

Учебник HTML для начинающих.

Версия для печати раздела "Учебник HTML" сайта www.webremeslo.ru

Введение

Этот учебник целиком и полностью посвящён языку HTML, без знаний которого не обойтись в создании какой либо веб-страницы. HTML язык - это основа, база которую просто так необходимо знать любому начинающему веб-мастеру! Этот язык, сам по себе (в чистом виде), достаточно простой, думаю его с лёгкостью может освоить любой "продвинутый" шестиклассник.. в тоже время он позволяет самостоятельно создавать полноценные страницы и сайты.

Рассчитан этот HTML учебник для начинающих "с нуля" людей, которые только только решили освоить веб-ремесло, поэтому старался писать его как можно более развернуто и что называется "доходчиво", в тоже время стараясь охватить весь курс до мелочей, так что может быть и опытный веб-мастер найдет в нём для себя что-то новое или будет использовать его в качестве "шпаргалки", изложенной в прилагаемом **справочнике тегов**: (<http://www.webremeslo.ru/spravka/spravka.html>). Где нет "воды" и нудноватого для уже опытного человека "разжёвывания" материала.

Ну и сразу к делу.. начнем с классики..

Откройте блокнот, да да, тот самый блокнот.. Пуск> Стандартные> Блокнот и напишите в нем следующий текст:

```
<html>
<head>
  <title>Моя первая страничка </title>
</head>
<body>
  Привет мир!!!
  <br>
  Меня зовут (здесь впишите Ваше имя), это моя первая страничка!
</body>
</html>
```

Далее, сохраните этот текст как **html** документ, название придумайте сами.. главное чтобы расширение было **html**

Повторюсь, чтобы не было казусов.. в блокноте кликаем по меню "Файл", выбираем "Сохранить как.." в строке "Имя файла" пишем: **index.html**, просто по умолчанию блокнот предлагает сохранить файл с расширением ***.txt**, а нам нужно расширение ***.html**

Далее открываем этот файл при помощи Вашего браузера.. ну к примеру того же Internet Explorer (правой кнопкой по нашему файлу.. "Открыть с помощью.." Internet Explorer)

Поздравляю, Вы написали свою первую страничку!!!

Если в Вашей душе загорелась искорка, проснулась тяга к знаниям и творчеству, предлагаю планомерно ознакомиться с главами этого учебника. В них на доступном языке с множеством примеров я постараюсь объяснить то, что мы только что написали, а так же как сделать этот документ более привлекательным, вставить в него графику, таблицы, прописать ссылки.. в общем, сделать его полноценным сайтом, заслуживающим место во всемирной паутине.

Глава 1

Что такое HTML язык и с чем его едят..

Начну, пожалуй, с начала..

HTML язык по своей сути не является языком программирования.. он является языком разметки гипертекстовых документов. Иными словами он отвечает за расположение в документе Ваших текстов, рисунков, таблиц.. предназначенных для жизни в сети Интернет. Заставить его посчитать, сколько будет дважды два невозможно, в нем нет логических функций, зато красиво и главное легко выложить информацию о том, что дважды два будет четыре - это запросто. Читается этот язык при помощи знакомых Вам программ, именуемых браузерами (обозревателями), которые "знают" стандартные команды html языка, и "пережевывая" их выводят на монитор компьютера документы в том виде, в котором хочет представить их веб мастер - составитель документа.

Теперь о командах - их называют дескрипторами, но чаще - тегами.

Вернемся к нашей первой страничке.. итак мы написали:

```
<html>
<head>
  <title>Моя первая страничка</title>
</head>
<body>
  Привет мир!!!
  <br>
  Меня зовут Карлсон, это моя первая страничка!
</body>
</html>
```

Так вот то что написано между <...> - называют тегами они не видны читателю, заглянувшему на Вашу страницу, зато хорошо видны браузеру, который наткнувшись на тег **<html>** понимает его как сигнал к тому, что далее будет документ, который необходимо прочитать и вывести на монитор в нужном виде.. а вот тег **</html>** говорит о том что документ закончился и от него, то бишь браузера, больше ничего не требуется и он может отдыхать с чистой совестью.

Так что же мы написали? и как это читает браузер?

Мысли браузера:

<html> - начало документа.. опять работы привалило..
<head> - смотрим словарь английского переводим "голова" ...в голове моей опилки не беда!! ... здесь указывается основная служебная информация о документе.. ну и что они там от меня хотят?
<title> - "название" значит.. это в шапке окна нужно написать его название:
Моя первая страничка - опять чайник тренируется..
</title> - все название закончилось.. можно идти дальше..
</head> - ага и в голове кроме названия больше никаких вздорных мыслей не держим..
<body> - "тело" документа все, что написано ниже выставляем на всеобщее обозрение
Привет мир!!! - как мило! Достали уже!!!
**
** - переносим текст на следующую строчку.. я даже догадываюсь, что в ней будет...
Меня зовут (здесь Ваше имя), это моя первая страничка! - ну да так и есть.. ничего

пооригинальнее придумать не могут..

</body> - что всё что ли? Больше ничего не отображать?

</html> - ну вот и конец!! Не будить, не кантовать, при пожаре выносить первым!!

Вот так примерно и происходит чтение нашей странички.. Как видите браузер довольно своенравный тип, поэтому команды ему нужно подавать чёткие и ясные, иначе он разругается.. причем в слух.. а по сему запомним следующие вещи:

1) Необходимо раз и навсегда запомнить что если есть открывающий тег **<...>** то обязательно должен быть и закрывающий **</...>**

Хотя есть и исключения как например у нас тег **
** - он закрытия не требует потому что говорит лишь о том, что следует писать с новой строки. Кстати попробуйте добавить еще пару тройку таких же перед строчкой "Меня зовут.." и Вы увидите, что в результате она заметно опустилась вниз.. (Ну естественно в блокноте сохраните изменения, а в браузере нажмите кнопку "обновить")

2) Все документы должны иметь вот такой шаблон кода:

<html>- начало документа
<head>- начало головы
</head>- закрытие головы
<body>- начало тела
</body>- закрытие тела
</html>- конец документа

Данные теги являются обязательными! Писать их необходимо всегда для каждой новой странички, и только в таком порядке! Не пытайтесь перевернуть всё с ног на голову..

3) О порядке:

Открывающий и закрывающий тег по типу **<...>** **</...>** представляет собой своего рода ёмкость, ящик в который могут складываться другие теги - ящички поменьше.. следовательно, согласно логики документ должен выглядеть так:

<Тег "большой ящик">
 <Тег "ящик средний">
 <Тег "ящик маленький">
 содержание
 </Тег "ящик маленький" >
 </Тег "ящик средний" >
</Тег "большой ящик">

Если писать, например, так:

<Тег "большой ящик">
 <Тег "ящик средний">
 <Тег "ящик маленький">
 содержание
</Тег "большой ящик">
 </Тег "ящик маленький">
 </Тег "ящик средний">

То получится белиберда, которая в голове не укладывается, "большой" уже закрыт, а из него "торчит" "маленький", который заперт в "среднем", а "содержание" рассыпалось везде.. Такое даже человеку трудно представить, что уж там про браузер говорить.. Чётко структурируйте код Вашей странички иначе ничего работать не будет..

Ну вот мы научились писать простой текст, начало положено! в следующей главе я расскажу о том, что с ним можно сделать..

Полезные советы:

- При написании сайта создайте папку в любом удобном месте на Вашем жёстком диске, называйте как угодно, лишь бы понятно было.. в эту папку сохраняете Ваши странички, присваивая им осмысленные имена.. варианты типа aaa.html, 123.html приведут к беспорядку и путанице... На данном этапе этот совет не покажется дельным, но в дальнейшем это значительно облегчит Ваш труд. Представьте к примеру хотя бы 20-30 таких файлов названия которых нужно держать в голове, чтобы связать их перекрёстными ссылками. Порядок, прежде всего!
- При написании кода советую придерживаться "хорошей манере письма", то есть писать теги "лесенкой" по мере вложенности одного тега в другой. Со временем Вы поймете, что читать код написанный вот так:

```
<html>
<head>
  <title>Моя первая страничка</title>
</head>
<body>
  Привет мир!!!
  <br>
  Меня зовут Карлсон, это моя первая страничка!
</body>
</html>
```

Значительно легче, чем так:

```
<html>
<head>
<title>Моя первая страничка</title>
</head>
<body>
Привет мир!!!
<br>
Меня зовут Карлсон, это моя первая страничка!
</body>
</html>
```

И уж тем более чем вот так:

```
<html><head><title>Моя первая страничка</title></head><body>Привет мир!!!<br>Меня зовут Карлсон, это моя первая страничка!</body></html>
```

Хоть это и дело привычки.. всё же лучше привыкнуть писать "разборчиво".

Глава 2

Редактируем текст на своей странице..

Вы наверняка не раз печатали документы в текстовых редакторах, так вот.. что можно сделать с текстом в том же Word, то же можно сделать и с текстом на нашей страничке.

В этой главе я расскажу о том, как изменить шрифт и его размер, как сделать текст жирным, курсивом, подчёркнутым и о том как его выравнивать по краям или центру документа.

Параграф.

Я думаю, Вы знаете, что такое параграф.. это такой кусочек текста, одно или несколько предложений, который в книгах обычно печатается с новой строки, тем самым, выделяя этот текст из основной массы, Книгу разбитую на параграфы легко читать, потому что, как правило, одному параграфу соответствует одна мысль или логическая часть текста.

Так вот, для того чтобы на странице сайта разбить текст на параграфы, необходимо воспользоваться тегом **<p>** - собственно параграф.

Параграф имеет атрибут **align** "выравнивание" который в свою очередь может быть равен тому ли иному значению.

Рассмотрим на примерах:

С помощью параграфа можно расположить наш текст по центру:

<p align="center">Привет мир!!!</p>

По левому краю:

<p align="left">Привет мир!!!</p>

По правому краю:

<p align="right">Привет мир!!!</p>

Или же обоим краям документа:

<p align="justify">Привет мир!!! - здесь нужен текст подлиней чтобы эффект был хорошо виден при открытии документа**</p>**

Давайте слегка изменим нашу первую страничку:

```
<html>
<head>
  <title>Мой первый сайт </title>
</head>
<body>
  <p align="center">Привет мир!!!</p>
  <br>
  <p align="justify">
```

Меня зовут Карлсон! Я в меру упитанный мужчина - это моя первая страничка!
Здесь я хочу найти себе новых друзей, для того чтобы вместе гулять по крышам! Я
очень очень сильно люблю варенье!!! С нетерпением буду ждать Вашего

приглашения на чай. Прилечу!!

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

Так уже лучше, не правда ли?

Запомним некоторые вещи:

1) Тег **<p>** не может содержать в себе других параграфов, то есть писать вот так:

```
<p>
  <p>
    </p>
</p>
```

Нельзя! - это нелогично, как может один параграф содержать в себе другой?

2) Так же, нельзя писать пустые теги без текста или других тегов.

<p> здесь, что-то обязательно должно быть!!!</p>

3) По умолчанию Ваш текст выравнивается браузером по левому краю, так что если Вам так и надо атрибут **align="left"** для параграфа можно не указывать.

4) Тег **<p>** подразумевает в себе перенос строки, если это Вам не нужно, используйте вместо тега **<p>** тег **<div>** данный тег является альтернативой тегу **<p>** пишется так:

```
<div align="center">Привет мир!!!</div>
<div align="left">Привет мир!!!</div>
<div align="right">Привет мир!!!</div>
<div align="justify">Привет мир!!!</div>
```

Все то же самое, только данный тег не будет переносить текст на следующую строку и в него можно вставлять тег **<p>** по принципу:

```
<div>
  <p align="left">Пишем слева</p>
  <p align="right">Пишем справа</p>
</div>
```

А вообще тег **<div>** многофункциональный.. и по своей сути он является пустым блоком-контейнером, который может содержать в себе как текст, так и другие теги.. в общем, хочу сказать, что работа с текстом это далеко не основная задача тега **<div>**, но об остальных возможностях данного тега мы поговорим позже..

5) Еще одним способом выравнивания текста по центру является использование тега **<center>** любое содержание взятое в данный тег выравнивается по центру экрана. Пишется так:

<center>Привет мир!!!</center>

Заголовки

В наборе тегов html языка имеется шесть типов заголовков:

<h1> Привет мир!!! </h1>

<h2> Привет мир!!! </h2>

<h3> Привет мир!!! </h3>

<h4> Привет мир!!! </h4>

<h5> Привет мир!!! </h5>

<h6> Привет мир!!! </h6>

Думаю с этим понятно.. **<hx>** тег заголовка, где значение **x** является величиной буковок (может быть от одного до шести). Помните, что после использования того или иного заголовка происходит перенос строки - на то он и заголовок.

Заголовки дело хорошее и достаточно удобное, но ими можно выделять только маленькие кусочки текста, а что если нам надо выделить весь текст?

Знакомимся тег **** в переводе на русский - "шрифт".. Тег **** помимо прочих атрибутов, о которых ещё пойдет речь, имеет атрибут **size** - размер.

Пишется и выглядит это так:

Привет мир!!!

Привет мир!!!

Привет мир!!!

**Привет мир!!! **

Привет мир!!!

Добавим эти теги на нашу страницу.

```
<html>
<head>
  <title>Мой первый сайт</title>
</head>
<body>
  <center><h2>Привет мир!!!</h2></center>
  <br>
  <p align="justify">
    <font size="+1">Меня зовут Карлсон! Я в меру упитанный мужчина - это моя первая
    страничка! Здесь я хочу найти себе новых друзей, для того чтобы вместе гулять по
    крышам! Я очень очень сильно люблю <font size="+2">варенье!!!</font> С нетерпением
    буду ждать Вашего приглашения на чай. Прилечу!!</font>
  </p>
</body>
</html>
```

Попробуйте поэкспериментировать с размерами шрифта и заголовком, подставьте свои значения, измените расположение тегов..

Цвет

Для придания страничке красивого вида не обойтись без палитры с красками..

В **html** языке используется своя палитра красок. Вот основные цвета, выглядят они так:

#000000 black	#ffffff white	#ff0000 red
#ffa500 orange	#ffff00 yellow	#008000 green
#00ffff cyan	#0000ff blue	#800080 purple

Полная палитра базовых красок приведена [здесь](http://www.webremeslo.ru/spravka/spravka1.html):
[\(http://www.webremeslo.ru/spravka/spravka1.html\)](http://www.webremeslo.ru/spravka/spravka1.html)

Один и тот же цвет можно задать двумя способами:
используя шестнадцатеричное значение цвета RGB - например **#008000** - зелёный цвет, либо
используя константы цвета - **green** (для тех кто учил французский, **green** -зелёный.)

Краски есть, давайте рисовать!

Мы уже знакомы с тегом **** у него есть еще один атрибут - **color**.

Так вот, если к примеру написать так:

****Привет мир!!!**** - То цвет шрифта станет красным.
Попробуйте..

А можно так:

****Привет мир!!!**** - Будет тоже самое..

Но советую писать всё же шестнадцатеричным числом, во-первых, по понятным причинам не для всех оттенков цветов есть своё название, а во-вторых, поговаривают, что не все браузеры знают названия тех или иных красок...

Есть еще один способ изменить цвет текста. Тег **<body> </body>** "тело" - имеет атрибут **text** - "текст" если присудить этому атрибуту любой цвет из доступной палитры то весь текст в нашей странице окрасится, кроме тех мест, где мы "принудительно" указали другой цвет.

В строчке, где стоит открывающий тег **<body>** пишем так:

<body text="#ff8c40 ">

Теперь весь текст у нас стал оранжевым кроме заголовка "Привет мир!!!" который мы отдельно перекрасили в красный.

