

ФЕДЕРАЛЬНОЕ АГЕНТСТВО ЖЕЛЕЗНОДОРОЖНОГО ТРАНСПОРТА

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Иркутский государственный университет путей сообщения»

Сибирский колледж транспорта и строительства

МЕТОДИЧЕСКИЕ УКАЗАНИЯ
К ПРАКТИЧЕСКИМ РАБОТАМ

по дисциплине

ОПЦ.15 Web-дизайн

по специальности

09.02.06 Сетевое и системное администрирование

базовая подготовка среднего профессионального образования

Иркутск 2024

Электронный документ выгружен из ЕИС ФГБОУ ВО ИргГУПС и соответствует оригиналу

Подписант ФГБОУ ВО ИргГУПС Трофимов Ю.А.

00920FD815CE68F8C4CA795540563D259C с 07.02.2024 05:46 по 02.05.2025 05:46 GMT+03:00

Подпись соответствует файлу документа



РАССМОТРЕНО:

Цикловой методической

комиссией специальностей 09.02.06 «Сетевое и системное администрирование»,

09.02.01 «Компьютерные системы и комплексы» и Информатика

Председатель ЦМК: Саквенко Т.В.

Протокол № 9

«11» апреля 2024 г.

Разработчик: Фитисова Н.Н., преподаватель высшей категории Сибирского колледжа транспорта и строительства ФГБОУ ВО «Иркутский государственный университет путей сообщения».

Оглавление

Практическая работа №1	Ошибка! Закладка не определена.
Практическая работа № 2	Ошибка! Закладка не определена.
Практическая работа №3	Ошибка! Закладка не определена.
Практическая работа №4	Ошибка! Закладка не определена.
Практическая работа №5	Ошибка! Закладка не определена.
Практическая работа №6	Ошибка! Закладка не определена.
Практическая работа №7	Ошибка! Закладка не определена.
Практическая работа №8	Ошибка! Закладка не определена.
Практическая работа №9	Ошибка! Закладка не определена.
Практическая работа №10	Ошибка! Закладка не определена.
Практическая работа №11	Ошибка! Закладка не определена.
Практическая работа №12.....	Ошибка! Закладка не определена.

Практическая работа	Количество часов
Практическая работа №1	4
Практическая работа № 2	4
Практическая работа №3	4
Практическая работа №4	4
Практическая работа №5	4
Практическая работа №6	4
Практическая работа №7	4
Практическая работа №8	2
Практическая работа №9	4
Практическая работа №10	4
Практическая работа №11	4
Практическая работа №12	3
Итого	45

Практическая работа №1

Форматирование текста на web-странице с помощью HTML

Цель: формирование умений применять теги логического и физического форматирования при создании Web-страниц.

Время выполнения: четыре академических часа.

Форма занятия: практическая работа.

Теоретический сведения:

С помощью тега `` можно изменить параметры шрифта. Для тега используются следующие параметры: **face**, **size** и **color**.

Параметр **Face** служит для задания гарнитуры шрифтов используемых для текста. Названий шрифтов можно указать несколько, через запятую. В этом случае, если первый указанный шрифт не будет найден, будет использоваться следующий по списку.

Пример 1. Использование параметра face

```
<font face="Arial, Helvetica, sans-serif">ТеКСТ будет написан шрифтом Arial.</font>
```

Size задает размер шрифта в условных единицах от 1 до 7. Средний размер, используемый по умолчанию принят 3. Размер шрифта можно указывать как абсолютной величиной (например, `size=4`), так и относительной (например, `size=+1`, `size=-1`). В последнем случае размер изменяется относительно базового.

Пример 2. Задание размера шрифта

```
<font size=1>Шрифт размера 1</font><br>  
<font size=2>Шрифт размера 2</font><br>  
<font size=3>Шрифт размера 3</font><br>  
<font size=4>Шрифт размера 4</font><br>  
<font size=5>Шрифт размера 5</font><br>  
<font size=6>Шрифт размера 6</font><br>  
<font size=7>Шрифт размера 7</font><br>
```

Шрифт размера 1

Шрифт размера 2

Шрифт размера 3

Шрифт размера 4

Шрифт размера 5

Шрифт размера 6

Шрифт размера 7

Color определяет цвет текста, который можно задавать с помощью названий цветов или в шестнадцатеричном формате.

Пример 3. Изменение цвета текста `П` первая буква этого предложения будет написана шрифтом Arial, красным цветом и увеличенной.

Первая буква этого предложения будет написана шрифтом Arial, красным цветом и увеличенной.

Видоизменение текста - средства его форматирования, такие как выбор начертания шрифта и использование эффектов, позволяющих менять вид текста. В таблице перечислены основные теги, которые применяются для изменения оформления текста.

Код HTML	Описание	Приме
Текст	Жирный текст	Текст
<i>Текст</i>	Курсивное начертание текста	<i>Текст</i>
<u>Текст</u>	Подчеркнутый текст	<u>Текст</u>
^{Текст}	Верхний индекс	e=mc ²
_{Текст}	Нижний индекс	H ₂ O
<strike>Текст</strike>	Зачеркнутый текст	Текст
<pre>Текст</pre>	Текст пишется как есть, включая все пробелы	Текст
Текст	Курсивный текст	<i>Текст</i>
Текст	Жирный текст	Текст

Обычно для создания верхнего или нижнего индекса используется тег **small**, делающий индекс меньше по размеру основного шрифта.

Пример 4. Создание нижнего индекса

<Б>Формула серной КИСЛОТЫ:</Б>

<i>H_{<small>2</small>}SO_{<small>4</small>}</i>

Формула серной кислоты:

H₂SO₄

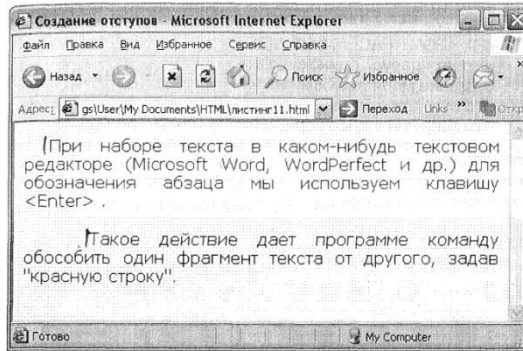
Выравнивание текста

Выравнивание текста определяет его внешний вид и ориентацию краев абзаца и может выполняться по левому, правому краю, по центру или по ширине.

Код HTML	Описание	Пример
<p>Текст</p>	Добавляет новый параграф, по умолчанию выровненный по левому краю. Перед параграфом автоматически добавляется пустая строка.	Текст
<p align=left>Текст</p>	Выравнивание по левому краю.	Текст
<p align=right>Текст</p>	Выравнивание по правому краю.	Текст
<p align=center>Текст</p>	Выравнивание по центру.	Текст
<p align=justify>Текст</p>	Выравнивание по ширине.	Текст по ширине

HTML спецсимволы

Символ	HTML	Комментарий
"	"	Прямая кавычка
&	&	Амперсанд
<	<	Знак "меньше"
>	>	Знак "больше"
™		Знак ТМ
	 	Неразрывный пробел
€	€	Евро
¢	¢	Цент
£	£	Фунт
¤	¤t;	Знак валюты
¥	¥	Йена
	¦	Вертикальная черта
§	§	Знак параграфа
©	©	Знак копирайта
«	&lquo;	Левая типографская кавычка
»	&rquo;	Правая типографская кавычка
¬	¬	Знак отрицания
®	®	Знак (R)
°	°	Знак градуса
±	&plum;	Знак "плюс минус"
²	²	Степень 2
³	³	Степень 3
μ	µ	Знак "микро"
¶	¶	Знак абзаца
·	·	Точка-маркер
¹	¹	Степень 1
¼	¼	Одна четвертая
½	½	Одна вторая
×	×	Крестик
÷	÷	Знак деления

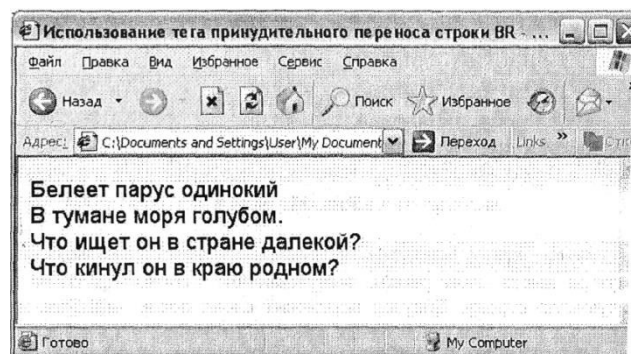


Задание № 3. Использование тега принудительного переноса строки

Запустите программу Блокнот и осуществите ввод следующего HTML кода:

```
<HTML>
<HEAD>
<TITLE> Использование тега принудительного переноса строки
<BR></TITLE>
</HEAD>
<BODY >
<P> <FONT FACE="Arial" SIZE="4">
Белеет парус одинокий <BR>
В тумане моря голубом <BR>
Что ищет он в стране далёкой? <BR>
Что кинул он в краю родном?
</FONT> </P>
</BODY>
</HTML>
```

Теперь сохраните код под любым именем, не забыв присвоить файлу расширение html или htm. Откройте файл в браузере. Результатом должно послужить следующее:



Задание № 4. Использование тегов <H1>—<H6>

Запустите программу Блокнот и осуществите ввод следующего HTML кода:

```
<HTML>
```

<HEAD>

<TITLE>Заголовки H1 – H6 </TITLE>

</HEAD>

<BODY>

<H1> Заголовок 1 </H1>

<H2> Заголовок 2 </H2>

<H3> Заголовок 3 </H3>

<H4> Заголовок 4 </H4>

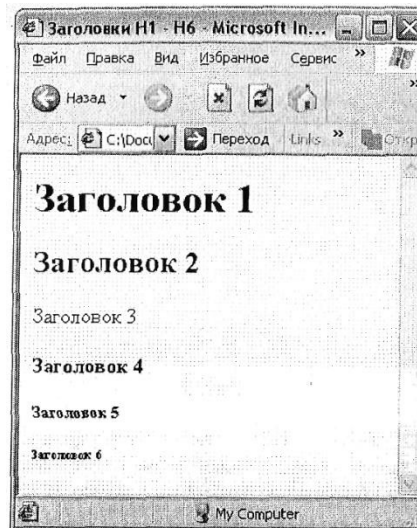
<H5> Заголовок 5 </H5>

<H6> Заголовок 6 </H6>

</BODY>

<HTML>

Теперь сохраните код под любым именем, не забыв присвоить файлу расширение html или htm. Откройте файл в браузере. Результатом должно послужить следующее:



Практическая работа № 2

Разметка web-страниц с использованием рисунков

Цель: формирование умений встраивания в HTML-документы графических изображений.

Время выполнения: четыре академических часа.

Форма занятия: практическая работа.

Для встраивания изображения в документ используется тег **IMG**, имеющий единственный обязательный параметр **src**, который определяет адрес файла с картинкой.

Теоретический сведения:

Файл с рисунком размещается в папке *images* корня сайта.

Для указания адреса изображения можно задавать как абсолютный, так и относительный адрес.

Пример 1 Вставка изображения в документ

```
<html>
```

```
<body>
```

```

```

 - это абсолютный адрес размещения изображения

```

```

 - адрес размещения изображения относительно корня сайта

```

```






 - адрес размещения изображения относительно текущего HTML-документа </body> </html>

Выравнивание изображений

Для изображений можно указывать их положение относительно текста или других изображений на веб-странице. Способ выравнивания изображений задается параметром **align** тега **IMG**. В таблице перечислены возможные значения этого параметра и результат его использования.

Наиболее популярные параметры - **left** и **right**, создающие обтекание текста вокруг изображения. Чтобы текст не прилегал плотно к рисунку, рекомендуется в теге **IMG** добавить параметр **hspace** и **vspace**, задающих расстояние до текста в пикселах.

Код HTML	Описание	Пример
<code></code>	Верхняя граница изображения выравнивается по самому высокому текстовому элементу текущей строки.	<p>Lorem ipsum dolor sit amet,  consec tetuer adipiscing elit...</p>
<code></code>	Верхняя граница изображения выравнивается по самому высокому элементу текущей строки.	<p>ipsum  Lorem dolor sit amet, consec tetuer adipiscing elit...</p>
<code></code>	Выравнивание середины изображения по базовой линии текущей строки.	<p>Lorem ipsum dolor sit amet,  consec-</p>

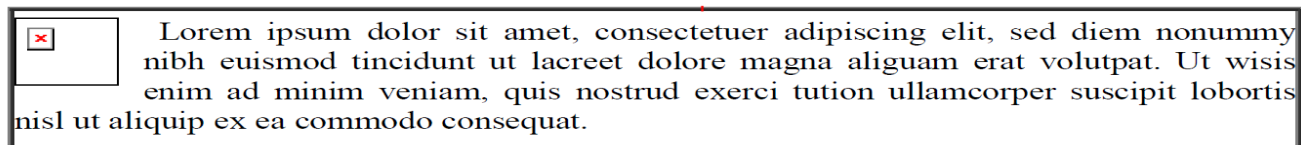
		tetuer adipiscing elit...
<code></code>	Выравнивание середины изображения по середине текущей строки.	<p>Lorem ipsum dolor sit amet,  consec tetuer adipiscing elit...</p>
<code></code>	Выравнивание изображения по базовой линии текущей строки.	<p>Lorem ipsum dolor sit amet,  consec tetuer adipiscing elit...</p>
<code></code>	Выравнивание нижней границы изображения по окружающему тексту..	<p>Lorem ipsum dolor sit amet, consec tetuer adipiscing elit. </p>
<code></code>	Выравнивает изображение по левому краю окна.	<p> Lorem ipsum dolor sit amet, consec tetuer adipiscing elit...</p>
<code></code>	Выравнивает изображение по правому краю окна.	<p>Lorem ipsum  dolor sit amet, consec tetuer adipiscing elit...</p>

Пример 2 Обтекание текста вокруг рисунка

```
<html>
```

```
<body>
```

```
Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam  
nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut  
wisit enim ad minim veniam, quis nostrud exerci tution ullamcorper suscipit lobortis
```



```
nisl ut aliquip ex ea commodo consequat. </body> </html>
```

Задание № 1. Выравнивание изображений

Запустите программу Блокнот и осуществите ввод следующего HTML кода:

```
<HTML>
```

```
<TITLE> Выравнивание изображений </TITLE>
```

```
<BODY> Выравнивание <IMG SRC=букет.gif ALIGN=top> по верхнему краю
```

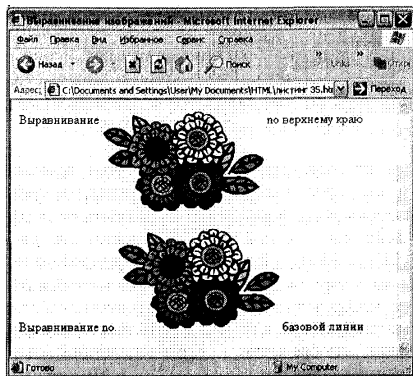
```
<P> Выравнивание по <IMG SRC= букет.gif ALIGN=BASELINE>базовой
```

линии

```
</BODY>
```

```
</HTML>
```

Теперь сохраните код под любым именем, не забыв присвоить файлу расширение **html** или **htm**. Откройте файл в браузере. Результатом должно послужить следующее:



Задание № 2. Выравнивание изображений

Запустите программу Блокнот и осуществите ввод следующего HTML кода:

```
<HTML>
```

```
<TITLE>Выравнивание изображений</TITLE>
```

```
<BODY>
```

```
<IMG SRC=верблюды.gif ALIGN=RIGHT>
```

```
<P ALIGN=JUSTIFY>
```

Верблюды – корабль пустыни. Это название закрепилось за ним с давних пор, когда караваны верблюдов ходили с грузами через пустыни в условиях безводья и недостатка пищи. Верблюды могут проходить через безводные пространства. На них навьючивают груз до 250 кг, с которым они могут делать переходы по 300 км в день со средней скоростью передвижения 10 км/час.

```
<IMG SRC=олень.gif ALIGN=LEFT>
```

```
<P ALIGN=JUSTIFY>
```

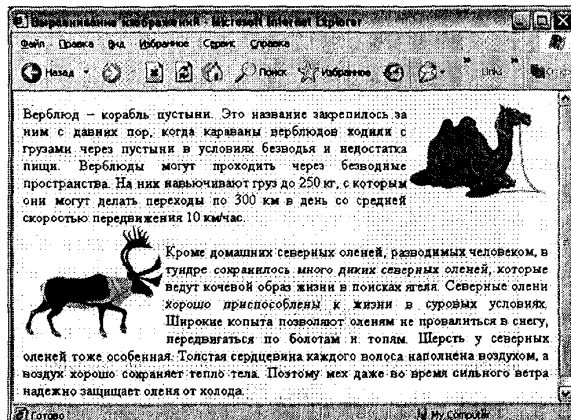
Кроме домашних северных оленей, разводимых человеком, в тундре сохранилось много диких северных оленей, которые ведут кочевой образ жизни в поисках ягеля. Северные олени хорошо приспособлены к жизни в суровых условиях. Широкие копыта позволяют оленям не провалиться в снег, передвигаться по болотам и топям. Шерсть у северных оленей тоже особенная. Толстая сердцевина каждого волоса наполнена воздухом, а воздух хорошо сохраняет тепло тела. Поэтому мех даже во время сильного ветра надежно

защищает оленя от холода.

