты не нужно повторно описывать в закрывающем теге;

– значения атрибутов записывают после знака равенства в кавычках "…" (кавычки можно опускать, если значение атрибута состоит из одного слова или числа).

В тег <body> могут входить атрибуты, определяющие параметры страницы, такие, как цвет фона, фоновый рисунок, цвет и размер основного шрифта и гиперссылок и т. п. Цвета этих элементов web-страницы указываются в шестнадцатеричном исчислении. Но существует и возможность словесного указания цвета (см. Приложение 2).

Включим в тег <body> следующие атрибуты и зададим им значения:

<body bgcolor="#00FFFF" text=red>

Атрибут **bgcolor** задает цвет фона, а **text** – цвет текста.

После сохранения и обновления страницы на бирюзовом фоне вы увидите красные буквы.

Для большинства документов хорошо выглядит в качестве фона ненавязчивый полупрозрачный рисунок (обои). Для того чтобы оформить страничку фоновым рисунком, в теге <body> нужно указать атрибут **background**="имя_файла". В этом случае графический файл с фоновым рисунком должен находиться **в том же каталоге, что html-страница**. В качестве фонового рисунка могут быть использованы только файлы формата GIF, JPG (JPEG) и PNG.

Пример использования этого атрибута:

<body background="back.jpg">

Регистр символов в имени файла имеет значение. Расширение указывать обязательно!

Большое значение имеет выбор рисунка: и цветовая насыщенность, и размер. На ярком рисунке будет не виден текст, а маленький рисунок на web-странице будет выложен плиткой.

Большие коллекции фоновых рисунков можно найти , например, на сайтах https://www.toptal.com/designers/subtlepatterns, http://bg.fantasyflash.ruи др.

Кстати, небольшой рисунок можно увеличить до необходимых размеров (с некоторой потерей качества), воспользовавшись графическими редакторами. Например, в состав MS Office 2003 и

поздних выпусков входит Microsoft Picture Manager. Открыв им картинку, выберите в главном меню опцию **Рисунок → Изменить размер**. В правой части появившегося окна задайте или выберите необходимый размер. Сохраните файл.

Рекомендуется все графические файлы, входящие в состав html-документа, хранить в отдельной папке, например images. Тогда в соответствующих атрибутах надо указать **весь путь**:

<body background="images/back.jpg">

Рекомендуется также названия всех файлов (текстовых, графических) вашего сайта писать строчными буквами по причине разного восприятия различными серверами регистров букв.

2. Управление текстом

2.1. Заголовки

Для задания заголовков в документе используются контейнерные теги <h1>, <h2>, <h3>,<h4>, <h5>, <h6> (в порядке убывания размера). По умолчанию браузеры выводят заголовки жирным начертанием.

Пример:

<h1>Заголовок</h1>

Заголовки можно выравнивать по левому краю (по умолчанию), по правому краю и по центру. Для этого используется атрибут **align** с параметрами соответственно **left, right, center**.

Пример:

<h1 align=center >Заголовок</h1>

Заголовок будет отцентрирован.

2.2. Абзацы (параграфы)

Для разбивки текста на абзацы используется контейнерный тег <**p**>. Встретив этот тег, браузер пропускает строку и начинает текст с новой строки. Это заметно облегчает восприятие текста, нежели сплошной текстовый блок – «кирпич».

Например,

 Текст абзаца

Текст абзацев можно выравнивать по левому краю (по умолчанию), по правому, по центру и по ширине.

Для этого используется атрибут **align** с параметрами **left, right, center, justify**.

Например, запись

 Текст абзаца

выровняет абзац по ширине.

Для выравнивания текста или встроенных картинок также можно использовать контейнерный тег ** **.

2.3. Цитата

Контейнерный тег <**blockquote**> предназначен для обозначения в документе цитаты из другого источника. Текст, обозначенный тегом <blockquote>, отступает от левого края документа на 8 пробелов.

Пример:

<blockquote>Текст цитаты</blockquote>

2.4. Списки

Списки бывают трех видов: неупорядоченные, упорядоченные и списки определений.

Неупорядоченные списки создаются тегами <**ul**> **</ul**>. Внутри помещается весь список. Отдельные элементы списка отмечаются тегом <**li**> (закрывающий тег не требуется). У команд <ul> и <li> можно указать атрибут **type**, который принимает одно из трех значений: **disc**, **square**, **circle**.

Пример.

В html-коде:	**На странице браузера:**
    Первый пункт Второй пункт Третий пункт <ul type=circle> Первый пункт <li type=square> Второй пункт Третий пункт 	• Первый пункт • Второй пункт • Третий пункт ○ Первый пункт ▪ Второй пункт ○ Третий пункт

Упорядоченные списки создаются тегами <**ol**> и </**ol**>.

У тега <ol> есть два атрибута **type** и **start**. Атрибут type задает тип нумерации:

1 – арабские числа 1, 2, 3, ...;

a – прописные буквы a, b, c, ...;

A – заглавные буквы A, B, C, ...;

I – маленькие римские числа i, ii, iii, ...;

I – большие римские числа I, II, III, ...

Атрибут start указывает, с какого номера начинать список.

У тега <li> можно указать тоже два атрибута: **type** и **value**. С помощью value можно задать номер элемента в списке.

Пример.

В html-коде:	**На странице браузера:**
<ol type=i start=5>	
Первый пункт	v. Первый пункт
Второй пункт	vi. Второй пункт
<li type=1> Третий пункт	7. Третий пункт
<ol type=I>	
Первый пункт	I. Первый пункт
Второй пункт	II. Второй пункт
<li value=10> Третий пункт	X. Третий пункт

Списки определений. Для создания списка используются теги <**dl**> и </**dl**>. Элемент списка создается двумя тегами **<dt>** и **<dd>.** Они не требуют закрывающих тегов. Первый тег задает термин, второй – определение. Для большей наглядности термин можно выделить жирным шрифтом.

Пример.

В html-коде:

```
<dl>
<dt> <b> WWW (World Wide Web) </b>
<dd> Всемирная Паутина – основная служба в сети Интернет, позволяющая получать доступ к информации на любых серверах, подключенных к сети
<dt> <b> DNS (Domain Name System) </b>
<dd> Система доменных имен – распределенная служба формирования имен узлов, используемая в сети Интернет, устанавливающая соответствие между именами узлов и доменов, с одной стороны, и IP-адресами, с другой стороны.
<dt> <b> HTTP (HyperText Transfer Protocol) </b>
<dd> Протокол передачи гипертекста – базирующийся на TCP/IP протокол передачи гипертекста, обеспечивающий доступ к документам на web-узлах.
</dl>
```

На странице браузера:

WWW (World Wide Web)
Всемирная Паутина – основная служба в сети Интернет, позволяющая получать доступ к информации на любых серверах, подключенных к сети

DNS (Domain Name System)
Система доменных имен – распределенная служба формирования имен узлов, используемая в сети Интернет, устанавливающая соответствие между именами узлов и доменов, с одной стороны и IP-адресами, с другой стороны.

HTTP (HyperText Transfer Protocol)
Протокол передачи гипертекста – базирующийся на TCP/IP протокол передачи гипертекста, обеспечивающий доступ к документам на web-узлах.

2.5. Выделение текста

Для выделения части текста **полужирным** шрифтом используются теги контейнерного типа

**** Текст ****,

курсивом – теги

**** Текст ****,

текстом, имитирующим шрифт пишущей машинки, – теги

<tt> Текст **</tt>.**

Тег **<sup>** определяет верхний индекс, а тег **<sub>** – нижний, например, формула $I_i = x_i^2 + a_{i,j}$ в html-коде будет выглядеть так:

I_i=x_i²+a_{i,j}

Обратите внимание, какой длинный и трудно читаемый (да и трудоемкий в исполнении) получился код. Поэтому вставка сложных формул в текст обычно осуществляется картинкой.

2.6. Управление шрифтами

Для управления шрифтовым оформлением web-страниц предназначен контейнерный тег ****. Атрибуты этого тега определяют размер, цвет и название шрифта. Текст, выделенный тегами , выглядит иначе, чем это задано по умолчанию или в теге <body>.

Пример.

Текст

Атрибут **face** задает название шрифта (по умолчанию Times New Roman). Обычно в Web используются немного разных видов шрифтов (именно они однозначно воспринимаются всеми браузерами). Это: Times New Roman, Arial, Courier New, Tahoma, Verdana.

Если же вы хотите сделать витиеватый заголовок – выполните его в графическом редакторе и вставьте картинкой.

Атрибут **size** определяет размер шрифта. Шрифт может иметь размер от 1 до 7. Можно прямо указать размер шрифта цифрой или указать смещение относительно базового значения (по умолчанию 3) в положительную или отрицательную сторону, например задание

size=+2

определит размер 5.

Атрибут **color** определяет цвет текста аналогично заданию цвета фона в теге <body>.

2.7. Спецсимволы

Как ввести в текст символы, которые отсутствуют на клавиатуре, например греческие буквы, или символы ×, ©, ®, или неразрывный пробел (в Word – сочетание клавиш Shift+Ctrl+Пробел), или знаки < , >, ", которые входят в состав тегов? Для этого используются особые последовательности символов (escape-последовательности), начинающиеся с символа & («амперсанд»).

Например, группа из 5…8 escape-последовательностей ** ** используется для получения абзацных отступов.

В таблицах 1 и 2 указаны некоторые escape-последовательности.