А вот атрибут тега **<body> bgcolor** и его значение задает цвет фона страницы

<body bgcolor="#40caff"> - залили всё голубым..

Раскрасьте свою страницу на свой лад.. пробуйте, экспериментируйте!

Сейчас Вам нужно понять принцип построения html кода иначе рано или поздно станете в тупик.. конечно, я и дальше буду выкладывать готовые примеры, но нет ничего лучше чем самостоятельная практика.

На данный момент у меня получилось вот так: .. а у Вас?

```
<html>
<head>
```



```

<title>Мой первый сайт </title>
</head>
<body text="#484800" bgcolor="#e8e8e8">
  <center>
    <h2>
      <font color="#008000">Привет мир!!!</font>
    </h2>
  </center>
  <p align="justify">
    <font size="+1">Меня зовут Карлсон! Я в меру упитанный мужчина - это моя
первая страничка! Здесь я хочу найти себе новых друзей, для того чтобы вместе гулять
по крышам! Я очень очень сильно люблю <font size="+2"
color="#ff0000">вареньеё!!!</font> С нетерпением буду ждать Вашего приглашения на
чай. Прилечу!!
    </font>
  </p>
</body>
</html>

```

Обратите внимание на то, как пишется код, если в теге присутствует два и более атрибута. В нашей строчке **<body text="#ff207b" bgcolor="#1a77f0">** у тега **<body>** два атрибута **text** и **bgcolor** мы просто пишем их подряд через пробел, не разделяя никакими другими знаками..

Стиль текста

Здесь все достаточно просто..

Итак, новые теги:

 	- Полужирный текст
<i> </i>	- Наклонный текст
<u> </u>	- Подчеркнутый текст
<strike> </strike>	- Перечеркнутый
<s> </s>	- Перечеркнутый (второй вариант, от первого ничем не отличается)
<tt> </tt>	- моноширинный шрифт
<small> </small>	- Малый
<big> </big>	- Большой
<sup> </sup>	- Верхний индекс
<sub> </sub>	- Нижний индекс

Ну думаю понятно.. текст заключённый между этими открывающими и закрывающими тегами приобретёт нужный нам стиль.

Вот пример на всякий случай...

```

<html>
<head>
  <title>Стили текста</title>
</head>
<body>
  <big><b><u>Научная статья.</u></b></big>
  <br>
  <br>
  Если разбавить дистиллированную воду Н<sub>2</sub>О сорока процентами

```

```

этилового спирта C<sub>2</sub>H<sub>5</sub>ОН то получится жидкость в
40<sup>о</sup> более известную широкой публике под названием <i>- водка.</i>
<br>
Впервые данную пропорцию <s>придлажил</s> предложил <tt>Дмитрий Иванович
МЕНДЕЛЕЕВ.</tt>
<br>
<br>
<br>
<small>Распитие спиртных напитков вредит вашему здоровью.</small>
</body>
</html>

```

Шрифт

Для того чтобы изменить шрифт документа необходимо дать указание браузеру, что мол от сел до сел показывать текст таким шрифтом. Для этого используем всё тот же тег **** и его атрибут **face** - лицо то бишь..

Пишется так:

****Эта строчка будет написана с помощью шрифта Arial****

Пример:

```

<html>
<head>
<title>Использование различных шрифтов</title>
</head>
<body>
<font face="arial">Мало кто знает, как много надо знать для того, что бы знать, как
мало мы знаем.</font>
<br>
<font face="times new roman">Мало кто знает, как много надо знать для того, что бы
знать, как мало мы знаем.</font>
<br>
<font face="comic sans ms">Мало кто знает, как много надо знать для того, что бы
знать, как мало мы знаем.</font>
<br>
</body>
</html>

```

Здесь необходимо отметить, что браузер использует библиотеку шрифтов, установленную на компьютере пользователя, и если вдруг указанного Вами шрифта в этой библиотеке не окажется, то он заменит его на тот который присутствует. Следовательно, чтобы не ударить face(ом) в грязь, не стоит указывать какие-то "экзотические" нестандартные шрифты, так как супер модный шрифт на компьютере Вашего друга сильно рискует превратиться в обыкновенный.

Предварительно отформатированный текст.

Если Вы обратили внимание, а если не обратили то знайте, что в браузерах текст набранный с помощью текстовых редакторов проходит "обработку" перед выводом его на экран компьютера. Так в набранном Вами тексте удаляются все переносы строк и лишние пробелы, пробелов между словами или просто символами может быть не более одного.

Проводится данная "обработка", для того чтобы на мониторах с разным расширением экрана текст переносился на следующую строку в тех местах где это действительно необходимо, а не там где были ранее расставлены пробелы и переносы строк.

Однако такая политика браузеров, в ряде случаев, не всегда оправдана.. Как например, написать стихи? Нет можно конечно после каждой строчки писать тег **
**, но есть вариант куда проще..

Знакомимся тег **<pre>**, текст заключённый в данный тег выводится браузерами на экран в том виде в котором он был набран, т.е. со всеми пробелами и переносами строк

Пример:

```
<html>
<head>
  <title>Пробелы и перенос строки</title>
</head>
<body>
  <pre>
      СЛОН.

  Дали туфельки слону.
  Взял он туфельку одну
  И сказал: - Нужны пошире,
  И не две, а все четыре!

      С. Я. Маршак.
  </pre>
</body>
</html>
```

Такие вот дела..

Полезные советы:

- Писать странички в блокноте конечно можно, но я Вам настоятельно рекомендую воспользоваться полноценным HTML редактором, благо их на рынке программного обеспечения великое множество. Я не буду давать их подробное описание, скажу лишь что они, как правило, несут в себе одинаковые стандартные наборы инструментов и отличаются друг от друга пожалуй только интерфейсом.. сравнивать их не берусь, так как они носят исключительно характер привычки пользования веб мастером тем или иным редактором html кода.

Расскажу про основные плюсы использования подобного редактора.

- Как правило, html редакторы имеют два окна, в одном Вы пишете код, а во втором сразу видите результат написанного! То есть теперь Вам не придётся постоянно сохранять текущий результат, чтобы открыть его браузером и оценить написанное.
- Как правило, они содержат в себе стандартные наборы шаблонов кода, в который Вы просто подставляете свою информацию, избавляя себя от прописывания тегов.
- Ну и "общие" удобства - сохранить, открыть, редактировать сразу несколько страниц сайта перелистывая их... много короче разных полезных штук.. одна подсветка синтаксиса чего стоит!

P.S. Однако, дальше буду писать эту книгу подразумевая, что Вы тренируетесь в стандартном блокноте... иначе мы друг друга не поймём.

- Немного об этике и здоровье глаз..

Если создаваемый Вами сайт его дизайн и назначение не требуют обратного, то советую избегать следующих вещей: Использовать яркие контрастные цвета фона и шрифта. Поверьте, когда у человека начинают слезиться глаза от чтения такой страницы, он резко теряет к ней интерес. Это же касается и размеров текста. Не стоит использовать огромные

заголовки, впрочем, как и содержание малюсенькими буквами - это напрягает, ищите золотую середину..

Глава 3

Рисунки.

В этой главе я расскажу о том как добавить на нашу страничку графические изображения фотографию например, или рисунок, а также о том, что можно с ними вытворять используя предложенный набор тегов html языка.

Путь к файлу

Итак, допустим Вы пишете сайт о себе любимом и хотите вставить в него свою фотографию, чтобы люди зашедшие на страничку смогли не только читать Ваши рукописи, но и вдоволь полюбоваться Вашим портретом.

Есть у Вас значит фотография, которая где-то лежит на Вашем жёстком диске.. копируем её и вставляем в ту папку (директорию) где уже лежит заготовка будущей страницы, ну то есть туда же, куда сохраняете блоком html документ. Так, вот для того чтобы теперь вставить её в нашу страничку к ней нужно указать путь.

делается это так:

Где **foto.jpg** это название Вашей фотографии в данном случае, так как фото лежит рядом с html документом, путь к ней мы не указываем.

Да, помните, тег **** не требует закрывающего тега!

Теперь приведу ряд примеров где путь указывается:

**** - Такая запись подразумевает, что в директории где расположен наш html документ есть папка myfoto в которой расположен файл foto.jpg

**** - Значит рядом с документом расположена папка myfoto, в ней еще одна папка с названием graphics, и уже в ней нужная нам фотография foto.jpg которую нужно выложить для всеобщего обозрения.

**** - А это значит, что фото размещено на уровень выше от документа

**** - Так соответственно на два уровня выше, сколько поставите../ настолько и поднимитесь.

Так же можно указывая место фотографии, ссылаться на тот или иной интернет ресурс

Ну что, давайте попробуем выложить фото.

Вот мой пример:

```
<html>
<head>
  <title>Моя первая страничка с фото</title>
</head>
<body text="#484800" bgcolor="#e8e8e8">
  <center>
    <h2> <font color="#008000"> Привет мир!!!</font> </h2>
  </center>
  <p align="justify">
```

```

<font size="+1">
  
  Меня зовут Карлсон! Я в меру упитанный мужчина - это моя первая страничка!
  Здесь я хочу найти себе новых друзей, для того чтобы вместе гулять по крышам! Я очень
  очень сильно люблю <font size="+2" color="#ff0000">вареньеё!!!</font> С нетерпением
  буду ждать Вашего приглашения на чай. Прилечу!!
</font>
</p>
</body>
</html>

```

Как видно из этого примера фотография появилась, но вместе со своим появлением сдвинула текст, а не хотелось бы..

Атрибуты тега

Поговорим о расположении изображений относительно текста.

Как и другие теги тоже имеет свои атрибуты.. уже знакомый нам атрибут **align** "выравнивание" применим и к данному тегу

 - фото слева от текста
 - фото справа от текста
 - текст выше фото
 - текст ниже фото
 - ну и соответственно текст посередине

Помимо **align** тег имеет еще ряд атрибутов, но сначала расскажу немного о пикселях. Пиксель (точка), наверное Вы не раз слышали это слово и думаю основная масса народа которая держала мышку в руках знает что это такое, но всё же.. Пиксель -это элементарная неделимая единица изображения. Каждый пиксель имеет свои координаты, например, самый верхний левый пиксель на мониторе имеет координаты x=1, y=1, а самый нижний правый в зависимости от разрешения монитора.. x=800, y=600 - будет соответственно при разрешении 800 на 600 точек. Впрочем, эта информация пригодится нам потом, а сейчас нужно усвоить, что все расстояния в графических изображениях меряются пикселями.. так картинка длиной 800 пикселей и шириной в 600 пикселей при указанном разрешении заполнит весь экран.

Теперь продолжим..

 - Атрибут **vspace** задаёт расстояние по вертикали от рисунка до текста, в данном случае мы задали расстояние в 15 пикселей
 - Расстояние по горизонтали соответственно
 - Ширина непосредственно самого изображения
 - Высота изображения. Если атрибуты **width** и **height** не использовать, то ширина и высота изображения по умолчанию будет равна реальным её размерам, без каких либо искажений.
 - Бордюр, рамка вокруг изображения и её толщина в пикселях.
 - **bordercolor** - это цвет рамки.
 -Атрибут **alt** - это описание изображения. Если навести курсор на наше фото и подержать его там несколько секунд, выскочит надпись -Это моя фотка!!!
 - альтернатива **alt** в данном случае.

А еще изображение можно сделать фоном страницы.. для этого используем атрибут **background** "фон" открывающего тега <body>

<body background="foto.jpg">

Теперь расскажу о том как разместить нашу фотографию в нужном месте страницы. На самом деле об этом я уже рассказывал.. вспомните вторую главу, где говорилось о том как можно выравнивать текст.. эти же теги применимы и к выравниванию изображения... так что, пожалуй, повторяться не буду, просто приведу пример, а Вы уж самими подумайте почему одно и тоже фото очутилось в том или ином месте..

```
<html>
  <head>
    <title>Расположение изображения</title>
  </head>
  <body>
    
    <center></center>
    <div align="left"></div>
    <br>
    <div align="right"></div>
    <br>
    <div align="center"></div>
  </body>
</html>
```

Итог главы:

[illegible]

Единственное, что Вам в этом примере не знакомо так это надпись - ** ** - это есть так называемый спецсимвол в данном случае - пробел.. о спецсимволах и их назначении я расскажу более подробно в 6 главе.

Честно говоря, мне мой пример не сильно по вкусу, неряшливо как-то.. Но цель показать, что можно делать с изображениями думаю достигнута..

Полезные советы:

- Используя графику, будьте осторожны.. не стоит размещать на своей странице большие весом мегабайтные изображения, пожалейте время и деньги посетителей страницы. Большой объём графики при низкой скорости интернета может привести к тому, что посетитель закроет Вашу страницу ещё до того момента как она загрузится.
- Помните, что есть пользователи, которые умышленно в своём браузере отключают загрузку рисунков, дабы сэкономить на трафике, а по сему рекомендую соблюдать следующие советы:
 - Если рисунки выполняют роль кнопок (см. Глава 5), обязательно подписывайте их (атрибут **alt**) к примеру: "Перейти на главную страницу" или "Скачать" дабы человек с отключенной графикой мог хоть как-то ориентироваться на Вашей странице.
 - Используя картинку в качестве фона документа дублируйте её заливкой фона каким либо подходящим для фона страницы цветом (атрибут **bgscolor** тега **<body>**)

вот так например:

```
<body bgscolor="#008000" background="fon.jpg">
```

Это будет значить, что если к примеру не загрузится Ваша изысканная зелёная текстура, фон будет просто зелёным - что лучше, нежели чем белый цвет в общем дизайнерском решении сайта.

Глава 4

Таблицы

Помимо прочих объектов в свой сайт Вы можете вставить таблицы.. и сразу забегаю вперёд скажу о том что они имеют не малую значимость при создании сайта. С помощью таблицы можно не только выложить ту или иную информацию, тарифную сетку или график дежурств к примеру, но и взяв её за основу полностью построить на ней свой сайт, таблицы порой незаменимы при верстке страницы, но об этом позже. А сейчас давайте научимся её рисовать..

Рисуем таблицу

Тег **<table>** задаёт начало и конец таблицы, любая таблица, как известно, состоит из строк и столбцов, для этого есть ещё два тега:

<tr> - строка таблицы

<td> - столбец таблицы

Вместе эти теги пишутся так:

```
<table>
  <tr>
    <td>ячейка</td>
  </tr>
</table>
```

Такая запись это самая маленькая таблица, в ней всего одна строка, содержащая один столбец - ячейку

Поставим перед собой задачу нарисовать таблицу из трёх строк и трёх столбцов, а заодно вспомним атрибут **border** "рамка", который добавит нам наглядности.

```
<html>
<head>
  <title>Таблица</title>
</head>
<body>
  <table border="1">
    <tr>
      <td>строка1 ячейка1</td>
      <td>строка1 ячейка2</td>
      <td>строка1 ячейка3</td>
    </tr>
    <tr>
      <td>строка2 ячейка1</td>
      <td>строка2 ячейка2</td>
      <td>строка2 ячейка3</td>
    </tr>
    <tr>
      <td>строка3 ячейка1</td>
      <td>строка3 ячейка2</td>
      <td>строка3 ячейка3</td>
    </tr>
  </table>
```

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

Разобрались? Если да, идем дальше..

Объединение ячеек.

Часто при работе с таблицами возникает необходимость объединить те или иные ячейки в одну.

На самом деле слово "объединить" здесь неуместно, использовал я его лишь для того, что бы вызвать ассоциации в Вашей голове. Вспомните тот же Word, добавив там таблицу Вы выделяете нужные ячейки и нажимаете кнопку "объединить".. здесь всё происходит не совсем так, и правильнее было бы говорить не "объединить" а "растянуть".

