

МИНИСТЕРСТВО НАУКИ И ОБРАЗОВАНИЯ РЕСПУБЛИКИ КАЗАХСТАН
КАСПИЙСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ТЕХНОЛОГИИ И
ИНЖИНИРИНГА ИМ.Ш.ЕСЕНОВА

ИНСТИТУТ МОРСКИХ ТЕХНОЛОГИИ

Кафедра «Вычислительная техника и программное обеспечение»

Балтабекова Ж.Т.

ИНТЕРНЕТ-ТЕХНОЛОГИИ

**Методические указания по выполнению лабораторных работ
для студентов специальности
050704 – «Вычислительная техника и программное обеспечение»**

Актау 2011г.

УДК 681.327(027)

СОСТАВИТЕЛЬ: Балтабекова Ж.Т. – Интернет - технологии. Методические указания по выполнению лабораторных работ для студентов специальности 050704 «Вычислительная техника и программное обеспечение» - Актау: КГУТиИ, 2011г., 52с.

РЕЦЕНЗЕНТ: к.ф.-м.н., Утебаев М.Н.

В данном методическом указании содержатся основные вопросы web-программирования на языках HTML, CSS, PHP. В частности, лабораторные работы рассматривают создание веб-страниц с использованием форматирования, редактирования текста, вставки таблиц, графических изображений, фонового рисунка, с нумерованными и маркированными списками, делить окно на фреймы, создавать гиперссылки и т.д. Также можно научиться написать на языке PHP программы для создания простейших графических изображений, использование математических функции и переменных.

Печатается по решению Учебно – методического совета Каспийского государственного университета технологии и инжиниринга им.Ш.Е. Есенова

Содержание

Введение.....	4
Лабораторная работа №1. «Создание простой страницы и отображение в браузере»	5
Лабораторная работа №2. «Способы форматирования HTML-документов.....	6
Лабораторная работа №3. «Списки (нумерованные и маркированные)».....	10
Лабораторная работа №4. «Работа с графическим изображением».....	13
Лабораторная работа №5. «Создание гиперссылок (ссылок)».....	17
Лабораторная работа №6. «Создание таблиц».....	20
Лабораторная работа №7. «Навигационные карты ссылок (Image map)»...	26
Лабораторная работа №8. «Фреймы».....	29
Лабораторная работа №9. «Создание форм».....	33
Лабораторная работа №10. «Создание кнопки очистки».....	37
Лабораторная работа №11 «Свойства шрифта, текста. Границы в CSS»...	38
Лабораторная работа №12 «Курсоры, полоса прокрутки в CSS».....	42
Лабораторная работа № 13 «Пишем первую программу(скрипт) на PHP».....	45

Лабораторная работа № 14 «Создание простого изображения на PHP».....	47
Лабораторная работа № 15 «Обрамление блока PHP-команд тегами. Функции».....	48
Список литературы.....	52

Введение

HTML - это текстовый вид файла, в котором то, что на страничках, перемежается невидимым для просмотра из [браузера](#) кодом. HTML - это не язык программирования, - он служит лишь для разметки странички, придания определенного вида тому или иному элементу, будь то таблица, текст или картинки.

Осуществляется это путем написания тегов и присвоения каждому элементу своих параметров, которые распознает браузер. Параметры эти могут быть заданы как для одного, так и для группы или типа элементов. Тип элементов может быть таким: таблицы, ячейки, ссылки, текст и т.п.

Параметры отображения элементов задаются при помощи [тегов*](#), в которых и задается желаемый вид того или иного элемента нашей странички. Используя DHTML создаются приложения, для которых ранее нужно было использовать JavaScript или CGI. Это защита кода. Код встроенный в HTML страницу можно посмотреть, использование VB эту проблему снимает, весь код находится в DLL. Вот так будет выглядеть собранный проект.

Динамический HTML (Dynamic HTML, или DHTML) - это развитие традиционного языка HTML, позволяющее создавать интерактивные, "живые" HTML-документы. Стремлению сделать Web-страницу динамичной

и интерактивной, то есть легко изменяющейся, управляемой пользователем, столько же лет, сколько и самому WWW.

Один из методов реализации идеи динамичности заставил разработчиков "дополнить" HTML и встроить в браузеры средства обработки специальных языков программирования, таких как JavaScript, JScript и VBScript. Сами сценарии (скрипты), т. е. программы, написанные на одном из этих языков, можно размещать вместе с основным содержанием и разметкой HTML-документа. Программы на этих языках перемещаются по сети Интернет в виде исходных текстов, а компилируются и выполняются после загрузки в конкретный браузер.

Лабораторная работа №1 по HTML

Тема: Создание простой страницы и отображение в браузере.

Цель работы: научиться создавать веб – страницы

Для того, чтобы отобразить HTML – документ, браузер принимает его из сети и выполняет команды встреченные в тегах. Сами теги не отображаются в окне браузера, представляется только результат обработки HTML – документа.

HTML – документ может начинаться с тега **<!DOCTYPE>**, указывающего на версию языка HTML, с помощью которого написан данный документ.

Например, указание на 4-ю версию языка выглядит таким образом:

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN">

1. Создание Web – страницы.

1) Загрузите текстовый редактор *WordPad* или *Блокнот*.

- 2) Установите режим *Формат – Перенос по словам*, чтобы текст страницы не выходил за пределы текущего окна.
- 3) Скопируйте исходный текст (без рамки) из лабораторной работы.

**Моя первая страница.
Брачное объявление**

Как Апполон, красив и статен,
Как ледокол «Ермак», силен,
Как герцог Букенгемский знатен,
Как десять ЭВМ, умен,
Как бой курантов, пунктуален,
Как меч Фемиды, справедлив,

Как Казанова, сексуален,
Как муравей, трудолюбив,
Как наркобизнес, обеспечен
И почитаем, как генсек,
Как В. Ульянов, человекен –
Таков я вкратце, человек.

Мне в женщине не надо много:
Лишь только б вышла за такого.

А. Костюшкин

4) Введите `<HTML>` непосредственно над текстом и `</HTML>` непосредственно под текстом страницы.

(Эти теги задают начало и конец страницы.)

5) Введите `<HEAD>` под дескриптором `<HTML>`

6) Дважды нажмите *Enter*

7) Наберите `</HEAD>`

(Эти теги задают служебную часть страницы, не отображаемую в окне браузера.)

8) Введите `<TITLE>` непосредственно под `<HEAD>`

9) Наберите текст названия (заголовка) Web – страницы (*Брачные объявления*)

10) Введите `</TITLE>` в конце заголовка.

11) Наберите `<BODY>` непосредственно под тегом `</HEAD>`

12) Введите `</BODY>` непосредственно над дескриптором `<HTML>`

(В этих тегах задается основной текст страницы). Сохраните учебный документ под именем *index.html* (или *index.htm*).

13) Обратите внимание на расстановку тегов в странице и запомните ее.

14) Откройте созданный файл с помощью браузера. Обратите внимание на разницу между исходным файлом и отображением в браузере.

15) Перейдите в окно редактора HTML (дальнейшие изменения в тексте HTML – страницы отобразятся в браузере по команде *обновить* – клавиша *F5*)

Вставьте в окно документа Блокнот:

2. Сохранение начального форматирования текста, пробелов и пустых строк.

- 1) Введите в тексте HTML – файла *index.htm* непосредственно под дескриптором `<BODY>` тег *сохранения форматирования* `<PRE>`
- 2) Введите непосредственно над дескриптором `</BODY>` тег `</PRE>`
- 3) Сохраните, просмотрите и проанализируйте результат в браузере.

Лабораторная работа №2 по HTML

Тема: Способы форматирования HTML - документов.

Цель работы: изучить теги форматирования.

Каждое действие желательно просматривать в браузере, чтобы лучше понять работу тегов и атрибутов HTML

1. Создание заголовков разных уровней.

Тег `<H?>` задает полужирным шрифтом заголовки различных размеров, размещая их по центру. Вместо знака ? проставляется одно из значений размеров заголовка от 1 до 7.

ЗАДАНИЕ 1:

1. Откройте ранее сохраненный файл *index.html*

2. Удалите теги сохранения форматирования <PRE> и </PRE> из текста.
3. Введите <H2> и </H2> непосредственно перед и после первого заголовка соответственно.
4. Введите <H3> и </H3> непосредственно перед и после второго заголовка.
5. Сохраните и проанализируйте полученный результат в браузере.
6. Уберите тег </H3> из текста документа. Как это отразилось в окне браузера.
7. Восстановите тег и сохраните документ.

2. Создание абзаца (параграфа). Переход на новую строку.

Тег <P> - задает разбивку текста на абзацы (параграфы). Имеет парный тег </P>, который задается при необходимости (каждый тег <p> добавляет пустую строку между абзацами).

**Тег
 - задает разбивку текста на отдельные строки.**

ЗАДАНИЕ 2:

- 1) Введите теги <P> и </P> в начале и в конце (желательно) каждого куплета стихотворения и автора.
- 2) Внутри куплетов перед каждой строчкой (кроме первых) введите тег
 .
- 3) Результат просмотрите в браузере.
- 4) Вставьте дополнительные пробелы между словами в заголовках (в первом – по одному, во втором – по три) командой ** (дополнительный пробел)**. Результат просмотрите в браузере.

3. Вертикальное и горизонтальное форматирование текста.

Атрибут ALIGN=? в теле тега <P> - задает вертикальное размещение текста (по умолчанию текст форматируется по левому краю).

Вместо знака вопроса проставляется одно из значений размещения: LEFT (по левому краю), CENTER (по центру), RIGHT (по правому краю), JUSTIFY (по ширине).

Пример:

```
<p align=right>Здравствуй, Дед мороз! </p>
```

Теги <CENTER>, </CENTER> - задают горизонтальное выравнивание текста по центру (то же, что и команда <p align=center>)

ЗАДАНИЕ 3:

- 1) Задайте горизонтальное выравнивание всего текста страницы.
- 2) Оставьте первые два куплета форматированными по левому краю страницы.
- 3) Разместите автора по правому краю страницы.

4) Сохраните и просмотрите результат в браузере.

4. Размеры полей страницы.

LEFTMARGIN = ?
RIGHTMARGIN = ?
TOPMARGIN = ?
BOTTOMMARGIN = ?

Атрибуты, использующиеся в теле тега <BODY>, задают размеры левого, правого, верхнего и нижнего поля страницы соответственно. Вместо знака вопроса проставляется размер полей страницы в пикселах

Пример:

<BODY TOPMARGIN =100 BOTTOMMARGIN =100> - установлены одновременно верхнее и нижнее поля страницы

Размером в 100 пикселей

ЗАДАНИЕ 4:

- 1) Изменяя размеры полей страницы, установите такие значения, чтобы она выглядела сбалансированной (перед просмотром эффектов разверните окно браузера на весь экран)
- 2) Сохраните и просмотрите результат в браузере.

5. Смысловое выделение текста.