Таблица 1. Наиболее распространенные escape-последовательности

Название	**Символ**	**&-имя**
Прямая кавычка	"	"
Амперсанд	&	&
Неразрывный пробел (невидимый символ)	◦	
Цент	¢	¢
Фунт	£	£
Копирайт	©	©
Зарегистрированный торговый знак	®	®
Градус	°	°
Плюс-минус	±	±
Знак умножения	×	×
Знак параграфа	§	§

Эти сочетания символов просто размещаются в нужном месте текста, ничем более не выделяясь, и будут интерпретированы

браузером как соответствующий символ. Точка с запятой, входящая в состав escape-последовательности, может опускаться.

Таблица 2. Греческий алфавит в HTML

Буква		**Название**	**Обозначение в HTML**	
прописная	**строчная**		**прописная**	**строчная**
Α	α	альфа	&#913;	&#945;
Β	β	бета	&#914;	&#946;
Γ	γ	гамма	&#915;	&#947;
Δ	δ	дельта	&#916;	&#948;
Ε	ε	эпсилон	&#917;	&#949;
Ζ	ζ	дзета	&#918;	&#950;
Η	η	эта	&#919;	&#951;
Θ	θ	тета	&#920;	&#952;
Ι	ι	йота	&#921;	&#953;
Κ	κ	каппа	&#922;	&#954;
Λ	λ	лямбда	&#923;	&#955;
Μ	μ	мю	&#924;	&#956;
Ν	ν	ню	&#925;	&#957;
Ξ	ξ	кси	&#926;	&#958;
Ο	ο	омикрон	&#927;	&#959;
Π	π	пи	&#928;	&#960;
Ρ	ρ	ро	&#929;	&#961;
Σ	σ	сигма	&#931;	&#963;
Τ	τ	тау	&#932;	&#964;
Υ	υ	ипсилон	&#933;	&#965;
Φ	φ	фи	&#934;	&#966;
Χ	χ	хи	&#935;	&#967;
Ψ	ψ	пси	&#936;	&#968;
Ω	ω	омега	&#937;	&#969;

2.8. Линия

Одиночный тег **<hr>** вставляет в текст горизонтальную разделительную линию.

Атрибуты тега:

width определяет длину линии в пикселях или процентах от ширины окна браузера;

size определяет толщину линии в пикселях;

align определяет выравнивание горизонтальной линии: =**left** – по левому краю документа, =**right** – по правому краю документа, =**center** –по центру документа (используется по умолчанию);

noshade определяет способ закраски линии как сплошной (не требует указания значения). Без данного атрибута линия отображается объемной;

color определяет цвет линии.

3. Вставка рисунков

Для размещения картинок на web-страницах существует одиночный тег **<img>**. Требования к графическому файлу – те же, что и для фонового рисунка (см. п. 1).

В html-коде нужно указать

<img src="путь" width=*n* height=*m* hspace=*h* и vspace=*v* border=*b* align=right alt="Подпись">

Атрибуты этого тега:

src – обязательный параметр. Указывает адрес (URL) файла с картинкой; ошибка в его написании (что очень часто встречается у начинающих) приводит к тому, что картинка упорно не хочет отображаться в браузере. Поэтому будьте внимательны;

width и **height** задают ширину *n* и высоту *m* изображения (в пикселях) соответственно. Имейте в виду, что если указанные значения не совпадают с реальным размером изображения, изображение масштабируется (порой с заметной потерей качества);

hspace и **vspace** определяют отступ картинки (в пикселях) по горизонтали (*h*) и вертикали (*v*) от других объектов документа. Просто необходимо при обтекании изображения текстом;

border определяет ширину рамки (*b*) вокруг изображения в пикселях;

align определяет способ выравнивания изображения в документе. Если его не указать, картинка раздвигает строки текста,

расположенные до и после нее и располагается слева (отцентрировать можно тегами).

Некоторые параметры атрибута align:

left – выравнивает изображение по левому краю документа, прилегающий текст обтекает изображение справа;

right – выравнивает изображение по правому краю документа, прилегающий текст обтекает изображение слева.

Атрибут **alt** определяет текст, отображаемый браузером при наведении курсора мыши на изображение или появляющийся на его месте, если браузер по каким-то причинам не может вывести изображение.

Примеры.

<img src="pict.gif" border=0 width=100 height=30> – рисунок расположен по центру страницы, рамка отсутствует, размеры 100×30 пикселей, подпись к рисунку отсутствует;

<img src="pic.jpg" width=120 height=90 align=left hspace=5 vspace=5 border=3 alt="Зимнее утро"> – рисунок размером 120×90 пикселей обтекается текстом (рисунок слева, текст справа) на расстоянии 5 пикселей от рамки толщиной 3 пикселя, при наведении появляется текст «Зимнее утро».

4. Гиперссылки

4.1. Текстовые гиперссылки

Для того чтобы объединить несколько страниц в один сайт, необходимо связать их гиперссылками. Это и есть основной принцип построения WWW.

Гиперссылка может быть присвоена любому элементу текста, расположенного на странице. Для организации гиперссылки существует контейнерный тег <A>. Между открывающим и закрывающим тегами располагается текст, который будет отображен подчеркнутым в окне браузера. Именно этот текст и будет объектом, которому назначена гиперссылка на какой-то

другой объект вашего сайта или глобальной сети[1].

Рассмотрим атрибуты этого тега.

href – определяет находящийся между начальным и конечным тегами текст как гипертекстовую ссылку (URL, или линк) на объект, указанный в значении данного параметра.

Возможные варианты задания значения:

href="http://www.tusur.ru" – создает ссылку на web-сайт или www-документ;

href="ftp://ftp.tomsk.ru/pub/" – создает ссылку на ftp-сайт или расположенные на нем папки и файлы;

href="mailto:abc@mail.ru" – запускает почтовую программу-клиент с заполненным полем имени получателя;

href="page1.html" – указывает на документ, находящийся в том же каталоге, что и тот, с которого происходит вызов. Если же этот документ находится в другом каталоге, то требуется указать полный путь (т.е. те же правила, что и для атрибута src тега <img>).

Например, линк **Документация** будет ссылаться на файл title.html в подкаталоге docs (относительно текущего).

href="#top" или **href="#bottom"** задает возврат на начало или конец web-страницы, что бывает полезно, если web-страница не входит целиком на экран монитора.

href="#имя" создает гиперссылку на часть текущего документа. Здесь имя – это помеченная область текущего документа, например отдельное слово-указатель, уникальное для данного документа.

Для того чтобы пометить эту область, используется тоже тег <a>, но с атрибутом **name**.

Атрибут **name** помечает находящуюся между начальным <a> и конечным тегами область документа как возможный объект для ссылки. В качестве значения нужно написать любое слово-указатель, уникальное для данного документа.

Например,

Глава 1.

---

[1] Поэтому не рекомендуется выделять подчеркиванием обычный текст на web-страницах – он же не станет от этого гиперссылкой, а рука пользователя потянется щелкнуть мышью по нему.

Теперь на область «Глава 1», помеченную именем «Chapter1», можно ссылаться при помощи линка

Глава 1.

Таким способом можно организовать гипертекстовое оглавление большого текстового документа, разбитого на разделы, главы, пункты, подпункты.

Еще один атрибут тега <a> – **target** указывается для того, чтобы определить, в каком окне должен открываться документ, на который ведет гиперссылка. Если параметр target не указан, новый документ будет открыт в том же окне браузера. Если указать параметр target с зарезервированным именем "blank", документ, на который указывает гиперссылка, будет открыт в новом окне, что рекомендуется делать при ссылках на другие сайты. Этот атрибут используется и во фреймах, но об этом чуть позже.

Для всех ссылок на странице можно задать цвета:

link – определяет цвет просто ссылки;

alink – определяет цвет активной ссылки (нажатой);

vlink – определяет цвет уже посещенной ссылки.

Это – атрибуты тега <body>, в нем они и размещаются. Их цветовые параметры – в таблице 1, например:

<body link="#0000FF" alink="#6495ED" vlink=red>

4.2. Картинка-гиперссылка

В качестве гиперссылки может использоваться и рисунок. Принцип ссылки тот же, что и для текстовой гиперссылки, только между открывающим и закрывающим тегами <a> помещают не текст, а тег <img> вместе со всем его содержимым (см. п. 3).

Пример.

<img src= "logo.jpg" width=70 height=50 border=2 alt="Щелкни, чтобы перейти на сайт ФИТ ТГУ">

Этим самым картинке размером 70×50 пикселей, находящейся в файле с названием logo.jpg, назначена гиперссылка на сайт факультета инновационных технологий ТГУ. При наведении на нее стрелка курсора трансформируется в изображение ладони с вытянутым

указательным пальцем и появляется надпись «Щелкни, чтобы перейти на сайт ФИТ ТГУ». Страница будет открываться в новом окне браузера.

А можно ли назначить различным частям одного графического изображения различные гиперссылки? А как разбить картинку (например, фотографию) на отдельные участки так, чтобы при подведении к ним курсора появлялись разные подписи? Для этого существуют специальные программы, например:

GeoHTML (http://www.fegi.ru/geohtml/), лицензия freeware.

MapEdit (http://www.boutell.com/mapedit/). лицензия shareware.

4.3. Картинка-навигационная карта

Прежде чем начинать работу над навигационной картой, поместите ту картинку, которую собираетесь превратить в карту, на html-страницу и посмотрите, удовлетворяют ли вас ее размеры. Подгоните размеры под требуемые, но не изменением параметров height и width, а с помощью графического редактора.

Создадим навигационную карту с помощью редактора **GeoHTML** (Приложение 3).