Эту задачу решают атрибуты **colspan** и **rowspan**

- **colspan** - определяет какое количество столбцов будет занимать данная ячейка
- **rowspan** - количество рядов занимаемое ячейкой

Предположим что в нашем примере нам необходимо "объединить" в третьей строке вторую и третью ячейку, для этого атрибуту **colspan** присваиваем значение 2 (растянуть на два столбца) и вставляем его в нужное место.

```
<html>
<head>
  <title>Таблица</title>
</head>
<body>
  <table border="1">
    <tr>
      <td>строка1 ячейка1</td>
      <td>строка1 ячейка2</td>
      <td>строка1 ячейка3</td>
    </tr>
    <tr>
      <td>строка2 ячейка1</td>
      <td>строка2 ячейка2</td>
      <td>строка2 ячейка3</td>
    </tr>
    <tr>
      <td>строка3 ячейка1</td>
      <td colspan="2">строка3 ячейка2</td>
      <td>строка3 ячейка3</td>
    </tr>
  </table>
</body>
</html>
```

Как видите ячейка растянулась на два столбца, но при этом сдвинула третью ячейку, которая собственно нам теперь не нужна, а по сему мы её просто напросто удаляем. Специально допустил эту оплошность, чтобы Вы поняли принцип данного действия.

Исправленный пример:

```
<html>
<head>
  <title>Таблица</title>
```

```

</head>
<body>
  <table border="1">
    <tr>
      <td>строка1 ячейка1</td>
      <td>строка1 ячейка2</td>
      <td>строка1 ячейка3</td>
    </tr>
    <tr>
      <td>строка2 ячейка1</td>
      <td>строка2 ячейка2</td>
      <td>строка2 ячейка3</td>
    </tr>
    <tr>
      <td>строка3 ячейка1</td>
      <td colspan="2">строка3 ячейка2</td>
    </tr>
  </table>
</body>
</html>

```

Теперь попробуем объединить весь первый столбец в одну ячейку, используя атрибут **rowspan**, ну точнее растянем ячейку первой строчки первого столбца на три строки и на сей раз сразу уберём лишнее.

```

<html>
  <head>
    <title>Таблица</title>
  </head>
  <body>
    <table border="1">
      <tr>
        <td rowspan="3">строка1 ячейка1</td>
        <td>строка1 ячейка2</td>
        <td>строка1 ячейка3</td>
      </tr>
      <tr>
        <td>строка2 ячейка2</td>
        <td>строка2 ячейка3</td>
      </tr>
      <tr>
        <td colspan="2">строка3 ячейка2</td>
      </tr>
    </table>
  </body>
</html>

```

Всё просто.. главное быть внимательным!

Размеры таблицы.

Если Вы самостоятельно тренировались с рисованием таблицы то наверняка обратили внимание на то, что размеры таблицы и ячеек по умолчанию ограничены вставленным в неё текстом.. и "ползают" себе как хотят. Вспомните про атрибуты **width** - ширина и **height** - высота, которые мы использовали для растягивания рисунков, они так же применимы к тегам **<table>**, **<tr>** и **<td>**. С этими атрибутами Вы знакомы, так что рассусоливать не буду.. просто приведу пример. В нём заданы размеры таблицы и отдельных её ячеек, а заодно и вся таблица выровнена по центру знакомым тегом **<center>**

Пример:

```
<html>
<head>
  <title>Таблица</title>
</head>
<body>
  <center>
    <table border="1" width="640" height="480">
      <tr>
        <td rowspan="3" width="150">строка1 ячейка1</td>
        <td height="30">строка1 ячейка2</td>
        <td>строка1 ячейка3</td>
      </tr>
      <tr>
        <td height="30">строка2 ячейка2</td>
        <td>строка2 ячейка3</td>
      </tr>
      <tr>
        <td colspan="2">строка3 ячейка2</td>
      </tr>
    </table>
  </center>
</body>
</html>
```

Обратите внимание на то, что длина и ширина заданы не для всех ячеек. А зачем? Если и так таблица будет выровнена по самым широким и длинным ячейкам.. на то она и таблица..

Еще не всё рассказал..

Значения атрибутов **width** и **height** в таблице могут указываться не только в пикселях, но и в процентах

Вот пример:

```
<html>
<head>
  <title>Таблица</title>
</head>
<body>
  <center>
    <table border="1" width="640" height="480">
      <tr height="25%">
        <td width="15%"> строка1 ячейка1</td>
        <td width="25%">строка1 ячейка2</td>
        <td width="60%">строка1 ячейка3</td>
      </tr>
      <tr height="50%">
        <td>строка2 ячейка1</td>
        <td>строка2 ячейка2</td>
        <td>строка2 ячейка3</td>
      </tr>
      <tr height="25%">
        <td>строка3 ячейка1</td>
        <td>строка3 ячейка2</td>
        <td>строка3 ячейка3</td>
      </tr>
    </table>
  </center>
</body>
</html>
```

```
</center>
</body>
</html>
```

В этом примере от общей ширины таблицы в 640 пикселей мы отдали 15% первой ячейке, 25% второй и 60% третьей. А по высоте из 480 пикселей по 25% первой и третьей строчке, а добрую половину места второй.. Короче справедливости в мире нет.. Ну до это Вам решать, сколько кому места отчуждать, главное, чтобы в сумме было 100%, иначе неразбериха будет.. смутное время.. революция..

Верстка страницы с использованием таблицы.

Посмотрите на этот пример с таблицей, он Вам ничего не напоминает??

```
<html>
<head>
  <title>Верстка страницы</title>
</head>
<body>
  <center>
    <table border="1" width="640" height="480">
      <tr>
        <td colspan="5" height="30"><center>Заголовок</center></td>
      </tr>
      <tr>
        <td height="30" width="20%"><center>кнопка1</center></td>
        <td width="20%"><center>кнопка2</center></td>
        <td width="20%"><center>кнопка3</center></td>
        <td width="20%"><center>кнопка4</center></td>
        <td width="20%"><center>кнопка5</center></td>
      </tr>
      <tr>
        <td valign="top">содержание:</td>
        <td colspan="4"><center>куча текста</center></td>
      </tr>
    </table>
  </center>
</body>
</html>
```

Как видите, таблица это хороший и главное удобный инструмент для верстки страницы..

Любая ячейка таблицы может служить самостоятельной ёмкостью для наполнения другими тегами и текстовым содержанием, а также иметь те или иные индивидуальные свойства - атрибуты.

Давайте освежим в голове уже пройденный материал, применив эти знания к таблице. Покажу на примерах, мы это уже проходили..

Добавим красок:

```
<html>
<head>
  <title>Таблица и краски</title>
</head>
<body>
  <center>
    <table width="300" height="300">
      <tr>
```

```

        <td bgcolor="#c0e4ff">1</td>
        <td bgcolor="#c5ffa0">2</td>
        <td bgcolor="#c0e4ff">3</td>
    </tr>
    <tr>
        <td bgcolor="#c5ffa0">4</td>
        <td bgcolor="#ffa0c5">5</td>
        <td bgcolor="#c5ffa0">6</td>
    </tr>
    <tr>
        <td bgcolor="#c0e4ff">7</td>
        <td bgcolor="#c5ffa0">8</td>
        <td bgcolor="#c0e4ff">9</td>
    </tr>
</table>
</center>
</body>
</html>

```

Выровняем текст:

Здесь появляется новый атрибут **valign** - Вертикальное выравнивание. До этого момента нам был знаком атрибут **align** - горизонтальное выравнивание.

```

<html>
<head>
    <title>Выравнивание в таблице</title>
</head>
<body>
    <center>
        <table width="300" height="300">
            <tr>
                <td bgcolor="#c0e4ff" valign="top">1</td>
                <td bgcolor="#c5ffa0" valign="top"><center>2</center></td>
                <td bgcolor="#c0e4ff" align="right" valign="top">3</td>
            </tr>
            <tr>
                <td bgcolor="#c5ffa0">4</td>
                <td bgcolor="#ffa0c5"><center>5</center></td>
                <td bgcolor="#c5ffa0" align="right">6</td>
            </tr>
            <tr>
                <td bgcolor="#c0e4ff" valign="bottom">7</td>
                <td bgcolor="#c5ffa0" valign="bottom"><center>8</center></td>
                <td bgcolor="#c0e4ff" align="right" valign="bottom">9</td>
            </tr>
        </table>
    </center>
</body>
</html>

```

Теперь о новых атрибутах.

Cellspacing - задаёт расстояние в пикселях между ячейками таблицы. Задав значение **cellspacing="0"** можно избавиться от "зазора" между ячейками.

Пример, увеличим "зазор":

```

<html>
<head>
<title>Расстояние между ячейками</title>
</head>
<body>
<center>
<table width="300" height="300" cellpadding="15">
<tr>
<td bgcolor="#c0e4ff" valign="top">1</td>
<td bgcolor="#c5ffa0" valign="top"><center>2</center></td>
<td bgcolor="#c0e4ff" align="right" valign="top">3</td>
</tr>
<tr>
<td bgcolor="#c5ffa0">4</td>
<td bgcolor="#ffa0c5"><center>5</center></td>
<td bgcolor="#c5ffa0" align="right">6</td>
</tr>
<tr>
<td bgcolor="#c0e4ff" valign="bottom">7</td>
<td bgcolor="#c5ffa0" valign="bottom"><center>8</center></td>
<td bgcolor="#c0e4ff" align="right" valign="bottom">9</td>
</tr>
</table>
</center>
</body>
</html>

```

Атрибут **cellpadding** (не путайте с **cellspacing**) - в пикселях задаёт поля ячеек (отступ от границ ячеек до текста)

Пример:

```

<html>
<head>
<title>Поля таблицы</title>
</head>
<body>
<center>
<table width="300" height="300" cellpadding="0" cellspacing="25">
<tr>
<td bgcolor="#c0e4ff" valign="top">1</td>
<td bgcolor="#c5ffa0" valign="top"><center>2</center></td>
<td bgcolor="#c0e4ff" align="right" valign="top">3</td>
</tr>
<tr>
<td bgcolor="#c5ffa0">4</td>
<td bgcolor="#ffa0c5"><center>5</center></td>
<td bgcolor="#c5ffa0" align="right">6</td>
</tr>
<tr>
<td bgcolor="#c0e4ff" valign="bottom">7</td>
<td bgcolor="#c5ffa0" valign="bottom"><center>8</center></td>
<td bgcolor="#c0e4ff" align="right" valign="bottom">9</td>
</tr>
</table>
</center>
</body>
</html>

```

Отдельно, пожалуй, еще следует обратить внимание на так называемые вложенные таблицы, я уже говорил про то, что каждая отдельно взятая ячейка может служить емкостью для других тегов.. и тег **<table>** не является исключением. По сути можно вставить одну таблицу в другую..

Смотрим пример:

```
<html>
<head>
  <title>Вложенная таблица</title>
</head>
<body>
  <center>
    <table width="600" height="400" border="1">
      <tr>
        <td colspan="2"><center>Заголовок</center></td>
      </tr>
      <tr>
        <td valign="top">содержание:</td>
        <td valign="top">
          <center>
            <br><br>
            В одну ячейку вложена целая таблица!!
            <br><br><br>
            <table width="200" height="200" cellspacing="0" cellpadding="15">
              <tr>
                <td bgcolor="#c0e4ff" valign="top">1</td>
                <td bgcolor="#c5ffa0" valign="top"><center>2</center></td>
                <td bgcolor="#c0e4ff" align="right" valign="top">3</td>
              </tr>
              <tr>
                <td bgcolor="#c5ffa0">4</td>
                <td bgcolor="#ffa0c5"><center>5</center></td>
                <td bgcolor="#c5ffa0" align="right">6</td>
              </tr>
              <tr>
                <td bgcolor="#c0e4ff" valign="bottom">7</td>
                <td bgcolor="#c5ffa0" valign="bottom"><center>8</center></td>
                <td bgcolor="#c0e4ff" align="right" valign="bottom">9</td>
              </tr>
            </table>
          </center>
        </td>
      </tr>
    </table>
  </center>
</body>
</html>
```

Такой вот растянутый код получился... Сложно? Запутанно? Может быть на первый взгляд оно так и есть, но ведь не зря же я приводил столько примеров! Если что то не понятно вернитесь назад, перечитайте, идите от простого к более сложному и тогда всё станет на свои места! Практика - залог успеха. Не ленитесь, попробуйте писать код самостоятельно..

Ну и уже по традиции подвожу итог главы.. Итак, что у Вас там за страничка получилась..?

У меня она уже вот такая:

```
<html>
<head>
```


указывать большую ширину страницы. По-моему ширина в 1024 пикселей будет оптимальным размером... хотя... ну да ладно.. решайте сами.

Глава 5

Ссылки

Думаю, Вам уже понятно о чём пойдет речь в этой главе.. и Вы знаете, что из себя представляет ссылка, если нет, то **нажмите сюда..**

(<http://www.webremeslo.ru/html/primer/primer29.html>) Существует несколько видов ссылок, а так же "механизмов" перехода по ним. В этой главе постараюсь подробно рассказать о том как прописать ссылки, а так же посвятить в тонкости дела по работе с ними.

Лирическое отступление:

Как-то в армии подходит ко мне начальник штаба и отдаёт приказ, цитирую:

Принеси мне вот тот документ, правда я не знаю где он и какой!! Что стоишь? Бегом давай!! Опаздываю я!!!

Так вот, к чему я это, чтобы браузер, как я тогда, не впал в ступор, ему необходимо знать: точное название документа, путь к документу, и место куда его принести, точнее где его открыть.

На данный момент с помощью блокнота мы создали только один HTML документ у меня он с именем index.html (зачем выбрал такое странное название index.html и зачем оно нужно смотрите в полезных советах) у Вас не знаю с каким, Вы сами имя придумывали, но думаю, Вы его помните и знаете где он лежит, если конечно Вы не мой начальник штаба :).. В этом документе мы попробуем создать ссылку на другой документ, которого у нас еще нет.. Так что прежде чем на него ссылаться, его нужно создать, благо Вы это уже умеете.

1. Открываем блокнот.
2. Пишем код на html языке. к примеру страничку с рядом фотографий.
3. Сохраняем его как html страничку в ту же рабочую папку, где уже есть созданный нами первый документ. Давайте, что б не путаться назовем его primer.html, да и первый тоже, пожалуй, переименуйте в index.html

Теперь я знаю, что у Вас два html документа index.html и primer.html и что теперь у Вас есть минимальный набор для дальнейшего обучения.

Текстовые ссылки.

Знакомимся, тег **<a>** (от anchor- якорь), в него можно заключить текст или рисунок, которые станут ссылкой на те или иные документы. Атрибут тега **<a> href** задаёт имя и путь к документу на который указывает ссылка.

Всё вместе пишется так:

Здесь мои фотки!!

Как Вы наверное поняли primer.html это имя нашего второго html документа, а надпись "Здесь мои фотки!!" это кусочек текста из файла index.html.

По аналогии с рисунками тег **** путь ссылки к открываемому документу прописывается теми же способами:

Здесь мои фотки!! - Такая запись подразумевает, что в директории, где расположен наш первый html документ есть папка stranica в которой расположен файл primer.html

Здесь мои фотки!! - А это значит, что файл myfoto.html

размещен на уровень выше от документа

`Здесь мои фотки!!` - документ расположен на сайте `www.site.ru..`

Ну что давайте попробуем? Ниже приведен пример сразу двух документов в которых прописаны ссылки указывающие друг на друга.