, - выделяет фрагмент, взятый в исходные парные теги полужирным шрифтом

<I>, </I> - выделяет фрагмент курсивным шрифтом

<U>, </U> - подчеркнутый шрифт

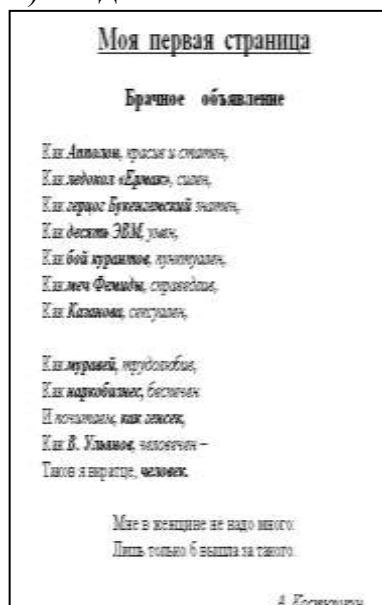
<STRIKE>, </STRIKE> - зачеркнутый шрифт

Например,

<P> <CENTER><U> Это полужирный зачеркнутый шрифт, имеющий горизонтальное выравнивание. </U></CENTER></P>

ЗАДАНИЕ 5:

- 1) Выделите текст в вашей страничке по примеру ниже.



Примечание:

Для выделения шрифта могут использоваться и другие теги:

 - полужирный или курсивный, для выделения особенно важных частей текста

 - курсив

- текст, для последующего удаления (зачеркнутый)

<INS> - придание смыслового оттенка тексту (курсив, полужирный)

6. Надстрочное и подстрочное начертание шрифта.

[,] - теги надстрочных символов

_, - теги подстрочных символов

Например,

<P> Это пример ^{над}строчных и _{под}строчных индексов</P>

Xⁿ+ Yⁿ= Zⁿ

ЗАДАНИЕ 6:

1) Добавьте в тексте документа после автора текущую дату.

2) Задайте ее верхним индексом.

3) Результат просмотрите в браузере.

Это пример ^{над}строчных и _{под}строчных символов
Xⁿ+Yⁿ=Zⁿ

7. Выбор типа шрифт и его размер

<H6> Сетевые адаптеры
<INS>\$45</INS>

, - задается перед блоком текста, шрифт

которого требуется изменить. Вместо знака ? поставьте одно или несколько названий или типов шрифта.

Примеры:

<P>Это пример одного из шрифтов</P>

<P>

Если первый из выбранных шрифтов не поддерживается браузером, то берется следующий по списку шрифт и т.д.</P>

<P>Это примеры типов шрифта</P>

<BASEFONT SIZE=?> - тег задает размер текста всей страницы, вместо знака ? проставляется одно из значений размера от 1 до 7

, - теги размера шрифта отдельного фрагмента

Пример 1:

<BODY>

<H3> Фрукты и овощи, Inc.</H3>

<BASEFONT SIZE =4>

<P>для всего текста задается размер шрифта, равный 4 </P>

</BODY>

Пример 2:

<BODY>

<P>Это шрифт размера 1</P>

<P>Увеличение на 2 единицы исходного размера текста</P>

<P>Уменьшение на 7 единиц исходного размера текста</P>

теги <SMALL>, </SMALL> и <BIG>, </BIG> соответственно уменьшение или увеличение исходного размера шрифта

Например:

<P>приходите к нам в субботу на <BIG> большую ярмарку!</BIG><SMALL>Дождя не предвидится.</SMALL></P>

ЗАДАНИЕ 7:

- 1) Задайте размер, равный 2 для текста всей страницы.
- 2) Задайте тип шрифта для всего текста «Bookman Old Style»; тип «Arial» для первого заголовка и тип «Courier New» для второго заголовка
- 3) Установите буквицу для всех первых букв стиха (кроме последних двух строчек) по примеру:

`Ж`или – были дед и баба, и была у них курочка Ряба.

- 4) Увеличьте размер шрифта для последних двух строчек при помощи тега `<big>` и задайте тип шрифта “Lucida Console”
- 5) Уменьшите размер шрифта для автора на 1 при помощи команды ``

8. Цвет фона и шрифта.

атрибут `BGCOLOR=?` в теле дескриптора `<BODY>` - задает цвет фона страницы, вместо знака вопроса проставляется код или название цвета (см. распечатку).

Атрибут `TEXT=?` в теле тега `<BODY>` - цвет всего текста страницы, вместо знака ? поставьте значение или код цвета.

`` - цвет отдельного фрагмента текста

Пример:

`<BODY BGCOLOR = blue Text=black>`, черные символы на голубом фоне`<P>` Этот пример задания цвета фрагмента текста с помощью кода`</P>`

`<P>` Этот пример задания цвета фрагмента текста с помощью названия`</P>`

ЗАДАНИЕ 8:

- 1) Подберите цвет фона вашей странички.
- 2) Задайте, исходя из цветовой гаммы фона, цвет заголовков и основного текста.

9. Вставка примечаний.

Примечания не отображаются в браузере, позволяют вносить в текст HTML – документа закладки, а также отменять действия команд без их удаления из документа.

`<!--текст комментария-->`

ЗАДАНИЕ 9:

- 1) Вставьте в конце вашей странички следующий комментарий: «Я еще вернусь к доработке этого документа»
- 2) Уберите любую строчку вашего документа с помощью комментария.
- 3) Результаты просмотрите в браузере.
- 4) Восстановите скрытую с помощью комментария строку.

Контрольные вопросы:

1. Какие теги используются для создания HTML-документа?
2. Какие теги используются для форматирования текста?
3. Как сохранить HTML-документ?

Лабораторная работа №3 по HTML

Тема: Списки (нумерованные и маркированные).

Цель работы: изучить теги списков

1) Нумерованные

, - обозначение начала и конца списка

, - разбиение на отдельные элементы списка

атрибут TYPE=? – стиль списка, вместо знака ? проставьте одно из значений таблицы (по умолчанию type=1).

Значение для TYPE	Пример обозначения списка
A	A, B, C,..
a	a, b, c, ...
I	I, II, III,...
i	i, ii, iii,...
1	1, 2, 3, ...

Пример обычного списка:

```

<BODY>
<H2> Оглавление</H2>
<OL>
<LI> Глава 1. На старт...</LI>
<LI> Глава 2. Основы Web – дизайна.
</LI>
<LI> Глава 3. Форматирование текста.
</LI>
...
</OL>

```

Атрибут START=? в теле тега - определяет начальный элемент списка, ≠1

Атрибут VALUE=? В теле тега или – назначает любому элементу списка требуемый номер

ЗАДАНИЕ 1:

- 1) Создайте HTML – документ под названием **spisky.html**, содержащий следующий текст (без тегов абзаца, переноса строк и форматирования):

Ивановы
Петровы
Сидоровы
Ольшанские
Кочетовы

- 2) Результат просмотрите в браузере.
- 3) Введите тег непосредственно над текстом и тег непосредственно под текстом.
- 4) Каждую фамилию (элемент списка) предварите тегом (закрывающий тег необязателен).

- 5) Сохраните и просмотрите результат в браузере.
- 6) Задайте атрибут **START=4** для фамилии **Ивановы**.
- 7) Задайте атрибут **VALUE=10** для фамилии **Ольшанские**.
- 8) Просмотрите результат в браузере.
- 9) Самостоятельно переопределите номера элементов данного списка.
- 10) Задайте начало нового списка непосредственно после слова **Ивановы** тегом **<ol type=a>**.
- 11) Задайте новые элементы списка при помощи тега ****:
Иванов Сергей
Иванова Лидия
Иванов Алексей
- 12) Результат просмотрите в браузере.
- 13) Задайте тег конца списка **** непосредственно после слова **Алексей**.
- 14) Переопределите тип значка списка для **Ивановой Лидии** на **A**
- 15) Результат просмотрите в браузере.
- 16) Самостоятельно достройте список для фамилии **Ольшанские** с типом списка **I**.

2) Маркированные списки

****, **** - начало и конец списка

****, **** - отдельные элементы списка

Атрибут **TYPE** в теле тега **** – значок списка (по умолчанию **type=disc**)

TYPE	значок
circle	○
disc	●
square	□

ЗАДАНИЕ 2:

- 1) В исходном документе **spisky.html** создайте еще один список (без форматирования и разбиения на строки):
Яблоки
Груши
Гранат
Апельсины
- 2) Просмотрите результат в браузере.
- 3) Введите тег **** непосредственно над словом **Яблоки** и тег **** непосредственно под словом **Апельсины**.
- 4) Определите элементы маркированного списка при помощи тега ****
- 5) Результат просмотреть в браузере
- 6) Задайте тег нового вложенного нумерованного списка **** непосредственно после слова **Яблоки**.
- 7) Введите элементы списка (тег ****):

**Ростовские
Московские
С Кавказа**

- 8) Закройте список тегом ``
- 9) Результат просмотрите в браузере.
- 10) Задайте новый вложенный маркированный список (тег `<ul type=circle>`) после слова **Ростовские**.
- 11) Запишите элементы этого списка (тег ``):
Зимние
Летние
Осенние
Весенние
- 12) Закройте список.
- 13) Просмотрите и проанализируйте результат в браузере.
- 14) Самостоятельно придумайте трехуровневый список со значками различных типов.

3) Списки определений.

<DL> - начало списка определений

<DT>, **<DD>** - начало текста термина и поясняющей его статьи.

Задание 3:

- 1) Скопируйте исходный код в конец тела страницы (перед тегом `</body>`):

```
<DL>
<DT>HTML
<DD>Термин HTML (HyperText Markup Language)
означает 'язык маркировки гипертекстов'.
Первую версию HTML разработал сотрудник
Европейской лаборатории физики элементарных частиц Тим Бернерс-
Ли.
<DT>HTML-документ
<DD>Текстовый файл с расширением *.htm (Unix-системы могут
содержать файлы с расширением *.html).
</DL>
```

- 2) Сохраните и просмотрите результат в браузере.
- 3) Самостоятельно создайте свои два элемента списка определений.

Контрольные вопросы:

1. Какие теги используют для создания нумерованного списка?
2. Какие теги используют для создания маркированного списка?
3. Какие теги используют для создания списка определений?

**Лабораторная работа №4 по HTML
Тема: Работа с графическим изображением.**

Цель работы: работа с графическими изображениями

1. Вставка графического изображения

**Тег ** - вместо знака ? запишите название и путь к графическому файлу (кавычки можно не прописывать)

Примечание:

Для удобства разместите картинку в ту же папку, что и документ-HTML, тогда путаницы не будет.

Названия каталогов (папок) не должно включать пробелы!

Например:

- 1) `` - картинка лежит в той же папке, в которой лежит HTML - документ. Вместо my.jpg мы можем подставить имя любой картинки с расширением .gif, .jpeg и .png.
- 3) `` - картинка лежит в поддиректории MY
- 4) `` - картинка лежит на уровень выше, а документ - HTML находится в поддиректории
- 5) `` - картинка лежит на другом сайте

- 2) `<p>
<center></center>` - в центре по горизонтали, на новой строке

Атрибут Alt позволяет вставлять текст вместо картинки при невозможности ее просмотреть.