После запуска программы выбираем **File | New Map…** Окно **New Map** («Новая карта») служит для создания новой карты. Оно содержит две страницы **Open an Image** («Открыть изображение») и **Look In Documen**t («Найти в документе») (рис. 2.5).

Страница **Open An Image** используется для открытия файла с диска (поле **Source**). Изменяя свойства Width и Height, можно задать ширину и высоту изображения. После нажатия кнопки «Ok» изображение появляется в поле программы и готово к редактированию.

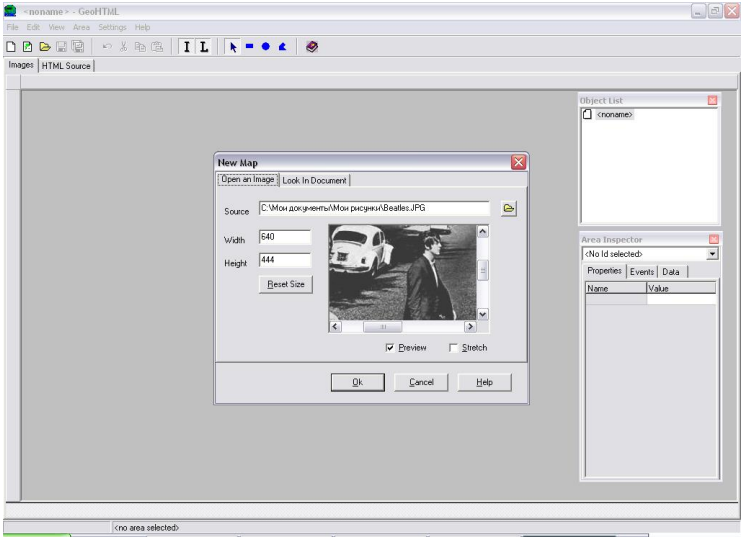


Рис. 2.5. Главное окно программы **GeoHTML** с выбранным графическим файлом.

Щелкая по изображению формы, выбираем «Прямоугольник», «Окружность» или «Многоугольник». Стрелкой осуществляется переход от одного объекта выделения к другому.

При работе с формой «Прямоугольник» ставите курсор в левый верхний угол предполагаемой области выделения и ведете его вправо-вниз.

При работе с формой «Окружность» ставите курсор в центр объекта и начинаете двигать его по радиусу, увеличивая обхват.

При работе с формой «Многоугольник» курсором обводите объект, щелкая мышью на углах поворота. Необходимо обязательно замкнуть ломаную линию! Объекты выделения покрываются синей сеткой (рис. 2.6).

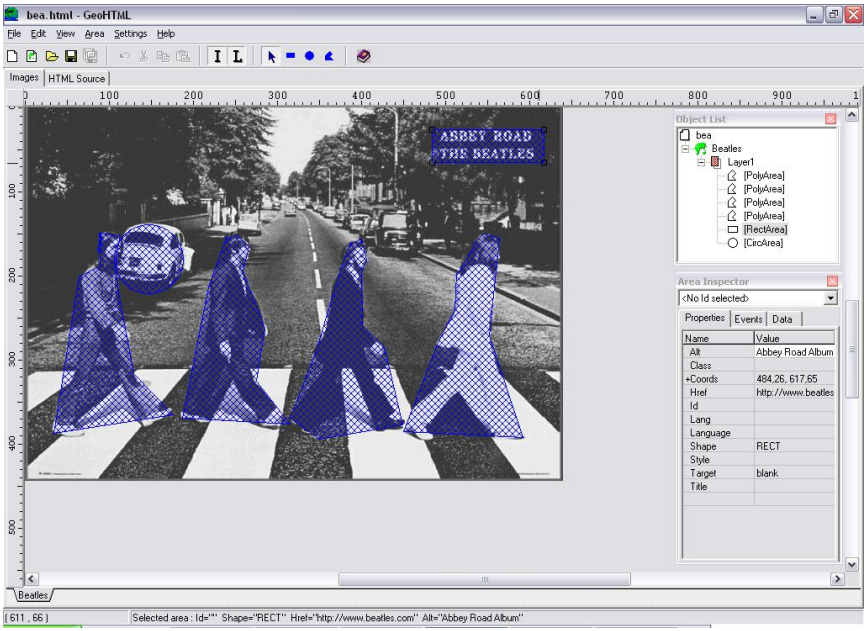


Рис. 2.6. Окно редактора **GeoHTML** с выделенными объектами. В нижнем правом углу – окно **Area Inspector.**

Чтобы отменить выделение, нужно выбрать стрелкой объект, щелкнуть правой кнопкой мыши и в контекстном меню выбрать **Delete Area**.

К выделенной области привязывается определенная информация. Это делается при помощи окна **Area Inspector** (на рис.2.6 справа). Атрибуты разделяются на **Свойства (Properties)** и **События (Events)**.

Свойство **Shape** – тип области. Свойству автоматически присваивается одно из трех значений: RECT (область прямоугольная), CIRCLE (область круглая) или POLY (область многоугольная). **Coords** – координаты области. Содержит целочисленные значения координат, записанные через запятую, и зависит от геометрического типа (свойства Shape) области. Единицей измерения этих координат являются пиксели (точки) экрана. **Href** – гипертекстовая ссылка. **Target** – задает окно, в котором откроется документ, указанный в атрибуте href. **ID** –

идентификатор области. Используется для обращения к свойствам области из кода сценария. **Alt** и **Title** – определяют выпадающий текст-подсказку (в зависимости от браузера – следует проверить!). **Language** – описание языка сценария, используемого в событиях области (например, «javascript»). **Lang** – описание ISO-языка для области (например, «text/javascript»). **Style** – определяет строку таблицы стилей для области. **Class** – определяет класс области, используемый в таблицах стилей (CSS).

Атрибуты из раздела **События** содержат операторы языков сценария (JavaScript или VBScript) и предназначены для продвинутых пользователей. Коды сценария выполняются, когда происходит соответствующее атрибуту событие.

После того, как определены все объекты, необходимо открыть закладку **HTML Source**, скопировать сгенерированный программой html-код, начиная с тега <img> и кончая тегом </map>, и вставить в нужное место на html-странице вашего сайта вместо кода вставки картинки. При необходимости подкорректируйте путь к рисунку.

Таким образом, выбранный рисунок превратился в навигационную карту, причем нужный код был сгенерирован автоматически.

Пример.

Для фотографии, изображенной на рис. 2.6, был сгенерирован следующий код:

```
<IMG SRC="images/Beatles.jpg" USEMAP="#Beatles" width=640 height=444 border="0" alt="Abbey Road Street">
<MAP NAME="Beatles">
  <AREA SHAPE="CIRCLE" HREF="http://www.avto.ru" TARGET="blank" ALT="Автомобиль" COORDS="147,180,41">
  <AREA SHAPE="RECT" HREF="http://www.beatles.com" TARGET="blank" ALT="Abbey Road Album" COORDS="484,26, 617,65">
  <AREA SHAPE="POLY" HREF="#" ALT="Джон Леннон" COORDS="538,157, 514,198, 513,197, 511,240, 489,284, 450,385, 592,392, 552,271, 564,155, 545,150, 532,163, 538,157">
  <AREA SHAPE="POLY" HREF="#" ALT="Ринго Старр" COORDS="387,160, 374,196, 345,292, 315,369, 350,394, 449,377,
```

427,284, 401,259, 400,196, 406,180, 400,156, 386,163, 387,160">

<AREA SHAPE="POLY" HREF="#" ALT="Пол Маккартни" COORDS="240,152, 218,213, 187,369, 315,376, 275,325, 285,262, 259,194, 266,165, 252,151, 238,157, 240,152">

<AREA SHAPE="POLY" HREF="#" ALT="Джордж Харрисон" COORDS="90,148, 89,188, 64,201, 64,201, 34,387, 176,365, 129,316, 110,214, 108,178, 113,152, 88,149, 90,169, 86,171, 90,148">

</MAP>

Видно, что были сформированы шесть областей выделения (автомобиль, надпись в правом верхнем углу и четыре фигуры), причем первым двум областям назначены гипертекстовые ссылки. Первая область сформирована окружностью, вторая – прямоугольником, остальные – многоугольником.

5. Таблицы

Таблицы, пожалуй, самый распространенный элемент web-дизайна.

При создании html-страниц таблицы используются не только для традиционных целей – табличного размещения текстовой и числовой информации, но и для компоновки элементов дизайна страницы – текста, графики. Например, в таблицу с невидимыми границами размещаются необходимые элементы: навигация, текст, картинки, таблицы и т.д. Пользователь может даже не догадаться, что на самом деле это таблица.

Сегодня практически любая Web-страница создана именно таким образом.

Для создания таблицы используется контейнерный тег **<table>**. Внутри тегов <table>…<table> располагается описание всех рядов и ячеек.

Для того чтобы границы таблицы были видимы, в теге <table> задают атрибут **border=*n***, где *n* – ширина границы в пикселях.

Таблицы в HTML формируются **построчно**.

Сначала – первый ряд со всем его содержимым, затем второй и т. д.

Каждый ряд формирует контейнерный тег **<tr>**. Тег <tr> начинает ряд в таблице, тег </tr> завершает ряд.

Внутри описываются все ячейки ряда. Для формирования ячеек первой строки (она обычно является строкой заголовков столбцов) используется контейнерный тег **<th>**. Внутри этих ячеек текст по умолчанию отцентрирован и выделен полужирным шрифтом.