Пример:

Файл index.html:

```
<html>
  <head>
    <title>Делаем ссылкой кусочек текста</title>
  </head>
  <body>
    <div align="center">
      <br><br><br><b>Скажи мне, милый ребёнок: в каком ухе у меня жужжит?</b>
      <br><br><br>
      В <a href="primer.html">правом</a> или <a href="primer.html">левом</a>?
    </div>
  </body>
</html>
```

Файл primer.html:

```
<html>
  <head>
    <title>Перешли по ссылке сюда</title>
  </head>
  <body>
    <br><br><br><br>
    <div align="center"><font size="+2">А вот и не угадал! У меня жужжит в обоих
ухах.</font></div>
    <br><br><br>
    <div align="center"><a href=" index.html">Ну я так не играю...</a></div>
  </body>
</html>
```

Из примера видно, что ссылки выделяются цветом, по умолчанию синеньким - ссылка, а красненьким - уже посещенная ссылка, эти цвета можно изменить с помощью уже хорошо известного нам открывающего тега `< body >` и его атрибутов.

link - цвет ссылки.

alink - цвет нажатой, активной ссылки.

vlink - цвет посещенной ссылки.

Пишется так:

```
<body link="#008000" alink="# ff0000 " vlink="# ffff00">
```

Продолжая говорить о цвете текстовой ссылки стоит упомянуть, что при необходимости можно принудительно выделять цветом как всю ссылку, так и отдельные её части (фразы слова буквы) знакомым тегом ` ` и его атрибутом **color**. Впрочем, это касается не только цвета так же отдельно можно задать размер, стиль и шрифт текста. Но помните, что манипуляции с цветом нужно проводить внутри тега `<a>вот здесь` а не за бортом, иначе цвет ссылки будет либо по умолчанию, либо так как прописано в теге `<body>`

Пример:

Файл index.html:

```
<html>
<head>
  <title>Радуга</title>
</head>
<body link="#008000" alink="# ff0000 " vlink="# ffff00">
  <center>
    <h3>Посмотрите на фразу которая поможет Вам запомнить места цветов в радуге</h3>
    <br>
    <a href="primer2.html">
      <font size="+1" color=ff0000>P</font>
      <font size="+2" color=ff8c40>A</font>
      <font size="+3" color=ffff00>Д</font>
      <font size="+3" color=#008000>У</font>
      <font size="+2" color=#0000ff>Г</font>
      <font size="+1" color=#800080>А</font>
    </a>
  </center>
</body>
</html>
```

Файл primer.html:

```
<html>
<head>
  <title>Радуга</title>
</head>
<body link="#008000" alink="# ff0000 " vlink="# ffff00">
  <center>
    <font size="+3">
      <font color=ff0000>Каждый</font>
      <font color=ff8c40>охотник</font>
      <font color=ffff00>желает</font>
      <font color=#008000>знать</font>
      <font color=#40caff>где</font>
      <font color=#0000ff>сидит</font>
      <font color=#800080>фазан</font>
    </font>
    <br><br><br>
    <a href="index.html">вернуться на главную</a>
  </center>
</body>
</html>
```

Такие вот дела...

Рисунок ссылка.

Ссылкой может являться не только текст, но и рисунок.. Здесь принцип такой же как и в текстовой ссылке, просто вместо текста мы заключаем рисунок который хотим сделать ссылкой.

Вот так:

```
<a href="primer3.html"></a>
```

Перед тем как привести очередной пример расскажу про то, как при переходе по ссылке открыть документ в новом окне браузера, до этого если Вы обратили внимание он у нас открывался в текущем, что не всегда удобно. Решает эту проблему атрибут **target** (цель) и его значение **_blank**.

Пишется так:

```
<a href="primer3.html" target="_blank">открыть в новом окне</a>
```

Пример:

Файл index.html:

```
<html>
<head>
  <title>кнопка</title>
</head>
<body>
  <center>
    <h1> Не в коем случае не нажимайте на эту кнопку!!!</h1>
    <br>
    <a href="primer3.html" title="Не нажимать!!!" target="_blank"></a>
  </center>
</body>
</html>
```

Файл primer.html:

```
<html>
<head>
  <title>итог..</title>
</head>
<body>
  <div align="center"><font size="+3">Ракеты ушли... Америки больше нет...</font></div>
</body>
</html>
```

Я надеюсь, Вы не додумались нажать большую красную кнопку? Ох уж этот русский менталитет.. :)

Что ещё можно почерпнуть из этого примера?

Во-первых. Мы применили уже знакомые атрибуты описания **title="Не нажимать!!!"** для тега **<a>** и одновременно **alt="Не нажимать!!!"** для тега **** если Вы помните, эти атрибуты позволяют сделать нам всплывающие подсказки. Но зачем я их продублировал? Один для ссылки, а другой для рисунка, если сам по себе рисунок является ссылкой?.. долгая история, когда ни будь напишу статью: "Спецификации HTML, браузеры и головная боль.." - а сейчас скажу лишь то, что разные браузеры не всегда воспринимают те или иные теги и их атрибуты. А описание для ссылки картинки нам просто так необходимо! Помните, я рассказывал, что есть люди, умышленно отключающие загрузку рисунков?- это для них.

Во-вторых. В примере использован атрибут тега **** - рамка рисунка. 0 - это значит выключить рамку, так как она портит весь вид страницы в данном примере, но бывают случаи когда рамки рисунка вполне актуальны и так сказать вписываются в страницу. Здесь нужно помнить, что когда рисунок является ссылкой, его рамка окрашивается в цвета ссылки: **link** - цвет ссылки, **alink** - цвет нажатой, активной ссылки, **vlink** - цвет посещенной ссылки тега **<body>**.

Ссылка на e-mail

Я тут Вас чуть-чуть с толку сбил, вроде бы договаривались вместе писать полноценный сайт, а я тут какие то мелкие примерчики выкладываю.. Не волнуйтесь! В конце главы покажу готовый сайт про Карлсона целиком. А сейчас думаю так будет удобнее Вам читать, и мне писать.. : Да и подразумеваю что Вы уже самостоятельно можете вставлять теги в нужные места.

Напишите мне письмо.. - строчка из того примера..

Для того что бы сделать текст или рисунок ссылкой на e-mail - почтовый ящик его нужно заключить в тег **<a>**, но не простой, а с использованием **mailto**

Пишем так:

** Напишите мне письмо.. **

Эта непривычная запись будет говорить что, кликнув по тексту ссылке "Напишите мне письмо.." посетитель сайта попадет в свою почтовую программу, которая выдаст ему бланк для отправки письма, где в строчке Кому: уже будет указан нужный нам почтовый ящик karlson@kruha.ru

Пример:

```
<html>
<head>
  <title>e-mail</title>
</head>
<body>
  <center><h2>
    <a href="mailto:karlson@kruha.ru"> Напишите мне письмо.. </a>
  </h2></center>
</body>
</html>
```

Помимо этого в почтовом бланке можно заранее прописать следующие вещи:

?subject= - Тема письма

&Body= - Текст сообщения

&cc= maluh@kruha.ru,maluh2@kruha.ru - Копии письма

&bcc= freken_bok@kruha.ru,freken_bok2@kruha.ru - Скрытые копии письма

Адреса ящиков для копий и скрытых копий письма пишутся через запятую.

Пример:

```
<html>
<head>
  <title>e-mail</title>
</head>
<body>
  <center><h2>
    <a href="mailto:karlson@kruha.ru?subject=А у вас молоко убежало..
    &Body=Текст письма &cc=maluh@kruha.ru &bcc=freken_bok@kruha.ru">Напишите
    мне письмо.. </a>
  </h2></center>
</body>
</html>
```

Закладки.

Закладки или якоря - это особый вид ссылок. Данные ссылки, как правило, не ведут к какому либо документу, а предназначены для навигации внутри страницы.

Предположим на странице в удобном месте находится некое содержание или меню.. по принципу:

Глава1
Глава2
Глава3

А далее идет большой текст с этими главами, так вот чтобы посетитель страницы нажав на одну из этих глав "перенёлся" в нужное место текста нам нужно сделать две вещи:

Присвоить индивидуальное имя каждой главе. Ищем, значит, в большом-большом тексте нужные главы и делаем их адресами ссылок закладок, присваиваем им имена.

Вот так:

```
<a name="glava1">Глава1 </a>  
<a name="glava2">Глава2 </a>  
<a name="glava3">Глава3 </a>
```

Имя можно присвоить любое необязательно glava1

А теперь собственно прописываем на них ссылки в нашем меню, содержании.

Вот так:

```
<a href="#glava1"> Глава1</a>  
<a href="#glava2"> Глава2</a>  
<a href="#glava3"> Глава3</a>
```

Замете перед каждым именем ставим знак **решётки #**.

Ладно, думаю, на примере будет понятнее:

```
<html>  
<head>  
  <title>Закладки</title>  
</head>  
<body>  
  <h2>А. С. ПУШКИН</h2>  
  <a href="#skazka1">Сказка о попе и работнике его Балде</a><br>  
  <a href="#skazka2">Сказка о рыбаке и рыбке</a><br>  
  <a href="#skazka3">Сказка о царе Салтане</a>  
  <h3><a name="skazka1">Сказка о попе и работнике его Балде</a></h3>  
  <pre>  
    Жил-был поп,  
    Толоконный лоб.  
    ... ..  
  </pre>  
  <h3><a name="skazka2">Сказка о рыбаке и рыбке</a></h3>  
  <pre>  
    Жил старик со своею старухой  
    У самого синего моря.  
    ... ..  
  </pre>
```



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

Файл myfoto.html:

```
<html>
<head>
  <title>Мой первый сайт</title>
</head>
<body text="#484800" bgcolor="#ffffff" background="graphics/fon.jpg">
  <center>
    <table width="640" height="480" cellpadding="5" cellspacing="2" border="1">
      <tr>
        <td colspan="3" height="30" bgcolor="#b2ff80">
          <center></center>
        </td>
      </tr>
      <tr>
        <td height="30" width="33%" bgcolor="#c0e4ff">
          <center><b><a href=" index.html">Обо мне!!!</a></b></center>
        </td>
        <td width="33%" bgcolor="#ffa0cf">
          <center><b><a href=" myfoto.html">Здесь мои фотки!!</a></b></center>
        </td>
        <td bgcolor="#c0e4ff">
          <center><b><a href="mailto:karlson@kruha.ru">Напишите мне
письмо..</a></b></center>
        </td>
      </tr>
      <tr>
        <td colspan="3" valign="top" bgcolor="#b2ff80">
          
          это я <br>
          
          и это я <br>
          
          снова я
        </td>
      </tr>
    </table>
  </center>
</body>
</html>
```

Фух.. ну вроде всё.. Думаю, на этом примере с Карлсоном мы распрощаемся, разросся он через чур.. да и как сайт с точки зрения построения имеет законченный вид...

Он улетел - но обещал вернуться!.. :)

Вполне нормальный сайт-визитка получился.. конечно, дизайн не вдохновляет, но зато теперь у Вас будет стимул сделать лучше!! Теперь этот сайт можно выложить в свет, как это сделать читайте в статье **Публикация сайта:**

(<http://www.webremeslo.ru/publication/publication2.html>)

Ну а мы пойдем дальше, впереди нас ждет еще много интересных всяких разных штук.

Полезные советы:

- Одна из Ваших страничек на сайте **обязательно** должна называться **index.html**. Именно файл с таким названием на Вашем сайте будет искать программа робот, когда человек введет имя Вашего сайта. Так как страница **index.html** будет открываться первой, делайте её главной. Остальные страницы можете называть, как угодно... больше никаких нюансов с именами нет.
- Про регистр.. Прописывая путь и имена документов помните, что, к примеру: Page.html, page.html и PAGE.html это имена разных документов! Это же касается имен закладок и рисунков. Всегда учитывайте регистр при написании кода, есть большая вероятность, что такие имена не будут распознаны тем или иным браузером. Возьмите за правило все писать и обзывать маленькими латинскими буквами, тогда риск человеческого фактора и капризов программ сведется к нулю.
- Правило трех кликов..

Старайтесь формировать "дерево ссылок" таким образом, чтобы посетитель сайта мог из любой его страницы попасть в любое место сайта за минимальное число переходов по ссылкам. Больше трёх переходов к нужному месту на сайте это уже не есть хорошо.. Бесконечные загрузки ненужных человеку страниц могут привести к нервному расстройству и преждевременному закрытию сайта. Берегите время деньги и нервы людей.

И ещё...

Спецсимволы

```

<td>
  &lt;html&gt;<br>
  &lt;head&gt;<br>
  &lt;title&gt;Моя первая страничка&lt;/title&gt;<br>
  &lt;/head&gt;<br>
  &lt;body&gt;<br>
  &lt;center&gt;&lt;h2&gt;Привет мир!!!&lt;/h2&gt;&lt;/center&gt;<br>
  &lt;br&gt;<br>
  Это моя первая страничка!<br>
  &lt;/body&gt;<br>
  &lt;/html&gt;<br>
</td>
</tr>
</table>
</body>
</html>

```

Пример в примере... навивает на философские мысли о бесконечности...

Ну вот теперь Вы знаете, как примерно выглядят все эти примерчики по "ту сторону экрана". Кстати, раз уж начал открывать военные тайны, данный пример является таблицей из двух ячеек, одна тоненькая слева выполняет сугубо декоративную функцию.. так вот если перед Вами встанет задача нарисовать ячейку таблицы без какого либо содержания вставляйте в неё знак пробела ** **. Помните правило `<тег>`здесь что то обязательно должно быть`</тег>`? Спецсимвол пробела один из выходов в данном случае.

Кроме выше указанных есть еще целый ряд спецсимволов, чего там только нет: знаки зодиака, карточные масти, палочки, точки, ёлочки, кругляшки, дроби.. Предназначены они уже не для каких-то "особых" задач, а просто выполняют свою роль ввиду отсутствия данных знаков на клавиатуре. Мою "коллекцию" спецсимволов можете **посмотреть здесь**:
<http://www.webremeslo.ru/spravka/spravka2.html>

Горизонтальная линия

Простой новый тег **<hr>** рисует в окне горизонтальную линию, не требует закрывающего тега. Часто применяется при верстке страницы в качестве декоративного элемента.

Имеет ряд атрибутов, **align** -выравнивание с одним из трёх значений (**center**, **left**, **right**) может быть применен, если задана длина линии атрибут **width** в пикселях или процентах. Так же можно задать толщину линии атрибут - **size**, цвет атрибут - **color**, и при необходимости отключить тень линии **noshade**.

Пример:

```

<html>
<head>
  <title>Горизонтальная линия</title>
</head>
<body>
  Это просто линия по умолчанию: &lt;hr&gt;
  <hr>
  Это линия без тени: &lt;hr noshade &gt;
  <hr noshade>
  Это линия окрашенная в красный цвет: &lt;hr color="#ff0000"&gt;
  <hr color="#ff0000">
  Линия длиной 250 пикселей: &lt;hr width="250"&gt;
  <hr width="250">
  Линия длиной 250 и толщиной 5 пикселей: &lt;hr width="250" size="5"&gt;
  <hr width="250" size="5">

```

```
Линия длиной 500, толщиной 50, синяя: <hr width="500" size="50"
color="#0000ff">
<hr width="500" size="50" color="#0000ff">
Примеры выравнивания:
<hr align="left" width="250" size="5">
<hr align="center" width="250" size="5">
<hr align="right" width="250" size="5">
<hr>
</body>
</html>
```

По моему достаточно просто и эффективно.