Например:

``

ЗАДАНИЕ 1:

1. Создайте на своем диске папки S:\Html\Images
2. Занесите в папку IMAGES любой понравившийся вам файл картинки
3. Скопируйте текст в рамке в Блокнот и сохраните как lab4.html в папке Html

```
<html>
<head>
<title> Мои странички </title>
</head>
<body text=green bgcolor=#aaaaaa>

<center>
<H3>Здравствуйте, это моя вторая страница.</H3>
<br>
<font color=#CC0000> Добро пожаловать!</font></center>
<p align=justify>
```

`` Я совсем недавно начал знакомиться с виртуальной жизнью, но мне по давней традиции тоже захотелось создать свою домашнюю страничку для моих новых виртуальных друзей и знакомых, чтобы они могли посмотреть мои фотографии, почитать обо мне, черкнуть пару строчек в мою гостевую книгу. `

` А может и просто случайный посетитель вдруг захочет познакомиться со мной, и у меня появится `` еще один виртуальный друг? ``

`

` На фотографии изображен(а) я. Качество картинка не очень хорошее, к сожалению, поэтому она не четкая и разглядеть черты моего лица немного проблематично. Но в целом заметно, что я вполне ничего. `

` Если ты так тоже думаешь, то давай как-нибудь встретимся, поболтаем, чайку попьем в кафешке? Кто знает, может быть мы и в реальной жизни станем друзьями.

`</p>`
`</body>`
`</html>`

4. Вместо знаков ? запишите путь к картинке и ее название
5. Снабдите картинку поясняющим текстом, если ее невозможно будет считать.
6. Посмотрите, что у вас получилось.

2. Оформление графического изображения.

Атрибут **BORDER =?** в теле тега `` задает рамку вокруг изображения с толщиной, выраженной в **пикселах (точках)**. Вместо знака вопроса задается значение толщины. Чтобы убрать рамку, необходимо задать толщину равной 0.

ЗАДАНИЕ 2:

Добавьте рамку вокруг вашего изображения произвольной толщины.

3. Размещение текста относительно графического изображения

3.1 Чтобы текст располагался весь рядом с картинкой, а не только одна его строчка, необходимо задать соответствующий параметр выравнивания, например:

атрибут **ALIGN=?** в теле тега `` - задает размещение картинки по отношению к тексту, где знак вопроса заменяется одним из значений: **left(слева от текста)**, **right(справа)**, **bottom(внизу от картинки)**, **top(вверху)**, **middle(посередине)**

Пример:

```

```

ЗАДАНИЕ 3:

Разместите текст странички справа от картинки.

3.2. *Дополнительные атрибуты размещения текста и изображения.*

1) Размеры изображения:

Атрибуты Width =? и Height =? в теле дескриптора **** - задает ширину или (и) высоту самой картинки (в пикселях и в % - по отношению к размеру окна браузера).

2) Расстояние от рисунка до текста:

Атрибуты Vspace=? и Hspace=? в теле дескриптора **** - задает расстояние (по вертикали или (и) по горизонтали) между текстом и рисунком в пикселях

ЗАДАНИЕ 4:

1. Измените размеры вашей картинки в сторону уменьшения.
2. Измените расстояние от текста до изображения, задав значения атрибутов **HSPACE=30** и **VSPACE=5**
3. Добавьте вторую картинку или фотографию в ваш документ с шириной, равной 130 пикселям, размещенной справа от текста странички.

2. *Добавление графического фона.*

Атрибут BACKGROUND =""? в теле тега **<BODY>**, вместо знака ? задайте путь к фону и его название.

Например:

```
<body text=#336699 bgcolor=#000000 background="ваш_фон.gif">
```

ЗАДАНИЕ 5:

1. Создайте папку HTML\Фоны на вашем сетевом диске.
2. Скопируйте один из понравившихся вам фонов в эту папку.
3. Добавьте параметры фона в HTML – документ и просмотрите результат.
4. *Построение разделительной горизонтальной линии.*

Линия задается тэгом **<Hr>** и не требует закрывающего тэга (ширина по умолчанию = 2 пикселям).

ЗАДАНИЕ 6:

Покажите размещения различных по параметрам линий на вашей страничке (см. примеры ниже)

Примеры использования:

- (1) **<Hr align=right>** (**center** или **left**) – линия размещается по правому краю (центру, слева).
- (2) **<Hr width=30%>** - ширина линии в процентах/пикселях по отношению к размеру окна браузера
- (3) **<Hr size=6>** - толщина линии в пикселях
- (4) **<Hr NoShade>** - отмена объемности

(5) `<Hr color=#cc0000>` - цвет линии, только в IE

Параметры могут употребляться одновременно.

ЗАДАНИЕ 7:

1. Занесите еще одну картинку в ваш файл, шириной, равной 130.
2. Расположите вторую картинку по другую сторону текста по отношению к первой картинке.

5. Параметр тэга *Br - Clear.*

Для отмены обтекания картинки фрагментом текста введите перед данным фрагментом тег:

`<BR CLEAR=?>`, вместо знака вопроса укажите один из параметров **Left**, **Right** или **All** (*прекращение обтекания текстом картинки либо слева, либо справа, либо везде*)

ЗАДАНИЕ 8:

1. Отмените обтекание у первой картинки.
2. Результат просмотрите и проанализируйте в браузере

6. Списки с графическими маркерами.

Например:

1) Способ введения графического маркера

```
<UL>
<BR><IMG SRC="рис.jpg"><I> Глава 1,</I>текст к главе 1
<BR><IMG SRC="рис.jpg"><I> Глава 2,</I>текст к главе 2
</UL>
```

2) Ширина расстояния от маркера до текста.

```
<UL>
<BR><IMG SRC="рис.jpg" HSPACE=10><I> Глава 1,</I>текст к главе 1
<BR><IMG SRC="рис.jpg" HSPACE=10><I> Глава 2,</I>текст к главе 2
</UL>
```

3) Выравнивание элементов относительно текста.

```
<UL>
<BR><IMG SRC="рис.jpg" ALIGN=middle><I> Глава 1,</I>текст к главе
1
<BR><IMG SRC="рис.jpg" ALIGN=middle><I> Глава 2,</I>текст к главе
2
</UL>
```

ЗАДАНИЕ 9:

1. Вставьте различные графические маркеры в файл picture.html, задав выравнивание относительно текста по центру, снизу и сверху.
2. Результат просмотрите и проанализируйте в браузере

Контрольные вопросы:

1. Какие теги используют для вставки графического изображения в документ?
2. Какие теги используют для создания фонового рисунка?
3. С помощью каких тегов можно редактировать рисунок?

Лабораторная работа №5 по HTML Тема: Создание гиперссылок (ссылок).

Цель работы: научиться создавать гиперссылки

1. Ссылки на отдельные документы

Ссылкой на другие документы (части странички) может быть текст (фраза, слово), а может быть и картинка.

`` текст (фраза или рисунок) для ссылки `` - команда, которая задает ссылки на документы, вместо знака вопроса проставляется путь и полное название документа, куда мы будем перемещаться по ссылке (принципы прописывания пути здесь такие же как в случае с картинками).

Примеры:

`мои фотографии` - документ лежит в той же директории (папке), что и документ, в котором мы ссылаемся на prf.html

`мои фотографии` - документ лежит в поддиректории /photos

`мои фотографии` - ссылаемся на сайт http://www.homepage.ru, где лежит нужный нам документ

`<p></p>` - пример ссылки с рисунка (графическая ссылка)

ЗАДАНИЕ 1:

3. Создайте новый документ **lab5.html** в той же директории (папке), где находится документ **lab4.html**, созданный ранее.
4. Файл lab5.html будет содержать 3 картинки в ряд, *размещенных по центру страницы, размером 130x60 и рамкой в 4 экранных пиксела.*
5. Введите в конце документа lab4.html фразу "*Хочешь посмотреть другие картинки?*"
6. Создайте ссылку с этой фразы документа lab4.html на документ lab5.html по примеру:
`
 Хочешь посмотреть другие картинки? `
7. Посмотрите и проверьте действие вашей ссылки в браузере.

8. Самостоятельно, используя примеры выше, создать ссылку со всех рисунков файла lab5.html на файл lab4.html.

2. Ссылки на объект в той же странице.

**** - атрибут NAME позволяет ставить закладки в документе, чтобы потом можно было строить ссылки именно на эти закладки, вместо знака вопроса задается название закладки.

наименование ссылки на закладку, вместо вопроса проставляется название закладки, определенное в NAME

Например:

(1) - ``

`переход на слово пылесосы в тексте`.

(2) - `` - ссылка на начало страницы

`Вернуться к началу документа `

(3) - `Модели пылесосов` -

указание метки страницы, на которую направлена ссылка из другого файла или страницы

ЗАДАНИЕ 2:

- 1) Откройте файл index.htm.
- 2) Используя буфер создайте 4 копии основного текста от тега `<body>` до `</body>` (без заголовков).
- 3) Создайте 2 закладки `` и `` в начале и в конце документа соответственно
- 4) Занесите в текст документа гиперссылку «Вернуться к началу документа» и ссылку «Перейти в конец документа». В результате первой ссылки должно происходить перемещение в начало документа, а при выборе второй ссылки – в конец исходного документа (см. примеры выше).
- 5) Проверьте результаты в браузере.
- 6) Самостоятельно создайте произвольную закладку в центре документа и постройте ссылку на этот объект.

2. Цвет ссылок.

атрибут LINK – основной цвет ссылок

в теле тега BODY

атрибут VLINK – цвет ссылок, уже выбиравшихся пользователем

атрибут ALINK – цвет активной (нажатой) ссылки

Например:

`<body text=#336699 bgcolor=#000000 link=red alink=green vlink=#339999>`

ЗАДАНИЕ 3:

- 1) Определите свой цвет для основных, уже выбиравшихся и активных ссылок.

5. Открытие Web – страниц в новом окне браузера.

Атрибут **TARGET=?** , где знак вопроса означает название окна (задается самостоятельно)

Например:

```
<br><h1><a href="prf.html" Target=prf>посмотреть мои фотографии</a></h1> - данные по  
ссылке отображаются в новом окне prf
```

```
<br><h1><a href="prf.html" Target=_blank>посмотреть мои фотографии</a></h1> -  
данные по ссылке отображаются в новом окне без имени  
наберите
```

Примечание:

Если необходимо, чтобы несколько гиперссылок отображали данные в общем окне, повторите одно и то же имя в Target для всех этих окон или введите между `<head>` `</head>` строку `<base target=?>`, где знак вопроса задает общее имя окна для всех ссылок сайта

ЗАДАНИЕ 4:

- 1) Создайте метку к слову "Здравствуйте" в тексте документа lab4.html.
- 2) Постройте ссылку на эту метку из любого другого документа формата .html, созданного вами ранее.

Все данные по ссылкам должны открываться в новом окне браузера.

ДОПОЛНИТЕЛЬНО:

3. Ссылки на отдельный файл.

Пример:

```
<br>Здесь архив песен Шарля Азнавура(формат MPEG3)  
<a href="Шарль Азнавур.exe">Щелкни, загрузи и наслаждайся</a>
```

4. Ссылки на адрес электронной почты.

`` - вместо знака вопроса проставляется соответствующий адрес электронной почты.

Примеры:

```
<a href=mailto:webmaster@classicalComposers.com>Сообщите, пожалуйста,  
по электронной почте.</a>
```

```
<a href="mailto:pochta@mail.ru"> pochta@mail.ru - пишите письма </a>
```

Контрольные вопросы:

1. С помощью какого тега создаются гиперссылки?
2. Какие атрибуты используются в гиперссылках?
3. Что означает атрибут <TARGET>?