</BODY>

</HTML>

Теперь сохраните код под любым именем, не забыв присвоить файлу расширение html или htm. Откройте файл в браузере. Результатом должно послужить следующее:



Задание № 3. Различие параметров

выравнивания

Запустите программу Блокнот и осуществите ввод следующего HTML кода:

<HTML>

<TITLE>Различие параметров выравнивания</TITLE>

<BODY>

Выравнивание ALIGN=TOP

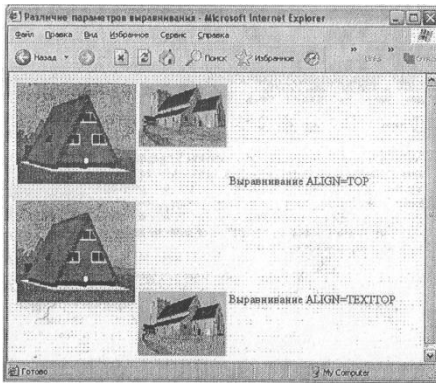
<p>

Выравнивание ALIGN=TEXTTOP

</BODY>

</HTML>

Теперь сохраните код под любым именем, не забыв присвоить файлу расширение html или htm. Откройте файл в браузере. Результатом должно послужить следующее:

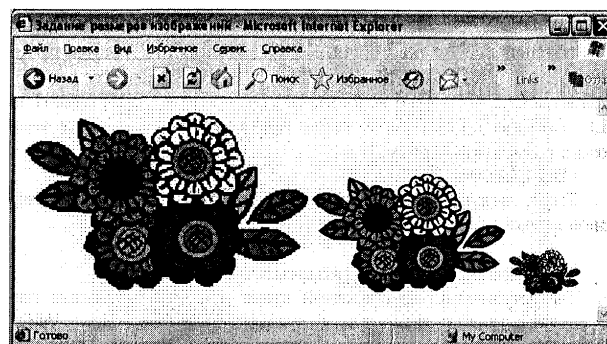


Задание № 4. Задание размеров изображений

Запустите программу Блокнот и осуществите ввод следующего HTML кода:

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

Теперь сохраните код под любым именем, не забыв присвоить файлу расширение html или htm. Откройте файл в браузере. Результатом должно послужить следующее:



Задание № 5. Использование параметров HSPACE и VSPACE

Запустите программу Блокнот и осуществите ввод следующего HTML кода:

```
<HTML>  
<TITLE>Использование параметров HSPACE и VSPACE</TITLE>  
<BODY>
```

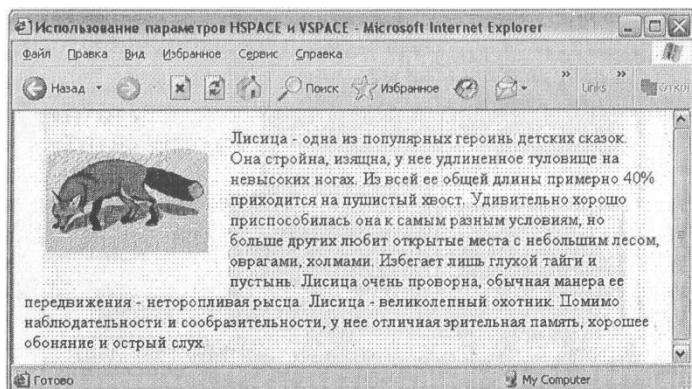
```
<IMG SRC=лиса.gif ALIGN=left HSPACE=20 VSPACE=20>
```

Лисица – одна из популярных героинь детских сказок. Она стройна, изящна, у нее удлиненное туловище на невысоких ногах. Из всей ее общей длины примерно 40% приходится на пушистый хвост. Удивительно хорошо приспособилась она к самым разным уровням, но больше других любит открытые места с небольшим лесом, оврагами, холмами, Избегает лишь глухой тайги и пустынь. Лисица очень проворна, обычная манера ее передвижения – неторопливая рысца Лисица – великолепный охотник: Помимо наблюдательности и сообразительности, у нее отличная зрительная память, хорошее обоняние и острый слух

```
</BODY>
```

```
</HTML>
```

Теперь сохраните код под любым именем, не забыв присвоить файлу расширение html или htm. Откройте файл в браузере. Результатом должно послужить следующее:



Задание № 6. Использование параметров HSPACE и VSPACE

Запустите программу Блокнот и осуществите ввод следующего HTML кода:

```
<HTML>
```

```
<TITLE>Помещение изображения в рамку</TITLE>
```

```
<BODY>
```

```
<IMG SRC=лиса.gif>
```

```
<IMG SRC=лиса.gif BORDER=1>
```

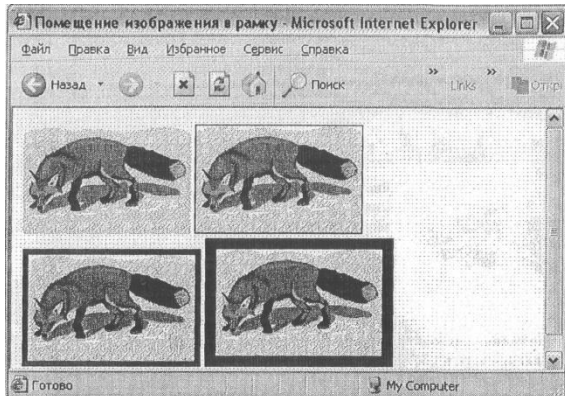
```
<IMG SRC=лиса.gif BORDER =5>
```

```
<IMG SRC=лиса.gif BORDER=10>
```

```
</BODY>
```

</HTML>

Теперь сохраните код под любым именем, не забыв присвоить файлу расширение html или htm. Откройте файл в браузере. Результатом должно послужить следующее:



Задание № 7. Использование альтернативного текста

Запустите программу Блокнот и осуществите ввод следующего HTML кода:

<HTML>

<TITLE>Использование альтернативного текста </TITLE>

<BODY>

</BODY>

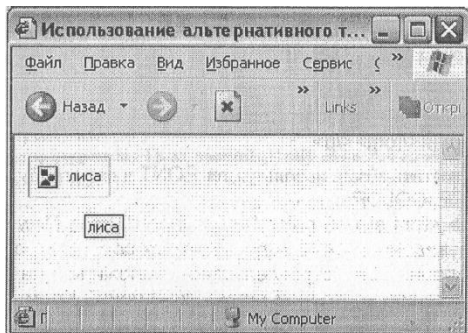
</HTML>

Практическая работа №3

Создание web-сайта на языке HTML. Вставка гиперссылок

Цель: формирование умений осуществления связей между страницами HTML.

Теперь сохраните код под любым именем, не забыв присвоить файлу расширение **html** или **htm**. Откройте файл в браузере. Результатом должно послужить следующее:



Время выполнения: четыре академических часа.

Форма занятия: практическая работа.

Теоретический сведения:

Для создания ссылки необходимо сообщить браузеру, что является ссылкой, а также указать адрес документа, на который следует сделать ссылку. Оба действия выполняются с помощью тега **A**, который имеет единственный параметр **href**. В качестве значения используется адрес документа (**URL**).

Адрес ссылки может быть абсолютным и относительным. Абсолютные адреса работают везде и всюду независимо от имени сайта или веб-страницы, где прописана ссылка.

Пример 1. Использование абсолютных ссылок

```
<html>
<body>
<a href=www.yandex.ru>Поисковая система Яндекс</a>
</body>
</html>
```

Относительные ссылки, как следует из их названия, построены относительно текущего документа или адреса. Примеры таких адресов:

1. /
2. /demo/
3. /images/pic.gif

4. ../help/me.html
5. manual/info.html

Первые две ссылки называются неполные и указывают веб-серверу загружать файл index.html (или default.html) находящемуся в корне сайта (пример 1) или папке demo (пример 2). Если файл index.html отсутствует, браузер, как правило, показывает список файлов, находящихся в данном каталоге. Слэш перед адресом говорит о том, что адресация начинается от корня сайта (пример 3), двоеточие - перейти на уровень выше в списке каталогов сайта (пример 4).

Пример 2. Использование относительных ссылок

```
<html>
<body>
<a href=images/xxx.jpg>Посмотрите на мою фотографию!</a><br>
<a href=tip.html>Как сделать такое же фото?</a>
</body>
</html>
```

Ссылки внутри страницы

Большие документы читаются лучше, если они имеют оглавление со ссылками на соответствующие разделы. Для создания ссылки следует вначале сделать закладку в соответствующем месте и дать ей имя при помощи параметра name тега A.

Пример 1. Создание внутренней ссылки

```
<html>
<body>
<a name=top></a> Друг уронил утюг в унитаз. И разбил его. Причем так разбил, что по назначению унитаз и использовать никак нельзя. Мгновением назад только что вот все было хорошо и вот уже дыра, да такая, что можно забыть, что есть такой предмет в доме. Махнул рукой нечаянно, а потом мучайся...
<a href=#top>Наверх</a>
</body>
</html>
```

Между тегами **** и **** отсутствует текст, так как требуется лишь указать местоположение перехода по ссылке, находящейся внизу страницы. Имя ссылки на закладку начинается символом # , после чего идет название закладки. Название выбирается любое, соответствующее тематике.

Можно, также, делать ссылку на закладку, находящуюся в другой веб-странице и даже другом сайте. Для этого в адресе ссылки надлежит указать ее адрес и в конце добавить символ решетки # и имя закладки.

Пример 2. Ссылка на закладку из другой веб-страницы

```
<html>
<body>
<a href=text.html#bottom>Перейти к нижней части текста</a>
</body>
</html>
```

Ссылка на новое окно

Если требуется сделать ссылку на документ, который открывается в новом окне браузера, используется параметр **target=_blank** тега **A**.

Создание нового окна обычно требуется в случаях, когда делается ссылка на другой сайт, в остальном лучше открывать документы в текущем окне, поскольку обилие окон может сбить читателя с толку.

Так как ссылки на текущее или новое окно ничем не отличаются друг от друга, на некоторых сайтах рядом со ссылкой ставят специальную иконку, показывающую, что документ открывается в новом окне.

Пример 2. Создание ссылки на новое окно

```
<html>
<body>
```

```
<a href=www.bsu.edu.ru>Обычная ссылка на сайт www.bsu.edu.ru</a><br>
<a href=www.bsu.edu.ru target=_blank>Ссылка открывает новое окно на сайт
www.bsu.edu.ru</a>
</body>
</html>
```

Задание № 1. Простейшей гиперссылки

Запустите программу Блокнот и осуществите ввод следующего HTML кода:

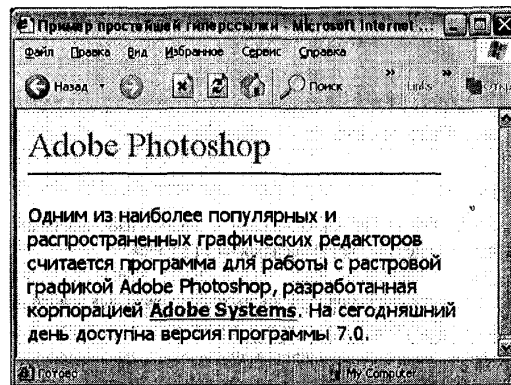
```
<HTML>
<HEAD>
<TITLE>Пример простейшей гиперссылки</TITLE>
</HEAD>
<BODY BGCOLOR="WFFFFFFF" TEXT="black" LINK="#FFOOOO"
ALINK="#FFOOOO" VLINK="blue">
<FONT COLOR="gray" SIZE="6"> Adobe Photoshop </FONT>
<HR COLOR="#003366" WIDTH="90%" ALIGN="left">
<P> <FONT FACE="Tahoma", "Verdana" SIZE="4"> Одним из наиболее
```

популярных и распространенных графических редакторов считается программа для работы с растровой графикой Adobe Photoshop, разработанная корпорацией [Adobe Systems](http://www.adobe.com). На сегодняшний день доступна версия программы 7.0 </P>

</BODY>

</HTML>

Теперь сохраните код под любым именем, не забыв присвоить файлу расширение html или htm. Откройте файл в браузере. Результатом должно послужить следующее:



Задание № 2. Гиперссылки с использованием графики в качестве указателя

Запустите программу Блокнот и осуществите ввод следующего HTML кода:

<HTML>

<HEAD>

<TITLE>Пример гиперссылки с использованием графики в качестве указателя</TITLE>

</HEAD>

<BODY BGCOLOR="#FFFFFF" TEXT="black" LINK="#FF0000" ALINK="#FF0000" VLINK="blue">

<P ALIGN="CENTER">

</P>

</BODY>

</HTML>

Теперь сохраните код под любым именем, не забыв присвоить файлу расширение html или htm. Откройте файл в браузере и ознакомьтесь с результатом.

Задание № 3. Гиперссылки с использованием в качестве указателя и текста, и графики

Запустите программу Блокнот и осуществите ввод следующего HTML кода:

<HTML>

<HEAD>

<TITLE>Пример гиперссылки с использованием в качестве указателя и текста, и графики </TITLE>

</HEAD>

<BODY BGCOLOR="WFFFFFFF" TEXT="black" LINK="#FFOOOO" ALINK="#FFOOOO" VLINK="blue">

 Adobe Photoshop

<HR COLOR="#003366" WIDTH="90%" ALIGN="left">

<P>

 Одним из наиболее популярных и распространенных графических редакторов считается программа для работы с растровой графикой Adobe Photoshop, разработанная корпорацией

<A HREF="<http://www.adobe.com/>">

 Adobe Systems
.

</P>

</BODY>

</HTML>

Теперь сохраните код под любым именем, не забыв присвоить файлу расширение html или htm. Откройте файл в браузере и ознакомьтесь с результатом.

Задание № 4. Составление внутренних гиперссылок

Запустите программу Блокнот и осуществите ввод следующего HTML кода:

```
<HTML>
<HEAD>
<TITLE>Пример составления внутренних гиперссылок</TITLE>
</HEAD>
<BODY BGCOLOR="#FFFFFF" TEXT="black" LINK="#FF0000"
ALINK="#FF0000" VLINK="blue">
  <FONT COLOR="gray" SIZE="6">Терн HTML </FONT>
  <HR COLOR="#003366" WIDTH="90%" ALIGN="left">
  <P> <A HREF="#font">FONT</A> </P>
  <P> <A HREF="#span">SPAN</A> </P>
  <P> <A HREF="#table">TABLE</A> </P>
  <HR ALIGN="center" WIDTH="95%" NOSHADE>
  <P ALIGN="JUSTIFY">
  <A NAME="font"> </A>
  FONT – один из основных тегов физического форматирования текста,
  отображающий свойства шрифтов.
</BODY>
</HTML>
```

Теперь сохраните код под любым именем, не забыв присвоить файлу расширение html или htm. Откройте файл в браузере и ознакомьтесь с результатом.

Задание № 5.

Используя внутренние ссылки, создать следующий словарь терминов:

Словарь терминов

[А](#) [Б](#) [В](#) [Г](#) [Д](#) [Е](#)

А

АВТЕНТИЧЕСКИЙ КАДАНС

кадансовый оборот, в котором заключительная тоническая гармония предваряется доминантовой

АЛИКВОТНЫЕ СТРУНЫ

резонирующие струны, к которым исполнитель не прикасается во время игры

АТАКТА

гармонический элемент на басу нижнего или верхнего вводного тона

[В начало](#)

Б

БАГАТЕЛЬ

небольшая нетрудная для исполнения пьеса

БАРТОКОВСКОЕ ПИЦЦИКАТО

сильный щипок струны с последующим ударом струны о гриф

БОНАНГ

набор из 10-12 гонгов разного размера

[В начало](#)

В

[В начало](#)

Г

[В начало](#)

Д

[В начало](#)

Практическая работа №4

Дополнительные элементы языка HTML для форматирования web-страниц. Списки

Цель: формирование умений создания списков посредством применения основных тегов.

Время выполнения: четыре академических часа.

Форма занятия: практическая работа.

Теоретический сведения:

Нумерованные списки представляют собой набор элементов с их порядковыми номерами. Вид и тип нумерации зависит от параметров тега **OL**, который и используется для создания списка. В качестве маркеров могут быть следующие значения: арабские цифры заглавные латинские буквы прописные латинские буквы заглавные римские цифры прописные римские цифры

Ниже, в таблице приведены различные параметры тега **OL** и результат их применения.

Код HTML	Пример
<pre> текст текст текст </pre>	<p>Нумерованный список с параметрами по умолчанию:</p> <ol style="list-style-type: none"> 1. текст 2. текст 3. текст
<pre><ol start="5"></pre>	<p>Нумерованный список начинающийся с пяти:</p> <ol style="list-style-type: none"> 5. текст 6. текст 7. текст
<pre><ol type="A"></pre>	<p>Нумерованный список с заглавными буквами латинского алфавита:</p> <ol style="list-style-type: none"> A. текст B. текст C. текст
<pre><ol type="a"></pre>	<p>Нумерованный список с прописными буквами латинского алфавита:</p> <ol style="list-style-type: none"> a текст b текст c текст
<pre><ol type="I"></pre>	<p>Нумерованный список с римскими буквами:</p> <ol style="list-style-type: none"> I. текст II. текст III. текст
<pre><ol type="i"></pre>	<p>Нумерованный список с прописными римскими буквами:</p> <ol style="list-style-type: none"> i. текст ii. текст iii. текст
<pre><ol type="1"></pre>	<p>Нумерованный список с арабскими цифрами:</p> <ol style="list-style-type: none"> 1. текст 2. текст 3. текст
<pre><ol type="I" start="7"></pre>	<p>Список с римскими цифрами начинающийся с семи:</p> <ol style="list-style-type: none"> IV. текст V. текст VI. текст

Маркированные списки

Маркированные списки позволяют разбить большой текст на отдельные блоки. Тем самым привлекается внимание читателя к тексту и повышается его читабельность. С учетом худшего восприятия текста с экрана монитора, чем печатного варианта, это является весьма полезным приемом.

Для установки маркированного списка используется тег **UL** и **LI** (Пример 1)

Пример 1. Создание маркированного списка

```
<html>
<head>
<body>
Что следует учитывать при тестировании сайта:
<ul>
<li>работоспособность всех ссылок</li>
<li>поддержку разных браузеров</li>
<li>читабельность текста</li>
</ul>
</body>
</html>
```

Ниже показан результат примера. Обратите внимание, что у маркированного текста

появляются отступы сверху и снизу. Чтобы от них избавиться,

Что следует учитывать при тестировании сайта:

- работоспособность всех ссылок
- поддержку разных браузеров
- читабельность текста

список можно делать без тега **UL**. При этом исчезнут и отступы текста перед маркерами.

Пример 2. Создание маркированного списка без отступов

```
<html>
<head>
<body>
Что следует учитывать при тестировании сайта:
<li>работоспособность всех ссылок</li>
<li>поддержку разных браузеров</li>
<li>читабельность текста</li>
</body>
</html>
```

Ниже показан результат примера 2.

Что следует учитывать при тестировании сайта:

- работоспособность всех ссылок
- поддержку разных браузеров
- читабельность текста

Маркеры могут принимать один из трех видов: круг (по умолчанию), окружность и квадрат. Для выбора типа маркера используется параметр **type="..."** тега **UL**. Вместо многоточия подставляется одно из трех значений указанных в

Код HTML	Пример
<code><ul type="disc"></code>	Что следует учитывать при тестировании сайта: <ul style="list-style-type: none"> • работоспособность всех ссылок • поддержку разных браузеров • читабельность текста
<code><ul type="circle"></code>	Что следует учитывать при тестировании сайта: <ul style="list-style-type: none"> <input type="checkbox"/> работоспособность всех ссылок <input type="checkbox"/> поддержку разных браузеров <input type="checkbox"/> читабельность текста
<code><ul type="square"></code>	Что следует учитывать при тестировании сайта: <ul style="list-style-type: none"> ■ работоспособность всех ссылок ■ поддержку разных браузеров ■ читабельность текста

таблице.

Список определений состоит из двух элементов — термина и его определения. Сам список задается с помощью контейнера `<dl>`, термин — тегом `<dt>`, а его определение — с помощью тега `<dd>`. Вложение тегов для создания списка определений продемонстрировано в примере 11.5.