Ячейки всех остальных строк формируются тегами **<td>...</td**>. По умолчанию текст в ячейке смещен влево и отцентрирован по вертикали. Ячейки таблицы обязательно должны содержать какие-либо символы. Если хотите оставить пустой, поместите туда неразрывный пробел .

Для формирования подписи к таблице используется контейнерный тег **<caption>.** Он должен присутствовать внутри тегов <table>...</table>, но снаружи описания какой-либо строки или ячейки. По умолчанию <caption> имеет атрибут **align=top**, но может быть явно установлен в **align=bottom**. Атрибут align определяет, где – сверху или снизу таблицы – будет поставлена подпись. Подпись всегда центрирована в рамках ширины таблицы.

Пример.
Строки html-кода

```
<table border="1">
<caption>Таблица </caption>
<tr>
<th>Заголовок столбца 1</th><th>Заголовок столбца 2</th>
<th>Заголовок столбца 3 </th>
</tr>
<tr>
<td>Содержимое ячейки 4</td><td>Содержимое ячейки 5</td>
<td>Содержимое ячейки 6</td>
</tr>
<tr>
<td>Содержимое ячейки 7</td><td>Содержимое ячейки 8</td>
<td>Содержимое ячейки 9</td>
</tr>
</table>
```

определяют простую таблицу следующего вида

Таблица		
Заголовок столбца 1	**Заголовок столбца 2**	**Заголовок столбца 3**

Содержимое ячейки 4	Содержимое ячейки 5	Содержимое ячейки 6
Содержимое ячейки 7	Содержимое ячейки 8	Содержимое ячейки 8

Рассмотрим основные атрибуты табличных тегов.

Атрибут **align**. Если он используется внутри тегов <tr>, <th> или <td>, то управляет положением данных в ячейках по горизонтали, если используется внутри тега <table> – управляет размещением таблицы на странице (аналогично изображению <img>).

Может принимать значения **left** (слева), **right** (справа) или **center** (по центру).

Атрибут **valign**. Он используется внутри тегов <tr>, <th> и <td> и определяет вертикальное размещение данных в ячейках. Может принимать значения **top** (вверху), **bottom** (внизу), **middle** (по середине).

Атрибут **nowrap** говорит о том, что данные в ячейке не могут логически разбиваться на несколько строк и должны быть представлены одной строкой.

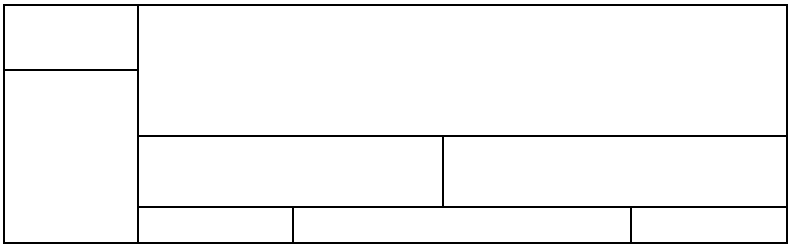
Атрибут **colspan** указывает, какое количество ячеек будет объединено **по горизонтали** для указанной ячейки. По умолчанию равен 1.

Атрибут **rowspan** указывает, какое количество ячеек будет объединено **по вертикали** для указанной ячейки. По умолчанию равен 1.

Умелое управление атрибутами colspan и rowspan позволяет создавать сколь угодно сложные таблицы.

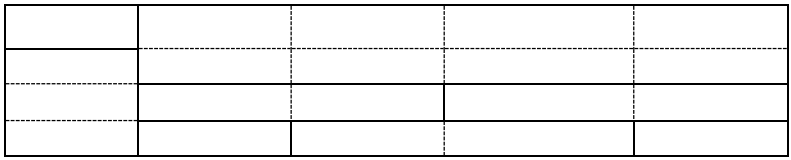
Рассмотрим на примере алгоритм их создания.

Запишем html-код, описывающий таблицу вида:



Продлим пунктиром все линии этой таблицы до ее краёв.

Получим таблицу из четырех строк и шести столбцов.



Теперь начинаем писать код. Сначала откроем таблицу и укажем толщину рамки:

<table border=1>

Каждая горизонтальная черта разделяет строки таблицы. Ячейка, обведенная сплошными линиями, – это одна пара <td>…</td>. Количество столбцов в такой ячейке записываем в colspan, а количество строк – в rowspan.

Если верхний левый угол ячейки находится не в рассматриваемой строке, то при записи кода эту ячейку игнорируем.

Открываем первую строку:

<tr>

Первая ячейка не делится ни на столбцы, ни на строки. Значит, так и запишем:

<td>Ячейка 1</td>.

Вторая ячейка делится на 4 столбца и 2 строки, поэтому записываем:

<td colspan=4 rowspan=2>Ячейка 2</td>.

</tr>

Строка закрывается, т.к. в ней ячейки закончились.

Переходим ко второй строке:

<tr>

Первая её ячейка делится на три строки.

<td rowspan=3>Ячейка 3</td>

Вторую ячейку мы уже записали, так что ее игнорируем (ее верхний левый угол находится в предыдущей строке).

Закрываем строку:

</tr>.

Переходим к третьей строке:

<tr>.

Первую ячейку игнорируем, вторая и третья делятся на две колонки. Пишем:

```
<td colspan=2>Ячейка 4</td>
<td colspan=2>Ячейка 5</td>
</tr>
```

Переходим к четвертой строке:

```
<tr>.
```

Первую ячейку игнорируем, вторая не делится, третья делится на две колонки, четвертая не делится. Пишем:

```
<td>Ячейка 6</td>
<td colspan=2>Ячейка 7</td>
<td>Ячейка 8</td>
</tr>
</table>
```

Запишем весь код полностью, добавив центрирование таблицы и центрирование текста в ячейках:

```
<table align=center border=1>
 <tr align=center >
  <td>Ячейка 1</td>
  <td colspan=4 rowspan=2>Ячейка 2</td>
 </tr>
 <tr align=center >
  <td rowspan=3>Ячейка 3</td>
 </tr>
 <tr align=center >
  <td colspan=2>Ячейка 4</td>
  <td colspan=2>Ячейка 5</td>
 </tr>
 <tr align=center >
  <td>Ячейка 6</td>
  <td colspan=2>Ячейка 7</td>
  <td>Ячейка 8</td>
 </tr>
</table>
```

Внешний вид таблицы кроме атрибута border определяется еще несколькими атрибутами.

Атрибут тега <table> **cellspacing** задает расстояние *n* между

ячейками (в пикселях) в таблице:

<table cellspacing=*n*>.

Атрибут тега <table> **cellpadding** задает расстояние *n* между границей (рамкой) ячейки и ее содержимым (в пикселях). Этот атрибут определяет данное расстояние для всех ячеек в таблице:

<table cellpadding= *n* >.

В теге <table> атрибут **width** задает ширину таблицы *n,* **height** – ее высоту *m*:

<table width=*n* height=*m*>.

Если размер таблицы не задан, то он будет минимально возможным для того, чтобы уместить ее содержимое. Если текста много, то когда ширина таблицы достигнет границ программы просмотра, браузер перенесет текст на следующую строку.

В теге <td> атрибут **width** задает ширину ячейки *n,* **height** – высоту *m*:

<td width=*n* height= *m* >

Уже известные нам атрибуты **bgcolor** и **background** определяют цвет фона или фоновый рисунок для всех ячеек таблицы:

<table background="путь к картинке">;

для строки таблицы:

<tr bgcolor=цвет>;

для ячейки таблицы:

<td bgcolor=цвет>,

причем последующий цвет перекрывает предыдущие.

Цвета могут быть заданы шестнадцатеричными RGB-значениями или стандартными названиями (приложение 2).

6. Фреймы

6.1. Страница-контейнер

Английское слово frame («фрейм») означает рамку, остов, оправу, кадр, словом, структурную единицу различных объектов.

В информатике фрейм – элемент языка HTML версии 3.0 и выше. Фреймы позволяют разделить web-страницу на несколько независимых окон и в каждом из них размещать отдельную web-страницу. При этом допускаются ссылки из одного окна в другое окно. Обычно фреймы применяются для организации меню, постоянно находящихся на экране, тем самым связывая отдельные

страницы в web-сайт. Чем мы сейчас и займемся.

Любой сайт, содержащий страницы с фреймами, начинается с написания специальной **странички-контейнера**, которая сама не показывается, но содержит в себе указания для организации фреймовой структуры (на сколько фреймов делить страницу и как) и ссылок на участвующие файлы.

Принципиальная особенность этой странички-контейнера – она не содержит тегов <body>…</body>. Их заменяют теги **<frameset>**… **</frameset>**.

Рассмотрим пример.

Наберем в текстовом редакторе Notepad++ html-код и сохраним его, как это описано в п.1, с именем **index.html**:

```
<html>
    <head>
    <title>Frame container</title>
    </head>
<frameset rows="10%,*">
    <frame src="up.html" name="logo" scrolling=no noresize>
<frameset cols="15%,*">
    <frame src="left.html" name="menu" scrolling=no noresize>
    <frame src="right.html" name="content" noresize>
</frameset>
</frameset>
</html>
```

Рассмотрим действие новых тегов и их атрибутов.

Контейнерный тег **<frameset>**… **</frameset>** описывает количество и размеры фреймов.

Атрибут **rows** задает горизонтальное разбиение страницы. Его параметры – это цифры и знаки, разделенные запятыми. Каждое значение определяет ширину области и может задаваться в процентах, пикселях и при помощи знака звездочки, обозначающей «все оставшееся пространство». Аналогично, параметр **cols** задает вертикальное разбиение страницы.