Лабораторная работа №6 по HTML Тема: Создание таблиц.

Цель работы: изучить теги таблицы, уметь создавать таблицы различной структуры

1. Построение простых таблиц и колонок

<table></table> - теги начала и конца таблицы

<tr></tr> - строка (и) таблицы

<th></th> - заголовки столбцов таблицы (обычная ячейка с центрованным полужирным шрифтом)

<td></td> - столбец (обычная ячейка) таблицы

Для создания таблицы необходимо выполнить следующие действия:

1.

```
<table>
<tr></tr>
<tr></tr>
```

Задаем количество строк таблицы

```
.....
</table>
```

2.

```
<table>
<tr>
```

```
<td></td>
<td></td>
<td></td>
```

Задаем количество столбцов (ячеек) таблицы в каждой строке

```
.....
</tr>
```

```
<tr>
<td></td>
<td></td>
<td></td>
```

```
.....
</tr>
.....

</table>
```

3.

```
<table>
```

```
<tr>
<td>1x1</td>
<td>1x2</td>
<td>1x3</td>
```

Внутри тегов ячеек вносим
необходимый текст в ячейки

```
.....
</tr>
.....

</table>
```

Примечание:



1. Если в ячейке задается заголовок столбца или строки таблицы, то целесообразнее использовать вместо тега **<td>** тег **<th>**

2. Для создания **колонок** (с центровкой по умолчанию) необходимо создать таблицу в **одну** строку.

ЗАДАНИЕ 1:

Постройте таблицу из *примера 1*:

Пример 1

1 столб	2 столб	3 столб
--------------------------	--------------------------	--------------------------

ец	ец	ец
1x1	1x2	1x3
2x1	2x2	2x3

2. *Обрамление таблицы (рамка).*

Атрибут **border=?** в теле тега **<table>**, где ? – это ширина рамки таблицы в пикселах.

ЗАДАНИЕ 2:

Установите рамку вокруг исходной таблицы шириной в 3 пиксела.

3. *Определение заголовка таблицы.*

Тег **<CAPTION>**

Например:

```
<TABLE BORDER=5>
```

```
<CAPTION>ДОМИНО</CAPTION>
```

ЗАДАНИЕ 3:

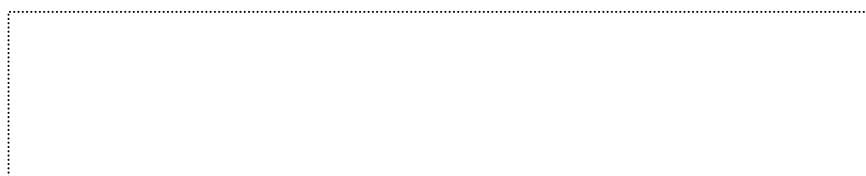
1. Добавьте заголовок «Мои первые шаги в построении таблиц» *в примере 1*
2. Задайте его курсивным, полужирным шрифтом.

4. *Цвет рамки и цвет ячеек таблицы.*

3.1. Атрибут **bordercolor=?** в теле тега **<table>**, где ? – это цвет рамки (название или код)

3.2. Атрибут **bgcolor=?**, где ? – задает цвет ячеек таблицы

Цвет можно задать для таблицы в целом, для ряда, для столбца (в пределе одного ряда).



Например:

1. `<table bgcolor=#FFCC33>` - цвет ячеек всей таблицы
2. `<tr bgcolor=#FFCC33>` - цвет ячеек строки
3. `<td bgcolor=#FFCC33>` - цвет отдельной ячейки строки

ЗАДАНИЕ 4:

1. Задайте зеленый цвет рамки таблицы *примера 1*.
2. Допишите код *примера 1*, задав цвет ячеек 2 и 3 строки таблицы в шахматном порядке.

5. Графический фон таблицы.

Атрибут `background="?"` задает фоновую картинку для таблицы (ряда, ячейки), где ? – это название файла фона и путь к нему.

Например:

1. `<table border=10 background="Мечта.jpg">` - цвет фона таблицы
2. `<th background="white.gif"> </th>` - цвет фона отдельной ячейки

ЗАДАНИЕ 5:

Подберите для ячеек первой строки таблицы из *примера 1* фоновую картинку.

6. Изменение размеров таблицы.

Атрибуты `width=?` и `height=?` - задают ширину и высоту ячеек таблицы (могут использоваться в любых тегах таблицы), вместо знака вопроса проставляется значение размера ячейки в пикселах или процентах(от общей ширины таблицы).

Атрибуты можно использовать для задания размеров всей таблицы, для одного ряда, для ячейки (столбца) в отдельности.

Например:

```
<table width=60> - задается ширина ячеек всей таблицы  
<td height=35 width=50 bgcolor=#FFCC33>1x1</td> - размер  
отдельной ячейки  
<tr width=30%> - размер ячеек всей строки
```

ЗАДАНИЕ 6:

1. Увеличьте в *примере 1* размеры (ширину и высоту) ячеек всей таблицы.
2. Измените размеры среднего столбца. Проанализируйте результат в браузере.
3. Измените длину всех ячеек 1 ряда таблицы. Что произошло с остальными рядами?

7. Выравнивание содержимого ячеек таблиц.

Атрибут **valign= middle (top, bottom)** в теле тегов `<tr>` и `<td>` задает вертикальное выравнивание содержимого таблиц (посередине, сверху, снизу).

Атрибут **align=center (right, left)** выравнивает содержимое ячейки по горизонтали.

Пример:

`<td valign=middle>` - выравнивание текста ячеек по центру

ЗАДАНИЕ 7:

Разместите ячейки таблицы *примера 1* в соответствии с рисунком:

1x1	1x2	1x3
2x1	2x2	2x3

8. Объединение ячеек.

Атрибуты **colspan=?** – объединение ячеек нескольких столбцов в пределах одной строки

Атрибут **rowspan** =? – объединение ячеек нескольких строк в пределах одного столбца

Вместо знака вопроса проставляется количество объединяемых ячеек.

Например:

<TH COLSPAN=3> - означает, что ячейка простирается на три колонки.

<TH ROWSPAN=2> - означает, что ячейка занимает две строки

Атрибут **NOWRAP** означает, что содержимое ячейки должно быть показано в одну строку.

Например:

<TH NOWRAP>

ЗАДАНИЕ 8:

1. Добавьте копию исходной таблицы документ, располагая ее ниже основной.
2. Измените таблицу №1 *примера1* по рисунку, используя параметр colspan (вылезшие при этом ячейки просто удалите из документа):

1x1	1x2	
2x1	2x2	2x3

В примере нет ячейки 1x3, ячейка 1x1 равна сама по себе двум ячейкам

3. Внесите изменения в таблицу №2 по рисунку, задавая значение для параметра rowspan.

1x1	1x2	1x3
2x1	2x2	

4. Перепишите код для 1 таблицы, по которому она примет вид:

1x1		1x2
2x1	2x2	

9. *Определение ширины границ ячеек таблицы.*

Атрибут **cellspacing=?** в теле тега <table>, вместо знака вопроса задается значение ширины границы в пикселах.

Атрибут **cellpadding=?** в теле тега <table>, устанавливает расстояние между рамкой ячейки и текстом в пикселах

ЗАДАНИЕ 9:

1. Установите для первой таблицы документа значение параметра cellspacing=0.
3. Задайте для второй таблицы значение параметра cellspacing=5
4. Результат просмотрите в браузере.
5. Установите для таблицы 1 значение параметра cellpadding=6.
6. Результат просмотрите в браузере.

10. *Обтекание таблицы текстом*

Атрибут **ALIGN=left (right)** в теле дескриптора <TABLE>

<BR CLEAR> (left, right, all) – прерывание обтекания слева, справа, везде.

Примечание:

Если таблица выровнена по центру, то обтекание текстом не происходит, например:

```
<table align=center border=1>
```

ЗАДАНИЕ 10:

1. Добавьте текст «Посмотрите, какие замечательные таблицы у меня получаются» в конце вашего документа.
2. Задайте обтекание текста так, чтобы он располагался между двумя таблицами.

ДОПОЛНИТЕЛЬНО:

Построение вложенных таблиц.

- Создайте таблицу, которая должна содержать в качестве элементов другие таблицы, и оставьте соответствующие ячейки пустыми.

- Щелкните в пределах текста описания ячейки, в которую необходимо вставить подчиненную таблицу.
- Опишите вложенную таблицу, руководствуясь общими правилами написания таблиц.
- При вводе HTML – кода руководствуйтесь собственной системой отступов.

Примечание:


Путем таблиц можно создать довольно интересные примеры:

Различные таблицы в IE



ЗАДАНИЕ 11:

1. Создать таблицу по исходному примеру.
2. Вставить рисунок небольших размеров на месте крестика в таблице (рисунок должен соответствовать тексту)
3. Добавьте пояснения к рисунку на случай невозможности его воспроизведения.
4. Придумайте заголовок к таблице.
5. Задайте графический фон для ячеек 1 строки таблицы и для документа в целом.
6. В остальных ячейках цвет установите по своему вкусу.
7. Результат работы предьявите преподавателю.

Расписание	Отелло	Гамлет	Цезарь
Генеральная репетиция	Следите за афишами		28 декабря 2003 года
Премьера			31 декабря 2003 года

Контрольные вопросы:

1. Какие теги используются для создания таблицы?
2. Какие теги служат для создания строк?

3. С помощью каких атрибутов можно объединять ячейки?

Лабораторная работа №7 по HTML Тема: Навигационные карты ссылок (Image map).

Цель работы: уметь создавать и работать с навигационными картами ссылок



Ранее по картинке мы могли задавать ссылку только на один документ.

Возможность, используя один тот же рисунок, перемещаться в различные документы позволяет Карта ссылок.

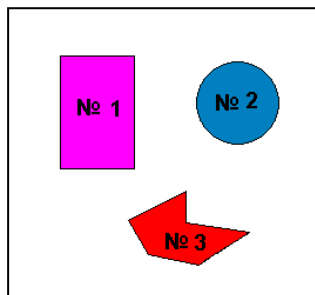
Она будто разбивает изображение (готовое или созданное самим пользователем) на отдельные участки, каждому из которых будет соответствовать своя гиперссылка.

Навигационные карты задаются тэгом **<Map>**.

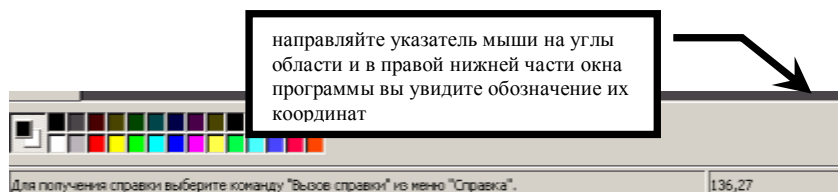
Тэг **<Map>** включает себя тэг(и) **<Area>**, которые определяют геометрические области внутри карты и ссылки, связанные с каждой областью (т.е. куда вы попадете при нажатии на какую-либо часть нашей карты).

Для создания карты выполните следующие действия:

1. Загрузите любой графический редактор (в нашем случае – это Paint).
2. Установите ширину и высоту страницы 200x200 точек (Рисунок-Атрибуты)
3. Зарисуйте картинку (без рамки) по примеру:



4. Определите координаты верхнего- левого и нижнего- правого углов прямоугольной области №1.