Пример 3. Общая структура списка определений

```

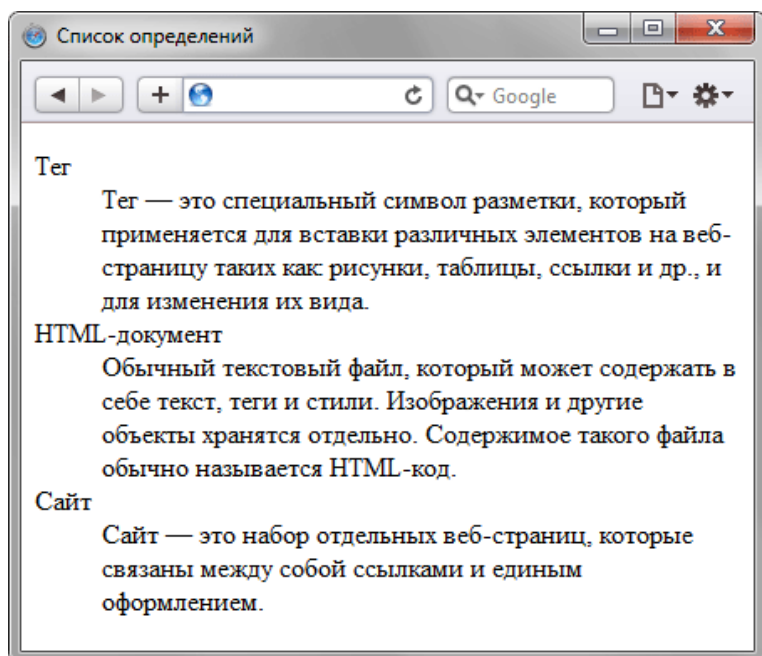
<dl>
  <dt>Термин 1</dt>
  <dd>Определение 1</dd>
  <dt>Термин 2</dt>
  <dd>Определение 2</dd>
</dl>

```

Список определений хорошо подходит для расшифровки терминов, создания глоссария, словаря, справочника и т.д. В примере 4 показано одно из возможных использований этого вида списка.

Пример 4. Создание списка определений

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
<title>Список определений</title>
</head>
<body>
<dl>
<dt>Тег</dt>
<dd>Тег — это специальный символ разметки, который применяется для
вставки различных элементов на веб-страницу таких как: рисунки,
таблицы, ссылки и др., и для изменения их вида.</dd>
<dt>HTML-документ</dt>
<dd>Обычный текстовый файл, который может содержать в себе текст,
теги и стили. Изображения и другие объекты хранятся отдельно.
Содержимое такого файла обычно называется HTML-код.</dd>
<dt>Сайт</dt>
<dd>Сайт — это набор отдельных веб-страниц, которые связаны между
собой
ссылками и единым оформлением.</dd>
</dl>
</body>
</html>
```



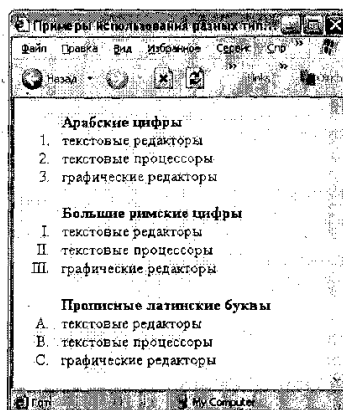
С помощью CSS этот список можно расширить и вместо встроенных символов использовать в качестве маркера рисунок.

Задание № 1. Примеры использования разных типов нумерованного списка

Запустите программу Блокнот и осуществите ввод следующего HTML кода:

```
<HTML>
<HEAD>
<TITLE>Примеры использования разных типов нумерованного
списка</TITLE >
</HEAD>
<BODY>
<OL TYPE="1">
<FONT SIZE="4"> <B>Арабские цифры</B> <FONT>
<LI>текстовые редакторы
< LI>текстовые процессоры
<LI>графические редакторы
</OL>
<OL TYPE="I">
<FONT SIZE="4"> <B>Большие римские цифры</B> </FONT>
<LI>текстовые редакторы
<LI>текстовые процессоры
<LI>графические редакторы
</OL>
<OL TYPE="A">
<FONT SIZE="4"> <B> Прописные латинские буквы</B> </FONT>
<LI>текстовые редакторы
<LI>текстовые процессоры
<LI>графические редакторы
</OL>
</BODY>
</HTML>
```

Теперь сохраните код под любым именем, не забыв присвоить файлу расширение html или htm. Откройте файл в браузере. Результатом должно послужить следующее:

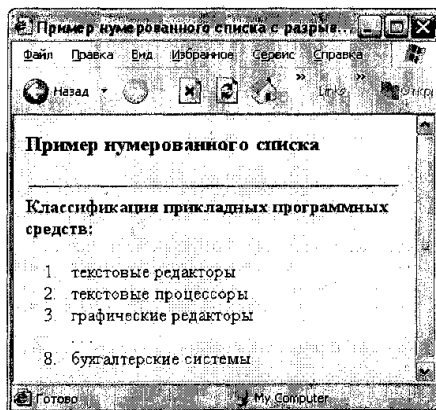


Задание № 2. Пример нумерованного списка с разрывом последовательности

Запустите программу Блокнот и осуществите ввод следующего HTML кода:

```
<HTML>
<HEAD>
< TITLE >Пример нумерованного списка с разрывом последовательности
</TITLE>
</HEAD>
<BODY>
<H3>Пример нумерованного списка</H3>
<HR ALIGN="CENTER" NOSHADE WIDTH="98%">
<FONT FACE="Tahoma"><B>Классификация прикладных программных
средств:</B> </FONT>
<OL>
< LI >текстовые редакторы
< LI >текстовые процессоры
< LI >графические редакторы
<BR>... .
<LI VALUE="8">бухгалтерские системы
</OL >
</BODY>
</HTML>
```

Теперь сохраните код под любым именем, не забыв присвоить файлу расширение html или htm. Откройте файл в браузере. Результатом должно послужить следующее:



Задание № 3. Пример использования графических маркеров

Запустите программу Блокнот и осуществите ввод следующего HTML кода:

```
<HTML>
```

```
<HEAD>
```

```
<TITLE>Пример использования графических маркеров </TITLE>
```

```
</HEAD>
```

```
<BODY>
```

```
<UL>
```

```
<FONT SIZE="4"> <B>Редакторы компьютерной графики: </B> </FONT>
```

```
<IMG SRC="marker.gif">CorelDRAW<BR>
```

```
<IMG SRC="marker.gif">Adobe Illustrator <BR>
```

```
<IMG SRC="marker.gif">Macromedia FreeHand <BR>
```

```
<IMG SRC="marker.gif">Adobe Photoshop <BR>
```

```
<IMG SRC="marker.gif">Macromedia Fireworks <BR>
```

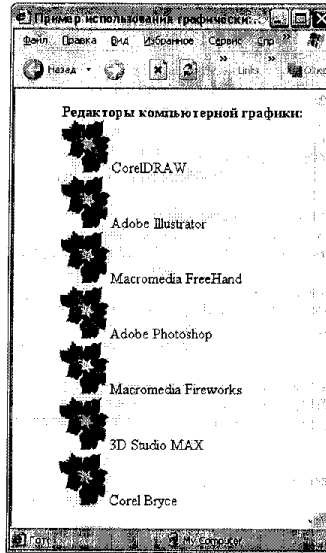
```
<IMG SRC="marker.gif">3D Studio MAX <BR>
```

```
<IMG SRC="marker.gif">Corel Bryce </UL>
```

</BODY>

</HTML>

Теперь сохраните код под любым именем, не забыв присвоить файлу расширение html или htm. Откройте файл в браузере. Результатом должно послужить следующее:



Задание № 4. Пример использования вложенных списков

Запустите программу Блокнот и осуществите ввод следующего HTML кода:

<HTML>

<HEAD>

<TITLE>Пример использования вложенных списков </TITLE>

</HEAD>

<BODY>

<UL TYPE="SQUARE">

 Редакторы компьютерной графики:

Векторные редакторы

<UL TYPE="DISC">

CorelDRAW

CorelDRAW

Corel PhotoPaint

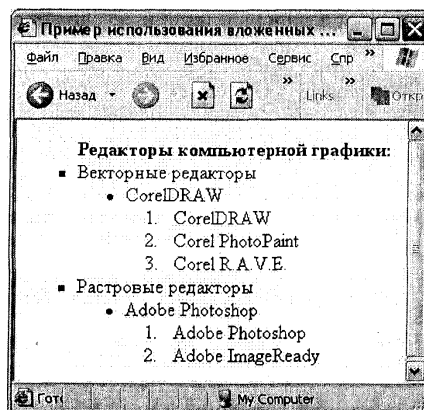
Corel R.A.V.E.

```

</OL>
</UL>
<LI>Растровые редакторы
<UL TYPE="DISC">
<LI>Adobe Photoshop
<OL>
<LI>Adobe Photoshop
<LI>Adobe ImageReady
</OL>
</UL>
</UL>
</UL>
</BODY>
</HTML>

```

Теперь сохраните код под любым именем, не забыв присвоить файлу расширение html или htm. Откройте файл в браузере. Результатом должно послужить следующее:



Задание № 5. Списки определений

Запустите программу Блокнот и осуществите ввод следующего HTML кода:

```

<HTML>
<HEAD>
<TITLE>Пример использования списков определений </TITLE>
</HEAD>
<BODY >

```


<DL>

<H3 ALIGN="CENTER">Редакторы HTML-документов </H3>

<DT> Визуальные Web-редакторы (WYSIWYG-редакторы)

<DD><SMALL>Редакторы, которые позволяют создавать HTML-документы без знания языка HTML. В том виде, как страница будет смотреться в рабочем окне программы, в конечном итоге его и будет отображать браузер.</SMALL>

<DT> HTML-редакторы

<DD> <SMALL> Программы, предназначенные для людей, разбирающихся в конструкциях языка HTML и позволяющие нажатием конкретных кнопок вставлять в текущий документ определенные теги или целые конструкции.</SMALL>

<DT> Текстовые редакторы

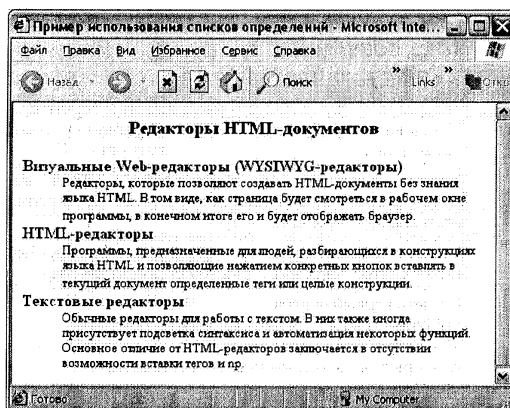
<DD> <SMALL>Обычные редакторы для работы с текстом. В них также иногда присутствует подсветка синтаксиса и автоматизация некоторых функций. Основное отличие от HTML-редакторов заключается в отсутствии возможности вставки тегов и пр. </SMALL>

</DL>

</BODY>

</HTML>

Теперь сохраните код под любым именем, не забыв присвоить файлу расширение html или htm. Откройте файл в браузере. Результатом должно послужить следующее:



Задание №6:

Отформатируйте списки , как показано ниже:

HTML

(от англ. *HyperText Markup Language* — «язык разметки гипертекста») стандартный язык разметки документов во Всемирной паутине. Большинство веб-страниц создаются при помощи языка HTML (или XHTML). Язык HTML интерпретируется браузером и отображается в виде документа, в удобной для человека форме

CSS

(от англ. *Cascading Style Sheets* — каскадные таблицы стилей) — технология описания внешнего вида документа, написанного языком разметки. Преимущественно используется как средство оформления веб-страниц в формате HTML и XHTML, но может применяться с любыми видами документов в формате XML, включая SVG и XUL.

PHP

(от англ. PHP: Hypertext Preprocessor — «PHP: препроцессор гипертекста», англ. Personal Home Page Tools[2] (устар.) — «Инструменты для создания персональных веб-страниц») — скриптовый язык [3], созданный для генерирования HTML-страниц на веб-сервере и работы с базами данных. В настоящее время поддерживается подавляющим большинством хостинг-провайдеров. Входит в LAMP — распространённый набор для создания веб-сайтов (Linux, Apache, MySQL, PHP (Python или Perl)).

Движки для построения блог-сайтов

- Wordpress
- Textpattern
- b2evolution
- Joomla

Функции PHP для работы с массивом

- XV. `array_change_key_case`
- XVI. `array_chunk`
- XVII. `array_combine`
- XVIII. `array_count_values`
- XIX. `array_fill`
- XX. `array_filter`
- XXI. `array_reverse`
- XXII. `array_search`

Смешанный список с автоматической нумерацией. Части света

1. Европа
 - Англия
 - Франция
 - Италия
 - Германия
2. Азия
 - Китай
 - Индия
 - Иран
3. Африка
 1. Египет
 2. Чан
 3. ЮАР
 4. Алжир
4. Америка

Практическая работа №5

Создание таблиц с использованием тегов

Цель: формирование умений создания таблиц посредством применения основных тегов.

Время выполнения: четыре академических часа.

Форма занятия: практическая работа.

Теоретический сведения:

Таблица состоит из строк и столбцов ячеек, которые могут содержать текст и рисунки. Обычно таблицы используются для упорядочения и представления данных, однако возможности таблиц этим не ограничиваются. С помощью таблиц удобно верстать макеты страниц, расположив нужным образом фрагменты текста и изображений.

Для добавления таблицы на веб-страницу используется тег-контейнер **TABLE**. Таблица должна содержать хотя бы одну строку и колонку.

Для добавления строк используются теги `<tr>` и `</tr>`. Чтобы разделить строки на колонки применяются теги `<td>` и `</td>`.

Параметры таблицы Для изменения вида и свойств таблицы используется множество параметров, которые добавляются в теге **TABLE**.

`<table параметр1=... параметр2=...>`

Описание параметров таблицы и их свойств описано ниже.

Свойство	Значение	Описание	Пример
align=	Left Right Center	Выравнивание таблицы	align=center
background=	URL	Фоновый рисунок	background=pic.gif
bgcolor=	#rrggbb	Цвет фона таблицы	bgcolor=#FF9900
border=	n	Толщина рамки в пикселах	border=2
bordercolor=	#rrggbb	Цвет рамки	border-color=#333333
bordercolor-dark=	#rrggbb	Тень рамки	bordercolor-dark=#f0f0f0
cellpadding=	n	Расстояние между ячейкой и ее содержимым	cellpadding=7
cellspacing=	n	Дистанция между ячейками	cellspacing=3
nowrap		Запрещает переносы строк в тексте	<table nowrap>
frame=	Void Above Below Lhs Rhs Hsides Vsides Box	Задание типа рамки таблицы	frame=hsides

valign=	Top Bottom	Выравнивание по высоте	valign=top
width=	n n%	Минимальная ширина таблицы, можно задавать в пикселах или процентах	width=90%
height	n n%	Минимальная высота таблицы, можно задавать в пикселах или процентах	height=18

Примечание

1. Таблица, если не указано особо, всегда выравнивается по левому краю;

2. Параметр **background**, отвечающий за рисунок фона, своеобразно понимается в разных браузерах. IE вставляет картинку во всю таблицу, если таблица по размеру больше фонового рисунка, он повторяется по горизонтали или вертикали. **Netscape** добавляет фоновое изображение в каждую ячейку таблицы;

3. По умолчанию, таблица выводится без рамки. Однако **Netscape** добавляет тонкую линию между ячеек. Чтобы ее не было, всегда указывайте параметр **border=0**;

4. Если ширина таблицы не указана, она подгоняется под содержание ячеек.

Задание 1

Создать 3 таблицы, используя приведенный ниже листинг

1.1 Создание таблицы

<HTML>

<HEAD> <TITLE>ТАБЛИЦА</TITLE>

</HEAD> <BODY> <TABLE border="2" align=center> <TR>

<TD colspan=2 align=center> Заголовок Таблицы.

</TD> </TR> <TR>

<TD align="center">

Первая ячейка первой строки

</TD>

<TD align="center">

Вторая ячейка первой строки

</TD> </TR> <TR>

<TD align="center">

Первая ячейка второй строки

</TD>

<TD align="center"> Вторая ячейка второй строки

</TD> </TR> </TABLE> </BODY> </HTML>

Тег <TABLE> задает таблицу

Атрибуты **border="2"** и **align=center** задают, соответственно, размер границ таблицы и выравнивание ее по центру страницы Тег <TR> задает строку таблицы

Тег <TD> задает ячейку таблицы

Атрибут **colspan=n** объединяет n ячеек по горизонтали (по столбцам) В результате получится следующая таблица, состоящая из двух столбцов и двух строк:

Заголовок Таблицы.	
1 ячейка первой строки	2 ячейка первой строки
1 ячейка второй строки	2 ячейка второй строки

1.2. Сложная таблица

```
<HTML> <HEAD>
```

```
<TITLE>СЛОЖНАЯ ТАБЛИЦА</TITLE>
```

```
</HEAD> <BODY>
```

```
<TABLE border="1" width="75%" align=center> <TR>
```

```
<TD width="66%" colspan="2"> <P align="center"> Две ячейки, объединенные  
по горизонтали
```

```
</TD > </TR > <TR >
```

```
<TD width="33%" rowspan="2" valign="middle" > Две ячейки, объединенные
```

по вертикали

```
</TD >
```

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

по левому краю

```
</TD> </TR > <TR >
```

```
<TD width="33%" > < P align="right" >по правому краю
```

```
</TD > </TR >
```

```
</TABLE > </BODY > </HTML >
```

Атрибут **colspan=n** объединяет n ячеек по горизонтали (по столбцам)

Атрибут **rowspan=n** объединяет n ячеек по вертикали (по строкам) Атрибут

valign="middle" выравнивает текст в ячейке по центру ячейки по вертикали

В результате получается следующая таблица:

Две ячейки, объединенные по горизонтали	
Две ячейки, объединенные по вертикали	по левому краю
	по правому краю

1.3. Более сложная таблица