Таким образом, в примере вся область сначала разделена тегом <frameset rows="10%,*"> на два фрейма по горизонтали (верхняя часть составляет 10% высоты области), а затем нижний фрейм

разделен, в свою очередь, тегом <frameset cols="15%,*"> по вертикали также на две фрейма, причем левая часть составляет 15% ширины области. Таким образом, создана трехфреймовая структура страницы.

Кроме указанных атрибутов в теге <frameset> используются атрибуты **frameborder=yes (no)** – указывает браузеру, отображать ли рамку у фреймов или нет; **border=*n*** – определяет ширину *n* рамки между фреймами в пикселях; **bordercolor="#RRGGBB"** – задает цвет рамки, которая разделяет отдельные фреймы.

Контейнерный тег **<frame>…</frame>** определяет источник информации для соответствующего фрейма.

Атрибут **src** задает URL страницы, которая будет помещена во фрейм. Обычно эти страницы и страница-контейнер находятся все вместе в одной папке, поэтому адрес состоит только из имени файла и его расширения.

В примере верхнему фрейму назначен файл **up.html**, нижнему левому фрейму – **left.html**, нижнему правому фрейму – **right.html**. Именно их содержимое появится на экране при запуске файла index.html.

Атрибут **name** позволяет задать каждому фрейму уникальное имя, по которому к нему можно будет обращаться. Ниже это будет использовано при создании гипертекстового меню.

Параметры атрибута **scrolling** позволяют управлять появлением полосы прокрутки: **no** – полосы прокрутки не будет ни при каких обстоятельствах; **yes** – полоса прокрутки будет всегда; **auto** – полоса прокрутки появится только тогда, когда она нужна (значение по умолчанию).

Атрибут **noresize** позволяет создавать фреймы без возможности изменения размеров. По умолчанию размер фрейма можно изменить при помощи мыши так же просто, как и размер окна Windows. Данную возможность noresize отменяет. Если у одного фрейма установлен атрибут noresize, то у соседних фреймов тоже не может быть изменен размер со стороны данного.

Итак, теперь если вы откроете файл index.html, то перед вами появится трехфреймовая страница примерно такого вида (рис. 2.5).

Такой вид можно наблюдать, если у вас нет файлов с именами up.html, left.html, right.html или вы ошиблись в их именах. Введя в код страницы-контейнера имена html-файлов, созданных вами в

процессе работы над заданием, вы получите их на экране монитора в соответствующем фрейме.

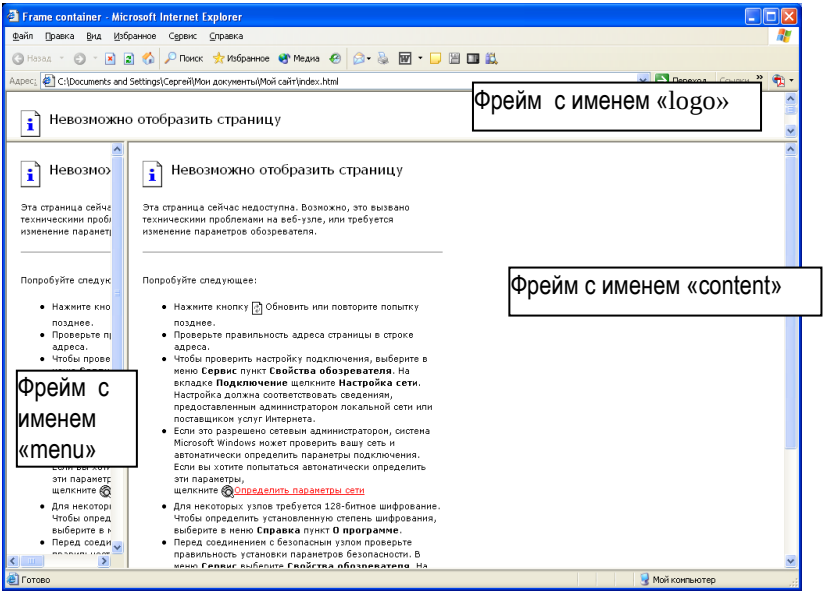


Рис. 2.5. Структура страницы с тремя фреймами

Внимание! Чтобы открыть страницу-контейнер на изменение html-кода, выберите в главном меню браузера опцию **Вид → Просмотр HTML кода**. Обновление – как обычно.

Но для реализации поставленной цели – организовать оглавление сайта, постоянно находящееся на экране, – еще необходимо создать html-страницу с гипертекстовым меню.

6.2. Страница меню

Пусть созданные вами html-страницы имеют имена 1.html, 2.html, 3.html и title.html.

Создадим простейший файл menu.html со страничкой, содержащей меню[1].

[1] Затем, естественно, вы ее «облагородите»…

```
<html>
<head>
<title>Menu Page</title>
</head>
<body>
<a href="1.html" target="content">Страница 1</a><br>
<a href="2.html" target="content">Страница 2</a><br>
<a href="3.html" target="content">Страница 3</a><br>
</body>
</html>
```

Действие тегов, входящих в код, вы уже знаете.

Первый тег <a> атрибутом **href="1.html"** определяет гиперссылку на html-страницу с именем 1.html. Атрибут **target="content"** определяет, в каком фрейме открывать эту страницу. Ранее, на странице-контейнере index.html это имя было назначено нижнему правому фрейму. Значит, там и откроется html-страница с именем 1.html.

Отметим, что атрибут target может принимать и другие значения:

target="_blank" определяет, что документ, полученный по ссылке, будет отображаться в новом окне браузера;

target="_self" определяет, что документ, полученный по ссылке, будет отображаться в том же фрейме, в котором находится ссылка;

target="_top" определяет, что документ, полученный по ссылке, будет отображаться на всей поверхности окна вне зависимости от наличия фреймов. Использование данного параметра удобно в случае вложенных фреймов.

Второй тег <a> атрибутом href="2.html" определяет гиперссылку на html-страницу с именем 2.html, которая тоже откроется в нижнем правом фрейме. И так далее.

Кстати, не забудьте сразу подкорректировать код странички-контейнера! В соответствии с назначенными выше именами файлов ее основное содержимое будет выглядеть так:

```
<frameset rows="10%,*">
    <frame src=title.html" name="logo" scrolling=no noresize>
<frameset cols="15%,*">
    <frame src="menu.html" name="menu" scrolling=no noresize>
    <frame src="1.html" name="content" noresize>
```

Таким образом, при запуске файла index.html в верхнем фрейме окажется содержимое файла title.html, в левом фрейме появятся подчеркнутые строчки-гиперссылки

Страница 1

Страница 2

Страница 3,

а в правом фрейме появится содержимое файла 1.html.

Файл index.html является стартовым файлом созданного web-сайта – так и принято по умолчанию в Интернет.

Использование фреймов заметно упрощает навигацию по сайту. На самом деле, очень удобно – на одной части страницы размещаешь меню, на другой – выкладываешь содержимое.

Но в последнее время использование фреймов становится все менее популярным решением. В качестве довода приводят неправильную обработку страниц, содержащих фреймы, некоторыми поисковыми машинами. Когда пользователь находит такой документ через поисковый сервер, то он попадает лишь на один из фреймов, но не видит истинного отображения страницы, что вызывает его полную дезориентацию. Да и сами попробуйте начать просмотр вашего сайта не со страницы с фреймами, а, например, со страницы 1.html. Ну, и где она, красивая фреймовая структура?

Эту проблему, конечно, можно обойти, поместив на каждой странице ссылку на страницу с фреймами. Но можно поступить по-другому: использовать **таблицы** для произвольного расположения элементов на странице и организации навигации по сайту.

Ведь как можно интерпретировать созданную выше трехфреймовую структуру в табличных терминах? Как таблицу из двух строк и двух столбцов, причем две верхних ячейки объединены. Создадим на каждой странице такую структуру в виде таблицы, и фреймы будут не нужны. Правда, будет много повторов (меню, заголовочная часть)…

В качестве информации к размышлению приведем слова Артемия Лебедева, авторитетного в РУНЕТе web-дизайнера (www.artlebedev.ru): «Многие начинающие веб-дизайнеры страшно боятся использовать фреймы, потому что они слышали, что это не хорошо. И не задумываются над вопросом: «А почему, собственно?». Решение простое – если вам надо, используйте. Но

помните одну вещь – если сайт большой, то для организации всей информации фреймы лучше не использовать. Они пригодятся для решения небольших конкретных задач интерфейсного характера».

7. Таблицы стилей

Многие рассмотренные выше теги имели атрибуты, с помощью которых можно было менять стили отображаемых объектов.

Стиль документа – это правило, которое определяет внешний вид того или иного элемента в документе: вид шрифта, цвет и размер объектов (символов, границ), формат текста и т.п.

Когда в составе сайта немного страниц, то эти html-возможности вполне устраивают разработчика. А если страниц много или очень много? Представьте себе, что, задав на всех страницах один и тот же фон или цвет ячеек таблиц, вы впоследствии решите их изменить. Очень долгая и нудная работа.

С появлением каскадных таблиц стилей (CSS – Cascading Style Sheets) этот процесс стал намного проще.

Идея CSS очень проста. В html-документе прямо ставится указание на то, как должен выглядеть тот или иной элемент, а при использовании CSS такие указания выносятся в отдельный блок, который может либо включаться в документ, либо читаться из внешнего файла.