5. Сохраните файл под именем **Области.bmp** на своем диске в каталоге **HTML**.
6. В конце ранее созданного файла **prf.html** в **центре новой строки** задайте вывод созданной картинке на экран при помощи тега ****
 Просмотрите результат в браузере и предъявите преподавателю.
7. Введите в конце документа **prf.html** теги для определения браузером карты ссылок:
<map>
<area> *<! тег <area> определяет зону отображаемой карты >*
</map>
8. Внутри тега **<area>** введите параметры для определения геометрического вида области:
<area shape=?> *<! атрибут <shape> определяет форму области>*
 вместо знака вопроса проставьте одно из значений, соответствующее виду вашей области:
rect (прямоугольник) , **circle (окружность)**, **poly (многоугольник)**
9. Добавьте в тег **<area>** атрибут **coords=?** для задания координат области (в пикселах, 1 точка=1 пикселу). Вместо знака вопроса проставьте координаты области, определенные ранее, выбрав необходимый вариант:
Coords=x1,y1,x2,y2 – координаты верхнего- левого (x1,y1) и нижнего- правого углов (x2,y2) прямоугольника
Coords=x0,y0,r – координаты центра (x0,y0) и радиуса (r) окружности
Coords=x1,y1,x2,y2,x3,y3,...,xN,yN – координаты всех углов прямоугольника.
10. Задайте переход с первой области карты на сайт института **http://iubip.ru/** , прописав параметры команды **Href=?** внутри тега **<area>**,
знак вопроса заменятся адресом Web – страницы (сайта), подлежащей (го) воспроизведению после щелчка по зоне.
11. В теле тега **<area>** определите атрибут **Alt=?**, позволяющий выводить на экран текстовое описание области (всплывающую подсказку).
 Вместо знака вопроса запишите для данного случая: *«Здесь вы можете попасть на сайт института»*
12. В общем случае ваша команда должна выглядеть так:
<area shape=? coords? Href=? Alt= Здесь вы можете попасть на сайт института>,

где на месте вопросов - соответствующие значения вашей области (см. выше).

13. Чтобы картинка стала ссылкой, необходимо указать имя карты:

<map name="?">

где ? – это произвольно заданное пользователем имя карты ссылок, н-р: karta_1

14. Свяжите название карты с картинкой при помощи атрибута

usemap="#?" в теле тега

знак вопроса – это установленное в <map> имя карты.

15. Задайте в <body> цвет основных, ранее выбиравшихся и текущих ссылок.

16. Сохраните все внесенные изменения.

17. Просмотрите и проверьте действие ссылок в браузере.

18. Результат предьявите преподавателю.

19. Самостоятельно внесите новым тегом <area> параметры остальных двух областей (в тегах указывайте ссылки на различные HTML-документы).

20. Результат предьявите преподавателю.

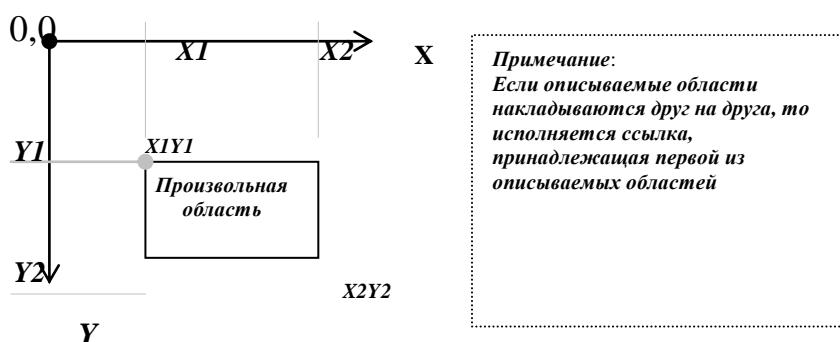
21. Введите в <map> новую команду:

<area shape=circle coords=x, y, z href=prf.html alt = произвольная область>

(вместо переменных X, Y, Z подберите «на глаз» такие координаты, чтобы эта область описывала центр рисунка).

При подборе координат учитывайте, что:

Начало координат (0,0) находится в левом верхнем углу монитора; по оси OX примерно 800 точек (пиксел); по оси OY приблизительно 600 точек.



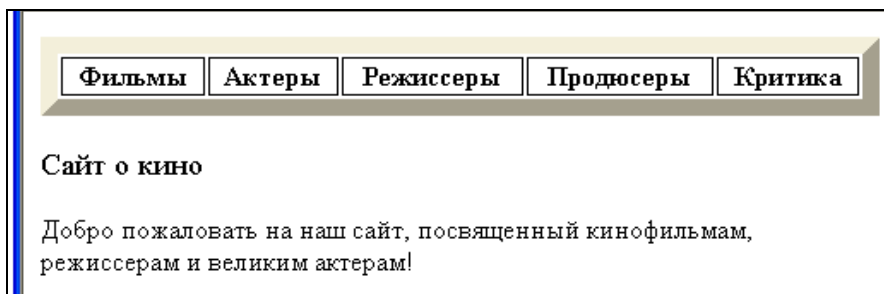
24. Остальные теги <area> возьмите в скобки комментария, для временного прекращения их действия.

25. Сохраните документ, перейдите в окно браузера и проверьте действие ссылки, щелкнув по центру вашего рисунка (*обратите внимание на действие ссылки*).

26. Самостоятельно измените координаты новой области так, чтобы она описывала правый верхний угол картинка.

Самостоятельно:

Создайте в файле `Моя_Карта.html` свою навигационную карту в виде меню (см. рис), по нажатию на какую-либо область вашей карты должно происходить перемещение на различные странички, отвечающие содержанию карты.



Контрольные вопросы:

1. С помощью каких тегов можно создавать навигационные карты ссылок?
2. Какие атрибуты используются для создания карт?
3. Что означает тег `<MAP>`?

Лабораторная работа №8

Тема: Фреймы.

Цель работы: научиться создавать фреймы

1. Основные понятия.

Frame-(рамка)- прямоугольные области экрана, каждая из которых содержит свой собственный HTML-документ. тэг, предназначенный для разделения Web-страницы на окна.

Фреймы прекрасно подходят для оформления следующих документов:

Оглавление. Если вы поместите на Web-страницу оглавление в виде вертикального столбца, то пользователь сможет обратиться к нему в любой момент.

Неподвижные элементы интерфейса. Можно зафиксировать на экране какое-то графическое изображение, например, логотип фирмы, в то время как остальная часть страницы будет прокручиваться в другом фрейме.

Формы и результаты. Можно создать форму в одном фрейме, а в другом отобразить результаты запроса.

2. Тэги фрейма:

1. `<frameset></frameset >`- используется для разделения экрана (заменяет тег `<body>`)

Атрибуты:

COLS	Делит экран по вертикали. Принимает значения в пикселях, в процентах или просто *. Значение * говорит о том, что конкретный столбец будет занимать всю остальную часть экрана
ROWS	Делит экран по горизонтали. Принимает значения в пикселях, в процентах или просто *. Значение * говорит о том, что конкретная строка будет занимать всю остальную часть экрана
FRAMEBORDER	Определяет наличие рамок, т.е. границ фреймов. Принимает значения "yes" или "no"
BORDER	Определяет ширину рамки в пикселях
BORDERCOLOR	Определяет цвет рамок. По умолчанию (если используется стандартная цветовая схема системы Windows) границы фреймов имеют тусклый, серый оттенок.

Примечание: В отдельном тэге <FRAMESET> имеет смысл использовать только один из атрибутов – COLS или ROWS. Это означает, что фреймовая структура будет состоять или только из столбцов, или только из строк. Чтобы создать строки внутри столбцов или столбцы внутри строк, вам потребуется вложенные контейнеры <FRAMESET> ... </FRAMESET>.

Например,

`<frameset rows=20%,80%>` - деление страницы на верхнее и нижнее окна с размерами 20% и 80% от высоты страницы.

`<frameset cols =20%,40%,* frameborder=yes border=5>`.

- `<frame>` - помещается в контейнер `<FRAMESET>...</FRAMESET>`, определяет, что именно должно отображаться в конкретном фрейме. Не имеет закрывающего тэга.

Атрибуты:

SRC	Определяет URL, связанный с конкретным фреймом
MARGINWIDTH	Определяет расстояние между содержимым фрейма и его границами справа и слева
MARGINHEIGHT	Определяет расстояние между содержимым фрейма и его границами сверху и снизу
SCROLLING	Определяет присутствие в окне фрейма полосы прокрутки. Может принимать значения "yes", "no" и "auto"(по умолчанию).
NORESIZE	Определяет границы фрейма как "жестко закрепленные"

**FRAMEBORDER,
BORDER,
BORDERCOLOR
NAME**

и не позволяет пользователю изменить размеры окна фрейма. Причем фиксируются размеры всех фреймов, имеющих с закрепленной общую границу.

Эти атрибуты связаны с рамками и аналогичны таким же в тэге <FRAMESET>

Определяет имя окна фрейма (произвольное). Позволяет контролировать процесс загрузки фреймов. Если окно фрейма имеет уникальное имя, то к нему можно непосредственно обратиться из других фреймов. Имена не должны начинаться с символа подчеркивания "_", иначе оно будет игнорироваться. Так как с этого символа начинаются некоторые служебные имена.

Например,

< frame name=frametop src=top.htm> - определены имя верхнего окна и имя файла, который загрузится в это окно.

<frameset cols=25%,75% frameborder=yes border=5 bordercolor=#008800>

<frame src=left.htm marginwidth=20 marginheight=20>

<frame src=right.htm name=main_window marginwidth=0 scrolling=no noresize>

</frameset>

Постановка задачи.

Создать страницу, разделенную на верхнее и нижнее окна. В верхнее окно поместить слова «Священный Байкал». В нижнем окне отобразить картинку,



подобранную самостоятельно по смыслу.

3. План разработки темы.

1. Образуйте, если она у вас еще не образована, новую папку, назвав ее, напр., *baikal*.
2. **Создание главного файла.**

Наберите следующий код:

```
<html>  
<head><title>FRAME ROWS</title></head>  
<frameset rows =20%,80%>  
<frame name = frametop src = top.htm>  
<frame name = framebottom src = bottom.htm>  
</frameset>
```

```
<html>
```

Сохраните этот файл под именем, например, *twowin.htm*.

3. Заполнение верхнего окна.

Наберите следующий код:

```
<html>
<head><title>Верхнее окно</title></head>
<body>
<h2>Священный Байкал </h2>
</body>
</html>
```

Сохраните этот файл обязательно под именем, которое вы указали в главном файле, а именно: *top.htm*.

Проверьте результат отображения этого файла в браузере.

4. Заполнение нижнего окна.

Наберите следующий код:

```
<html>
<head><title>Нижнее окно</title></head>
<body>
<img src = .....>      (вместо многоточия вставьте имя файла
вашей картинки)
</body>
</html>
```

Сохраните обязательно под тем именем, которое вы указали в главном файле, а именно: *bottom.htm*.

Проверьте результат отображения в браузере

4. Загрузите в Web-браузер главный файл (*twowin.htm*) и убедитесь, что заполнены оба окна Web -страницы.