```

<HTML> <HEAD>
<TITLE>УЧЕБНАЯ ТАБЛИЦА</TITLE>
</HEAD>
<BODY>
<TABLE WIDTH="50%" BORDER="6" CELLSPACING="6"
CELLPADDING="20" ALIGN="center" BORDERCOLORLIGHT="Lime"
BORDERCOLORDARK="Green" BGCOLOR="#DFFDFD"> <THEAD
BGCOLOR="Aqua">
  <TR><TH COLSPAN="3">УЧЕБНАЯ ТАБЛИЦА</ TH ></ TR >
</THEAD>
<TBODY>
<TR>
  <TD WIDTH="33%">Это первая ячейка</TD> <TD WIDTH="33%">Это
вторая ячейка</ TD > <TD ROWSPAN="3">А это три ячейки третьего столбца
объединились в одну большую</TD>
  </TR> <TR>
  <TD COLSPAN="2">Это единственная ячейка второй строки, объединяющая
оба столбца</TD>
  </TR> <TR>
  <TD>Это первая ячейка третьей строки</TD>
  <TD WIDTH="33%">А это вторая ячейка третьей строки</TD >
  </TR> </TBODY>
<TFOOT BGCOLOR="Yellow">
<TR><TD COLSPAN="3" ALIGN="center">
<SMALL>конец</SMALL></TD></TR>
</TFOOT>

```

</TABLE>

</BODY>

</HTML>

Атрибут **CELLSPACING="6"** задает свободное пространство между ячейками таблицы

Атрибут **CELLPADDING="20"** задает свободное пространство между данными в ячейке и ее границами

Атрибут **BORDERCOLORLIGHT="Lime"** задает цвет левого и верхнего углов таблицы

Атрибут **BORDERCOLORDARK="Green"** задает цвет правого и нижнего углов таблицы

Атрибут **BGCOLOR="#DFFFDF"** задает цвет фона таблицы В результате получается следующая таблица:

УЧЕБНАЯ ТАБЛИЦА		
Это первая ячейка	Это вторая ячейка	А это три ячейки третьего столбца объединились в одну большую
Это единственная ячейка второй строки, объединяющая оба столбца		
Это первая ячейка третьей строки	А это вторая ячейка третьей строки	
конец		

Задание 1

Создать турнирную таблицу, представленную ниже на рисунке.

Для задания кавычек используйте специальные символы **«** ("«") и **»** ("»"), для создания длинного тире используйте специальный символ **—** ("—").

**ЧЕМПИОНАТ ГОРОДА
ПО ХОККЕЮ**

ТАБЛИЦА РЕЗУЛЬТАТОВ

	Команда	1	2	3	4	5	6	Шайбы	Очки
1	«РОТОР»		6:4	2:3	7:3	2:1	0:0	17—11	7
2	«РОТАТОР»	4:6		5:4	5:3	5:5	2:2	21—20	6
3	«УРОЖАЙ»	3:2	4:5		0:1	11:2	14:4	32—14	6
4	«МАЧТА»	3:7	3:5	1:0		1:1	10:0	18—13	5
5	«МЕЧТА»	1:2	5:5	2:11	1:1		8:3	17—22	4
6	«ОКАЗИЯ»	0:0	2:2	4:14	0:10	3:8		9—34	2

Задание 2

- задать структуру страниц web-сайта при помощи таблиц.

Алгоритм задания:

1. Задать структуру страниц web-сайта, определив необходимые информационные блоки и задав их положение на странице. Так, например, для книжного магазина определяем следующие информационные блоки:

- Заголовок
- Навигация
- Новинки
- Основная информация
- Реквизиты

2. Привести структуру web-страницы. Для книжного магазина структура web-страницы представлена на рисунке

1. Заголовок сайта	
2. Навигация	4. Основная информация
3. Новинки	
5. Реквизиты	

3. Создать таблицу, соответствующую заданной структуре web-страницы. Представить страницы web-сайта, определенные в предыдущей работе, с использованием таблицы, задающей структуру сайта.

4. Просмотреть результат в браузере



Практическая работа №6

Создание фреймов с использованием тега FRAME

Цель: формирование умений создания web-страниц, содержащих фреймы.

Время выполнения: четыре академических часа.

Форма занятия: практическая работа.

Теоретический сведения:

Фреймы - это окна независимого просмотра HTML-документов. Иногда бывает очень удобно использовать навигацию по странице в виде меню, оформленного в отдельном фрейме, и основного окна, где будет представлена вся основная информация, определяемая действиями пользователя в области меню.

Для создания фрейма используется тег **FRAMESET**, который заменяет тег **BODY** в документе и используется для разделения экрана на области. Внутри данного тега находятся теги **FRAME**, которые указывают на HTML- документ, предназначенный для загрузки в область.

Пример 1. Создание простого фрейма

```
<HTML>
<FRAMESET cols="30%, 70%" border=0>
<FRAME src="menu.html" name="Menu">
<FRAME src="main.html" name="Main">
</FRAMESET>
</HTML>
```

В приведенном примере присутствует объявление двух фреймов, которые будут располагаться вертикальными полосами и занимать соответственно 30 и 70 процентов рабочей области. Вертикальное расположение устанавливается параметром **cols="..."**, а для горизонтальных полос используют параметр **rows="..."**. Параметр **border="..."** определяет границу между фреймами. Определение документа, изначально загружаемого при открытии этого файла (этот документ является HTML-страницей.), задается атрибутом тега **<FRAME>** **src="..."**, при этом необходимо указать параметр **name="..."**, позволяющий задать "имя" созданной области в виде последовательности латинских букв и цифр, использованной как значение этого атрибута. Это имя можно использовать, чтобы загружать новые документы в ранее созданную область. Для этого в тег

<A>, определяющий гиперссылку, необходимо добавить атрибут **target="..."**, значение которого совпадает с ранее определенным именем области. При переходе по данной гиперссылке новый документ загрузится в указанный фрейм. Например, предположим, что начальная страница Web-узла состоит из двух фреймов: слева располагается навигационная панель, а справа - текущая страница. Если правой области присвоено имя, используемое во всех ссылках, имеющихся в левой области, то щелчок на любой ссылке навигационной панели приведет к обновлению информации в соседней области, оставляя навигационную панель без изменений. В данном документе находятся только ссылки на другие (существующие) HTML-документы, которые будут загружены браузером при открытии страницы с фреймами. Файл **menu.html** имеет следующую структуру:

```
HTML>
<HEAD>
<TITLE></TITLE>
</HEAD>
<BODY>
<H1>Первый фрейм</H1>
</BODY>
</HTML>
```

Файл **main.html** также имеет указанную выше структуру. Следует сказать, что оба этих файла должны располагаться в том же каталоге, где расположен файл с фреймами.

Задание № 1. Разветвлённая фреймовая структура

Запустите программу Блокнот и осуществите ввод следующего HTML кода:

```
<HTML>
<HEAD>
<TITLE>Пример разветвленной фреймовой структуры </TITLE>
</HEAD>
<FRAMESET COLS="30%, 70%" FRAMEBORDER="0"
FRAMESPACING="0" BORDER="0">
```

```

<FRAME NAME="left" SRC="left.html" SCROLLING="yes"
MARGINWIDTH="10" MARGINHEIGHT="10">
<FRAMESET ROWS="*, 2*" FRAMEBORDER="0" FRAMESPACING="0"
BORDER="0">
<FRAME NAME="top" SRC="top.html"> <FRAME NAME="bottom"
SRC="bottom.html">
</FRAMESET>
</HTML>

```

Теперь сохраните код под любым именем, не забыв присвоить файлу расширение html или htm. Откройте файл в браузере и ознакомьтесь с результатом.

▲ Пример описывает фреймовую структуру, состоящую из двух вертикальных и двух горизонтальных фреймов.

Конструкция

```

<FRAMESET COLS="30%, 70%" FRAMEBORDER="0"
FRAMESPACING="0" BORDER="0">

```

показывает, что окно браузера поделено на две части: 30% и 70%, соответственно. Следующий за ней тег <FRAME> задает свойства левого фрейма.

Правый фрейм разбит еще на два фрейма, на этот раз горизонтальных, для чего дается отдельная конструкция:

```

<FRAMESET ROWS="*, 2*" FRAMEBORDER="0" FRAMESPACING="0"
BORDER="0">

```

После этого посредством тега <FRAME> дается описание двух упомянутых горизонтальных фреймов.

Задание № 2. Файл верхнего фрейма "menu.html"

Запустите программу Блокнот и осуществите ввод следующего HTML кода:

```

<HTML>
<BODY BGCOLOR="red" TEXT="black" LINK="blue" VLINK="blue">
<H3>Ссылки верхнего фрейма</H3>
<HR>
<FONT FACE="Tahoma" SIZE="2">

```

```

<UL TYPE="square">
  <LI> <A HREF="text.html" TARGET="left">Файл с текстом в левом нижнем
фрейме</A>
  <LI> <A HREF="text.html" TARGET="right">Файл с текстом в правом
нижнем фрейме</A>
  <LI> <A HREF="text.html" TARGET="menu">Файл с текстом в верхнем
фрейме</A>
  <LI> <A HREF="text.html" TARGET="_top">Файл с текстом в полном
окне</A>
  <LI> <A HREF="text.html" TARGET="_blank:">Файл с текстом в новом
окне</A>
  <LI> <A HREF="text.html" TARGET="_self">Файл с текстом в текущем
фрейме</A>
</OL>
</FONT>
</BODY>
</HTML>

```

▲ Первая ссылка откроется в левом нижнем фрейме по причине указания конструкции TARGET="left" ("left" в данном случае – это внутреннее имя данного фрейма).

Вторая ссылка откроется в правом нижнем фрейме (указано внутреннее имя фрейма "right").

Третья ссылка будет открыта в этом же окне, так как "menu" – это имя текущего фрейма, из которого открываются гиперссылки.

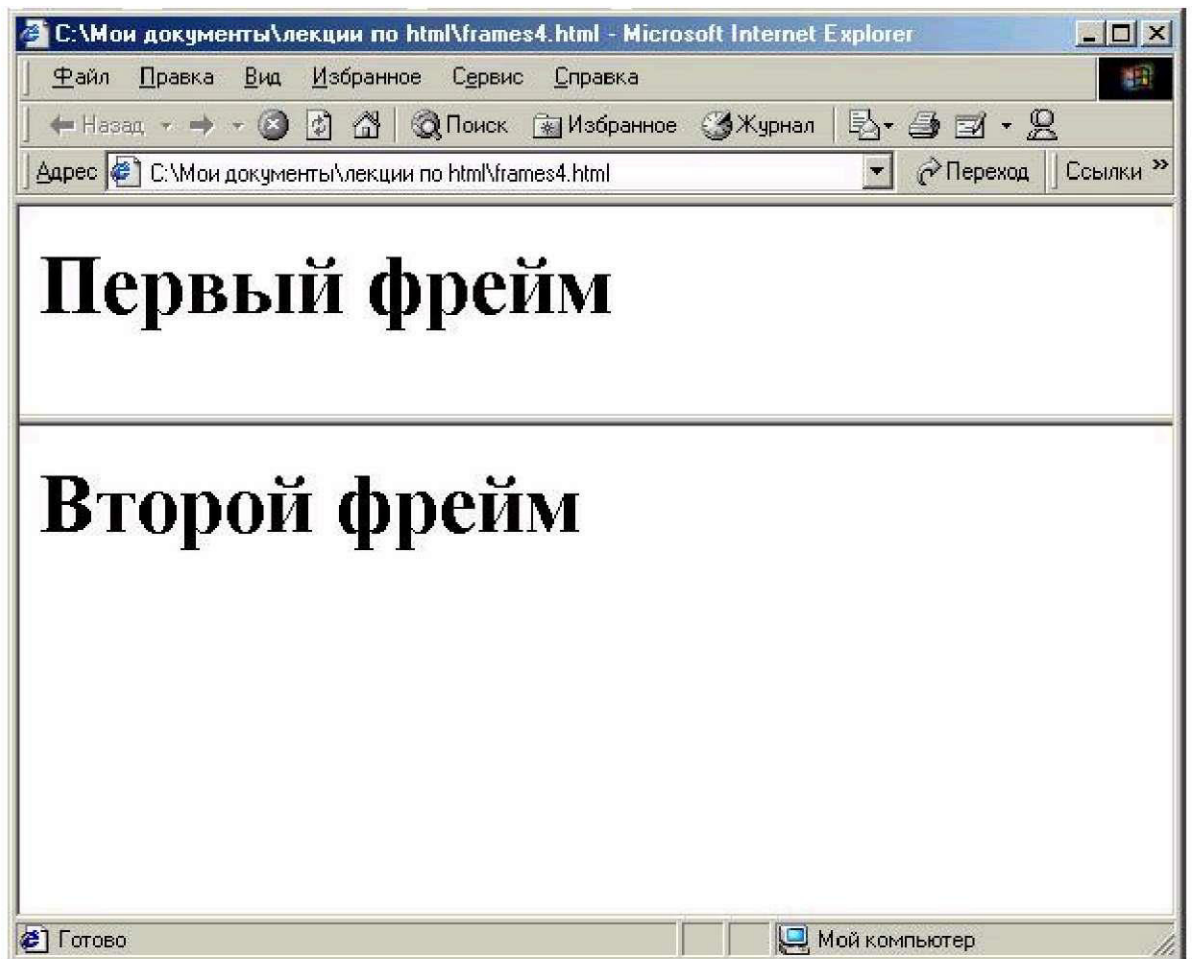
Четвертая ссылка откроется в полном окне браузера (конструкция "top").

Пятая ссылка будет открыта в отдельном новом окне поверх фреймовой структуры (действие "_blank").

Наконец, последняя ссылка откроется в текущем фрейме (аналогично дейст

Задание 3

Расположить эти фреймы по вертикали, используя атрибут rows="...\ \ В результате у вас должно получиться следующее:



Задание 4

Распространенное явление - комбинация вертикальных и горизонтальных фреймов.

`<FRAMESET cols="*, 55%">` символ * означает все оставшееся место

`<FRAME src="homepage.htm" name="Frame1">`

`<FRAMESET rows="15%, 15%, 70%">`

`<FRAME src="menu.htm" name=" Frame2">`

`<FRAME src="menu2.htm" name=" Frame3">`

`<FRAME src="menu3.htm" name=" Frame4">`

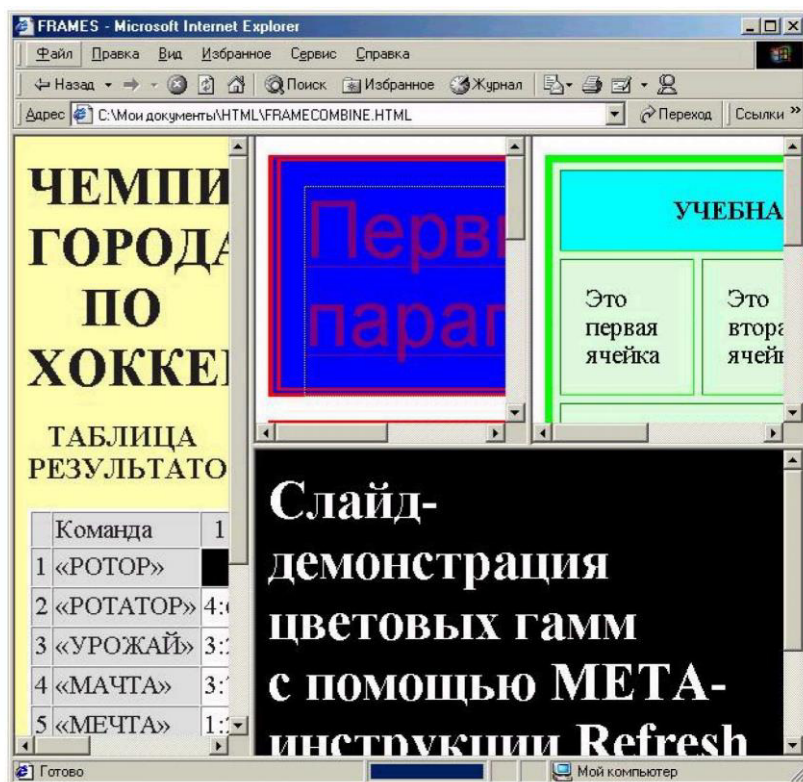
`</FRAMESET> </FRAMESET>`

Используя вложенные теги **FRAMESET**, разбейте страницу следующим образом:

Задание 5

Можно в качестве значения параметра SRC тега `<FRAME>` задать имя файла, который, в свою очередь, содержит описание структуры фреймов. В данном

случае тег <FRAME> будет использован для указания на HTML-документ, который является фреймовой структурой и используется в качестве отдельного фрейма.



Используя атрибут **target="..."**, сделать в фреймах гиперссылки, по нажатию на которые в эти же или в другие фреймы произойдет загрузка других страниц.

Практическая работа №7

Создание формы с использованием тега FORM

Цель: формирование умений создания web-страниц, содержащих пользовательские формы.

Время выполнения: четыре академических часа.

Форма занятия: практическая работа.

5.1.1 Формы. Общие сведения

Формы предназначены для организации взаимодействия с пользователем. Они позволяют вводить текст, осуществлять выбор из предложенных значений при помощи списков или кнопок, организовывать интерактивный обмен информацией между Web-страницей и сервером.

Как правило, форма работает совместно с установленным на сервере сценарным приложением, обрабатывающим введенную информацию. Механизм обработки форм представлен на рисунке 5.1.

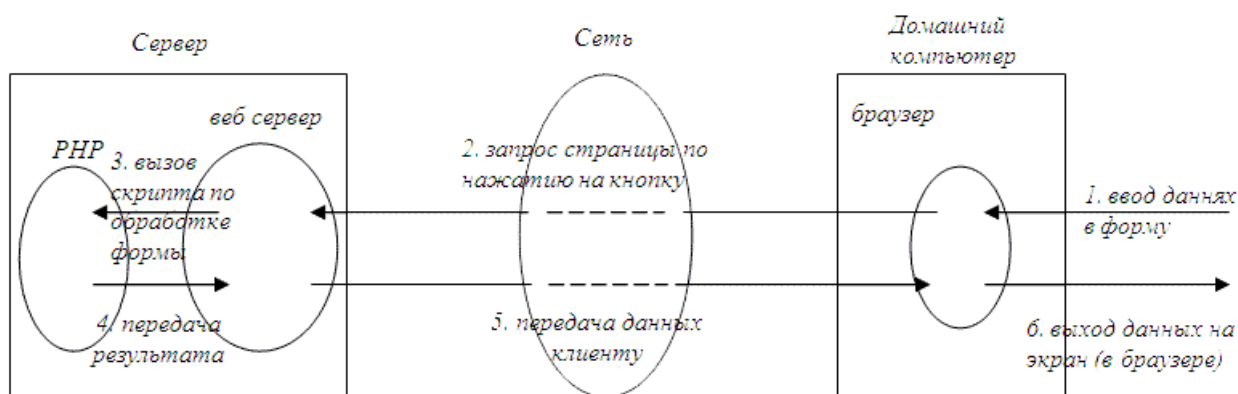


Рисунок 5.1 – Обработка данных формы сервером

Наиболее широко формы применяются для поиска, регистрации, заполнения анкет, тестов и т.д. Форма определяется с помощью тегов `<form></form>`, между которыми располагаются поля ввода, кнопки, а также все необходимые элементы оформления формы. Для указания параметров в теге `<form>` используются атрибуты (см. таблицу 4.1).

Таблица 5.1 — Атрибуты тега

Атрибут	Назначение
action	адрес программы или документа, которые обрабатывают данные формы
method	метод передачи данных по протоколу http. Возможные варианты значений метода: GET — способ передачи, когда данные, введенные в форму добавляются в строку запроса
name	имя формы. Используется для обращения к форме с помощью javascript.
target	имя окна или фрейма, куда обработчик будет загружать возвращаемый результат
enctype	тип информации формы. Различается в случае использования символично-цифровой информации и в случае использования файлов

5.1.1 Элементы управления формы

Сама форма служит только обязательным контейнером для размещения элементов управления. Внешний вид элементов управления зависит от установленных значений атрибутов.

Поле ввода

Однострочное поле ввода и используется, когда необходимо, чтобы пользователь ввел в форму данные в произвольной форме но ограниченные по объему. Для получения поля ввода используют тег **<input type="text">**.

Тег **<input>** является одним из разносторонних элементов формы и позволяет создавать разные элементы интерфейса и обеспечить взаимодействие с пользователем. Главным образом предназначен для создания текстовых полей, различных кнопок, переключателей и флажков.

Основной параметр тега **<input>**, определяющий вид элемента — **type**. Он позволяет задавать следующие элементы формы: текстовое поле (**text**), поле с паролем (**password**), переключатель (**radio**), флажок (**checkbox**), скрытое поле (**hidden**), кнопка (**button**), кнопка для отправки формы (**submit**), кнопка для очистки формы (**reset**), поле для отправки файла (**file**) и кнопка с изображением

(**image**). Для каждого элемента существует свой список параметров, которые определяют его вид и характеристики.

Кроме атрибута `type` тег имеет следующие атрибуты (см. таблица 4.2):

Таблица 5.2 — Атрибуты тега

Атрибут	Назначение
<code>align</code>	определяет выравнивание, в случае если <code>type="image"</code>
<code>alt</code>	определяет альтернативный текст для кнопки с изображением (<code>type="image"</code>)
<code>checked</code>	предварительно активированный переключатель или флажок
<code>disabled</code>	блокирует доступ и изменение элемента (поле недоступное)
<code>maxlength</code>	максимальное количество символов, разрешенных в тексте
<code>name</code>	имя поля, предназначено для того, что бы отличать один элемент управления от другого. Имя может быть указано по всем правилам именования переменных
<code>readonly</code>	разрешает только чтение с текстового элемента управления
<code>size</code>	ширина текстового поля в символах
<code>value</code>	значение элемента. Значение переменной указанной в атрибуте <code>name</code>

Переключатель Женский
 Мужской

Переключатели определяют поля выбора одного значения из нескольких доступных, для каждой позиции переключателя создается свой тег `<input type="radio">`. Группируются переключатели при помощи одинакового имени, задаваемого атрибутом `name`.

Флажки. Подписаться на рассылку

Флажки используются, когда необходимо, чтобы пользователь выбрал один или несколько вариантов из ограниченного числа вариантов выбора. Флажки в форме не зависят друг от друга, их можно установить или сбросить в любой

комбинации. Для каждого флажка необходимо задать свое уникальное имя при помощи атрибута name. Создаются флажки тегом `<input type="checkbox">`. Для установки флажка при загрузке страницы необходимо указать атрибут `checked="checked"`.

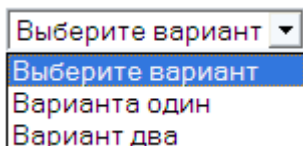
Командные кнопки.

Командная кнопка отправки (`type="submit"`) используются для выполнения пересылки данных формы на сервер. Командная кнопка сброса (`type="reset"`) возвращает форму к исходному состоянию (очищает форму).

Поле выбора файла.

Поле выбора файла (`type="file"`) создает поле для выбора файла, который будет загружен на сервер вместе с информацией формы. Рядом с полем ввода отображается командная кнопка “Обзор...”, открывающая стандартное диалоговое окно выбора файла. Если форма подразумевает загрузку файлов на сервер, то в атрибутах формы должны быть установлены следующие атрибуты `method="post"` и `enctype="multipart/form-data"`.

Списки выбора.



Списки выбора бывают двух типов: раскрывающиеся списки (выпадающие меню) и списки с множественным выбором. Независимо от типов списков описываются они одинаково с помощью пары тегов `<select>/select>`. Отдельные элементы списка задаются с использованием тега `<option>`. Тип списка определяется при помощи атрибута `multiple` тега `<select>`

Задание № 1. Варианты использования элемента текстовой

Запустите программу Блокнот и осуществите ввод следующего HTML кода:

```
<HEAD>
```

```
<TITLE>Варианты использования элемента текстовой строки</TITLE>
```

```
</HEAD>
```

```
BODY BGCOLOR="#FFFFFF" TEXT="black" LINK="#00FF00"
```

```
ALINK="#00FF00" VLINK="blue">
```

<H3>Пожалуйста, сообщите свои данные:</H3>

<FORM NAME="mail" ACTION="/cgi-bin/mail.cgi" METHOD="post"> Ваше имя:

<INPUT TYPE="text" SIZE="20" MAXLENGTH="50" NAME="name">

Ваш E-mail:

<INPUT TYPE="text" SIZE="30" MAXLENGTH="35" NAME="email">

Ваш сайт:

<INPUT TYPE="text" SIZE="40" MAXLENGTH="60" NAME="www" >

Комментарий:

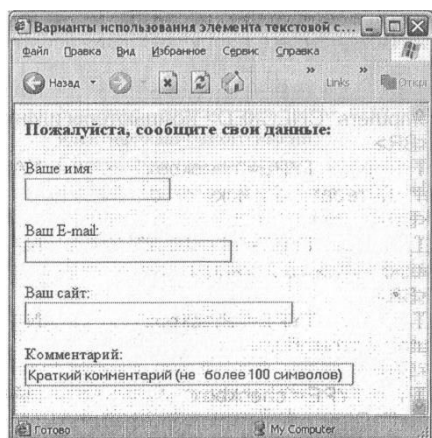
<INPUT TYPE="text" SIZE="50" MAXLENGTH="100" NAME="comment"
VALUE="Краткий комментарий (не более 100 символов)">

</FORM>

</BODY>

</HTML>

Теперь сохраните код под любым именем, не забыв присвоить файлу расширение html или htm. Откройте файл в браузере и результатом должно послужить следующее:



Задание № 2. Использование элемента опции выбора

Запустите программу Блокнот и осуществите ввод следующего HTML кода:

<HTML>

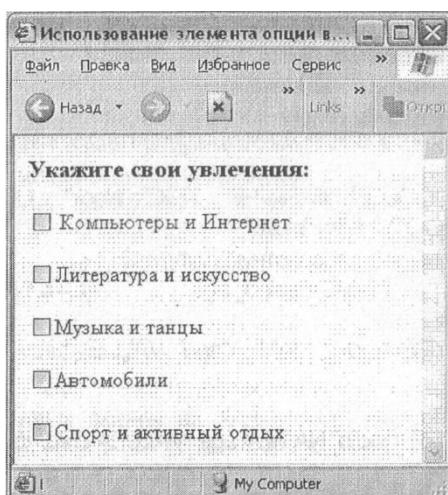
<HEAD>

```

<TITLE>Использование элемента опции выбора</ TITLE >
</HEAD>
<BODY   BGCOLOR="#FFFFFF"   TEXT="black"   LINK="#OOFFOO"
ALINK="#OOFFOO" VLINK="blue">
  <H3>Укажите свои увлечения:</H3>
  <FORM   NAME="select_hobby"   ACTION="/cgi-bin/hobby.cgi"
METHOD="get">
    <INPUT   TYPE="checkbox"   NAME="hobby"   VALUE="computers"
CHECKED> Компьютеры и Интернет <BR><BR>
    <INPUT   TYPE="checkbox  NAME="hobby"  VALUE="art">Литература и
искусство <BR> <BR>
    <INPUT   TYPE="checkbox  NAME="hobby"  VALUE="music">Музыка и
танцы <BR><BR>
    <INPUT   TYPE="checkbox  NAME="hobby"  VALUE="auto">Автомобили
<BR><BR>
    <INPUT   TYPE="checkbox"  NAME="hobby"  VALUE="sport">Спорт и
активный отдых </FORM>
  </BODY>
</HTML>

```

Теперь сохраните код под любым именем, не забыв присвоить файлу расширение html или htm. Откройте файл в браузере и результатом должно послужить следующее:



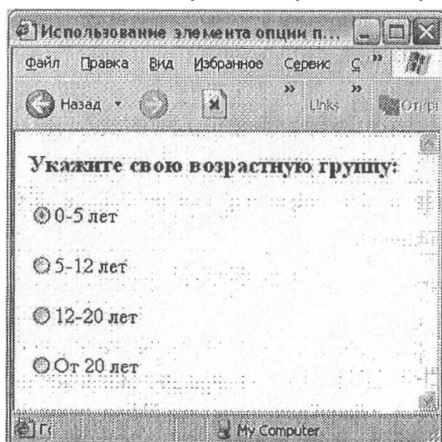
▲ Как видно из рисунка, пользователь может выбирать несколько вариантов поля CHECKBOX, значение каждого из которых будет передано программой-обработчиком на Web-сервер.

Задание № 3. Использование элемента опции переключения

Запустите программу Блокнот и осуществите ввод следующего HTML кода:

```
<HTML>
<HEAD>
<TITLE>Использование элемента опции переключения</TITLE >
</HEAD>
<BODY BGCOLOR="#FFFFFF" TEXT="black" LINK="#OOFFOO" ALINK="#OOFFOO"
VLINK="blue">
<H3>Укажите свою возрастную группу:</H3>
<FORM NAME="select_age" ACTION="/cgi-bin/age.cgi" METHOD="get">
<INPUT TYPE="radio" NAME="age" VALUE="baby" CHECKED>0-5 лет
<BR> <BR>
<INPUT TYPE="radio" NAME="age" VALUE="child">5-12 лет <BR> <BR>
<INPUT TYPE="radio" NAME="age" VALUE="junior">12-20 лет <BR> <BR>
<INPUT TYPE="radio" NAME="age" VALUE="adult">От 20 лет
</FORM>
</BODY>
</HTML>
```

Теперь сохраните код под любым именем, не забыв присвоить файлу расширение html или htm. Откройте файл в браузере и результатом должно послужить следующее:



Задание № 4. Использование кнопок отправления и сброса данных

Запустите программу Блокнот и осуществите ввод следующего HTML кода:

```
<HTML>
<HEAD>
```

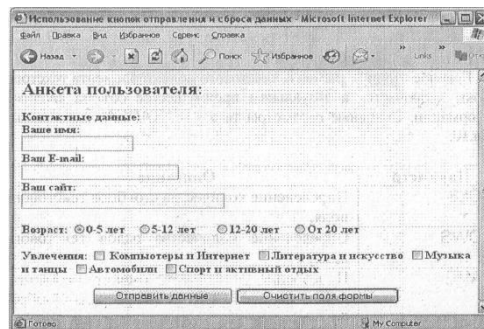


```

<P ALIGN ="center">
<INPUT TYPE="submit" VALUE="Отправить данные">&nbsp;
<INPUT TYPE="reset" VALUE="Очистить поля формы">
</FORM>
</BODY>
</HTML>

```

Теперь сохраните код под любым именем, не забыв присвоить файлу расширение html или htm. Откройте файл в браузере и результатом должно послужить следующее:



Задание № 5. Применение тега <TEXTAREA>

Запустите программу Блокнот и осуществите ввод следующего HTML кода:

```

<HTML>
<HEAD>
<TITLE>Использование текстового поля</TITLE>
</HEAD>
<BODY BGCOLOR="#FFFFFF" TEXT="black" LINK="#00FF00"
ALINK="#00FF00" VLINK="blue">
<H3>Пожалуйста, сообщите свои данные:</H3>
<FORM NAME="mail" ACTION="cgi-bin/mail.cgi" METHOD="get"> Ваше
имя:<BR>
<INPUT TYPE="text" SIZE="30" MAXLENGTH="35" NAME="name"> <BR>
Ваш E-mail:<BR>
<INPUT TYPE="text" SIZE="30" MAXLENGTH="35" NAME="email"> <BR>
Комментарий:<BR>

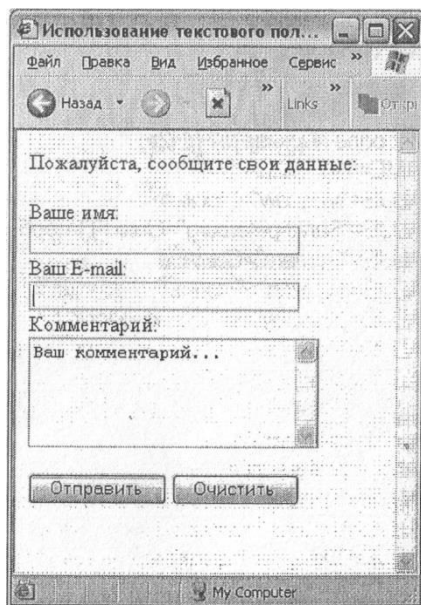
```

```

<TEXTAREA COLS="25" ROWS="5" NAME="comment">Ваш
комментарий...</TEXTAREA>
<BR> <BR>
<INPUT TYPE="submit" VALUE="Отправить">&nbsp;
<INPUT TYPE="reset" VALUE="Очистить">
</FORM>
</BODY>
</HTML>

```

Теперь сохраните код под любым именем, не забыв присвоить файлу расширение html или htm. Откройте файл в браузере и результатом должно послужить следующее:



Задание № 6. Использование выпадающего меню и списка наименований

Запустите программу Блокнот и осуществите ввод следующего HTML кода:

```

<HTML>
<HEAD>
<TITLE>Использование      выпадающего      меню      и      списка
наименований</TITLE>
<HEAD>
<BODY BGCOLOR="#FFFFFF" TEXT="black" LINK="#00FF00"
ALINK="#00FF00" VLINK="blue">
<FORM NAME="anketa" ACTION="/cgi-bin/anketa.cgi" METHOD="get">

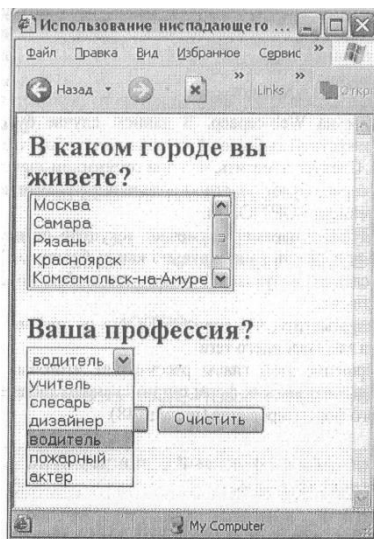
```

```

<H2>В каком городе вы живете?</H2>
<SELECT NAME="city" SIZE="5">
<OPTION VALUE="Moscow">Москва
<OPTION VALUE="Sankt-peterburg">Санкт-петербург
<OPTION VALUE="Samara">Самара
<OPTION VALUE="Ryazan">Рязань
<OPTION VALUE="Krasnoyarsk">Красноярск
<OPTION VALUE="Komsomolsk">Комсомольск-на-Амуре
<OPTION VALUE="Vladivostok">Владивосток
</SELECT>
<H2>Ваша профессия?</H2>
<SELECT NAME="profession">
<OPTION VALUE="Teacher">учитель
<OPTION VALUE="Technik">слесарь
<OPTION VALUE="Designer">дизайнер
<OPTION VALUE="Driver" SELECTED>водитель
<OPTION VALUE="Fireman">пожарный
<OPTION VALUE="Actor">актёр
</SELECT>
<BR><BR>
<INPUT TYPE="submit" VALUE="Отправить">&nbsp;
<INPUT TYPE="reset" VALUE="Очистить">
</FORM>
</BODY>
</HTML>

```

Теперь сохраните код под любым именем, не забыв присвоить файлу расширение html или htm. Откройте файл в браузере и результатом должно послужить следующее:



Задание № 7. Итоговый пример по организации пользовательских форм

Запустите программу Блокнот и осуществите ввод следующего HTML кода:

```
<HTML>
<HEAD>
<TITLE>Итоговый пример по организации пользовательских форм</TITLE>
<HEAD>
<BODY BGCOLOR="#FFFFFF" TEXT="black" LINK="#00DFF00"
ALINK="#00FF00" VLINK="blue">
<H2 ALIGN="center">ПОЛЬЗОВАТЕЛЬСКАЯ АНКЕТА</H2>
<FORM NAME="user_anketa" ACTION="/cgi-bin/anketa.cgi" METHOD="posf
ENCTYPE="multipart/form-data">
<TABLE BORDER="0" CELLPADDING="5" CELLSPACING="3"
ALIGN="center" WIDTH="98%">
<TR>
<TH BGCOLOR="#999999" WIDTH="50%">Личные данные</TH>
<TH BGCOLOR="#999999" WIDTH="50%">Увлечения</TH>
</TR>
<TR>
<TD VALIGN="top" BGCOLOR="#EEEEEE" WIDTH="50%">
Ваше имя:<BR>
<INPUT TYPE="text" SIZE="30" MAXLENGTH="35" NAME="name"> <BR>
Ваш пароль:<BR>
```

```

        <INPUT      TYPE="password"      SIZE="30"      MAXLENGTH="35"
NAME="password">
    <BR>
    Ваша фотография:<BR>
    <INPUT TYPE="file" SIZE="18" NAME="photo">
    </TD>
    <TD VALIGN="top" BGCOLOR="#EEEEEE" WIDTH="50%">
    <INPUT      TYPE="checkbox"      NAME="hobby"      VALUE="computers"
CHECKED>Компьютеры и Интернет
    <BR>
    <INPUT  TYPE="checkbox"  NAME="hobby"  VALUE="art"Литература  и
искусство
    <BR>
    <INPUT  TYPE="checkbox"  NAME="hobby"  VALUE  ="music">Музыка  и
танцы
    <BR>
    <INPUT TYPE="checkbox" NAME="hobby" VALUE ="auto">Автомобили
    <BR>
    <INPUT  TYPE="checkbox"  NAME="hobby"  VALUE="sport">Спорт  и
активный отдых
    </TD>
    </TR>
    <TR>
    <TH BGCOLOR="#999999" WIDTH="50%">Место жительства</TH>
    <TH BGCOLOR="#999999" WIDTH="50%"> Комментарии</TH>
    </TR>
    <TR>
    <TD VALIGN="top" BGCOLOR="#EEEEEE"WIDTH="50%">
    <SELECT NAME="city" SIZE="5">
    <OPTION VALUE="Moscow">Москва
    <OPTION VALUE="Spb">Санкт-Петербург
    <OPTION VALUE="Samara">Самара

```