Такой простой подход сразу дает массу преимуществ. Прежде всего, значительно облегчается изменение внешнего вида сайта или отдельных его элементов – достаточно изменить определение соответствующего стиля в единственном CSS-файле, и эти изменения распространятся на весь сайт. Второе преимущество – сокращение размеров документов. Ну, и третье – таблицы стилей позволяют отделить логическую структуру страниц от визуального представления (оформления). Ведь язык HTML изначально был разработан для разметки структуры документа.

Более того, в новой спецификации HTML 4.0 было рекомендовано использовать для оформления именно каскадные таблицы стилей. При этом многие теги и атрибуты (собственно говоря, почти все, относящиеся к оформлению) были отменены. Это, конечно, не означает, что всем нужно немедленно прекратить их использование, они будут еще долго поддерживаться, но к

рекомендациям WWW-консорциума следует прислушаться (http://www.w3.org/TR/REC-CSS1/).

Существует **три способа добавления CSS в html-документ**:

1) подключение внешней стилевой таблицы;

2) использование внутренней стилевой таблицы, действующей только в данном документе;

3) указание стиля непосредственно внутри тега.

7.1. Внешняя таблица стилей

Внешняя стилевая таблица создается с помощью текстового редактора, например того же Notepad++. Только нужно сохранить этот файл с расширением .css (например, style.css).

В этом файле объявляются правила стилей для разных объектов. Они записываются с самой первой строки блокнота, ничем особым не выделяясь.

Например,

H1 {color:blue}

Это правило говорит о том, что все заголовки первого уровня (выделяемые тегами <h1>… </h1>) будут иметь синий цвет[1].

Каждое правило состоит из двух частей:

– **селектор** (здесь: **H1**) определяет тег в html-документе (заголовок, параграф, таблица и ее элементы, ссылки, и т. д.), для которого назначается стиль;

– **определение** (здесь: **color:blue**) устанавливает стиль, применяемый к данному селектору (тегу).

Определение, в свою очередь, тоже состоит из двух частей: свойства (color) и значения (blue). Они записываются в фигурных скобках и отделяются друг от друга двоеточием. Определения перечисляются через точку с запятой. Селекторы тоже могут перечисляться, отделяясь запятой.

Например:

H1, H2 {color:red; font-family:Arial}

В данном примере задается одинаковое оформление для заголовков первого и второго уровней. Они будут иметь красный цвет и шрифт Arial.

[1] В CSS принято названия тегов писать прописными буквами.

Некоторые свойства и их значения приведены в таблицах 3, 4, 5.

В таблицах стилей также могут применяться **контекстные селекторы**. Они позволяют задать стилевое оформление для тега, который находится внутри другого тега.

Пусть в CSS заданы стили

H4 {color:grey}

H4 I {color:green} .

Два подряд (через пробел) записанных селектора H4 и I и есть контекстный селектор.

Тогда в строке

<h4> Наследование родительских свойств </h4>

текст, заключенный между тегами …, будет отображаться зеленым цветом, в то время как остальной текст между тегами <h4>…</h4> будет серого цвета. И это будет выполняться для любых участков текста, где встретится такая комбинация.

Таблица 3. Параметры шрифта

Свойство	**Описание**	**Значения**	**Примеры и примечания**
font-family	Задает шрифт(ы) или семейство шрифтов	Названия шрифтов в порядке приоритетности. Перечисляются через запятую.	BODY { font-family: Verdana, Arial, sans-serif} В данном примере страница будет оформляться шрифтом Verdana. Если этого шрифта на машине пользователя нет, то браузер будет искать Arial, а если нет и его, то используется любой шрифт из семейства sans-serif
font-style	Задает способ начертания шрифта	normal, italic, oblique	H1 { font-family: Helvetica; font-style: italic; font-weight: bold}

font-weight	Задает толщину шрифта	bold (400) normal (700) Числовые значения от 100 до 900, с шагом, кратным 100. Можно употреблять относительные значения bolder и lighter	
font-size	Задает размер шрифта	Точный размер: px – пиксели, pt – пункты (1pt = 1/72 in), in – дюймы, mm – миллиметры. Относительное изменение: larger, smaller	H1{ font-weight: bold; font-size: 30pt }

Таблица 4. Параметры текста

Свойство	**Описание**	**Значения**	**Примеры и примечания**
text-decoration	Задает параметры начертания: подчеркивание, надчеркивание, зачеркивание	none, underline, overline, line-through	A:visited { text-decoration: none }
text-align	Задает горизонтальное выравнивание текста	left, right, center, justify	P {text-align: justify}
text-indent	Задает абзацный отступ	px – пиксели, pt – пункты, mm – миллиметры, em – символ максимальной ширины	P {text-align: justify; text-indent: 5em}
vertical-align	Задает отображение текста по вертикали; верхние (2^{10}) и	top, middle, bottom, super, sub,	TD {vertical-align: top} <P>H<SPAN STYLE="vertical-align : sub;">2</SPAN>O</P> В браузере – H_2O

	нижние (H_2O) индексы		

Таблица 5. Параметры цвета и фона

Свойство	Описание	Значения	Примеры
color	Задает цвет текста	Название цвета – red; шестнадцатеричное значение составляющих цвета – #FF0000	H1 {color:red} A:link {color: #8B4513}
background-color	Задает цвет фона	TD { background-color: #F5F5DC }	
background-image	Задает фоновое изображение	BODY { background-image: URL (picture /back.gif)}	
background-repeat	Задает способ повторения фонового изображения	repeat – повторять по вертикали и горизонтали. no-repeat – не повторять. repeat-x – повторять по горизонтали. repeat-y – повторять по вертикали	BODY { background-image: URL (picture/back.gif); background-repeat: repeat-x }

Итак, наберем в блокноте строки:

BODY {font-family: Verdana}

H1, H2 {color:red; font-family:Arial}

P {text-align: justify; text-indent: 3em}

TD {background-color: yellow}

и сохраним под именем style.css. Таблица стилей создана. Как теперь ее применить?

7.2. Применение внешней таблицы стилей

Чтобы применить созданную таблицу стилей, вставляем в заголовочную часть **каждого html-документа,** в котором хотим

применить стили (т.е. внутрь тегов <head>…</head>), одиночный тег **<link>** с соответствующими атрибутами и значениями:

<link href="style.css " rel="stylesheet" type="text/css">

Атрибут **href="style.css"** указывает путь к файлу со стилями.

Атрибут **rel="stylesheet"** указывает, является ли подгружаемые таблицы стилей постоянными, предпочитаемыми или альтернативными. В данном случае установлены постоянные таблицы стилей для документа (подробнее об атрибуте rel и его значениях – в спецификации HTML 4.0 (http://www.citforum.ru/internet/html40/)).

Атрибут **type="text/css"** – тип подгружаемой информации, в данном случае указывается, что это CSS.

Теперь браузер отобразит текст шрифтом Verdana, заголовки первого и второго уровня – шрифтом Arial красного цвета, текст параграфов (тот, который охвачен тегами …) будет выровнен по ширине страницы, а абзацный отступ будет длиной в 3 символа. Фон ячеек таблиц будет желтым.

Но описанной выше стилевой таблице не хватает гибкости. Как быть, например, если хочется один параграф выполнить в одном стиле, а другой – в другом? Для большей гибкости при оформлении страниц можно создать **свои собственные классы**, дать им имена и задать для них стили.

Классы задаются двумя способами.

Способ 1. Привязка класса к определенному тегу.

Например, в CSS записывается:

P.oba {text-align: justify; font-style: italic}

P.sprava {text-align: right; color:red }

Это означает, что задано два класса стилей, которые можно применять только к параграфам.

Здесь oba и sprava — названия классов. Названия можно давать любые, но лучше осмысленные. Обратите внимание на синтаксис (между названием тега и названием класса ставится точка).

Чтобы теперь использовать эти классы, нужно добавить атрибут **class** в начало параграфа с соответствующим значением, например

<p class="oba">Параграф оформлен с помощью класса oba .

Или

<p class="sprava">Параграф оформлен с помощью класса sprava .

Способ 2. Создание произвольного класса, который можно присваивать любому тегу на странице.

В CSS помещают стиль без упоминания тега:

.cvet1 {color: #EE82EE}

На html-странице помещают атрибут class:

<h6 class="cvet1">Цвет заголовка задан с помощью класса .cvet1 </h6>

или

<body class="cvet1">Цвет текста всей страницы задан с помощью класса .cvet1 </body>

При работе с классами сначала рекомендуется определить стиль по умолчанию для всего документа. Этот стиль по умолчанию задается для тега <body>. А уже потом по мере необходимости добавлять стили для разных тегов.

Из примеров можно понять, что атрибут class нужно «прикрепить» к какому-то тегу. Часто он используется с тегами параграфа …. . Но есть еще один подобный контейнерный тег – ** **:

<div **class="oba"**>Абзац

Это тег-разделитель, он определяет фрагмент текста, который можно наделить какими-либо стилевыми свойствами, расположив его между тегами и и задав стиль. Новый тег , так же как и тег , начинает новый абзац, но в **отличие от тега после тега пропуск строки не производится**, поэтому текст выглядит «как обычно».

Но возникает вопрос, а как применить стиль к отдельному слову (группе слов) в тексте документа? Тег … не используешь, поскольку в этом случае следующий после выделения текст начнется на новой строке.

В таких ситуациях вместо тега следует использовать контейнерный тег **** :

выделено три слова

Этот тег, кстати, бессмысленно применять без атрибутов, поскольку тогда вообще не будет никакого эффекта. Как правило,

он применяется с атрибутами class, style (о последнем речь пойдет ниже).