ДОПОЛНИТЕЛЬНО:

1. Разделите новую Web-страницу на левое и правое вертикальные окна
2. Разделите страницу на три части: пусть верхняя часть будет горизонтальной полосой, а нижняя пусть будет разделена на две вертикальные колонки (при этом разделительная линия будет выглядеть подобно букве Т).

Подсказка. Примените так называемый вложенный фрейм (он выделен жирным шрифтом):

```
<Frameset rows =20%, 80%>
<Frame Name=Frame1 src=f1.htm >
  <Frameset cols=40%,60%>
    <Frame name=Frame2 src=f2.htm>
    <Frame name=Frame3 src=f3.htm>
  </Frameset>
```

</Frameset>

3. Закрепите разделительную линию, чтобы пользователь не смог бы ее перемещать.
4. Сделайте так, чтобы разделительная линия не была бы видна.
5. Замените заполнение окон: текст на изображение, картинку на текст.
6. Уберите полосу прокрутки из верхнего фрейма.

Контрольные вопросы:

1. С помощью каких тегов можно создавать фреймы?
2. Какие атрибуты используются для фреймов?
3. Что означает тег <frameName>?

Лабораторная работа №9 Тема: «Создание форм»

Цель работы: научиться создавать формы

Форма

это электронный вариант анкеты на бумаге, содержащий поля для ввода текста, варианты ответов на вопрос, выпадающие меню с вариантами и другие элементы формы и предназначенная для связи владельца сайта и посетителей.

Давайте сначала посмотрим как выглядит обычная форма на сайте:

Ф.И.О.

Пол муж. жен.

Возраст

Ваши комментарии:

Кнопка "Очистить" удаляет введенные данные из полей формы. Кнопка "Отправить" посылает данные одним из выбранных методов автору сайта. Вот общая схема для создания любой формы.

<FORM>

теги полей формы

</FORM>

Итак, какие бывают типы полей формы и для чего они предназначены?

Текстовая строка (TYPE=TEXT)

это строка, предназначенная для ввода небольшого количества символов, используемая для ввода пользователем информации произвольного характера.

Пример:

```
<INPUT TYPE=TEXT NAME=Текст SIZE=20 VALUE="Здесь текст">
```

Результат:

Здесь тек

Таким образом, текстовая строка создается при помощи тега `<INPUT>` с использованием параметра `TYPE=TEXT`. Что означают остальные параметры этого тега?

NAME

Имя переменной, которая будет передана серверу. Результат введенной пользователем строки будет присвоен переменной с таким именем. Вообще говоря создание форм в HTML тесно связано с программированием на сервере. Но эта тема не будет обсуждаться в данном руководстве. Вместо этого, вы сможете воспользоваться уже готовой программой на этом сервере и получать данные от своих посетителей через эту программу.

SIZE

Ширина текстовой строки в символах. В нашем примере значение этого параметра равно 20. Это значит что ширина строки для ввода равна 20 символам. Пользователь может ввести и большее количество символов, но видеть он будет только 20. С помощью стрелок на клавиатуре можно перемещаться к концу или к началу введенного текста.

VALUE

Значение этого параметра оказывается заранее впечатанным в текстовую строку. Его можно изменять или удалять.

Вариант ответа (TYPE=RADIO)

Среди нескольких вариантов ответа пользователь может выбрать только один вариант.

Пример:

Пол:

```
<INPUT TYPE=RADIO NAME=Sex VALUE="Мужской">Муж.
```

```
<INPUT TYPE=RADIO NAME=Sex VALUE="Женский">Жен.
```

Результат:

Пол: Муж. Жен

Как видите варианты ответа создаются при помощи тега `<INPUT>` с использованием параметра `TYPE=RADIO`. Обратите внимание, что значение

параметра NAME должно быть одинаковым, иначе это будут разные вопросы.

VALUE

Это значение будет передаваться в качестве значения переменной (в нашем примере, если выбрать Муж., то значение переменной Sex будет равно 'Мужской').

Подтверждение (TYPE=CHECKBOX)

Используется для отметки некоторого значения или варианта ответа при помощи проставления галочки

Пример:

Пол:

```
<INPUT TYPE=CHECKBOX NAME=AddMe VALUE="Подписаться">
```

Да, я хочу получать новости вашего сайта

Результат:

Да, я хочу получать новости вашего сайта

Выпадающее меню (SELECT)

Используется для предоставления нескольких вариантов ответа в виде выпадающего меню, для выбора одного варианта.

Пример:

Возраст:

```
<SELECT NAME=Age>
```

```
<OPTION VALUE=10>меньше 10
```

```
<OPTION VALUE=11-20>11-20
```

```
<OPTION>больше 21
```

```
</SELECT>
```

Результат:

Возраст:

Выпадающее меню создается при помощи тега <SELECT>. Варианты выпадающего меню создаются при помощи тега <OPTION>

NAME

Параметр тега <SELECT>, задающий название переменной. Этой переменной будет присвоено значение выбранного варианта ответа. В нашем случае переменная называется Age.

VALUE

Параметр тега <OPTION>, означающий значение, которое нужно присвоить переменной в случае, если будет выбран данный вариант. Если этот параметр не указан (как в третьем теге <OPTION>), то значение переменной будет таким, какой текст указан после тега

<OPTION>. Если в нашем примере выбрать третий вариант, то значение переменной Age будет равно 'больше 21', а если выбрать первый вариант, то '10'.

Текст указанный после тега <OPTION> будет отображен в качестве варианта в выпадающем меню на экране. Кстати, тег <OPTION> закрывать не нужно.

Текстовое поле (TEXTAREA)

это поле, предназначенное для ввода большого количества символов, используемое для ввода пользователем информации произвольного характера.

Пример:

Ваши комментарии:

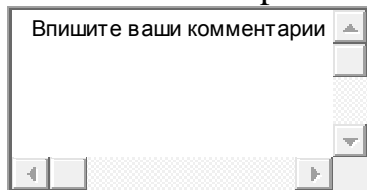
```
<TEXTAREA NAME=Comments COLS=20 ROWS=5>
```

Впишите ваши комментарии

```
</TEXTAREA>
```

Результат:

Ваши комментарии:



Текстовое поле создается при помощи тега <TEXTAREA>. Все что стоит между открывающим и закрывающим тегом текстового поля будет впечатано в это текстовое поле

COLS

Ширина текстового поля в символах

ROWS

Высота текстового поля в строках

Задание для самостоятельного выполнения:

1. Составьте следующее выпадающее меню (назовите переменную **Rabota**, параметр **VALUE** не используйте.):

Студент
Студент
Директор
Бизнесмен
Менеджер
Безработный

2. Составьте кусок HTML кода так, чтобы результат был следующим (назовите переменные как **Pol**, **Rabota** и **Podpiska**. параметры **VALUE** и **SIZE** не используйте. Для переноса строки используйте тег **
**):

Pol: Man Woman

Job:

yes, please subscribe me

Контрольные вопросы:

1. С помощью каких тегов можно создавать формы?
2. Какие атрибуты используются для форм?
3. Что означает тег `<TEXTAREA >`?

Лабораторная работа №10

Тема: «Создание кнопки очистки»

Цель работы: научиться создавать кнопку очистки

Для того, чтобы пользователь смог заполнить и отправить форму, нужно создать кнопку подтверждения, а также можно создать кнопку очистки формы, нажав на которую можно очистить все поля формы. В начале этой главы вы уже видели как выглядят кнопка отправки и очистки формы. Вверху этой страницы вы можете увидеть как выглядят кнопки для отправки формы. Точно так же выглядит и кнопка для очистки полей формы.

Кнопка отправки (TYPE=SUBMIT)

Предназначена для подтверждения пользователем введенных в форму данных и отправки этих данных на сервер.

Пример:

```
<INPUT TYPE=SUBMIT VALUE="Отправить форму">
```

Результат:

Кнопка с надписью 'Отправить форму'

Кнопка отправки создается при помощи тега `<INPUT>` с использованием параметра `TYPE=SUBMIT` В результате будет создана кнопка на которой

будет написано значение параметра. При нажатии на кнопку данные формы будут отправлены на сервер. VALUE

Кнопка очистки (TYPE=RESET)

Предназначена для очистки всех введенных в форму пользователем данных.

Пример:

```
<INPUT TYPE=RESET VALUE="Очистить форму">
```

Результат:

Кнопка с надписью 'Очистить форму'

Кнопка очистки полей формы создается при помощи тега <INPUT> с использованием параметра TYPE=RESET. В результате будет создана кнопка на которой будет написано значение параметра VALUE. При нажатии на кнопку все поля формы будут очищены.

Задание, для самостоятельного выполнения:

Ф.И.О.

Пол муж. жен.

Возраст

Ваши комментарии:

Очистить

Контрольные вопросы:

1. С помощью каких тегов можно создать форму очистки?
2. Какие атрибуты используются?

Лабораторная работа № 11

Тема: Свойства шрифта, текста. Границы в CSS

Цель работы: изучить свойства шрифта и текста. Научиться создавать границы различного вида.

font-family	Используется для указания шрифта или шрифтового семейства, которым будет отображаться элемент. P {font-family: Times New Roman, sans-serif;}
-------------	---

font-style	Задает способ начертания шрифта: normal - Нормальный (по умолчанию), italic - Курсив, oblique - Наклонный. P {font-style: italic;}
font-variant	Задает варианты начертания шрифта: normal - Нормальный (по умолчанию), small-caps - Все буквы заглавные). P {font-variant: small-caps;}
font-weight	Определяет степень жирности шрифта с помощью параметров: normal, bold, bolder, lighter, 100, 200, 300, 400, 500, 600, 700, 800, 900 B {font-weight: bolder;}
font-size	Устанавливает размер шрифта. Параметр может указываться как в относительной (проценты), так и абсолютной величине (пункты, пиксели, сантиметры) H1 {font-size: 200%;} H2 {font-size: 150px;} H3 {font-size: 400pt;}

Примечания

font-family - Можно указать до трех семейств шрифта. Это необходимо, так как, если на компьютере клиента отсутствует, скажем, семейство Verdana, а в правиле указано только оно, то разработчик не может предположить, как страница будет выглядеть на экране пользователя. В том, случае, когда в правилах указано несколько семейств, браузер будет последовательно перебирать все перечисленные, пока не найдет того, которое присутствует на машине клиента. Браузер при запуске сам распределяет имеющиеся семейства для подстановки вместо стандартных.

font-weight - Значения bolder и lighter являются относительными. Т.е. если элемент является дочерним к элементу со значением font-weight: bold и правило font-weight для него задано lighter, то он будет отображен нормальным шрифтом.

font-size - В абсолютных размерах можно использовать некоторые predefined значения: xx-small | x-small | small | medium | large | x-large | xx-large. При использовании относительных размеров задаются размеры шрифта относительно родительского элемента. Возможные значения: larger, smaller. И наконец, можно использовать точные значения - Задаются размеры шрифта в единицах длины, допускаемых в рекомендациях W3C. Это: px - логические пиксели, pt - логические пункты, em - единицы размера шрифта. em подразумевает, что размер шрифта устанавливается относительно размера шрифта в родительском элементе. 1em равно размеру шрифта родительского элемента. Либо используются проценты - Задаются размеры шрифта в процентах по отношению к родительскому элементу