```

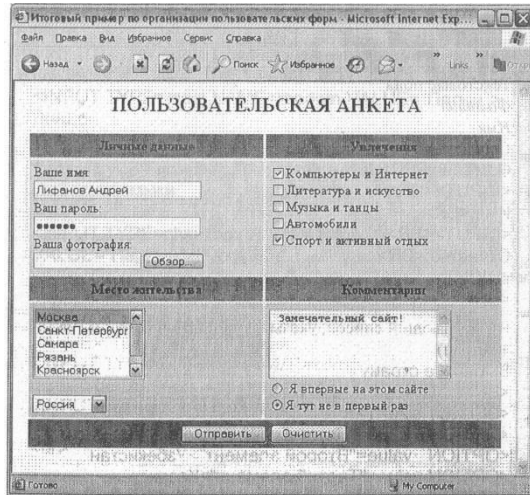
<OPTION VALUE="Ryazan">Рязань
<OPTION VALUE="Krasnoyarsk">Красноярск
<OPTION VALUE="Komsomolsk">Комсомольск-на-Амуре
<OPTION VALUE="Vladivostok">Владивосток
</SELECT>
<BR><BR>
<SELECT NAME="country">
<OPTION VALUE="Russia">Россия
<OPTION VALUE="England">Англия
<OPTION VALUE="USA">США
<OPTION VALUE="Germany">Германия
<OPTION VALUE="France"> Франция
</SELECT>
</TD>
<TD VALIGN="top" BGCOLOR="EEEEEE" WIDTH="50%">
<TEXTAREA COLS="25" ROWS="5" NAME="comments">
</TEXTAREA>
<BR>
<INPUT TYPE="radio" NAME="site" VALUE="first" CHECKED> Я впервые
на этом сайте
<BR>
<INPUT TYPE="radio" NAME="site" VALUE="not first" Я тут не в первый раз
</TD>
</TR>
<TR>
<TD COLSPAN="2" ALIGN="center" BGCOLOR="#999900">
<INPUT TYPE="submit" VALUE="Отправить">&nbsp;
<INPUT TYPE="reset" VALUE="Очистить">
</TD>
</TR> .
</TABLE>
</FORM>

```

</BODY>

</HTML>

Теперь сохраните код под любым именем, не забыв присвоить файлу расширение html или htm. Откройте файл в браузере и результатом должно послужить следующее:



▲ В данном примере были использованы следующие элементы пользовательских форм:

- текстовая строка;
- поле для ввода пароля;
- поле выбора локального файла для загрузки на Web-сервер;
- опция выбора нескольких вариантов данных;
- опция переключения между несколькими вариантами данных;
- текстовое поле;
- ниспадающее меню;
- список наименований;
- кнопка отправления данных;
- кнопка сброса данных.

Задание 8 - создать HTML форму для ввода информации на web-сайт.

Алгоритм задания:

1. Задать структуру HTML формы для ввода информации на web-сайт, определив необходимые поля и их типы. Так, например, для осуществления заказа в книжном магазине определяем форму со следующими полями:

- Автор (text)

- Название (text)
- Количество (text)
- Адрес доставки (text)
- Способ доставки (radio)
- Способ оплаты (select)
- Дополнительные пожелания (textarea)
- Ознакомлен с условиями доставки (checkbox)
- Отправить (submit)

2. Создать страницу form.html со структурой, определенной в лабораторной работе №3.

3. В блок основной информации поместить HTML форму, используя таблицу для выравнивания названий полей формы и ее элементов

The screenshot shows a web browser window with the address bar displaying a local file path. The page content includes a header 'Книжный магазин', a navigation menu with links like 'Главная', 'Новинки', 'Каталог книг', 'Форма заказа', and 'Контакты'. The main section is titled 'Форма заказа' and contains the following form elements:

- Text: 'Для заказа товара заполните поля следующей формы. Все поля обязательны для заполнения.'
- Form fields:
 - Автор:
 - Название:
 - Количество:
 - Адрес доставки:
 - Способ доставки: почтой, курьером
 - Способ оплаты:
 - Дополнительные пожелания:
 - Ознакомлен с условиями доставки:
- Submit button:

On the left side, there is a 'Новинки' section featuring a book cover for 'iCon Steve Jobs' by Jeffrey S. Young and William T. Stahn, with the text 'iCone: Steve Jobs' below it.

At the bottom of the page, there is an email address: **E-mail** info@bookshop.com.ua

Практическая работа №8

Создание сайта с внедрением мультимедийных и *карт-изображений*

Цель: формирование умений создания web-страниц, содержащих мультимедийных. и созданию карт-изображений.

Время выполнения: два академических часа.

Форма занятия: практическая работа.

Теоретические сведения:

Internet Explorer поддерживает тег **<MARQUEE>**, который позволяет создать так называемую бегущую строку, т. е. эффект прокручивания текста в заданном поле. Характеристики бегущей строки задаются следующими атрибутами:

- **WIDTH** — ширина поля бегущей строки в пикселах или процентах от ширины окна;

- **HEIGHT** — высота поля бегущей строки (в пикселах);

- **HSPACE, VSPACE** — интервалы по горизонтали и вертикали между текстом строки и краями ее поля (в пикселах);

- **ALIGN** — положение текста бегущей строки в ее поле; возможные аргументы:

- **TOP** (вверху);

- **BOTTOM** (внизу);

- **MIDDLE** (посередине);

- **DIRECTION** — определяет направление движения; возможные аргументы:

- **LEFT** (справа налево);

- **RIGHT** (слева направо);

- **BEHAVIOR** — характер движения строки; возможные аргументы:

- **SCROLL** — текст появляется от одного края и скрывается за другим;

- **SLIDE** — строка вытягивается из одного края поля и останавливается у другого края;

- **ALTERNATE** — задает переменное направление движения, от одного края к другому, а затем обратно;

- **LOOP** — количество повторений текста в бегущей строке, задаваемое числом; если необходимо «бесконечное» количество повторений, то следует задать аргумент в виде ключевого слова **INFINITY**.

Атрибут **LOOP** не влияет на поведение бегущей строки, если для атрибута **BEHAVIOR** заданы аргументы **ALTERNATE** или **SLIDE**;

- **SCROLLAMOUNT** — устанавливает длину (в пикселах) «прыжка» те кета за один такт; при большом значении этого параметра текст движется рывками, а при малом — замедленно;

- **SCROLLDELAY** — определяет величину паузы между тактами перемещения текста в миллисекундах;

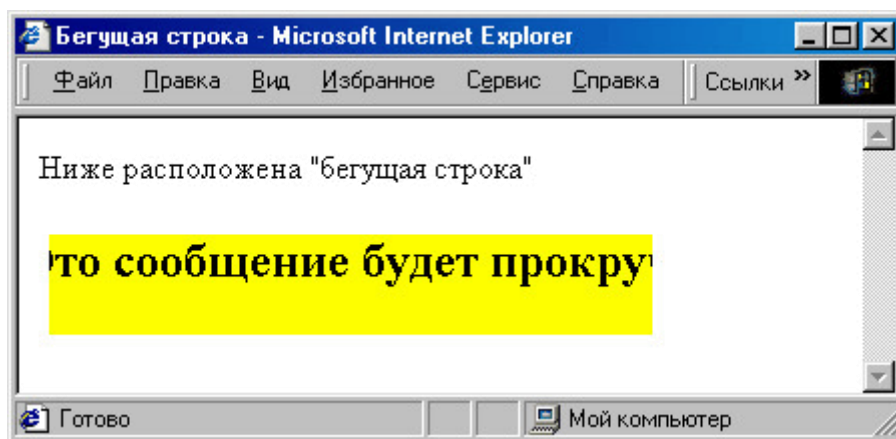
- **BGCOLOR** — устанавливает цвет поля бегущей строки, задаваемый шестнадцатеричным числом или именем.

Соотношения между длиной текста, размером шрифта и скоростью перемещения, при которых бегущая строка выглядит хорошо, подбираются опытным путем.

В следующем примере бегущая строка «бесконечно» прокручивается на желтом поле шириной 75% от ширины окна браузера.

```
<HTML> <HEAD><TITLE>Бегущая строка</TITLE> </HEAD>  
<BODY>
```

```
Ниже расположена "бегущая строка"<MARQUEE HEIGHT=50  
WIDTH=75% HSPACE=5 VSPACE=5 ALIGN=TOP BGCOLOR=YELLOW  
DIRECTION=LEFT LOOP=INFINITE BEHAVIOR=SCROLL SCROLLAMOUNT=5  
SCROLLDELAY=100> <H1> Это сообщение будет прокручиваться на экране  
справа налево</H1> </MARQUEE> </BODY> </HTML>
```



Задание № 1. Создание карт-изображений

— Создайте папку под именем Карты-изображения

– В созданной папке создайте папку и под именем 1, скопируйте в неё картинки, прежде полученные по сети от преподавателя

– Создайте HTML-документ и присвойте ему имя 3, введя следующий HTML-код:

```
<html>
<body bgcolor="#FFD8C0">
<script>
alert("Ну здесь нет ничего особенного! Продолжать?")
</script>
<hr size=4 noshade color=black>
<hr size=4 noshade color=black>
<h1><p align=center> <em> <strong> "Один"
</h1> </strong> </em> </p>
<p align="center">
<img src=1/2.bmp align=center border="none" width=362 height=434
usemap=#18>
</p>
<br>
<hr size=4 noshade color=black>
<hr size=4 noshade color=black>
</font>
</body>
</html>
```

Результатом должно послужить следующее:



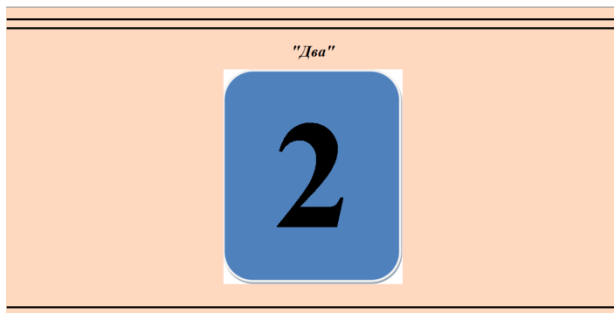
– Создайте HTML-документ и присвойте ему имя 4, введя следующий HTML-код:

```

<html>
<body bgcolor="#FFD8CO">
<hr size=4 noshade color=black>
<hr size=4 noshade color=black>
<h1><p align=center> <em> <strong> "Два"
  </h1> </strong> </em> </p>
<p align="center">
<img src=1/3.bmp align=center border="none" width=362 height=434
usemap=#18>
</p>
<br>
<hr size=4 noshade color=black>
<hr size=4 noshade color=black>
</font>
</body>
</html>

```

Результатом должно послужить следующее:



– Создайте HTML-документ и присвойте ему имя 5, введя следующий HTML-код:

```

<html>
<body bgcolor="#FFD8CO">
<hr size=4 noshade color=black>
<hr size=4 noshade color=black>
<h1><p align=center> <em> <strong> "Три"
  </h1> </strong> </em> </p>
<p align="center">

```

```

<img src=1/4.bmp align=center border="none" width=362 height=434
usemap=#18>
</p>
<br>
<hr size=4 noshade color=black>
<hr size=4 noshade color=black>
</font>
</body>
</html>

```

Результатом должно послужить следующее:



– Создайте HTML-документ и присвойте ему имя 6, введя следующий HTML-код:

```

<html>
<body bgcolor="#FFD8C0">
<hr size=4 noshade color=black>
<hr size=4 noshade color=black>
<h1><p align=center> <em> <strong> "Четыре"
</h1> </strong> </em> </p>
<p align="center">
<img src=1/5.bmp align=center border="none" width=362 height=434
usemap=#18>
</p>
<br>
<hr size=4 noshade color=black>
<hr size=4 noshade color=black>
</font>
</body>

```

</html>

Результатом должно послужить следующее:



– Создайте HTML-документ и присвойте ему имя 7, введя следующий HTML-код:

```
<html>
<body bgcolor="#FFD8CO">
<hr size=4 noshade color=black>
<hr size=4 noshade color=black>
<h1><p align=center> <em> <strong> "Пять"
</h1> </strong> </em> </p>
<p align="center">
<img src=1/6.bmp align=center border="none" width=362 height=434
usemap=#18>
</p>
<br>
<hr size=4 noshade color=black>
<hr size=4 noshade color=black>
</font>
</body>
</html>
```

Результатом должно послужить следующее:

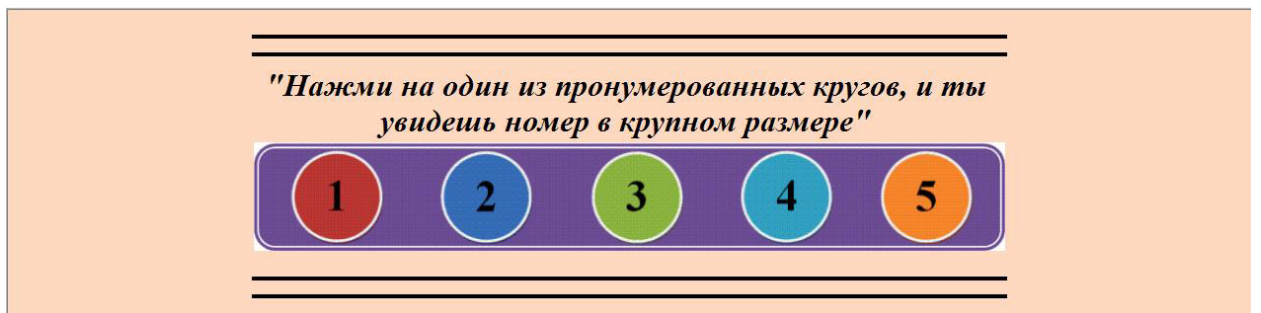


– Создайте HTML-документ и присвойте ему имя 2, введя следующий HTML-код:

```
<html>
<body bgcolor="#FFD8C0">
<script>
alert("Вы уверены, что хотите загрузки этой страницы?")
</script>
<table border="0"align="center" width="771">
<tr>
<td>
<hr size=4 noshade color=black>
<hr size=4 noshade color=black>
</td>
</tr>
<tr>
<td>
<h1><p align=center> <em> <strong> "Нажми на один из пронумерованных
кругов, и ты увидешь номер в крупном размере"
</h1> </strong> </em> </p>
</td>
</tr>
<tr>
<td>
<p align="center">
<img src=1/1.bmp align=center border="none" width=761 height=110
usemap=#18>
</p>
```

```
<map name=18>
<area shape=circle coords=84,54,44 href=3.html alt="Один">
<area shape=circle coords=235,54,44 href=4.html alt="Два">
<area shape=circle coords=388,54,44 href=5.html alt="Три">
<area shape=circle coords=539,54,44 href=6.html alt="Четыре">
<area shape=circle coords=681,54,44 href=7.html alt="Пять">
</map>
<hr size=4 noshade color=black>
<hr size=4 noshade color=black>
<tr>
<td>
</font>
<table>
</body>
</html>
```

Результатом должно послужить следующее:



Практическая работа №9

Форматирование web-страниц с использованием таблиц стилей

Цель: обучить созданию web-страниц, с использованием возможностей каскадных таблицей стилей для форматирования текста.

Время выполнения: четыре академических часа.

Форма занятия: практическая работа.

Теоретический сведения:

С помощью CSS можно определять стиль и вид текста. Аналогично тому, что используется тег **FONT**, задающий свойства шрифта, но стили обладают большими возможностями и позволяют сократить код HTML. **Свойства шрифта**

Свойство	Значение	NC	IE	Описание	Пример
font-family	имя шрифта	4+	4+	Задает список шрифтов	p {font-family: Arial, serif}
font-style	Normal	4+	4+	Нормальный шрифт	p {font-style: italic}
	Italic	4+	4+	Курсив	
	Oblique		4+	Наклонный шрифт	
font-variant	Normal Small-caps		4+ 4+	Капиталь (особые прописные буквы)	p {font-variant: small-caps}
font-weight	Normal	4+	4+	Нормальная жирность	p {font-weight: bold}
	Lighter	4+	4+	Светлое начертание	
	Bold	4W	4+		
	Bolder	4W	4+	Полужирный	
	100-900	4+	4+	Жирный 100-светлый шрифт, 900-самый жирный	
font-size	Размер шрифта			нормальный размер	font-size: normal
	Normal	4+	4+		font-size: 12pt
	Pt	4+	4+	пункты	font-size: 12px
	Px	4+	4+	пиксели	font-size: 120%
	%	4+	4+	проценты	

NC - Netscape Communicator, IE - Microsoft Internet Explorer; 4+ означает, что свойство поддерживается в браузере 4 версии и выше. 4W - работает только под Windows.

Замечание:

Когда размер шрифта задается абсолютными значениями, т.е. указывается конкретное значение шрифта в пунктах или пикселах, то изменить эту величину с помощью опции браузера *Вид | Размер шрифта* невозможно. Если шрифт установлен слишком мелким, то исправить этот недостаток читателю простыми средствами не представляется возможным. Поэтому лучше задавать размер шрифта в процентах (Пример 1).

Пример 1. Задание свойств шрифта с помощью CSS

```
<html> <style>
H1 { font-family: Arial, Helvetica, Verdana, sans-serif; font-size: 150%; font-weight:
```