Кстати, кроме стилевых свойств элементов HTML и классов, можно определять свойства так называемых **идентификаторов**. Дело в том, что у каждого элемента HTML может быть атрибут **id** – его уникальное имя в данном документе (этот атрибут играет большую роль при создании динамических страниц). Для того чтобы определить в таблице стилей свойства элемента, имеющего определенный идентификатор id, следует записать, например, так:

#cool {color: white; background-color: black}

Тогда элемент по имени **cool** (например) будет наделен указанными свойствами (в данном случае белым цветом символов на черном фоне). Только не забывайте, что в каждом HTML-документе каждый идентификатор должен быть уникальным, то есть не может быть двух и более элементов с одинаковым значением атрибута id.

7.3. Внутренняя таблица стилей

Внутренняя таблица стилей внедряется непосредственно в html-документ при помощи тега **<style>**, в котором размещается код таблиц стилей (**<style>**код таблицы стилей**</style>**), и действует только внутри одного документа.

Обратите внимание, тег <style>, как и тег <link>, размещается внутри тега <head>…</head>, а не <body>!

Пример.

```
<head>
…
<style type="text/css">
BODY {background-color: #BABAAO; color: blue}
P.oba {text-align: justify; font-style: italic}
P.sprava {text-align: right; color:red}
H1, H2 {color:red; font-family:Arial}
</style>
…
</head>
```

Атрибут type="text/css" здесь имеет тот же смысл, что и в теге <link>, а правила записи стилей – те же, что были рассмотрены выше для внешних таблиц.

Еще один пример.

Как оформить в таблицах стилей разные цветовые **настройки для гиперссылок**? Гиперссылка может быть в одном из четырех состояний: непосещенная; посещенная; активная (в момент нажатия на нее); реагирующая (при наведении на нее курсора мыши).

В стилевых таблицах ссылки оформляются с помощью псевдоклассов селектора A:

```
<style type="text/css">
A:LINK {color: #8B4513} /* непосещенная ссылка*/ ;
A:VISITED {color: #2E8B57} /* посещенная ссылка*/ ;
A:ACTIVE {color: #F4A460} /* активная ссылка*/ ;
A:HOVER {color: Olive; text-decoration: none} /* в момент наведения*/ .
</style>
```

Аналогично записывается и для внешней стилевой таблицы.

Порядок следования стилей должен быть именно таким, иначе работать не будет. Пробелы между селектором и названием псевдокласса не допускаются.

Заодно здесь показано, как в таблицах стилей оформляются комментарии: они охватываются символами «слэш-звездочка» и «звездочка-слэш».

Обратите внимание на свойство text-decoration: none – при наведении мыши текст будет неподчеркнутым (по умолчанию для ссылок указано свойство text-decoration: underline).

Если нужно в одном документе задать разное оформление для разных ссылок, то используются классы, которые комбинируются с псевдоклассами, например:

```
A.menu:LINK {свойство:значение; ...}
A.menu:VISITED {свойство:значение; ...}
```

В этом примере создан класс menu.

Это имя затем указывается в атрибуте class тега <a> на html-странице:

текст ссылки

7.4. Указание стиля внутри тега

Третий способ внедрения CSS в html-документ заключается в указании стиля непосредственно внутри тега при помощи атрибута **style** (не путать с тегом!), например:

<h1 style="color:red; font-family:Arial">…</h1> .

Сразу отметим, что существует иерархия стилей: стиль, заданный во внутренней таблице, отменяет стиль, заданный во внешней таблице, а стиль, заданный внутри тега, отменяет табличные стили. Поэтому они и получили название «каскадные таблицы стилей».

Этот способ используется мало, так как теряет все достоинства таблиц стилей. Ведь применяется он только к одному конкретному тегу.

ЗАДАНИЕ

Используя описанные выше теги HTML, создать при помощи текстового редактора Notepad++ тематический Web-сайт по вопросам дисциплины «Мировые информационные ресурсы».

Перечень тем

1) Российские государственные библиотечные информационные ресурсы: структура, формирование, распространение.

2) Российские государственные научно-технические информационные ресурсы: структура, формирование, распространение.

3) Российские государственные архивные информационные ресурсы: структура, формирование, распространение.

4) Российские государственные правовые информационные ресурсы: структура, формирование, распространение.

5) Российские государственные статистические информационные ресурсы: структура, формирование, распространение.

6) Геоинформационные системы: структура, ведущие производители, сферы применения.

7) Электронные информационные ресурсы органов власти Томской области.

8) Государственная регистрация информационных ресурсов: принципы, правила, организация.

9) Биржевая и финансовая информация: обработка, распространение, ведущие российские и зарубежные поставщики.

10) Медиаимперии: характеристика ведущих мировых поставщиков новостной и развлекательной информации.

11) Поисковые системы Интернета: история, услуги, принципы работы.

12) Информационная безопасность: методы и способы защиты информационных ресурсов.

13) Электронные библиотеки в Интернете: классификация, ресурсы, услуги.

14) Wiki-технология: история, принципы, перспективы развития.

15) Семантический Web (WWW второго поколения): концепция, составляющие, перспективы развития.

16) Международное сотрудничество в информационной сфере: правовые и экономические аспекты.

17) Классификация персональных данных и информационных систем с соответствии с ФЗ «О персональных данных»

18) Базовые государственные информационные ресурсы: правила формирования и использования

19) Информационные услуги: классификация, ведущие поставщики.

20) Государственная информационная политика: российская и зарубежная практика.

21) Управление Интернет-сообществом: принципы, структура, проблемы.

22) Проблемы сохранения цифровых информационных ресурсов.

23) Мировой рынок информации: сегменты, характеристика, тенденции развития.

24) Федеральная информационная адресная система РФ: структура, оператор, порядок работы

25) ФГИС «Единая система нормативно-справочной информации»: структура, оператор, порядок работы

625) Data Mining: методы и средства поиска, обнаружения, извлечения и анализа текстовых, аудио- и других данных.

27) Интернет вещей: история: структура, перспективы развития.

28) Индустриальный интернет: структура, перспективы развития.

29) Облачные технологии: структура, перспективы развития.

30) Туманные технологии: структура, перспективы развития.

31) Технологии блокчейна: структура, перспективы развития.

32) NBIC-конвергенция: инновационный тренд XXI века.

Требования к содержанию сайта:

1) на главной странице поместить название работы, информацию об авторе (ФИО, город, учебная группа, вуз), информацию о руководителе работы (ФИО, должность), название дисциплины, по которой выполнена работа;

2) информацию по теме найти в Интернете с помощью поисковых систем Google, Yandex и др. и поместить на разных страницах сайта в соответствии со структурой материала;

3) информационное наполнение сайта (контент) должно достаточно полно раскрывать тему, в то же время не следует перегружать сайт детальными подробностями – лучше сделайте соответствующую ссылку;

4) на отдельной странице привести гиперссылки на электронные ресурсы, которые были использованы при выполнении работы (сайт с указанием конкретного ресурса).

Требования к оформлению сайта:

1) организовать систему навигации по страницам сайта (гипертекстовое меню переходов на тематические разделы (страницы) сайта, гипертекстовые переходы со страницы на страницу («линейная навигация»), гипертекстовое оглавление отдельных страниц с большим по объему текстом, гипертекстовые переходы на начало страницы);

2) использовать фреймы для создания четырехоконной структуры страницы, например:

Здесь «шапка» страницы	
Здесь меню	Здесь основная часть
Футер сайта	

Меню выполнить в виде гипертекстового перечня названий страниц сайта;

3) использовать как однородную цветную заливку страницы, так и фоновую картинку;

4) при оформлении текста на страницах:

– использовать разный шрифт, цвет, размер, нижний и верхний индексы, выделение курсивом, полужирным, выравнивание слева, справа, по центру;

– большой по объему текст разбить на пункты, подпункты, абзацы;

– вставить списки (нумерованные и ненумерованные);

– использовать спецсимволы, например   © " § « &raquo. Обозначить свои авторские права на созданный web-сайт: © Иванов И.И., 2017;

– использовать таблицы для форматирования текста;

– использовать таблицу каскадных стилей CSS;

5) вставить графические объекты (рисунки, фотографии, графики) и оформить интерактивную подпись к ним. Использовать рисунок как гиперссылку. Использовать различные режимы обтекания картинки текстом. Все используемые рисунки поместить в отдельную папку;

6) с помощью прилагаемой программы GeoHTML сформировать на одном из рисунков навигационную карту.