Примеры

Курсивный шрифт

ВСЕ БУКВЫ ЗАГЛАВНЫЕ

Жирный шрифт **bolder**

Жирный шрифт **bold**

Жирный шрифт на 200 % 

text-decoration	<p>Устанавливает эффекты оформления шрифта, такие, как подчеркивание или зачеркнутый текст</p> <p>H4 {text-decoration: underline;} - подчеркивание A {text-decoration: none;} - стандартный текст I {text-decoration: line-through;} - зачеркивание B {text-decoration: overline;} - надчеркивание</p>
text-transform	<p>Задаёт преобразование регистра текста при отображении</p> <p>H4 {text-transform: capitalize;} - Первая буква каждого слова преобразуется в заглавную A {text-transform: uppercase;} - Все буквы преобразуются в заглавные I {text-transform: lowercase;} - Все буквы преобразуются в строчные B {text-decoration: none;} - Отменяет установленные преобразования</p>
text-align	<p>Определяет выравнивание элемента. Возможные значения: left, right, center, justify</p> <p>P {text-align: justify} H5 {text-align: center}</p>
text-indent	<p>Устанавливает отступ первой строки текста. Чаще всего используется для создания параграфов с табулированной первой строкой.</p> <p>P {text-indent: 50pt;}</p>
line-height	<p>Управляет интервалами между строками текста.</p> <p>P {line-height: 50 % }</p>
word-spacing	<p>Устанавливает интервалам между словами. Можно использовать отрицательные значения</p> <p>P {word-spacing: 50 % }</p>
letter-spacing	<p>Устанавливает интервалам между буквами</p> <p>P {letter-spacing: 50 pt}</p>
vertical-	<p>Устанавливает вертикальное положение элемента. Может</p>

align	принимать следующие значения: baseline middle sub super text-top text-bottom top bottom P {vertical-align: 50 pt}
-------	--

Примеры

Подчеркнутый заголовок (underline)

[Это ссылка без подчеркивания \(none\)](#)

~~Перечеркнутый курсивный текст (line-through)~~

Текст с подчеркиванием сверху (overline)

каждая первая буква слова большая

Заголовок по центру

Абзац с отступом для лучшего восприятия текстов при оформлении ваших страничек


Текст с уменьшенным интервалом на 50% между строками. Данным метод можно применять в декоративных целях

Увеличим промежутки между словами

Уменьшим промежутки между буквами

Vertical-align

Формула спирта

C₂H₅OH 

Границы

Спецификация CSS2 описывает атрибут стиля border, который позволяет задать вид границ вокруг объектов. Атрибут border может быть применен к любым объектам. Могут быть установлены следующие параметры границы:

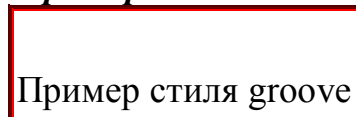
border-width	Ширина границы. Может быть задана значением в пикселях или с помощью зарезервированных слов thin (тонкая), medium (средняя), thick (толстая). table {border-width: 2px;}
border-color	Цвет границы. Может принимать значение transparent для задания невидимой, но имеющей ширину, границы, что иногда полезно table {border-color: green;}
border-style	Задаёт стиль рисования границы. Может принимать следующие значения: none (по умолчанию), hidden, dotted, dashed, solid, double, groove, ridge, inset, outset

	<pre>table {border-style: dashed;} table {border-style: groove;}</pre>
border-collapse	<p>Задаёт стиль рисования таблицы. Может принимать следующие значения:</p> <p><i>separate</i> (по умолчанию) - Ячейки отделены друг от друга</p> <p><i>collapse</i> - ячейки не имеют промежутков между собой</p> <p><TABLE STYLE="border-collapse:collapse"></p>

Примечание

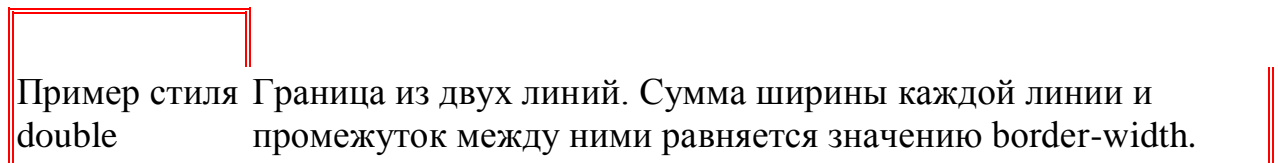
Спецификация CSS2 позволяет задать стили для каждой из сторон границы. Например: border-сторона[-атрибут]. Где сторона может принимать следующие значения: top (верхняя граница), bottom (нижняя граница), left (левая граница) и right (правая граница). Таким образом { border-bottom-width: 1px } будет означать, что ширина нижней границы равна 1 пикселу.

Примеры



Верхняя часть заголовка зеленого цвета (dotted)

Верхняя граница окрашена в синий цвет Пример стиля dashed



Пример использования тега DIV

Пример атрибутов границы элемента DIV

Здесь приводится код примера

```
<DIV STYLE="background: silver;
border-style: dashed;
border-width: thick;
border-top-color: red;
border-right-color: green;
border-bottom-color: blue;
border-left-color: yellow;">Пример атрибутов<br>
границы элемента DIV</DIV>
```

Это обычный вид таблицы	Ячейки отделены друг от друга
Теперь попробуйте щелкнуть мышкой внутри	Ячейки должны

**Контрольные вопросы:**

1. Свойства шрифта?
2. Свойства текста?
3. Какие виды границ существуют?

Лабораторная работа №12
Тема: Курсоры, полоса прокрутки в CSS

Цель работы: изучить виды курсоров, полосы прокрутки в CSS

Internet Explorer позволяет задавать стили для курсоров. Некоторые стили доступны только для IE 6.

all-scroll	Курсор со стрелками во все четыре стороны и точкой в центре, показывающий на возможность скроллинга страницы в любом направлении I { cursor: all-scroll; }
auto	По умолчанию. Браузер определяет самостоятельно, какой курсор требуется в данном контексте I { cursor: auto; }
col-resize	Курсор со стрелками влево-вправо и вертикальной разделяющей полоской. Используется для индикации возможности изменения размеров по горизонтали H4 { cursor: col-resize; }
crosshair	Курсор-крест H4 { cursor: crosshair; }
default	Стандартный курсор, используемый системой H4 { cursor: default; }
hand	Рука с вытянутым указательным пальцем. Используется при гиперссылке H4 { cursor: hand; }
help	Стрелка с вопросительным знаком. H3 { cursor: help; }
move	Курсор со 4 стрелками, показывающий возможность перемещения H2 { cursor: move; }
no-drop	Рука с перечеркнутым кружочком. Нельзя сбросить объект в текущую позицию курсора TD { cursor: no-drop; }

not-allowed	Перечеркнутый круг. Данная операция не поддерживается TD { cursor: not-allowed; }
pointer	Идентична стилю hand TD { cursor: pointer; }
progress	Песочные часы, показывающие на продолжение операции TD { cursor: progress; }
row-resize	Курсор со стрелками вверх-вниз и вертикальной разделяющей полоской. Используется для индикации возможности изменения размеров по вертикали TD { cursor: row-resize; }
text	Текстовый курсор-каретка TD { cursor: text; }
url(uri)	Ваш собственный курсор. Поддерживаются файлы .cur и .ani TD { cursor:url(eflogo.cur); }
vertical-text	Горизонтальная текстовая каретка для вертикального текста TD { cursor: vertical-text; }
wait	Курсор, показывающий, что система занята и требуется подождать TD { cursor: wait; }
*-resize	Курсоры, показывающие возможность потянуть за край окна. Вместо символа * используйте N, NE, NW, S, SE, SW, E, или W, определяющие направление стрелок TD { cursor: n-resize; }

Примеры

Проведите курсоры над ячейками таблицы для демонстрации стилей

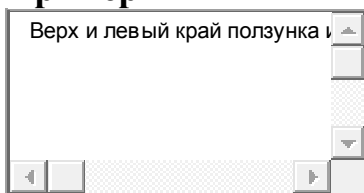
all-scroll	auto	col-resize	crosshair	default	hand
help	move	no-drop	not-allowed	pointer	progress
row-resize	text	vertical-text	wait		
n-resize	s-resize	ne-resize	sw-resize		
nw-resize	se-resize	e-resize	w-resize		
url(new.cur)	url(kiss.ani)				

Данные настройки применимы ко всем элементам, имеющим полосы прокрутки: сама страница (элемент BODY), текстовый блок (TEXTAREA) и т.д.

scrollbar-3dlight-color	Определяет или устанавливает цвет верха и левой части ползунка и кнопок со стрелками на полосе прокрутки body { scrollbar-3dlight-color: green; }
-------------------------	--

scrollbar-arrow-color	Устанавливает или определяет цвет стрелок на кнопке со стрелками body { scrollbar-arrow-color: red; }
scrollbar-base-color	Устанавливает или определяет цвет основных элементов ползунка: ползунок, кнопок со стрелками, дорожки для ползунка, если не определены параметры в scrollbar-face-color body { scrollbar-base-color: green; }
scrollbar-darkshadow-color	Устанавливает или определяет цвет тени для ползунка и кнопок со стрелками body { scrollbar-darkshadow-color: red; }
scrollbar-face-color	Устанавливает или определяет цвет ползунка и кнопок со стрелками. Также, если вы не задали параметр SCROLLBAR-TRACK-COLOR, у вас изменится цвет дорожки body { scrollbar-face-color: green; }
scrollbar-highlight-color	Устанавливает или получает цвет подсветки, создающий эффект объёмности. Это цвет, который окаймляет освещённую часть кнопочки. Когда кнопка не нажата, то цвет заливает левый верхний угол и стороны между ним, когда нажата - нижний правый угол body { scrollbar-highlight-color: green; }
scrollbar-shadow-color	Схоже с scrollbar-darkshadow-color body { scrollbar-shadow-color: green; }
scrollbar-track-color	Устанавливает или получает цвет дорожки для ползунка body { scrollbar-track-color: aqua; }

Примеры



Контрольные вопросы:

1. Какие виды курсоров существуют?
2. Виды полос прокрутки?

Лабораторная работа №13

Тема: Пишем первую программу (скрипт) на PHP

Цель работы: научиться программировать на PHP

Знакомство с любым языком программирования начинается с программы Hello, World! Текст программы приведен в листинге 1.

Листинг 1. Первая программа (скрипт) на **PHP**

```
echo "<html><body><h1>";  
echo "Hello, $my_name";  
echo "</h1></body></html>";  
?>
```

Прежде, чем запустить программу, ее нужно правильно «установить» на сервере. Для этого сохраните сценарий под именем `first.php` и скопируйте его в каталог `DocumentRoot` вашего сервера. Если вы работаете под **Linux**, таким каталогом будет `/var/www/html` (в старых версиях — `/home/httpd/html`). Если вы работаете под Windows, каталогом **DocumentRoot** будет `c:\www\html`.

Выполнив вышеуказанные рекомендации по установке, в строке браузера введите `http://localhost/first.php?my_name=Denis`. В результате в окне браузера вы увидите **Hello, Denis** (см. рис. 1).