Бегущая строка

Тег **<marquee>** заставляет текст помещённый в него двигаться в том или ином направлении, проще говоря делает его бегущей строкой. Бегущая строка имеет ряд настроек скроллинга, которые задаются следующими атрибутами:

behavior - определяет тип скроллинга, может иметь следующие значения:

- **alternate** - колебательные движения от края к краю
- **scroll** - прокручивание текста. текст будет выходить за рамки экрана и снова появляться с противоположной его стороны
- **slide** - прокручивание текста с остановкой.

scrollamount - скорость бегущей строки от 1 до 10.

loop задает количество прокруток бегущей строки.

direction - направление движения текста. значения:

- **up** - вверх,
- **down** - вниз,
- **left** - влево,
- **right** - вправо.

bgcolor - цвет фона бегущей строки,

height - высота строки,

width - ширина строки.

Пример:

```
<html>
<head>
  <title>Бегущая строка</title>
</head>
<body>
  <div align="center"><h2>Бегающие строки</h2></div>
  <marquee>Бегущая строка по умолчанию</marquee>
  <marquee direction="right">Бегущая строка слева направо</marquee>
  <marquee behavior="alternate">Бегущая строка бежит от края к краю</marquee>
  <marquee scrollamount="10">Бегущая строка со скоростью 10</marquee>
  <marquee scrollamount="1">Бегущая строка со скоростью 1</marquee>
  <marquee direction="right" loop="2">Эта строка будет прокручиваться только два
  раза</marquee>
  <marquee behavior="slide">Бегущая строка с остановкой</marquee>
```

```

<marquee bgcolor="#b40000">Бегущая строка с фоном</marquee>
<marquee width=400>Бегущая строка с ограничением ширины прокрутки</marquee>
<marquee direction="up">Бегущая строка снизу вверх</marquee>
<marquee hspace="300">Бегущая строка с отступами от границ</marquee>
</body>
</html>

```

Ну думаю, сами разберётесь, что к чему. Попробуйте использовать различные сочетания атрибутов для достижения нужного Вам эффекта. Добавлю лишь то, что с текстом бегущей строки можно делать то же самое, что и с обыкновенным текстом: менять размер, цвет, стиль, шрифт, сделать фразу из строки ссылкой.. прописывая нужные теги внутри тега **<marquee>**

Комментарии

Комментарии используются для того, чтобы облегчить свой собственный труд, а так же упростить понимание другим человеком кода Вашей странички. Пишутся они непосредственно в коде html документа, однако браузер не выводит эти "пометки" на экран. Для того чтобы заставить браузер игнорировать какой либо текст его необходимо заключить вот в такой набор символов:

```
<!-- -->
```

По принципу:

```
<!-- здесь может быть любой текст -->
```

Использовать комментарии очень полезно, разрабатывая большие и сложные по своей структуре документы, для того, чтобы потом самому с легкостью в нём ориентироваться..

Пример:

```

<html>
<head>
<title>Мой первый сайт</title>
</head>
<body text="#484800" bgcolor="#ffffff" background="graphics/fon.jpg">
<center>
<!-- Начало таблицы -->
<table width="640" height="480" cellpadding="5" cellspacing="2" border="1">
<tr>
<td colspan="3" height="30" bgcolor="#b2ff80">
<center></center>
</td>
</tr>
<!-- Строка с меню -->
<tr>
<td height="30" width="33%" bgcolor="#ffa0cf">
<center><b><a href=" index.html">Обо мне!!!</a></b></center>
</td>
<td width="33%" bgcolor="#c0e4ff">
<center><b><a href=" myfoto.html">Здесь мои фотки!!</a></b></center>
</td>
<td bgcolor="#c0e4ff">
<center><b><a href="mailto:karlson@kruha.ru">Напишите мне
письмо..</a></b></center>
</td>
</tr>
<!-- Конец меню -->

```

[illegible]

Полезные советы:

- Писать сайты на своём компьютере конечно дело хорошее, но пора задумываться о том, как Ваше творчество разместить в сети Интернет. О том как это сделать читайте в статье **Публикация сайта: (<http://www.webremeslo.ru/publication/publication2.html>)**. А совет собственно заключается в том, что на данном этапе обучения Вам пора обзавестись своим собственным местом (хостом) "тренировочным плацдармом" на каком либо сервере.. а также собственным именем сайта (доменом)... для начала, это место и имя могут быть бесплатными, даже я бы сказал должны быть бесплатными.. Там, в этом месте на сервере, Вы будите тренироваться выкладывать свои страницы., смотреть как и с какой скоростью они загружаются, так же Вы сможете показать своё творение миру (а можете и не показывать) и друзьям. Почему этим стоит заняться именно на данной стадии обучения? Да потому что с помощью того, что мы изучили, Вы уже можете делать вполне пристойные сайты!! нет конечно до конца обучения ещё рано, да и предела совершенствования мастерства как известно нет.., но это послужит хорошей тренировкой и так сказать поднимет Ваш морально боевой дух!!)) Знаете как это приятно, увидеть свою работу в действии!? Но, ещё раз повторюсь, о том как это дело проверить читайте в статье "Публикация сайта".

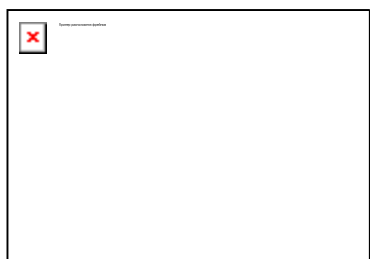
Глава 7

Фреймы.

Итак, фреймы.. для чего они нужны и какую пользу в себе несут? Об этом я и постараюсь рассказать в этой главе, ну и естественно о том, как их внедрять и работать с ними..

Часто при создании сайта возникает необходимость открывать в одном окне браузера одновременно несколько HTML документов.. так вот фреймы как раз таки и созданы для того чтобы определить рабочие области для каждого документа. Кроме того, фреймы - это хороший инструмент, с помощью которого можно заниматься версткой страницы, они служат достойной "альтернативой" табличному способу верстки страницы.. в кавычки взял слово "альтернатива" потому, что это совершенно иной способ построения сайта со своими достоинствами и недостатками и его достаточно трудно сопоставлять с доселе привычным построением сайта.. но обо всем по порядку..

Ну что, начнем? Допустим нам необходимо в одном окне браузера открыть сразу три HTML документа и расположить их к примеру вот таким способом:



Что нам для этого нужно? Для начала естественно нам необходимо создать три отдельных html документа которые мы собственно и будем открывать в одном окне.. Пусть первый документ будет нести в себе графическое изображение и будет выполнять роль логотипа назовём его logotype.html , вторым документом будет некое содержание.. назовём menu.html , ну и третьим документ с кучей текста. text.html. Конечно Вы можете придумать свои названия, как впрочем и содержание документов, но пока лучше всё же копируйте мои.. так будет удобнее и мне и Вам.

Итак, есть три файла logotype.html, menu.html и text.html.. которые нам необходимо разместить под одной крышей, а крыши у нас как раз таки ещё пока и нет..

Пишем "крышу", главный документ, к которому будем подключать наши файлы. Раз он будет главным, присваиваем ему имя index.html.

```
<html>
<head>
  <title>фреймы</title>
</head>
<body>
</body>
</html>
```

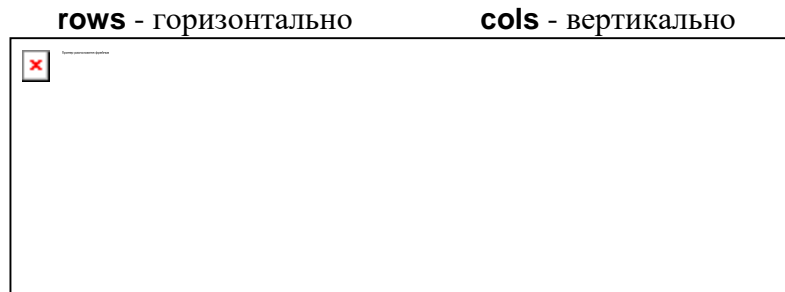
Вот привычная для нас структура документа которую мы переживали ещё в самом начале нашего обучения.. Фреймы ломают сложившиеся стереотипы! Структура фрейм документа имеет вот такой вид:

```
<html>
<head>
  <title>фреймы </title>
</head>
```

```
<frameset>
</frameset>
</html>
```

Отсутствие тега **<body>** компенсирует новый тег **<frameset>** - установить фрейм или набор фреймов.. с этим товарищем мы собственно и будем работать дальше.

Тег **<frameset>** имеет атрибуты **rows** и **cols** - эти атрибуты указывают браузеру на то, как следует размещать фреймы в окне браузера



В нашем случае нам необходимо горизонтальное размещение фреймов.. значит пишем так:

```
<html>
<head>
  <title>фреймы </title>
</head>
<frameset rows="15%,15%,70%">
</frameset>
</html>
```

Проценты после знака равенства - это не что иное, как размеры наших окошек- фреймов в одном большом окне браузера, у нас предполагается размещение трёх окон - следовательно и значений через запятую тоже три. Вспомните как мы задавали размеры ячеек для таблицы, здесь тот же принцип.. Как и в случае с ячейками таблицы, размеры фреймов можно указывать в процентах от общей площади и в пикселях.

Вот несколько примеров написания:

<frameset rows="15%,15%,70%"> - такая запись говорит о том, что будет три горизонтальных окошка где последнее займет 70% площади окна браузера, а первые два по 15%.

<frameset cols="200,240,200"> - здесь три окна расположены вертикально ширина которых указана в пикселях.

<frameset rows="100,*,180"> - такая запись значит, что первое и третье окно будут иметь ширину 100 и 180 пикселей соответственно, а второй фрейм займет всю оставшуюся площадь.

С размещением документов покончили, теперь собственно осталось их подключить и насладиться первым результатом..

Тег **<frame>** и его атрибут **src** укажет браузеру путь к html документу который следует открыть в отведенном для него месте. У нас имеется три отдельных документа `logotype.html`, `menu.html` и `text.html` теперь к каждому нужно прописать путь, что мы собственно и делаем. (предполагаю, что у Вас все четыре файла находятся в одной папке и пути к ним имеют самую простую запись по типу: **<frame src="logotype.html">**)

Пример:

Файл index.html

```
<html>
<head>
  <title>фреймы</title>
```

```

</head>
<frameset rows="15%,15%,70%">
  <frame src="logotype.html">
  <frame src="menu.html">
  <frame src="text.html">
</frameset>
</html>

```

Файл logotype.html

```

<html>
  <head>
    <title>фреймы</title>
  </head>
  <body>
    
  </body>
</html>

```

Файл menu.html

```

<html>
  <head>
    <title>фреймы</title>
  </head>
  <body bgcolor="#e8e8e8" background="fon1.jpg">
    <font face="Monotype Corsiva" size="6" color="#ff0000">Меню:</font><br>
    <font face="Comic Sans MS" size="3">
      <a href="text.html">Суп из шампиньонов</a><br>
      <a href="text1.html">Бобы в горшочке по-итальянски</a><br>
      <a href="text2.html">Австралийский летний салат</a><br>
      ... ..
    </font>
  </body>
</html>

```

Файл text.html

```

<html>
  <head>
    <title>фреймы</title>
  </head>
  <body bgcolor="#f5f5f5">
    <center><font face="Comic Sans MS" size="3">Суп из шампиньонов</font></center>
    <br>
    куча текста..
  </body>
</html>

```

В данном примере я выложил все четыре HTML документа где index.html является основным, а остальные три, это подключаемые страницы, каждая из которых в принципе может работать автономно и содержать в себе всё что угодно картинки, таблицы, тексты, ссылки.. сделал я это для того что бы дать Вам понять принцип работы с фреймами. В дальнейших примерах я буду выкладывать лишь головной файл с фреймами, (а то уж больно много места на странице они все вместе занимают), а Вы знайте что у меня "где то там" есть все остальные файлы содержащие в себе графику, тексты, может быть ещё что ни будь.. ну и не отставайте за мной потихоньку редактируйте свои собственные странички к будущему тренировочному сайту.. Не знаю про что Вы будите писать свой, а я тут пообещался помочь одной девушке в написании сайта посвящённого кулинарии..) так что решил так сказать сразу на двух зайцев замахнуться.. на девушку зайку и сайт..) как-то так..))

В выше изложенном примере мы расположили все окна горизонтально, поменяв атрибут **rows** на **cols** можно расположить их в вертикальном порядке. А как быть, если требуется разместить наши окна

вот так?:

или так?:

или даже так?:



Выход есть. Разберём на примерах.

Начнём с первого случая.. что мы видим? А видим мы две строки, где вторая поделена на два столбца.

А теперь по порядку:

<frameset rows="20%,80%"> - делим окно браузера на две строки
<frame src="logotype.html"> - в первую строку загружаем наш логотип
<frameset cols="30%,70%"> - а вторую строку делим на два столбца
<frame src="menu.html"> - в первом будет меню
<frame src="text.html"> - а во втором куча текста
</frameset> - закрываем тег деления на столбцы
</frameset> - закрываем тег деления на строки

Вообще то говорить столбцы и строки неверно, так как фреймы никакого отношения к таблицам не имеют кроме как визуального подобия, правильно говорить горизонтальные и вертикальные фреймы.. ну это я так изрекаюсь что б Вам понятнее было..

Ну да ладно, смотрим пример:

```
<html>
<head>
  <title>фреймы</title>
</head>
<frameset rows="15%,85%">
  <frame src="logotype.html">
  <frameset cols="20%,80%">
    <frame src="menu.html">
    <frame src="text.html">
  </frameset>
</frameset>
</html>
```

Во втором случае у нас имеется два столбца в котором второй поделен на две строки, значит будем писать так:

<frameset cols="30%,70%"> - делим окно на два столбца
<frame src="menu.html"> - в первом будет содержание
<frameset rows="20%,80%"> - а второй делим на две строки
<frame src="logotype.html"> - логотип
<frame src="text.html"> - и основной текст
</frameset> - закрываем деление на строки
</frameset> - закрываем деление на столбцы

Пример:

```
<html>
  <head>
    <title>фреймы</title>
  </head>
  <frameset cols="20%,80%">
    <frame src="menu.html">
    <frameset rows="15%,85%">
      <frame src="logotype.html">
      <frame src="text.html">
    </frameset>
  </frameset>
</html>
```

Третий случай немного посложней, но бояться его не стоит.. тем более что именно такое построение лично я выбрал для сайта про кулинарное искусство, чуть ниже объясню почему. Что мы имеем? три столбца причем во второй столбец по сути заключён наш первый случай..