```
light }
```

```
</style> <body>
```

```
<H1>Заголовок</H1> Обычный текст </body> </html>
```

Ниже приведен результат примера 1.

Заголовок

Обычный текст

В таблице Примера 2 даны некоторые параметры и результат их применения.

Пример 2. Результат использования различных параметров шрифтов

Пример	Пример	Пример	Пример	Пример
font-family: Verdana, sans-serif; font-size: 120%; font-weight: light	font-size: large; font-weight: bold	font-family: Arial, sans-serif; font-size: x-small; font-weight: bold	font-variant: small-caps	font-style: italic; font-weight: bold

Свойства текста

Кроме изменения параметров шрифтов, можно управлять и свойствами всего текста. Значения свойств приведены в таблице.

Свойство	Значение	NC	IE	Описание	Пример
line-height	Normal	4W	4+	Интерлиньяж (межстрочный интервал)	line-height: normal
	Множитель	4+	4+		line-height: 1.5
	Точно	4+	4+		line-height: 12px
	%	4+	4+		line-height: 120%
text-decoration	None	4+	4+	Убрать все оформление	text-decoration: none
	Underline	4+	4+	Подчеркивание	
	Overline		4W	Линия над текстом	
	line-through	4+	4+	Перечеркивание	
	Blink	4+			

				Мигание текста	
text-transform	None	4+	4W	Убрать все эффекты	text-transform: capitalize
	Capitalize	4+	4W	Начинать с прописных	
	Uppercase	4+	4W	ВСЕ ПРОПИСНЫЕ	
	Lowercase	4+	4W	все строчные	
text-align	Left	4+	4+	Выравнивание текста	text-align: justify выравнивание по ширине
	Right	4+	4+		
	Center	4+	4+		
	Justify	4+	4W		
text-indent	Точно	4+	4+	Отступ первой строки	text-indent: 15px;
	%	4+	4+		text-indent: 10%

NC - Netscape Communicator, **IE** - Microsoft Internet Explorer; **4+** означает, что свойство поддерживается в браузере 4 версии и выше. **4W** - работает только под Windows.

Ниже, в Примере 3 приведены некоторые параметры текста и результат их применения.

Пример 3. Результат использования различных параметров текста

Пример: и это все о нем	Пример: текст по центру	Пример: Это не ссылка, а просто текст	Пример: отступ первой строки	Пример: полуторный межстрочный интервал
text-transform: capitalize	text-align:center	text-decoration: underline	text-indent: 20px	line-height: 1.5

Границы и рамки

Спецификация CSS описывает несколько свойств, с помощью которых можно создавать границу вокруг различных элементов и управлять ее видом. Границы - одна из наиболее слабых сторон CSS, т.к. браузеры содержат большое количество ошибок и по разному интерпретируют параметры. Старшие версии браузеров отображают рамки вокруг элементов более корректно.

Свойство	Значение	NC	IE	Op	Описание	Пример
padding-top padding-right padding-bottom padding-left padding	Значение %	4+	4+	3+	Отступ от границы элемента до его содержимого	table {padding: 15px 15px}
border-top-width border-right-	Thin Medium Thick	4+	4+	3+	Ширина границы	P {border-top-width: 4px}

width border-bottom-width border-left-width border-width	Значение					
border-color	цвет	4+	4+	3+	Цвет границы	P {border-color: red}
border-style	None Dotted Dashed Solid Double Groove Ridge Inset Outset	4+ 6 6 4+ 4+ 4+ 4+ 4+ 4+	4+ 5.5+ 5.5+ 4+ 4+ 4+ 4+ 4+	3+	Стиль рамки	table {border-style: double}
border-top border-right border-bottom border-left	border-top-width border-style цвет	6	4+	3+	Определяет толщину, стиль и цвет каждой границы	table {border-top: solid 4px red; border-left: solid 4px blue}
border	см. выше	4+	4+	3+	Задаёт толщину, стиль и цвет рамки	table {border: solid 4px red}

NC - Netscape Communicator, IE - Microsoft Internet Explorer, Op - Opera; 4+ означает, что свойство поддерживается в браузере 4 версии и выше.

Типы рамок

Для управления видом рамки предоставляется восемь значений параметра border-style

Первые две - **dotted** и **dashed** стали поддерживаться браузерами Netscape и Internet Explorer только со старших версий. <p style="color: yellow;

background-color: deepskyblue; text-decoration: underline; text-transform: uppercase;

border: pink inset 25;

PADDING: 20; font-size: larger; line-height: 40px; text-align: center;"> ... </p>



Атрибут **style="..."** задает стилевое оформление абзаца Атрибут **color: yellow;** задает цвет текста

Атрибут **background-color: deepskyblue;** задает цвет фона для абзаца Атрибут **text-decoration: underline;** задает подчеркивание для текста Атрибут **text-transform: uppercase;** задает режим заглавных букв для текста

Атрибут **border: pink inset 25;** задает рамку вокруг абзаца, соответственно, розовую выпуклую толщиной 25 пикселей Атрибут **PADDING: 20;** задает Атрибут **font-size: larger;** задает размер шрифта АТРИБУТ **line-height: 40px;** задает межстрочный интервал АТРИБУТ **text-align: center;** задает выравнивание текста внутри абзаца по центру

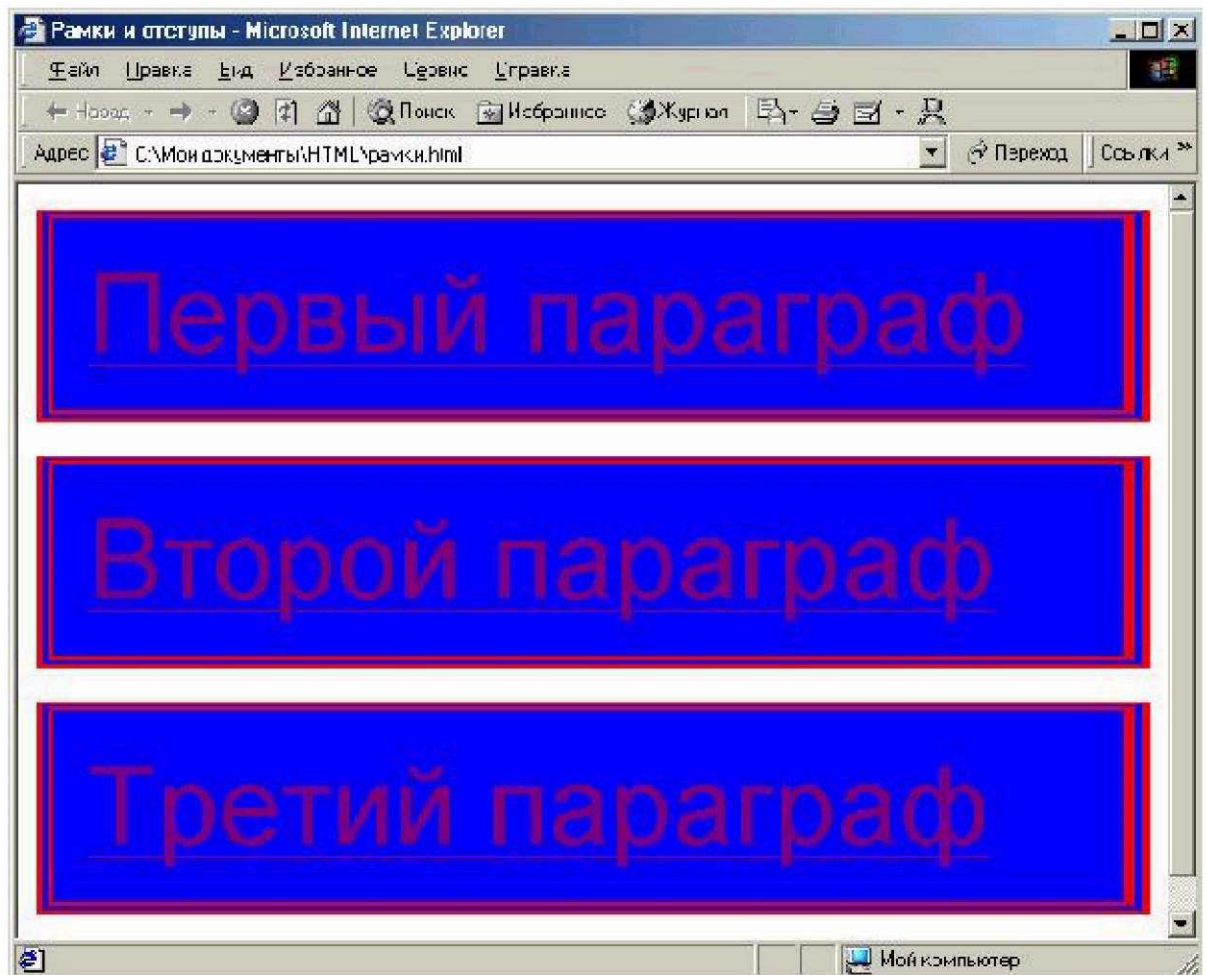
Задание 1

Создать в графическом редакторе Paint небольшой рисунок, который будет использоваться в качестве маркера списка. Создать маркированный список со своим маркером, используя стилевое свойство **LIST-STYLE-IMAGE: url(имя_рисунка)**.

Например, `<UL style="LIST-STYLE-IMAGE: url(star2.jpg)">`

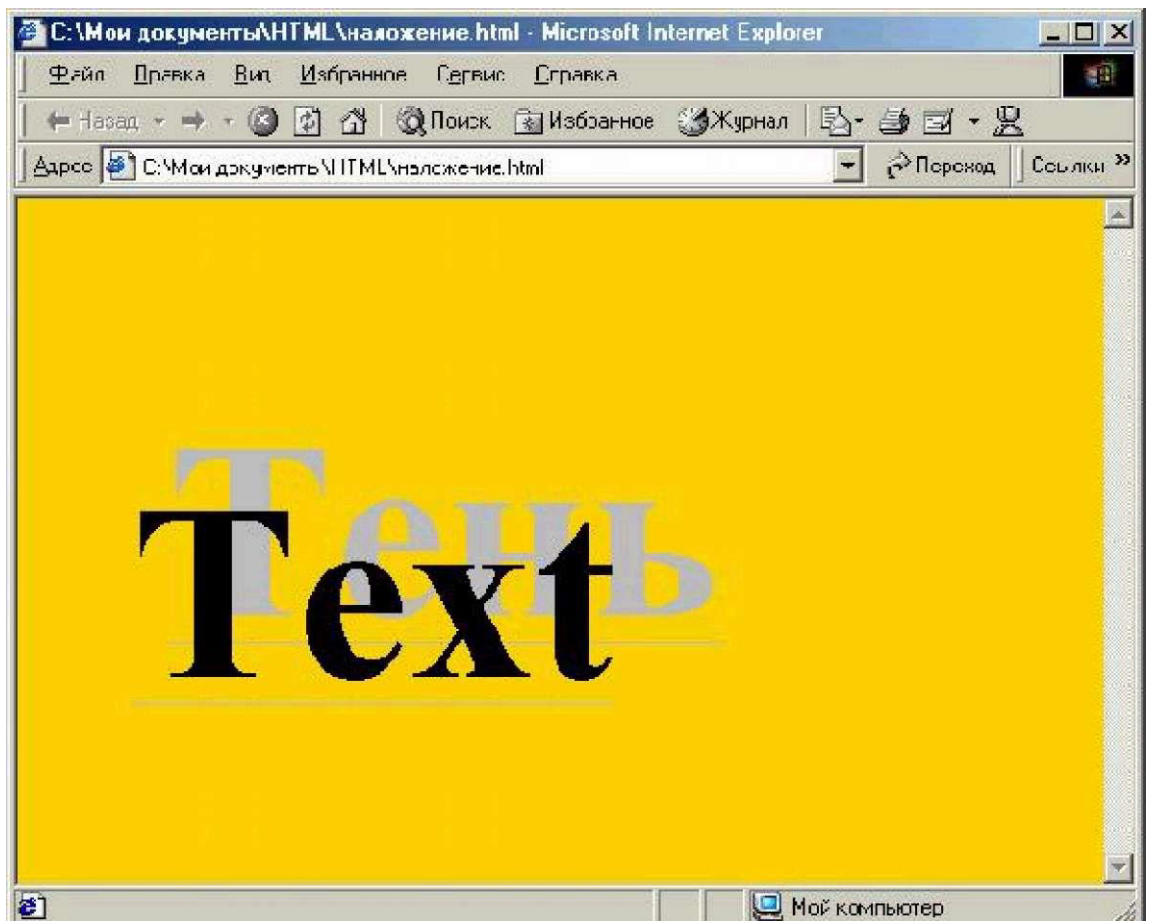
Задание 2

Используя свойства блоков текста, создать следующую страницу:



Задание 3

Значения отступов вокруг объектов можно указывать как положительные, так и отрицательные. Таким образом, можно использовать данную возможность для наложения одного слоя текста на другой. Например, можно создать текст с тенью, без использования трехмерного изображения. Создайте два стиля, которые



отличаются цветом и размером отступов вокруг них. Используя отрицательные значения отступов и подбирая нужное значение, можно добиться того, что верхний слой как бы напозаает на предыдущий. Используйте контейнер `<DIV>` и свойства `margin-top` и `margin-left`. В результате у вас должно получиться следующее:

Практическая работа №10

Создание динамического меню навигации при помощи правил CSS.

Цель: формирование умений применения элементов HTML (списки и гиперссылки), а также свойства CSS (модель отображения, псевдоклассы гиперссылок) для организации системы навигации на веб-странице.

Время выполнения: четыре академических часа.

Форма занятия: практическая работа.

Теоретический сведения:

Навигация

Меню — это неотъемлемая часть каждого сайта, ну или почти каждого. Меню (или навигация) позволяет быстро перемещаться по основным разделам сайта, что очень похоже на оглавление в книгах. Особенно сходство заметно на картах сайта (разновидность навигации).

Меню может быть:

Горизонтальным

Вертикальным

Многоуровневым (выпадающим или раскрывающимся)

Содержащим пиктограммы.

Меню может быть очень разнообразным и сверстать его можно очень по-разному: таблицами, блоками, просто ссылками и др. и большинство из них будут соответствовать принципам блочной (бестабличной) верстки. Но в профессиональных кругах почему-то считается, что навигация должна быть на основе неупорядоченных списков — теги `` и ``.

Почему в навигации следует использовать ul?

Обратимся лучше за помощью к авторитетам. W3C о списках:

Списки могут содержать:

не упорядоченный перечень

упорядоченный перечень

навигацию

определения

Как использовать списки для построения навигации?

Список в не обработанном виде может выглядеть пугающе не похожим на меню, который нарисовал дизайнер:

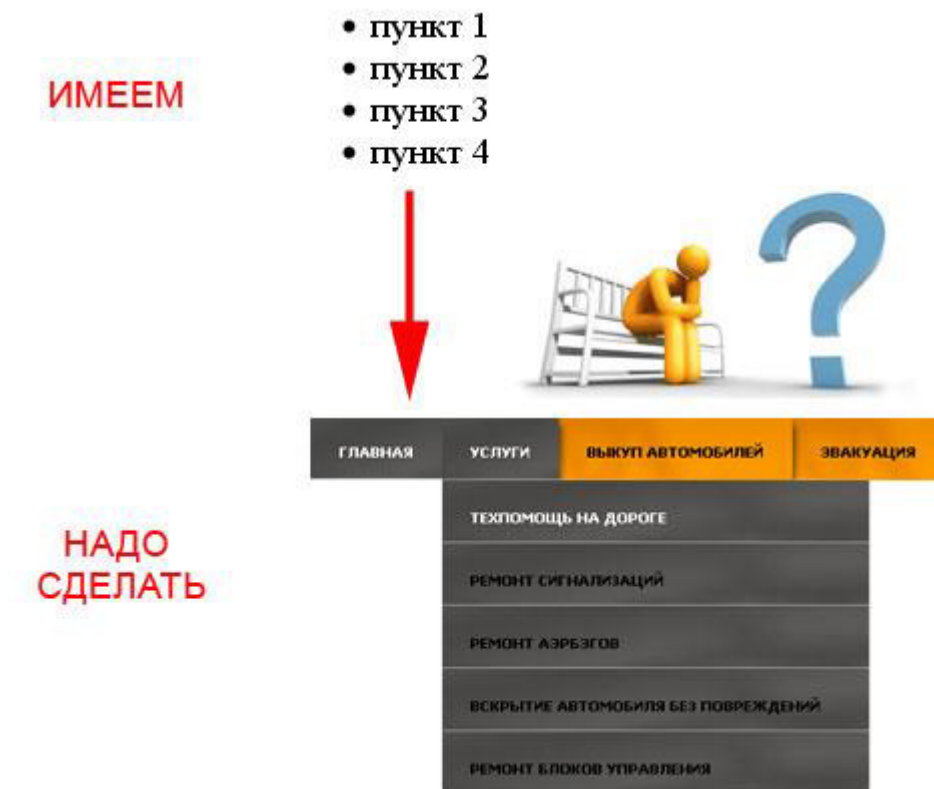


Рис. 5.1.

Создание меню на основе списков с помощью CSS

Отчаиваться не стоит, стоит учиться. CSS способно творить чудеса. Пойдем по пути от простого к сложному и реализуем несколько меню.

Простое горизонтальное меню

HTML:

```
<ul class="menu_nav">
  <li><a href="#">Главная</a></li>
  <li><a href="#">HTML и CSS приемы</a></li>
  <li><a href="#">Обучение</a></li>
  <li><a href="#">Справочники</a></li>
</ul>
```

CSS:

```

.menu_nav {
    list-style: none; /* прячем маркеры */
}
.menu_nav li {
    display: inline; /* выстраиваем элементы списка в один ряд */
    margin-right: 15px; /* делаем отступ чтобы пункты меню не
сливались */
}

```

Посмотреть пример. Напмню, тем, кто забыл, **list-style** задает вид маркера, а так же позволяет его скрыть.

По умолчанию элементы списка (****) - блочные элементы, а каждый такой элемент начинается с новой строки. Чтобы выстроить их в одну линию - меняем отображение элемента на строчный (**display: inline**) -это самый простой и предсказуемый способ выстроить элементы в один ряд.

Горизонтальное меню со ссылками в виде блоков

Теперь, используя тот же HTML-код усложним задачу:

```

.menu_nav {
    list-style: none; /* прячем маркеры */
}
.menu_nav li {
    float: left; /* выстраиваем элементы списка в один ряд */
    margin-right: 15px; /* делаем отступ чтобы пункты меню не сливались */
}
a {text-decoration: none;} /*Убираем подчечкивание ссылки*/
.menu_nav li a {
    display: block; /* меняем отображение на блок, чтобы иметь возможность
задавать внутренние отступы */
    padding: 3px 5px;
    background: #ff7f50;
}

```

```

    color: #000;
    position: relative; /* для IE6, чтобы ссылка была кликабельной по всей
своей площади */
}
.menu_nav li a:hover {
    background: #d2b48c;
    color: #fff;
}

```

Посмотреть пример. В данном примере выстраиваем меню в ряд используя свойство **float**, при этом не стоит забывать, что теперь следующие за меню () элементы, будут пытаться его обтекать. Это можно устранить используя **clearfix**, просто **clear** или любой другой способ обновления потока.

Выполнить задание!

Меню с картинками

HTML:

```

<ul class="menu_nav">
    <li class="delic"><a href="#">Delicious</a></li>
    <li class="face"><a href="#">Facebook</a></li>
    <li class="linked"><a href="#">Link</a></li>
    <li class="rss"><a href="#">RSS</a></li>
</ul>

```

CSS:

```

.menu_nav { list-style: none; /* прячем маркеры */
margin: 30px;
}
.menu_nav li {
    float: left; /* выстраиваем элементы списка в один ряд */
    margin-right: 15px; /* делаем отступ чтобы пункты меню не сливались */
    margin: 10px;
}

```