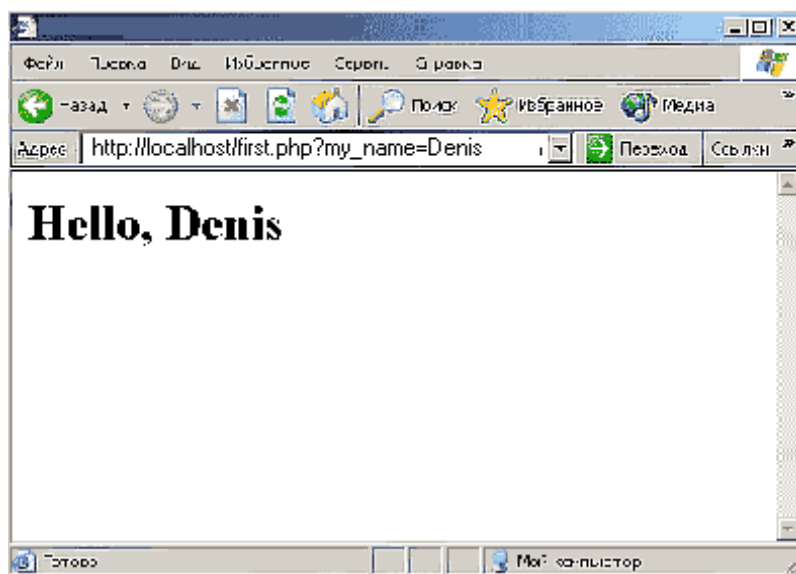


Рис. 1. Первая PHP-программа

Если теперь страницу, отображенную в браузере, просмотреть в виде **HTML**-кода, то можно увидеть следующее:

```
<html><body><nl>  
Hello, Denis  
</hl></body></html>
```

Как видите **PHP**-код выполнен, подставил нужное значение переменной `$my_name` и выдал готовую **HTML**-страницу.

Теперь рассмотрим саму программу. Код **PHP** заключается в специальные теги . После тега начала сценария **echo**. Оператор **echo** осуществляет вывод информации. Выводимая оператором **echo** строка заключается в кавычки. Первый оператор выводит теги **HTML, BODY и HI**.

Второй оператор `echo "Hello, $my_name";` выводит слово **Hello** и значение переменной **\$my_name**. Чтобы как-то отличать переменные от обыкновенных строк, перед именем переменной ставится знак доллара. И это касается не только строк, в любом случае перед именем переменной нужно ставить знак доллара. Присваивание переменной какого-либо значения осуществляется оператором присваивания, который обозначается значком «=» (равно). Примеры использования переменных: `$1 = 0; $d=7; $i=5 + $d// $1=12`

Еще один пример:

```
$word1 = "Hello";  
$word2 = "world";  
echo "$word1, $word2";  
echo "$word1, word2";
```

Первый оператор выведет в браузер строку «**Hello, world**», а второй — «Hello, word2». В этом примере мы определили две переменные — **\$word1** и **\$word2**. Так как мы забыли дописать символ доллара во втором операторе, **PHP** воспринял «word2» как обыкновенную строку, а не как переменную.

Третий оператор нашей программы в особых комментариях не нуждается, как и тег окончания **PHP**-программы (?>).

Модифицируем немного наш текст программы **first.php**, чтобы он выглядел так:

```
<?  
echo "<html><body><hl>";  
echo "Hello, $my_name";  
echo "<br> $var";  
echo "</hl></body></html>";  
?>
```

Если мы не передадим программе параметр **\$var**, то есть, как обычно, запустим программу только с параметром **\$my_name**, то в окне браузера мы увидим сообщение о том, что переменная `$var` не определена. Для того чтобы избавиться от этой ошибки, непосредственно перед оператором `echo " $var";` добавьте строку: `$var=" "`;

Контрольные вопросы:

1. Структура программы на PHP?
2. Какие операторы используются для создания программы на PHP?
3. Как описываются переменные?

Лабораторная работа № 14

Тема: «Создание простого изображения на PHP»

Цель работы: научиться создавать изображения на PHP

Для создания изображения необходимо воспользоваться функцией `imagecreate()`, которой в качестве обязательных целочисленных параметров передаются ширина и высота создаваемого изображения. В качестве единиц измерения используются пиксели.

Пример создания простого изображения:

```
<?php
header("Content-type: image/gif");
$image=imagecreate(50,50);
$color=imagecolorallocate($image, 255, 0, 0);
imagefilledrectangle ($image, 0, 0, 50, 50, $color);
imagegif($image);
imagedestroy($image);
?>
```

Сначала требуется послать пользователю HTTP-заголовок, определяющий тип пересылаемого содержимого. Так как будет создаваться изображение формата GIF, следует использовать заголовок `Content-type: image/gif`, который пересылается при помощи функции `header()`. После этого необходимо создать изображение при помощи функции `imagecreate`.

После этого при помощи функции `imagecolorallocate()` выбирается основной цвет изображения. После того как изображение создано и для него выбран цвет, можно приступить к отрисовке содержимого. В данном случае закрасенный квадрат создается при помощи функции `imagefilledrectangle()`. Эта функция создает прямоугольник, заполненный указанным цветом.

После того как изображение создано, его следует вывести на экран. Для этого в примере использовалась функция `imagegif()`, которая выводит изображение в формате GIF. По окончании работы с изображением следует освободить ресурсы, которое оно занимало, при помощи функции `imagedestroy()`.

Для подписания изображения можно использовать следующий код:

Красный квадрат

```
<br>
```

```

```

Пример создания треугольника:

```
<?php
header("Content-type: image/gif");
$image=imagecreate(200, 200);
$white=imagecolorallocate($image, 255, 255, 255);
$blue=imagecolorallocate($image, 0, 0, 255);
$points=array(0, 0, 150, 0, 100, 150);
imagefilledpolygon($image, $points, 3, $blue);
imagegif($image);
imagedestroy($image);
?>
```

Контрольные вопросы:

1. Как создаются простые изображения на PHP?
2. Какие функции используются для создания изображения?
3. Что означает функция imagecolorallocate ()?

Лабораторная работа №15

Тема: Обрамление блока PHP-команд тегами. Функции.

Цель работы: изучить функции в PHP

Могут использоваться такие варианты тегов:

- Стандартные - <?php ?>
- Короткие <? ?>
- ASP <% %>
- Программные <SCRIPT LANGUAGE="php"> </SCRIPT>

Функция print ()

Функция print () - выводит данные, естественно - в основном в окно браузера. Может использоваться без скобок, в этом смысле она исключение из правил.

Итак. Мы уже знаем достаточно, чтобы написать **первый скрипт**. Создаваем файл с **расширением php**. В файле пишем следующий текст:

```
<?php
print "Первый скрипт";
?>
```

Вызываем файл, как говорилось выше http://localhost/имя_файла. Видим, естественно только - Первый скрипт. Но не в этом прелесть. Из окна браузера откройте "**В виде HTML**" и тут не увидите ни тегов ни функции print. То есть весь текст скрипта обрабатывается на сервере и клиенту отдает только то, что сами желаем.

Итак. Следующее, что надо понять, это что текст самого скрипта (или его частей) правильно включать в стандартную HTML структуру. То есть правильнее весь текст документа с первым скриптом должен выглядеть примерно так:

```
<HTML>
  <HEAD>
    <TITLE>
      Заголовок страницы
    </TITLE>
  </HEAD>

  <BODY>

    <?php
      print "Первый скрипт";
    ?>

  </BODY>
</HTML>
```

Естественно, включение в документ любых тегов и "прочего" не ограничено.

ПЕРЕМЕННЫЕ

Имена переменных в PHP должны начинаться с \$. В имени могут быть буквы (латинские), цифры и подчеркиватели. Нельзя использовать. Простейший пример использования переменных (для простоты и лаконичности здесь и далее примеры приводятся без HTML структуры):

```
<?php
$a="Слово 1";
$b="Слово 2";
print "$a $b";
?>
```

Типы данных (переменных)

В PHP поддерживаются следующие типы данных:

- Integer - целое
- Double - с плавающей точкой
- String - строка
- Boolean - булева

- Array - массив
- Object - объект (но к нему мы не скоро доберемся).

Вывести тип переменной позволяет функция **gettype()**. Попробовать на практике разные типы переменных позволяет простенький пример, приведенный ниже:

```
<?php
$a=1;
print "$a ";
print gettype($a);
print "<br>";
$b=1.5;
print "$b ";
print gettype($b);
print "<br>";
$c="слово";
print "$c ";
print gettype($c);
print "<br>";
$d=true;
print "$d ";
print gettype($d);
?>
```

Запустите его и в разных строках у Вас должны появиться типы переменных - Integer, затем Double, String и Boolean. На этом примере также видно, что теги, вписанные в функцию print срабатывают вполне корректно. Ну и напоследок -

Изменение типов переменных

Все просто - изменить тип переменной позволяет функция **settype(переменная, новый тип)**. Приведенный ниже пример поясняет изменение типов. Сначала присваиваем переменной не целое число ($\$a=5.5$), затем преобразуем его в строку, затем в целое и в заключении - в boolean:

```
<?php
Error_Reporting(E_ALL & ~E_NOTICE);
$a=5.5;
print "$a ";
print gettype($a);
print "<br>";
settype ( $a, string );
```

```
print "$a ";
print gettype($a);
print "<br>";
settype ( $a, integer );
print "$a ";
print gettype($a);
print "<br>";
settype ( $a, boolean );
print "$a ";
print gettype($a);
?>
```

Непонятная строка `Error_Reporting(E_ALL & ~E_NOTICE)` в начале скрипта необходима, чтобы сервер не выдавал предупреждений об ошибках.

Контрольные вопросы:

1. Какие типы переменных используются в PHP?
2. Как изменить тип переменных?
3. Какие варианты тегов могут использоваться в PHP?

Список использованной литературы

1. Петров В.Н. Информационные системы. Учебник. СПб: Издательство "Питер", 2002.
2. Прохоров А.Н. Интернет: как это работает.- СПб: БХВ-Петербург, 2004.
3. Филимонов А.Ю. Протоколы Интернета.- СПб: БХВ-Петербург, 2003.
4. Найк Д. Стандарты и протоколы Интернета. Пер. с англ.- М.: 1999.
5. Ганеев Р.М. Проектирование интерактивных WEB-приложений. М.: 2001.
6. А. Павлов. CGI-программирование: учебный курс. - СПб: Питер, 2000.
7. А.В. Картузов, Д.В. Николенко. Программируем на языке Java: краткий курс.- СПб: Наука и техника, 2001.
8. Ливингстон Д., Белью К., Браун М. Perl 5. Web - профессионалам: Пер. с англ. - К.: Издательская группа BHV, 2001.
9. Косентино К. PHP. Web - профессионалам: Пер. с англ. - К.: Издательская группа BHV, 2001.
10. Э. Кингсли-Хью, К. Кингсли-Хью. JavaScript 1.5. Учебный курс.- СПб: Питер, 2001.
11. Бишоп Д., Java 2. Питер, 2002
12. Д. Вебер, Технология Java в подлиннике: Пер. с англ.- СПб.: БХВ-Петербург, 2000.
13. В.Г. Олифер, Н.А. Олифер. Компьютерные сети. Учебник.- СПб: Питер, 1999.