<frameset cols="*,800,*"> - делим окно на три столбика (обратите внимание второй столбец занимает ровно 800 пикселей а два крайних точного размера не имеют и поделят пополам оставшееся место)

<frame src="dekor.html"> - в первый столбец загружаем html документ который будет выполнять сугубо декоративную функцию

<frameset rows="15%,85%"> -

<frame src="logotype.html"> -

<frameset cols="20%,80%"> - во второй столбец вставляем наш "первый случай"

<frame src="menu.html"> -

<frame src="text.html"> -

</frameset> -

</frameset> -

<frame src="dekor.html"> - в третий столбец загружаем всё тот же файл с декорациями

</frameset> - захлопываем

Посмотрите пример, а потом я объясню почему избрал путь с пятью окнами

Пример:

```
<html>
  <head>
    <title>фреймы</title>
  </head>
  <frameset cols="*,800,*">
    <frame src="dekor.html">
    <frameset rows="120,*">
      <frame src="logotype.html">
      <frameset cols="200,600">
        <frame src="menu.html">
        <frame src="text.html">
      </frameset>
    </frameset>
  </frameset>
  <frame src="dekor.html">
</frameset>
</html>
```

Так почему пять окон? Помнится мне, я уже писал про то, что у разных пользователей сети Интернет на мониторах выставлены разные разрешения и соответственно на разных экранах наш сайт будет выглядеть по-разному.. и если нет конкретных размеров все наши рисунки, тексты, таблицы что называется "поплывут" у тех пользователей, разрешения мониторов которых отлично от Вашего. Когда мы верстали сайт с помощью таблицы, вопрос с размерами страницы решался при помощи присуждения этой таблице конкретной ширины и высоты, к сожалению, с фреймами так поступить нельзя.. даже если указывать ширину фреймов не в процентах, а в пикселях, всё равно последний столбец растянется на оставшуюся ширину окна браузера и страница, что называется, потеряет "товарный вид". Так как же быть? Приходится идти на маленькие хитрости.. Задав центральному столбцу (в котором у нас собственно и размещена вся страница) размер 800 пикселей мы раз и навсегда определи его ширину, а безразмерные первый и третий столбцы помимо декора выполняют роль своего рода "пружин" на которых подвешен центральный столбец.. Так, у людей с маленьким разрешением монитора эти окна/поля будут узкими, а людей с большим разрешением широкими, таким образом центральный столбец никак не пострадает и теперь мы можем с полной уверенностью размещать в нём какие либо объекты, с точной привязкой по месту, не переживая за их дальнейшую судьбу. Сравните первый пример, где три окна и третий, где их уже пять, неправда ли так лучше?

На этом с размещением и размерами фреймов закончим.. и так здесь долго задержались.. идём дальше.

Приводим фреймы в опрятный вид.

В последнем нашем примере первое, что бросается в глаза, это целая куча полос прокруток, которые стоят где надо и не надо.. Давайте от них избавимся, ну можно и оставить кое где.. Делается это при помощи атрибута **scrolling** - тега **<frame>**, он может иметь одно из трёх значений:

- **no**- никогда не показывать полосу прокрутки,
- **yes**- всегда показывать,
- **auto**- показывать в том случае если она необходима.

Пример:

```
<html>
<head>
  <title>фреймы</title>
</head>
<frameset cols="*,800,*">
  <frame src="dekor.html" scrolling="no">
  <frameset rows="120,*">
    <frame src="logotype.html" scrolling="no">
    <frameset cols="200,600">
      <frame src="menu.html">
      <frame src="text.html">
    </frameset>
  </frameset>
  <frame src="dekor.html" scrolling="no">
</frameset>
</html>
```

Поля фреймов или иначе расстояние от границ фрейма до текста или картинки, как в нашем случае, задаются в пикселях при помощи атрибутов **marginwidth** и **marginheight** тега **<frame>**

Пример:

```
<html>
<head>
  <title>фреймы</title>
```

```

</head>
<frameset cols="*,800,*">
  <frame src="dekor.html" scrolling="no">
  <frameset rows="120,*">
    <frame src="logotype.html" scrolling="no" marginwidth="0" marginheight="0">
    <frameset cols="200,600">
      <frame src="menu.html">
      <frame src="text.html" marginwidth="10" marginheight="10">
    </frameset>
  </frameset>
  <frame src="dekor.html" scrolling="no">
</frameset>
</html>

```

Поговорим немного о рамках вокруг наших фреймов.

Если Вы обратили внимание, то в последнем примере наведя курсор на границу фрейма он, курсор то бишь, приобретает другой вид (захватить и переместить) и теперь эту границу можно перетаскать в какую либо сторону удерживая левой кнопкой мыши. Иногда такая "мобильность" границ фреймов играет на руку веб-мастеру, но чаще всё же мешает.. Для того, чтобы запретить пользователю играть с размерами окон для тега **<frame>** придумали атрибут **noresize**

Пример:

```

<html>
<head>
  <title>фреймы</title>
</head>
<frameset cols="*,800,*">
  <frame src="dekor.html" scrolling="no" noresize>
  <frameset rows="120,*">
    <frame src="logotype.html" scrolling="no" marginwidth="0" marginheight="0" noresize>
    <frameset cols="200,600">
      <frame src="menu.html" noresize>
      <frame src="text.html" marginwidth="10" marginheight="10" noresize>
    </frameset>
  </frameset>
  <frame src="dekor.html" scrolling="no" noresize>
</frameset>
</html>

```

А вот давно нам знакомый атрибут **border** задаёт ширину в пикселях этих самых рамок между фреймами.. пишется он внутри тега **<frameset>**. Как и раньше значение **border="0"** вовсе избавит нас от рамок.

Пример:

```

<html>
<head>
  <title>фреймы</title>
</head>
<frameset cols="*,800,*" border="0">
  <frame src="dekor.html" scrolling="no" noresize>
  <frameset rows="120,*" border="0">
    <frame src="logotype.html" scrolling="no" marginwidth="0" marginheight="0" noresize>
    <frameset cols="200,600" border="0">
      <frame src="menu.html" noresize>
      <frame src="text.html" marginwidth="10" marginheight="10" noresize>
    </frameset>
  </frameset>

```

```

</frameset>
<frame src="dekor.html" scrolling="no" noresize>
</frameset>
</html>

```

Фреймы и ссылки.

Настало время оживить наш сайт ссылками, да вот беда, знакомая нам обыкновенная ссылка **Бобы в горшочке по-итальянски** откроет данный документ в том же фрейме где она и находится, в нашем случае непосредственно в фрейме с содержанием, а само содержание в момент перехода по этой ссылке уйдёт в небытие.. можете взглянуть на **корявый пример..** кликнете в нём на любую ссылку.. Что бы впредь такого не происходило, необходимо указывать браузеру в каком фрейме следует открывать необходимый нам документ, ну если конечно Вам и в правду не нужно открывать его в этом же фрейме.

Помнится мы уже знакомились, в главе посвящённой ссылкам, с атрибутами **name**- имя и **target**- цель, применяются они также и при работе с фреймами механизм немного изменён а так почти тоже самое. первым делом фрейму в котором мы хотели бы открывать какие либо документы необходимо присвоить индивидуальное имя.

пишется так:

<frame src="text.html" name="osnovnoe"> имя можно придумать любое.. главное его не забыть..

А теперь в документе с ссылками (в нашем случае это файл menu.html) необходимо указать цель то есть собственно определить в каком окне стоит открывать тот или иной документ

Пишется так:

Бобы в горшочке по-итальянски

Ну я думаю Вам уже не стоит объяснять, что прежде чем ссылаться на какие либо документы, их необходимо создать.. в моём случае файлы (рецепты) имеют имена text.html, text1.html, text2.html...

Посмотрите пример:

Файл index.html

```

<html>
<head>
<title>фреймы</title>
</head>
<frameset cols="*,800,*" border="0">
<frame src="dekor.html" scrolling="no" noresize>
<frameset rows="120,*" border="0">
<frame src="logotype.html" scrolling="no" marginwidth="0" marginheight="0" noresize>
<frameset cols="200,600" border="0">
<frame src="menu.html" noresize>
<frame src="text.html" name="osnovnoe" marginwidth="10" marginheight="10"
noresize>
</frameset>
</frameset>
<frame src="dekor.html" scrolling="no" noresize>
</frameset>
</html>

```



```

Файл menu.html
<html>
  <head>
    <title>фреймы</title>
  </head>
  <body bgcolor="#e8e8e8" background="fon1.jpg" link="#b40000" alink="#900000"
vlink="#900000">
    <font face="Monotype Corsiva" size="6"color="#ff0000">Меню:</font><hr>
    <font face="Comic Sans MS" size="3">
      <a href="text.html" target="osnovnoe">Суп из шампиньонов</a><hr>
      <a href="text1.html" target="osnovnoe">Бобы в горшочке по-итальянски</a><hr>
      <a href="text2.html" target="osnovnoe">Австралийский летний салат</a><hr>
      ... ..
    </font>
  </body>
</html>

```

Так же как и раньше документ можно открыть в отдельном окне. Напомню, пишется так:

```
<a href="text1.html" target="_blank">Бобы в горшочке по-итальянски</a>
```

Или же присвоив атрибуту **target** значение **_top** открыть его в этом же окне браузера, но на весь экран.. "обнулив" при этом всё что бы там не находилось.. пишется так:

```
<a href="text1.html" target="_top">Бобы в горшочке по-итальянски</a>
```

Такой вот сайтик получился.. конечно над ним ещё работать и работать.. к тому же по моей задумке он будет иметь несколько иную структуру в плане навигации по сайту, в нем будет ещё целая куча страниц, красивое меню, но что касается фреймовой структуры, думаю она останется прежней..

Плавающий фрейм

Иногда в страницу содержащую в себе не фреймовую структуру необходимо в отдельном окне вставить другой HTML документ или даже ряд таковых документов. Для выполнения этой цели существует тег **<iframe>** - так называемый плавающий фрейм.

Данный тег имеет ряд атрибутов:

src - обязательный атрибут, указывающий путь к открываемой странице
width - ширина плавающего фрейма в пикселях или процентах
height - высота плавающего фрейма

scrolling - показ полосы прокрутки

- **no** - никогда не показывать полосу прокрутки,
- **yes** - всегда показывать,
- **auto** - показывать в том случае если она необходима.

align - выравнивание плавающего фрейма

- **left** - слева
- **right** - справа
- **top** - выше
- **bottom** - ниже

frameborder - наличие рамки вокруг плавающего фрейма

- **1** - включить рамку
- **0** - выключить рамку

Всё вместе пишется так:

```
<iframe src="primer.html" width="300" height="250" align="left" scrolling="auto"
frameborder="1" ></iframe>
```

Пример документа с плавающим фреймом:

```
<html>
<head>
  <title>Плавающий фрейм</title>
</head>
<body>
  <center><h2>Плавающий фрейм</h2></center>
  В эту страницу введён так называемый "плавающий фрейм".
  В отдельном окне он открывает для показа другой документ html.
  <br>
  <iframe src="primer.html" width="250" height="250" align="left" scrolling="auto"
frameborder="1"></iframe>
  <br>
  ... ..
  <br>
</body>
</html>
```

Noframes

Некоторые браузеры не поддерживают фреймовую структуру документа или неправильно её интерпретируют, кроме того зачастую пользователи в настройках своих браузеров умышленно отключают поддержку фреймовой структуры html документа. И хотя процент таких браузеров и пользователей невелик, но тем не менее они есть.

Теперь представьте, что Вы построили свой сайт с помощью фреймовой структуры и вот некоторые посетители, может быть сами не подозревая в чем проблема, пытаются открыть Ваш сайт, а их браузер выдает ошибку! Что они подумают о Вашем сайте? я думаю, что то типа: "Фу.. ерунда какая то.. больше сроду сюда не зайду!".

Для того что бы дать понять пользователю, что его браузер/настройки браузера не поддерживают фреймы существует тег **<noframes>**.

Тег **<noframes>** выводит текст, заключенный в него в том случае, если браузер пользователя не поддерживает фреймы или они принудительно выключены в его настройках. Если же фреймы поддерживаются браузером пользователя, то данный тег попросту игнорируется.

Пример:

```
<html>
<head>
  <title>фреймы</title>
</head>
<frameset cols="*,800,*" border="0">
  <noframes>Извините, но Ваш браузер не поддерживает фреймы..</noframes>
  <frame src="dekor.html" scrolling="no" noresize>
  <frameset rows="120,*" border="0">
    <frame src="logotype.html" scrolling="no" marginwidth="0" marginheight="0" noresize>
```

```

<frameset cols="200,600" border="0">
  <frame src="menu.html" noresize>
  <frame src="text.html" marginwidth="10" marginheight="10" noresize>
</frameset>
</frameset>
<frame src="dekor.html" scrolling="no" noresize>
</frameset>
</html>

```

Результат примера будет заметен, если Ваш браузер и в правду не поддерживает фреймы (тут я надолго задумался.. :) если это так, то зачем вообще читать эту главу?) или же Вы в качестве эксперимента выключили поддержку фреймов в своём браузере.

Тег **<noframes>** должен быть расположен внутри тега **<frameset>**

С плавающим фреймом всё ещё более просто, достаточно написать нужный текст между **<iframe>** **</iframe>** и данная надпись будет выводиться на экран, в том случае если браузер не поддерживает фреймы.

Вот так:

```

<iframe src="primer.html" width="300" height="250" align="left" scrolling="auto"
frameborder="1">Извините, но Ваш браузер не поддерживает фреймы..</iframe>

```

Полезные советы:

- Перед тем как начать создавать страницу с использованием фреймовой структуры проанализируйте её макет, размеры каждого окна, наличие или отсутствие в них полос прокруток и т. д. После этого можно создавать подключаемые HTML файлы особо не переживая за их взаимное расположение относительно друг друга..
- Используйте тег **<noframes>**. Помните что если в Вашем браузере сайт работает и отображается так как Вы и задумывали то у других пользователей дела могут обстоять иначе!

Глава 8

Навигационные карты

Мы с Вами уже говорили о том как сделать рисунок ссылкой.. в этой главе поговорим о том как сделать фрагмент, область рисунка ссылкой на тот или иной документ, а так же о том как сделать различные области одного и того же рисунка ссылками на различные документы.

Для осуществления данной задачи нам не обойтись без так называемых навигационных карт.

Предположим, у нас имеется такой рисунок:



..и нам необходимо сделать, так что бы пользователь, нажав на одну из этих "кнопок" переходил по ссылке в тот или документ.. что нам для этого нужно?

В первую очередь обозначим наш рисунок не как обыкновенное графическое изображение, а как навигационную карту присвоив этому рисунку своё индивидуальное имя. Делается это при помощи атрибута **usemap** тега **** (я думаю Вам не стоит напоминать о том что тег **** имеет обязательный атрибут **src** который указывает путь к той или иной картинке ****)

Назовём наш рисунок/карту именем **panel**. Это будет выглядеть так:

```

```

Не забываем по правилам синтаксиса поставить знак **#** решётки перед именем..

Ну а теперь, собственно, составим навигационную карту. Она задаётся тегом **<map>** у которого есть атрибут **name** - имя.. улавливаете к чему я веду? Ну как Вы наверное уже догадались, укажем на базе какого рисунка мы будем строить нашу навигационную карту указав его имя..

```
<map name="panel">  
</map>
```

Теперь займемся тем, что укажем браузеру области рисунка которые будут являться ссылками, а заодно пропишем пути переходов по этим ссылкам. Эту задачу берет на себя тег **<area>**, закрывающего тега он не требует, и имеет следующие атрибуты, с которыми мы будем работать:

href - указывает путь к открываемому документу (точно так же как в теге **<a>**)
shape - форма области рисунка которая будет служить ссылкой. Может иметь одно из трёх значений:

- **rect** - прямоугольная область
- **poly** - область представляет собой некий многоугольник
- **circle** - область заданная окружностью

coords - координаты формы

Прямоугольная область

Теперь наша карта приобретает такой вид:

```
<map name="panel">
<area href="primer1.html" shape="rect" coords="15,15,82,82">
</map>
```

По сути теперь "зелёная" квадратная кнопка стала рабочей.

В чем Вы можете убедиться нажав на неё в этом примере:

```
<html>
<head>
  <title>Навигационная карта</title>
</head>
<body>
  <center>
    
  </center>
  <map name="panel">
    <area href="primer1.html" shape="rect" coords="15,15,82,82">
  </map>
</body>
</html>
```

Немного поспешил с примером толком не объяснив суть написанного.. Заострим внимание на атрибутах тега **<area>**.

href="primer1.html" - здесь думаю понятно, это путь к документу, который должен открыться при нажатии на "зелёную кнопку".

Так как кнопка у нас квадратная, а квадрат как известно это "правильный" прямоугольник, форму области рисунка назначаем прямоугольной **shape="rect"**.