```

}
.menu_nav li a {
    display: block;
    text-align: center;
    padding-top: 128px; /* внутренний отступ чтобы текст был вне фона */
    color: #0000cd;
    background-position: center top; /* выравниваем фон вверх и по центру */
    background-repeat: no-repeat;
    width: 120px; /* размеры указываем чтобы картинки полностью
отобразились */
    height: 50px;
    margin: 10px;
}
.delic a { background-image: url(image/Furry-Cushion-Delicious_64.png); }
.face a { background-image: url(image/Furry-Cushion-Facebook_64.png); }
.linked a { background-image: url(image/Furry-Cushion-LinkedIn_64.png); }
.rss a { background-image: url(image/Furry-Cushion-RSS_64.png); }
.menu_nav li a:hover { color: #ff8c00; }

```

Посмотреть пример. Данную задачу (как и впрочем многие другие) можно реализовать разными способами. Прием, приведенный выше, на мой взгляд самый простой. Обратите внимание, что в меню кликабилен не только текст ссылки, но и сама картинка. И при этом используется минимум элементов.

Задание: Создать следующее меню с помощью CSS:



Главная

Html

○ **Основы**

- **Ссылки**
- **Картинки**
- **Списки**
- **Таблицы**

Photoshop

- **Основы**
- **Кнопки**
- **Логотипы**
- **Web-разметка**

CSS

- **Основы**
- **Ссылки**
- **Картинки**
- **Меню**
- **Верстка**

Javascript

- **Основы**
- **Операторы**
- **Функции**
- **Циклы**
- **Скрипты**

Практическая работа №11

Верстка web-страниц с использованием таблиц стилей.

Цель: Ознакомиться с различными способами верстки. Получить навыки работы в них.

Время выполнения: четыре академических часа.

Форма занятия: практическая работа.

Теоретический сведения:

.

Плоская верстка

Если Вы в качестве базовых элементов страницы не используете ни таблицы ни блоки, то это и есть плоская верстка.

Табличная верстка

Достоинства и недостатки табличной верстки:

Основные достоинства таблиц:	Недостатки таблиц:
Простое создание колонок "Резиновая" верстка "Склейка" изображений Однозначная трактовка браузерами	Долгая загрузка Громоздкий код Плохая индексация поисковиками Нет разделения содержимого и оформления Несоответствие стандартам Невозможность конвертации в мобильную версию

Пример табличной верстки

Вот что мы хотим получить:

ЛОГОТИП		
Меню		
menu1	menu1	menu1
Субменю		
<ul style="list-style-type: none"> • Ссылка • Ссылка • Ссылка 	контент-верх	суб-контент-верх
текст	контент-низ	суб-контент-низ
ПОДВАЛ		

Рис. 6.1. Пример табличной верстки



Впрочем, у Вас уже достаточно знаний, чтобы выполнить данный пример самостоятельно:

По макету с рис. 6.1. сверстать страницу табличной версткой.



Важно!

Применение табличной верстки Целесообразно в том случае, если

Высота колонок должна быть одинаковой

Макет должен занимать всю высоту окна браузера, независимо от объема информации

Нет времени на сложную верстку

Блочная верстка

Слой — это элемент веб-страницы, созданный с помощью тега `<div>`, к которому применяется стилевое оформление. Благодаря этому тегу HTML-код распадается на ряд четких наглядных блоков, за счет чего верстка слоями называется также блочной версткой. Код при этом получается более компактным, чем при табличной верстке, к тому же поисковые системы его лучше индексируют.

Сейчас я Вам продемонстрирую как разметить страницу с рис. 6.2.

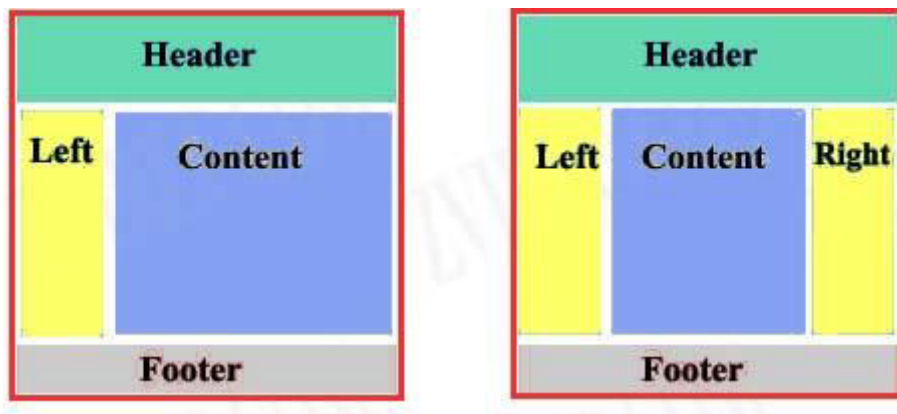


Рис. 6.2. Пример для

блочной разметки

На рис 6.2. представлены варианты для резиновой верстки с двумя и тремя колонками. Давайте выполним разметку двух колонок по шагам:

Шаг 1

Сначала, как обычно, выполним разметку HTML

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=windows-1251">
<title>Блочная верстка</title>
<link href="style_1.css" rel="stylesheet" type="text/css">
</head>
<body>
  <div id="container">

      <div id="header">
          <h1>Header</h1>
      </div>

      <div id="left">
          <h3>left Content</h3>
  
```



```

        </div>

        <div id="content">
            <h1>Main Content </h1>
            <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit.
            Praesent aliquam, justo convallis luctus rutrum, erat nulla
fermentum

            Integer turpis arcu, pellentesque eget, cursus et, fermentum ut,
sapien.

            Fusce metus mi, eleifend sollicitudin, molestie id, varius et,
nibh.

            Donec nec libero.</p>
            <h1>Content</h1>
            <p>Lorem ipsum dolor sit amet, consectetur adipiscing
elit.</p>

            <p>Quisque ornare risus quis ligula Phasellus tristique
            purus a augue condimentum adipiscing. Aenean sagittis.
            Etiam leo pede, rhoncus venenatis, tristique in, vulputate at,
odio.</p>

            </div>
            <div id="footer">
            <p><strong>Footer</strong></p>
            </div>
        </div>
    </body>
</html>

```

Обратите внимание на присутствие базового (главного) контейнера, который уже содержит все остальные. Наличие этого главного контейнера позволяет

регулировать общие параметры страницы, что значительно облегчает верстку слоями.

Результат получится вот такой:

Header

left Content

Main Content

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Praesent aliquam, justo convallis luctus rutrum, erat nulla fermentum Integer turpis arcu, pellentesque eget, cursus et, fermentum ut, sapien. Fusce metus mi, eleifend sollicitudin, molestie id, varius et, nibh. Donec nec libero.

Content

Lorem ipsum dolor sit amet, consectetur adipiscing elit.

Quisque ornare risus quis ligula. Phasellus tristique purus a augue condimentum adipiscing. Aenean sagittis. Etiam leo pede, rhoncus venenatis, tristique in, vulputate at, odio.

Footer

Рис. 6.3. Пример для блочной разметки. Шаг 1.

Оно и понятно, ведь стиль-то CSS у нас не написан. Итак, займемся стилями.

Шаг 2

```
/* CSS Document */
body, html {
margin:0px; /*Это хорошая практика обнулять поля и отступы, т.к.
различные браузеры их по разному воспринимают. */
padding:0px;
text-align:center; /*Выравниваем макет (содержимое элемента body
будет посередине) по центру в старых версиях браузеров */
}
#container{
margin:0 auto; /*выравниваем макет по центру в современных браузерах */
width:650px;
}
```

```

/*Здесь пишем стили для шапки сайта */
#header{
background-color:#64dab2;
}

/*Здесь пишем стили для левой колонки сайта */
#left{
background-color:#ffff66;
}

/*Здесь пишем стили для блока контента */
#content{
background-color:#83a0f3;
}
#content h1 {
margin:0px; /* Обнуляем отступы для заголовка первого уровня,
находящегося в блоке контента.*/
}

/*Здесь пишем стили для подвала сайта */
#footer{
background-color:#ccc;
}

```

После применения данного стиля style_1.css результат будет следующим:



Рис. 6.4. Пример для блочной разметки. Шаг 2. Первоначальное стилевое оформление

Теперь необходимо блоки Left и Content поставить рядом, как на [рис. 6.2](#). Меняем ширину блоку Left на 150px и устанавливаем для него обтекание:

```
#left{  
...  
width:150px; /*ширина колонки */  
float:left; /*обязательное выравнивание по левому краю,  
с включением обтекания*/  
}
```

Смотрим результат, получаем не совсем то, что хотели (*рис. 6.4. а*):

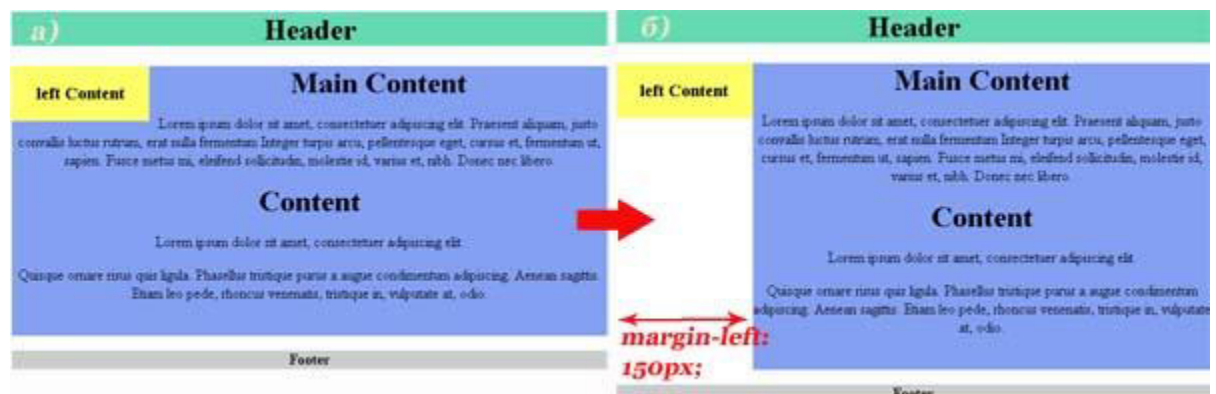


Рис. 6.4. Шаг 2. Обтекание блоков

Для того, чтобы из [рис. 6.4.а](#)) получить [рис. 6.4. б](#)) необходимо сдвинуть блок Content вправо на ширину блока Left:

```
#content{  
background-color:#83a0f3;  
margin-left:150px;  
}
```

То же самое можно сделать иначе, установив блоку Left `width:20%` и `float:left;`, а блоку Content `width:80%` и `float:right;` (см. пример в лекциях)

При этом не забудьте поставить для блока footer значение `clear:both;`, иначе при увеличении количества текста в предыдущих блоках может получиться седующий эффект:

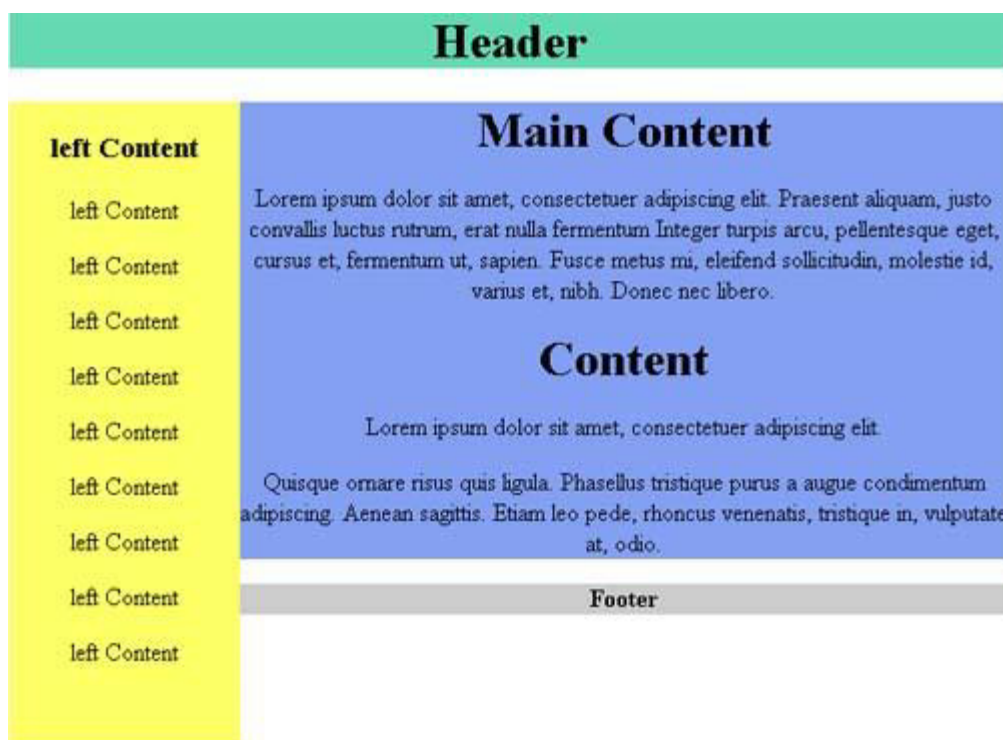


Рис. 6.5.

Необходима отмена обтекания блоков



Итак, Ваша задача

По макету с рис. 6.2 (б) разработать трехколонный блочный макет, используя в качестве базы двухколонный и растянуть содержимое на всю ширину страницы.

И последнее: Проверьте свой трехколоночный макет: хорошо ли он минимизируется? Не скачут ли блоки? Тем, кто забыл о свойстве `min-width` рекомендую установить его для блока Content и снова проверить минимизацию Вашего макета.



Важно!

При верстке слоями учитывайте блочную модель, и помните, что, когда Вы задаете ширину блока `width`, а затем добавляете ему границы `border` и отступы `margin, padding`, то к величине блока добавляются значения границ и отступов и блок расширяется:

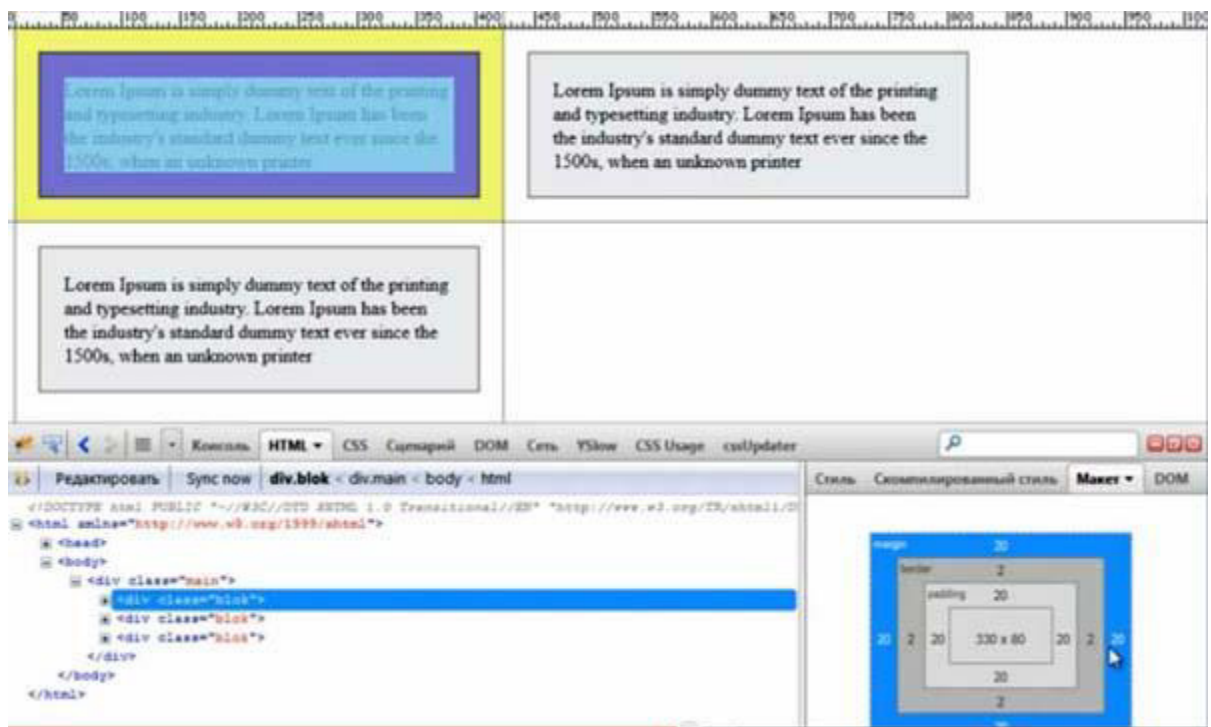


Рис. 6.6. Инспектирование блочной модели в FireBug.

Центрировать картинку в резиновом дизайне можно, например, так:

```
img {display: block; width: auto; height: auto; margin: 0 auto; }
```

Практическая работа №12

Размещение web-сайта в Интернете

Цель: формирование умений регистрации своего сайта в поисковых системах;
- размещения сайта на серверах в Интернете.

Время выполнения: три академических часа.

Форма занятия: практическая работа.

Задание

1. Найти сервер, позволяющий бесплатно разместить сайт, и зарегистрироваться на нем.
2. Разместить на сервере www.narod.ru ваш сайт.
3. Зарегистрировать ваш сайт в поисковых системах.

Алгоритм выполнения:

Задание №1

Найти сервер, позволяющий бесплатно разместить сайт

1. Для поиска серверов, позволяющих бесплатно разместить ваш сайт, запустите web-браузер, например, Internet Explorer, перейдите на web-сайт поисковой системы, например, www.aport.ru, и задайте образ поиска Бесплатный хостинг. По результатам поиска вы увидите, что таких серверов сотни.

Какой из них выбрать? При выборе хостинга (то есть места размещения вашего сайта) надо учесть несколько важных параметров:

- скорость и надежность работы сервера;
- способ загрузки на сервер: по протоколу HTTP, FTP или по адресу электронной почты;
- возможность размещения CGI, SSI и других сценариев, выполняемых на сервере;
- физическое расположение сервера - удаленность сервера оказывает сильное влияние на скорость;

- многие серверы очень любят вставлять свою рекламу в сайты.

Рассмотрим пример с регистрацией и размещением вашего сайта на сервере <http://www.narod.ru/> - одном из популярных серверов, бесплатно предоставляющих всем желающим неограниченное дисковое пространство для размещения сайтов, ftp-доступ и доменное имя вида name.narod.ru. Кроме этого, есть возможность завести себе электронный ящик с адресом name@narod.ru. Можно построить сайт по шаблонам, которых более 100, предоставляются шаблонные гостевые книги, форумы, чаты, дизайн которых можно полностью изменять. К ограничениям этого сервера относится запрет на использование CGI, SSI.

2. Загрузите сайт <http://www.narod.ru/> и, щелкните на ссылке Что здесь дают, познакомьтесь в бесплатными услугами на данном сервере. Чтобы воспользоваться этими услугами, необходимо зарегистрироваться, то есть заполнить анкету, поэтому после просмотра информации щелкните на кнопке Зарегистрироваться.

3. Для ознакомления с условиями использования бесплатной услуги щелкните на кнопке Пользовательское соглашение. Внимательно изучите текст соглашения, описание услуг, ваши обязательства.

4. После изучения соглашения закройте окно web-обозревателя с текстом соглашения и заполните бланк регистрационной анкеты.

Щелкнув на ссылке Вернуться в редактированию, введите новое имя (логин), введите пароль, отредактируйте записи во всех полях и щелкните на кнопке Зарегистрироваться.

5. На следующем шаге регистрации введите имя вашего сайта и щелкните на кнопке Сохранить.

После регистрации в окне web-обозревателя выводится подтверждение.

6. Для входа на сервер в качестве зарегистрированного пользователя и выбора имени вашего сайта на www.narod.ru введите в поле Логин в окне Вход

зарегистрированное имя, а в поле Пароль укажите пароль и щелкните на кнопке Войти.

На следующем шаге введите имя вашего сайта (можно оставить по умолчанию ваше зарегистрированное имя), установите флажок Подписаться на общие новости сайта или Подписаться на анонсы «Народной газеты» и щелкните на кнопке Сохранить.

7. После окончания регистрации сайта в окне web-обозревателя выводится подтверждение. Если нужно заменить настройки доступа, щелкните на ссылке Изменить настройки доступа, ознакомьтесь с различными вариантами настроек доступа, выберите подходящий, подтвердите изменения вводом пароля и щелкните на кнопке Сохранить.

8. Для изменения настроек почты щелкните на ссылке Настройки почты, измените настройки и щелкните на кнопке Сохранить изменения. Итак, регистрация завершена. Теперь для входа на сервер www.narod.ru в качестве зарегистрированного пользователя вы можете выбрать в окне Вход варианты: Мой сайт, Моя мастерская, Моя почта и т.п.