Приложение 1. Таблица основных тегов языка HTML

Основные теги документа	
<html></html>	Указывает программе просмотра страниц, что это HTML документ
<head></head>	Определяет место, где помещается различная информация, не отображаемая в теле документа. Здесь располагается тег названия документа <title>, теги стилевых таблиц <link> и <style>и теги для поисковых машин <meta>
<body></body>	Определяет видимую часть – «тело» документа
<title></title>	Помещает название документа в оглавление программы просмотра страниц
<body bgcolor ="?">	Устанавливает цвет фона документа. ? – значение цвета в виде RRGGBB или текстовом виде. Например: FF0000 – красный (Red) цвет
<body text="?">	Устанавливает цвет текста документа. ? – значение цвета в виде RRGGBB или текстовом виде. Например: 000000 - черный цвет (Black)
<body link="?">	Устанавливает цвет гиперссылок. ? – значение цвета в виде RRGGBB или текстовом виде. Например: 00FF00 - зеленый цвет (Green)
<body vlink="?">	Устанавливает цвет гиперссылок, на которых вы уже побывали. ? – значение цвета в виде RRGGBB или текстовом виде. Например: 333333 (Gray) - серый цвет
<body alink="?">	Устанавливает цвет гиперссылок при нажатии
Теги форматирования текста	
<pre></pre>	Обрамляет предварительно отформатированный текст
	Создает САМЫЙ БОЛЬШОЙ заголовок
	Создает самый маленький заголовок
	Создает жирный текст
	Создает наклонный текст
<tt></tt>	Создает текст, имитирующий стиль печатной машинки
<cite></cite>	Используется для цитат, обычно наклонный текст
	Используется для выделения из текста слова (наклонный или жирный текст)

	Используется для выделения наиболее важных частей текста (наклонный или жирный текст)
	Устанавливает размер текста в пределах от 1 до 7
	Устанавливает цвет текста, используя значение цвета в виде RRGGBB
<marquee height="10" width="600" loop="2" bgcolor="#99CCFF">Текст </marquee>	Режим бегущей строки. Текст становится бегущей строкой. height, width – атрибуты текста, loop задает количество пробегов текста, bgcolor задает цвет фона
<p>	Создает новый параграф
<p align=?>	Выравнивает параграф относительно одной из сторон документа, значения: left, right, или center
 	Вставляет перевод строки
<blockquote> </blockquote>	Создает отступы с обеих сторон текста.
<dl></dl>	Создает список определений
<dt>	Определяет каждый из терминов списка
<dd>	Описывает каждое определение
	Создает нумерованный список
	Определяет каждый элемент списка и присваивает номер
	Создает ненумерованный список
	Предваряет каждый элемент списка и добавляет кружок или квадратик
	Тег, используемый для форматирования больших блоков текста HTML документа, также используется в таблицах стилей
Графические элементы	
<img src="name">	Добавляет изображение в HTML-документ
<img src="name" align=?>	Выравнивает изображение к одной из сторон документа, ? принимает значения: left, right, center; bottom, top, middle
<img src="name" border=?>	Устанавливает толщину рамки вокруг изображения
<hr>	Добавляет в HTML-документ горизонтальную линию

<hr size=?>	Устанавливает высоту (толщину) линии
<hr width=?>	Устанавливает ширину линии, можно указать ширину в пикселах или процентах
<hr noshade>	Создает линию без тени.
<hr color=?>	Задает линии определенный цвет. Значение RRGGBB
Таблицы	
<table></table>	Создает таблицу
<tr></tr>	Определяет строку в таблице
<td></td>	Определяет отдельную ячейку в таблице
<th></th>	Определяет заголовок таблицы (нормальная ячейка с отцентрованным жирным текстом)
<table border=#>	Задает толщину рамки таблицы
<table cellspacing=#>	Задает расстояние между ячейками таблицы
<table cellpadding=#>	Задает расстояние между содержимым ячейки и ее рамкой
<table width=#>	Устанавливает ширину таблицы в пикселах или процентах от ширины документа
<tr align=?> или <td align=?>	Устанавливает выравнивание ячеек в таблице, принимает значения: left, center, или right.
<tr valign=?> или <td valign=?>	Устанавливает вертикальное выравнивание для ячеек таблицы, принимает значения: top, middle, или bottom.
<td colspan=#>	Указывает количество столбцов, которые объединены в одной ячейке (по умолчанию=1)
<td rowspan=#>	Указывает количество строк, которые объединены в одной ячейке (по умолчанию=1)
<td nowrap>	Не позволяет программе просмотра делать перевод строки в ячейке таблицы
Гиперссылки	
	Создает гиперссылку на другие документы или часть текущего документа
 	Создает гиперссылку вызова почтовой программы для написания письма автору документа
	Отмечает часть текста как цель для гиперссылок в документе
	Создает гиперссылку на часть текущего документа

Фреймы	
<frameset></frameset>	Заменяет тег <body> в документе, определяющем структуру фреймов на веб-странице
<frameset rows="value,value">	Определяет строки в таблице кадров, высота которых определена количеством пикселов или в процентном соотношении к высоте таблицы кадров
<frameset cols="value,value">	Определяет столбцы в таблице кадров, ширина которых определена количеством пикселов или в процентном соотношении к ширине таблицы кадров
<frame>	Определяет единичный кадр или область в таблице кадров
<noframes></noframes>	Определяет, что будет показано в окне браузера, если он не поддерживает кадры
<frame src="URL">	Определяет, какой из HTML-документов будет показан в кадре
<frame name="name">	Указывает имя кадра или области, что позволяет перенаправлять информацию в этот кадр или область из других кадров
<frame marginwidth=#>	Определяет величину отступов по левому и правому краям кадра; должно быть равно или больше 1
<frame marginheight=#>	Определяет величину отступов по верхнему и нижнему краям кадра; должно быть равно или больше 1
<frame scrolling=VALUE>	Указывает будет ли выводиться линейка прокрутки в кадре; значение value может быть "yes," "no," или "auto". Значение по умолчанию для обычных документов – auto
<frame noresize>	Препятствует изменению размеров кадра

Приложение 2. Цветовая палитра

Aliceblue F0F8FF	Antiquewhite FAEBD7	Aqua 00FFFF	Aquamarine 7FFFD4
Azure F0FFFF	Beige F5F5DC	Bisque FFE4C4	Black 000000
Blanchedalmond FFEBCD	Blue 0000FF	Blueviolet 8A2BE2	Brown A52A2A
Burlywood DEB887	Cadetblue 5F9EA0	Chartreuse 7FFF00	Chocolate D2691E
Coral FF7F50	Cornflowerblue 6495ED	Cornsilk FFF8DC	Crimson DC143C
Cyan 00FFFF	Darkblue 00008B	Darkcyan 008B8B	Darkgoldenrod B8860B
Darkgray A9A9A9	Darkgreen 006400	Darkkhaki BDB76B	Darkmagenta 8B008B
Darkolivegreen 556B2F	Darkorange FF8C00	Darkorchid 9932CC	Darkred 8B0000
Darksalmon E9967A	Darkseagreen 8FBC8F	Darkslateblue 483D8B	Darkslategray 2F4F4F
Darkturquoise 00CED1	Darkviolet 9400D3	deeppink FF1493	Deepskyblue 00BFFF
Dimgray 696969	Dodgerblue 1E90FF	Firebrick B22222	Floralwhite FFFAF0
Forestgreen 228B22	Fuchsia FF00FF	Gainsboro DCDCDC	Ghostwhite F8F8FF
Gold FFD700	Goldenrod DAA520	Gray 808080	Green 008000
Greenyellow ADFF2F	Honeydew F0FFF0	Hotpink FF69B4	Indianred CD5C5C

Indigo 4B0082	Ivory FFFFF0	Khaki F0E68C	Lavendar E6E6FA
Lavenderblush FFF0F5	Lawngreen 7CFC00	Lemonchiffon FFFACD	Lightblue ADD8E6
Lightcoral F08080	Lightcyan E0FFFF	Lightgoldenrodyellow FAFAD2	Lightgreen 90EE90
Lightgrey D3D3D3	Lightpink FFB6C1	Lightsalmon FFA07A	Lightseagreen 20B2AA
Lightskyblue 87CEFA	Lightslategray 778899	Lightsteelblue B0C4DE	Lightyellow FFFFE0
Lime 00FF00	Limegreen 32CD32	Linen FAF0E6	Magenta FF00FF
Maroon 800000	Mediumauqamarine 66CDAA	Mediumblue 0000CD	Mediumorchid BA55D3
Mediumpurple 9370D8	Mediumseagreen 3CB371	Mediumslateblue 7B68EE	Mediumspringgreen 00FA9A
Mediumturquoise 48D1CC	Mediumvioletred C71585	Midnightblue 191970	Mintcream F5FFFA
Mistyrose FFE4E1	Moccasin FFE4B5	Navajowhite FFDEAD	Navy 000080
Oldlace FDF5E6	Olive 808000	Olivedrab 688E23	Orange FFA500
Orangered FF4500	Orchid DA70D6	Palegoldenrod EEE8AA	Palegreen 98FB98
Paleturquoise AFEEEE	Palevioletred D87093	Papayawhip FFEFD5	Peachpuff FFDAB9

Peru CD853F	Pink FFC0CB	Plum DDA0DD	Powderblue B0E0E6
Purple 800080	Red FF0000	Rosybrown BC8F8F	Royalblue 4169E1
Saddlebrown 8B4513	Salmon FA8072	Sandybrown F4A460	Seagreen 2E8B57
Seashell FFF5EE	Sienna A0522D	Silver C0C0C0	Skyblue 87CEEB
Slateblue 6A5ACD	Slategray 708090	Snow FFFAFA	Springgreen 00FF7F
Steelblue 4682B4	Tan D2B48C	Teal 008080	Thistle D8BFD8
Tomato FF6347	Turquoise 40E0D0	Violet EE82EE	Wheat F5DEB3
White FFFFFF	Whitesmoke F5F5F5	Yellow FFFF00	YellowGreen 9ACD32

Приложение 3. Программное обеспечение, необходимое для выполнения лабораторной работы

№	Программа	Назначение	Дистрибутив	Объем, Мб	Лицензия
1	Notepad++ v.7.5.1	Текстовый редактор. Удобен для написания html-кода	npp.7.5.1.Installer .exe	2,8	freeware
2	GeoHTML v.2.1	Программа создания изображения–навигационной карты	geohtm21.exe	1,2	freeware