А теперь самое интересное **coords="15,15,82,82"** - координаты.. Для прямоугольника они задаются двумя точками по принципу "X1,Y1,X2,Y2" Где X1,Y1- первая точка и соответственно X2,Y2 - вторая. Координаты указываются в пикселях.. Наш рисунок/карта имеет размеры 300 на 100 пикселей, её самый верхний левый пиксель имеет координаты X=0,Y=0, а самый нижний правый X=300,Y=100. Если не понятно, окупёмся в геометрию за пятый класс..

Взгляните на рисунок:



Так, выбрав прямоугольную форму **shape="rect"** для нашей области в виде квадратной кнопки мы указали координаты её верхнего левого и нижнего правого пикселей.. которых вполне достаточно для обозначения "рабочей" области всей кнопки.

Полигон (многоугольник).

Займёмся "жёлтой кнопкой", она у нас представлена в виде треугольника. Для того чтобы выделить её "рабочую" область из нашего рисунка, присвоим атрибуту **shape** значение **poly** - полигон, которое определит эту область как некий многоугольник, где координаты через запятую будут являться его вершинами, их может быть сколь угодно много "X1,Y1,X2,Y2,X3,Y3,X4,Y4...X124,Y124" фигура образованная этими точками вершинами углов может иметь вид любого многоугольника как правильного, так и неправильного. В нашем случае угла всего три, на то он и треугольник, следовательно его координаты будут заданы тремя парами значений "X1,Y1,X2,Y2,X3,Y3"

Значит всё вместе пишем так:

```
<area href="primer2.html" shape="poly" coords="148,15,185,82,110,82">
```

А вот рисунок который наглядно показывает откуда берутся координаты точек..



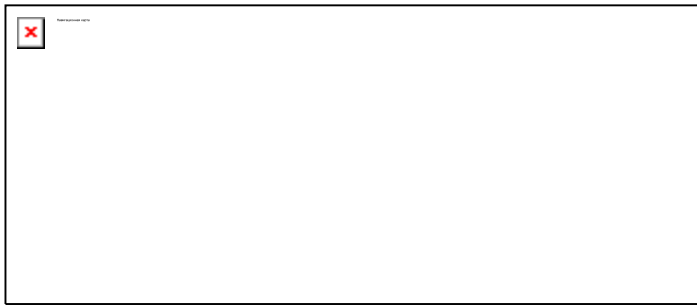
Пример:

```
<html>
<head>
  <title>Навигационная карта</title>
</head>
<body>
  <center>
    
  </center>
  <map name="panel">
    <area href="primer1.html" shape="rect" coords="15,15,82,82">
    <area href="primer2.html" shape="poly" coords="148,15,185,82,110,82">
  </map>
</body>
</html>
```

Окружность

Ну и последняя "красная кнопка" она у нас круглая.. значит форма области будет круглой **shape= "circle"**. На сей раз с координатами дело обстоит немного иначе. Нам понадобится три значения X,Y,R. X и Y это координаты центра нашей окружности, а R - это длина радиуса в пикселях.

Вот рисунок:



А вот пример:

```
<html>
<head>
  <title>Навигационная карта</title>
</head>
<body>
  <center>
    
  </center>
  <map name="panel">
    <area href="primer1.html" shape="rect" coords="15,15,82,82">
    <area href="primer2.html" shape="poly" coords="148,15,185,82,110,82">
    <area href="primerk3.html" shape="circle" coords="250,50,33">
  </map>
</body>
</html>
```

Доводим до ума..

Теперь немного о том, что ещё желательно было бы сделать с нашей картой прежде чем "монтировать" её в какую либо страницу.

Определим фиксированные размеры рисунка-карты атрибутами **width** и **height**

Напишем альтернативный текст, как для всего рисунка карты, так и для её отдельных областей используя атрибут **alt**, а также описание элементов атрибутом **title**.

Избавимся от рамки бордюра.. сделаем **border="0"**..(ну если Вам с бордюром больше нравится, можете этого не делать.. я не настаиваю..)

В конце должно выглядеть примерно вот так:

```
<html>
<head>
  <title>Навигационная карта</title>
</head>
<body>
  <center>
    
  </center>
  <map name="panel">
    <area href="primer1.html" shape="rect" coords="15,15,82,82" alt="Зелёная кнопка"
    title="Зелёная кнопка">
    <area href="primer2.html" shape="poly" coords="148,15,185,82,110,82" alt="Жёлтая
    кнопка" title="Жёлтая кнопка">
    <area href="primer3.html" shape="circle" coords="250,50,33" alt="Красная кнопка"
```

```
title="Красная кнопка">
  </map>
</body>
</html>
```

Пересечение областей

Иногда удобно формировать "рабочую" область изображения "смешивая" разные формы..

Предположим, наша очередная кнопка имеет такой вид:



Конечно, можно определить форму как многоугольник, но придется задать уйму координат для округлой части такой кнопки (ну это если нужна особая точность в навигационной карте).

А можно определить в этом примере две формы прямоугольник **rect** и окружность **circle** как показано на рисунке:



А в коде указать путь к одному и тому же документу:

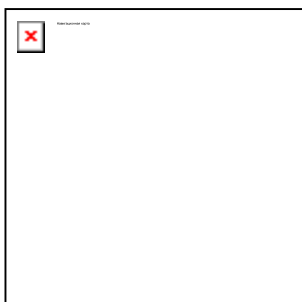
Пример:

```
<html>
<head>
  <title>Навигационная карта</title>
</head>
<body>
  <center>
    
  </center>
  <map name="panel1">
    <area href="primer4.html" shape="circle" coords="206,50,33" title="Синяя
кнопка">
    <area href="primer4.html" shape="rect" coords="80,17,206,83" title="Синяя
кнопка">
  </map>
</body>
</html>
```


Однако стоит помнить о том, что если бы ссылки с разных форм вели к разным документам, то нажимая на область пересечения наших фигур посетитель будет переходить по той ссылке, чья область в коде указана первой.. в нашем случае это окружность.

"Не область"

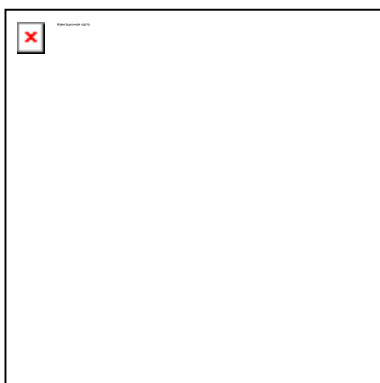
Рассмотрим на примере.. предположим необходимо изготовить вот такую кнопку:



Как быть с отверстием в ней?

Тег **<area>** помимо атрибута **href** имеет противоположенный по значению атрибут **nohref** - неактивная область, то есть если пользователь нажмет на такую область, то ровным счетом ничего не произойдет, чего нам собственно и необходимо добиться при изготовлении "отверстия" в нашей карте.

Карта будет задана двумя областями, неактивной окружностью **circle** и активной прямоугольной областью **rect** и как показано на рисунке чертеже иметь следующие координаты:



Как уже говорилось ранее, при пересечении областей наибольшим приоритетом будет пользоваться та область, которая в коде внутри тега **<map>** указана первой

Следовательно, пример будет иметь такой вид:

```
<html>
<head>
  <title>Навигационная карта</title>
</head>
<body>
  <center>
    
  </center>
  <map name="panel2">
    <area nohref shape="circle" coords="76,74,35" title="дырка">
    <area href="primer5.html" shape="rect" coords="25,22,126,124" title="Дырявая кнопка">
  </map>
```

Карта на сервере.

Отрывок из справочника: (<http://www.webremeslo.ru/spravka/spravka.html>)
(атрибуты тега ****):

usemap - изображение является навигационной картой на стороне клиента.
ismap - изображение является навигационной картой на сервере.

Непонятно? Тогда читаем..

С атрибутом **usemap** мы вроде бы разобрались.. Смотрите, пользователь (клиент) открыв Вашу страницу, одновременно со всем остальным содержанием, загружает "на свою сторону" как сам рисунок, так и навигационную карту к нему и всё это дело обрабатывается его браузером.

А вот атрибут **ismap** тега **** говорит браузеру о том, что к данному изображению на сервере, ну там где Вы выложили или собираетесь выложить свой сайт, есть навигационная карта. И теперь, когда посетитель (клиент) кликнет по какой либо области рисунка с таким атрибутом, браузер запомнит координаты этого клика и отправит их на сервер, где специальная программка обработает эти данные и перенаправит пользователя на тот адрес который указан в карте на сервере, в соответствии с полученными от браузера клиента координатами точки.

Пишется так:

```
<a href="http://www.site.ru/imagemaps/karta"></a>
```

Где **** адрес навигационной карты на каком либо сайте.. размещённом на том или ином сервере..

Всё равно непонятно? Если да, то тогда не стоит заморачиваться по этому поводу.. используйте **usemap**, на мой взгляд такое решение будет лучшим в большинстве случаев при использовании навигационных карт.

Полезные советы:

- Что бы без труда определить координаты той или иной точки на Вашей навигационной карте откройте рисунок графическим редактором тем же Paint к примеру.. там, когда будете двигать указателем по рисунку, увидите две меняющиеся цифры, это и есть координаты пикселя X и Y на данном рисунке. В Paint эта панелька расположена в нижней части экрана.
- При создании страницы с навигационной картой тег **** всегда должен быть выше самой карты **<map>** То есть писать вот так:

```
<map name="panel">  
  <area href="primer1.html" shape="rect" coords="20,20,80,80">  
</map>
```

```

```

..можно, но не нужно.. потому что при загрузке страницы могут возникнуть проблемы, так как карта с разметкой уже загрузилась, а самого рисунка ещё нет..

- И ещё что касается загрузки..

```

```

В этом месте кода может быть всё что угодно длинный текст, таблицы, графика.. но

всё же лучше здесь ничего не писать

```
<map name="panel">  
  <area href="primer1.html" shape="rect" coords="20,20,80,80">  
</map>
```

А писать здесь, так как во время загрузки страницы, пользователь не дождавшись её окончания, может пытаться нажимать на кнопки указанные в навигационной карте которая к этому времени еще не загрузилась..

Глава 9

Списки

В этой главе пойдёт речь о списках, которые могут быть:

- неупорядоченные(маркированные)
- упорядоченные (нумерованные)
- и являться списком определений

Такое вот нестандартное начало главы.. зато теперь Вам понятно, что есть список в html.

Ну что ж давайте пройдемся по нашему списку.

Неупорядоченные списки

Неупорядоченный список задаётся тегом ****. Любой список состоит из элементов, "пунктов", каждый элемент в свою очередь задаётся тегом **** после которого собственно и идёт текст нужного нам "пункта". Для тега **** закрывающий тег **** необязателен.

Вместе данные теги приобретают следующий вид:

```
<ul>
  <li> Пункт первый..
  <li> Пункт второй..
  <li> Пункт третий..
</ul>
```

Пример неупорядоченного списка:

```
<html>
<head>
  <title>неупорядоченный список</title>
</head>
<body>
  Купить и доставить домой:
  <ul>
    <li> 2 кг. клубней картофеля
    <li> 1 булку ржаного чёрного хлеба
    <li> пачку макарон "спагетти" (длинные не менее 200мм. тонкие)
    <li> 1 кг. шлифованного риса (сложно объяснить.. спросишь..)
    <li> 1 дес. куриных яиц.
    <li> 1 пачку 500г. Натрия двууглекислого (Сода)
  </ul>
  Смотри ничего не перепутай.. целую..
</body>
</html>
```

Теги **** и **** имеют атрибут **type** который присуждает элементу списка или же всему списку целиком определённый стиль.

Может иметь одно и трёх значений:

- **disk** - кружок, диск (по умолчанию)

- **circle** - полый круг
- **square** - квадрат

Пример:

```
<html>
<head>
  <title>стили неупорядоченного списка</title>
</head>
<body>
  В этом списке каждый элемент имеет свой стиль:
  <ul>
    <li type="disk"> кружок, диск (по умолчанию)
    <li type="circle"> полый круг
    <li type="square"> квадрат
  </ul>
  А здесь стиль задан всему списку
  <ul type="circle">
    <li> Пункт 1
    <li> Пункт 2
    <li> Пункт 3
  </ul>
</body>
</html>
```

Упорядоченные списки

Упорядоченный или нумерованный список задаётся тегом ****, так же как и в неупорядоченном списке, элемент списка присуждается тегом ****.

Построение кода полностью схоже с неупорядоченным списком, поэтому сразу пример:

```
<html>
<head>
  <title>упорядоченный список</title>
</head>
<body>
  Купить товары в следующем порядке:
  <ol>
    <li> Водка
    <li> Пиво
    <li> Сырок (необязательно)
  </ol>
  Жду!!!
</body>
</html>
```

А вот атрибут **type** в сочетании с упорядоченным списком может иметь следующие значения:

- **A** - Заглавные буквы
- **a** - Строчные буквы
- **I** - Заглавные римские цифры
- **i** - Строчные римские цифры
- **1** - Арабские цифры (по умолчанию)

Вот пример их применения:

```

<html>
<head>
  <title>Стили упорядоченного список</title>
</head>
<body>
  Арабские цифры
  <ol type="1">
    <li>Во-первых
    <li>Во-вторых
  </ol>
  Строчные буквы
  <ol type="a">
    <li>Во-первых
    <li>Во-вторых
  </ol>
  Заглавные буквы
  <ol type="A">
    <li>Во-первых
    <li>Во-вторых
    <li>В-третьих
  </ol>
  Строчные римские цифры
  <ol type="i">
    <li>Во-первых
    <li>Во-вторых
    <li>В-третьих
    <li>В-четвёртых
  </ol>
  Заглавные римские цифры
  <ol type="I">
    <li>Во-первых
    <li>Во-вторых
    <li>В-третьих
    <li>В-четвёртых
  </ol>
</body>
</html>

```

В упорядоченном списке есть ещё один атрибут **start** его числовое значение говорит о том с какого номера следует строить упорядоченный список.

Пример:

```

<html>
<head>
  <title>Начало упорядоченного списка</title>
</head>
<body>
  <ol type="1" start="24">
    <li>Сразу переходим к двадцать четвёртому пункту!!
    <li>Идём дальше
    <li>И дальше
  </ol>
  Аналогично можно "стартовать" при любом стиле упорядоченного списка
  <ol type="I" start="8">
    <li>Сразу переходим к восьмому пункту..
    <li>Идём дальше
    <li>И дальше
  </ol>

```

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

Списки определений

Со списком определений дело обстоит немного иначе, нежели чем с уже знакомыми списками. Задаётся данный вид списка тегом **<dl>**. Пункты списка определений размечаются тегом **<dt>**, а определения этих пунктов тегом **<dd>**.

Всё вместе пишется по следующей схеме:

```
<dl>
<dt>
<dd>
</dl>
```

Пример:

```
<html>
<head>
  <title>Список определений</title>
</head>
<body>
  <dl>
    <dt> Слово коса может иметь следующие определения:
      <dd> сельскохозяйственный инструмент
      <dd> хитрая девичья причёска
      <dd> отмель реки
    <dt> Слово ключ тоже имеет несколько значений:
      <dd> гаечный
      <dd> источник, родник
  </dl>
</body>
</html>
```

Собственно вот и вся премудрость..

Полезные советы:

Психология посетителя страницы.

- Не могу, не согласится с тем утверждением, что посетители страницы чаще просматривают документы, нежели чем вдумчиво читают их.. По-крайней мере так происходит практически всегда при первом визите посетителя на страничку. Первое на что обращает внимание пользователь просматривая какой либо документ это: заголовки в тексте, выделенные фрагменты текста, и списки. Умелое и ненавязчивое размещение таковых элементов залог того что посетитель окунётся в чтение Вашего документа более глубоко.

- Что касается списков.. следует понимать что: Маркированные перечни **** используются тогда, когда порядок следования пунктов не важен.. и делаются они для того чтобы изложенная информация воспринималась легче. А нумерованные списки **** наоборот, применяются тогда, когда важен именно порядок следования пунктов, например - когда список является изложением последовательности каких либо действий.

Глава 10

Meta теги

В первой главе этого учебника, об общем построении html документа, я говорил о том, что все html документы должны иметь вот такой шаблон кода:

<html> - начало документа
<head> - начало головы
</head> - закрытие головы
<body> - начало тела
</body> - закрытие тела
</html> - конец документа

Где между тегами **<body>** **</body>** указывается информация предназначенная для вывода на экран в нужном нам виде, а между тегами **<head>** **</head>** исключительно служебная информация предназначенная для поисковых систем и браузеров тех или иных пользователей. Так что же это за информация такая и для чего она нужна? Отвечу, планомерно и порционно в этой главе.

С тегом **<title>** мы уже знакомы, с помощью него мы указываем имя документа в заголовке страницы. Теперь новый тег **<meta>** (закрывающего тега не требует) с помощью него мы и будем указывать эту самую служебную информацию на нашей страничке.

<meta> тег имеет следующие атрибуты:

- **http-equiv** - указывает браузеру как следует обработать основное содержание документа, точнее на основе каких данных.
- **name** - информационное имя. (применяется в паре с атрибутом **content**)
- **content** - информационное содержание, связанное с мета именем (**name**)

Теперь на примерах будем вникать в суть дела.

Кодировка символов и язык

Пример (очень нужный и важный):

```
<meta http-equiv="Content-Type" Content="text/html; Charset=Windows-1251">
```

Сначала расскажу, зачем необходима эта строка в заголовке html документа. Данная запись указывает браузеру кодировку, в которой была написана данная страница - формат документа и раскладку клавиатуры, в данном случае это кириллица для Windows. Если эту строку не писать в заголовке страницы, то есть большая вероятность что весь текст на Вашей странице отобразится в виде непонятных человеку "иероглифов" у разных пользователей тех или иных браузеров. Конечно, пользователь может применить к такому документу команду в браузере **Вид->Кодировка->Кириллица**, но он может не знать о данной функции, да и зачем утруждать человека данным действием.

Теперь разберём по "слогам" нашу запись:

<meta http-equiv="Content-Type" - указываем, что в этом мета теге мы будем заниматься
Content-Type - типом содержания
Content="text/html;" - а именно его текстом
Charset=Windows-1251"> - документ для Windows - Кириллица, где **1251** кодировка раскладки клавиатуры, так например Английская клавиатура будет задаваться **Charset=Windows-1252**

В настоящее время продвинутые веб-мастера рекомендуют использовать кодировку UTF 8

То есть писать в голове документа вот так:

```
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
```

Пример:

```
<meta http-equiv="Content-Language" Content="ru">
```

В этой строчке говорится о том, что язык **Language** документа является русским
Content="ru"

Неправильная установка языка и раскладки клавиатуры может привести к печальным последствиям.

Информация о документе

Пример:

```
<meta name="author" Content="Остап Бендер">  
<meta name="copyright" Content="""Рога и копыта" Остап Бендер">
```

Данные метаописатели предназначены для заявления об авторских правах непосредственно в заголовке html кода, так **name="author"** указывает имя автора страницы, а **name="copyright"** авторское право (копирайт) в котором может указываться фамилия, имя, отчество автора сайта, название фирмы, бренда.. и т. д. Кроме того, включив в заголовок документа такое описание, Вы значительно упростите задачу поисковой машине при поиске Вашего сайта по имени автора, названию фирмы, бренду...

Пример:

```
<meta name="Generator" Content="Microsoft Notepad">
```

Если хотите можете указать с помощью какого html редактора была написана данная страница.

Описание страницы и ключевые слова

Пример:

```
<meta name="description" Content="Производим закупку по выгодным ценам рогов и копыт!">
```

Description - краткое описание страницы. Данное описание частенько используется поисковыми системами для вывода в результатах поиска, по какому либо запросу, информации о сайте и его назначении.

Пример:

```
<meta name="keywords" Content="рога, копыта, рожки, рог, копыто, копытце, закупка, покупка, приобретение, выгодно, продать, купить, сбыть, реализовать, корова, бык, коровьи, бычьи, оплата, деньги, наличные, цена, цене">
```

Keywords - ключевые слова веб-страницы, опять таки предназначены для поисковых машин.

Представьте, что Вы ищете в какой либо поисковой системе сайт с информацией о том, где можно продать те же рога и копыта :) Какие слова и фразы Вы будите вводить в строке "Поиск"? ну наверно что то типа: "Где продать коровьи рога?" или "Реализовать копыта по выгодной цене" Так вот если определить ключевые слова и так сказать предугадать мысли потенциального посетителя можно надеяться на то, что та или иная поисковая система выдаст ссылку на Ваш сайт в первых строчках результата поиска. Конечно ввод данного метоописателя не есть гарант того что именно Ваш сайт займет первые места в поиске по данным словам, но всё же не стоит им пренебрегать. Впрочем, оптимизация и раскрутка сайта это отдельная тема для разговора.

Помните, что описание **description** не должно превышать по длине более 200 символов, а ключевые слова **keywords** 1000 символов, иначе это может пагубно отразится при индексации Вашей страницы поисковыми системами.

Адрес

Пример:

```
<meta name="Publisher-Email" Content="Ваш_e-mail@сервер.домен">  
<meta name="Publisher-URL" Content="http://www.Ваш_сайт/">
```

Думаю понятно.. здесь указывается адрес Вашего почтового ящика **Publisher-Email** и адрес сайта **Publisher-URL**

Обновление страницы

Пример:

```
<meta name="revisit-after" Content="15 days">
```

Если некая страница на Вашем сайте подразумевает постоянное обновление и/или дополнение информационным содержанием, то хорошо было бы включить данное описание в заголовок данной страницы. Такое введение позволит программе роботу своевременно посещать Ваш сайт и индексировать его содержание. В нашем примере мы заявили о том, что собираемся обновлять содержание на странице не менее одного раза в 15 дней, можете не сомневаться программа робот возьмет Ваши планы себе на заметку и будет приходить "к Вам в гости" раз в пятнадцать дней, для того чтобы проверить ничего ли у Вас не изменилось..

Время действия документа и кэш

Пример:

```
<meta http-equiv="expires" content="Sun, 24 Jan 2010 12:28:36 GMT+03:00">
```

Для того чтобы ускорить загрузку страницы, а так же сэкономить трафик современные браузеры сохраняют посещаемые пользователем страницы в кэш (на жёсткий диск), и при повторном посещении загружают их не с сервера, а непосредственно с кэша. На самом деле такая функция хороша собой.. но есть одно "но", дело в том что в браузере может отображаться уже устаревшая информация, какой либо страницы. Представьте, к примеру, Ваш сайт представляет собой некое периодическое новостное интернет издание, а пользователь получит, вместо самых свежих новостей, уже устаревшую информацию, ту которая хранится у него в кэше!! и не разобравшись в чем "беда" примет Ваш сайт за "мертвый" заброшенный и никем не обновляемый.

Для того чтобы принудительно заставить браузер загружать ту или иную страницу не с жёсткого диска, а с сервера необходим мета тег с данным синтаксисом, где указывается день

недели, число месяц год время (чч:мм:сс) и часовой пояс(**GMT+03:00** - время Московское + три часа). День недели и время дня можно не указывать. Теперь при чтении страницы браузером страница будет грузиться с сервера, если указанная дата и время настало или просрочено, и напротив из кэша если указанное время еще не наступило.

Ниже на всякий случай приведены таблицы сокращений от Английских слов на месяцы и дни недели

Месяцы :	От Английского:	Сокращения:	Дни недели:	От Английского:	Сокращения:
Январь	January	Jan	Понедельник	Monday	Mon
Февраль	February	Feb	Вторник	Tuesday	Tue
Март	March	Mar	Среда	Wednesday	Wed
Апрель	April	Apr	Четверг	Thursday	Thu
Май	May	May	Пятница	Friday	Fri
Июнь	June	Jun	Суббота	Saturday	Sat
Июль	July	Jul	Воскресенье	Sunday	Sun
Август	August	Aug			
Сентябрь	September	Sep			
Октябрь	October	Oct			
Ноябрь	November	Nov			
Декабрь	December	Dec			

Атрибуту **content** можно присвоить значение **"0"** **<meta http-equiv="Expires" content="0">** в этом случае страница всегда будет загружаться с сервера.

И еще.. некоторые поисковые роботы могут отказаться индексировать документ с заведомо устаревшей датой. - не искушайте судьбу..

Пример:

```
<meta http-equiv="pragma" content="no-cache">
```

А такая запись вовсе запретит браузеру кэшировать данную страницу.

Пример:

```
<meta name="robots" content="Index,follow">
```

Данный мета тег предназначен для подачи поисковому роботу той или иной команды.

Список возможных команд роботу:

- **Index** - индексировать страницу
- **Noindex** - не индексировать страницу
- **Follow** - проследивать гиперссылки на странице
- **Nofollow** - не проследивать гиперссылки на странице
- **All** - индексировать страницу и проследивать гиперссылки на странице (по умолчанию)
- **None** - не индексировать страницу и не проследивать гиперссылки на странице

Автоматический переход на другую страницу

Пример:

```
<meta http-equiv="Refresh" content="10; URL=http://www.mysite/index.html">
```

Если вдруг по каким либо причинам Вы задумаете поменять URL адрес Вашего сайта то хорошо было бы на старом месте оставить страницу вроде этой:

```
<html>
  <head>
    <meta http-equiv="Content-Type" Content="text/html; Charset=Windows-1251">
    <meta http-equiv="Refresh" content="10; URL=http://www.mysite/index.html">
    <title>Переадресация</title>
  </head>
  <body>
    <font size="+1">
      Адрес сайта был изменен, через 10 секунд Ваш браузер будет автоматически
      перенаправлен по новому адресу:<br>
      <a href="http://www.mysite.ru/index.html"><b>http://www.mysite.ru/</b></a><br>
      Нажмите <a href="http://www.mysite.ru/index.html">здесь</a> для того чтобы выполнить
      переход немедленно.<br>
      Приносим извинения за доставленные неудобства.
    </font>
  </body>
</html>
```

Разберём и осмыслим строчку из примера:

```
<meta http-equiv="Refresh" content="10; URL=http://www.mysite/index.html">
```

meta http-equiv="Refresh" - Refresh (восстановление) указывает браузеру что данную страницу необходимо обновить

content="10; - обновить через заданное количество секунд (в нашем случае десять)

URL=http://www.mysite/index.html" - адрес новой/другой страницы на которую следует перейти.

Пример:

```
<meta http-equiv="Refresh" content="30">
```

А вот если в заголовке **Refresh** URL адрес упустить, как показано в примере, то тогда браузер будет постоянно через каждые 30 секунд (ну или не 30.. сколько пропишите через столько и будет..) обновлять содержимое данной страницы.

Такой метод широко используется в новостных лентах, где информация идет так сказать потоком и требует постоянного обновления.

Эффекты при переходе по ссылке

Пример:

```
<meta http-equiv="Page-Enter" Content="RevealTrans(Duration=1.0, Transition=0)">
<meta http-equiv="Page-Exit" Content="RevealTrans(Duration=3.0, Transition=23)">
```

Данные заголовки создают визуальные эффекты при переходе с одной страницы на другую.

- **Page-Enter** - Эффект появления страницы
- **Page-Exit** - Эффект исчезновения страницы

В которых:

- **Duration** - время действия эффекта в секундах
- **Transition** - Один из номеров предлагаемых эффектов (от 0 до 23) перечисленных в таблице:

Номер	Описание эффекта	Номер	Описание эффекта
0	Прямоугольники внутрь	12	Растворение
1	Прямоугольники наружу	13	Вертикальная панорама внутрь
2	Круг внутрь	14	Вертикальная панорама наружу
3	Круг наружу	15	Горизонтальная панорама внутрь
4	Наплыв наверх	16	Горизонтальная панорама наружу
5	Наплыв вниз	17	Уголки влево - вниз
6	Наплыв вправо	18	Уголки влево - вверх
7	Наплыв влево	19	Уголки вправо – вниз
8	Вертикальные жалюзи	20	Уголки вправо – вверх
9	Горизонтальные жалюзи	21	Случайные горизонтальные полосы
10	Шажки горизонтальные	22	Случайные вертикальные полосы
11	Шажки вертикальные	23	Случайный выбор эффекта

Пример:

Файл page1.html

```
<html>
<head>
<meta http-equiv="Content-Type" Content="text/html; Charset=Windows-1251">
<meta http-equiv="Page-Enter" Content="RevealTrans(Duration=1.0, Transition=12)">
<title>Эффекты перехода страниц</title>
</head>
<body bgcolor="#c5ffa0">
<center>
<h2>На заметку:</h2>
<font size="+1">Эффекты перехода с одной страницы на другую работают не во
```

всех браузерах.<hr>

Нажмите на "Перейти" чтобы перейти к следующей странице
и оценить эффект перехода от одной странице к другой.

"Перейти"

</center>

</body>

</html>

Файл page2.html

<html>

<head>

<meta http-equiv="Content-Type" Content="text/html; Charset=Windows-1251">

<meta http-equiv="Page-Enter" Content="RevealTrans(Duration=2.0, Transition=23)">

<title>Эффекты перехода страниц</title>

</head>

<body bgcolor="#c0e4ff">

<center>

<h2>На заметку:</h2>

Эффекты открытия и закрытия веб-страниц будут видны только при
переходе

от одной страницы к другой или же при помощи кнопок "назад" "вперёд".

При первом открыти страницы, а также во время перезагрузки

эффекты перехода видны не будут.<hr>

Нажмите на "Перейти" чтобы перейти к следующей странице

и оценить эффект перехода от одной странице к другой.

"Перейти"

</center>

</body>

</html>

Полезные советы:

- Ещё раз напомним о том что мета теги стоит применять умело и грамотно особенно это касается команд для робота и кодировки символов, иначе весь Ваш труд может пойти насмарку..

- Заголовок **Refresh** (автоматический переход на другую страницу) можно использовать не совсем стандартно.. Некоторые авторы используют его для создания своего рода "презентации" слайд шоу, где сменяющиеся страницы и есть кадры презентации. Представьте заходит человек на такой сайт а тут ему "Откинетесь на спинку кресла и расслабьтесь..":) а далее сами по себе пошли картинки, графики, тексты.. а последняя страница тупиковая где пользователь берёт сайт "в свои руки" или же может замыкаться на первую. Только всегда помните о золотом правиле веб-мастера: Главное не переборщить!

Глава 11

Заголовок <!DOCTYPE>

Ухитрился я таки перевернуть всё с ног наголову.. вообще то о заголовке **<!DOCTYPE>** нужно было говорить в самом начале данного учебника.. Ну до ничего ещё есть шанс всё исправить..

Но прежде чем продолжить обучение немного отвлечёмся.. ответе на вопрос:

Какие из этих слов, согласно правилам грамматики, написаны с ошибкой?: **Шіопоть, Шепоть, Шёпот, Шепот, Шопот**. Уверенны?

Но, подождите, я ведь не сказал, каких именно правил нужно придерживаться!

Так на старославянском было единственно верным писать "Шіопоть" потом, следуя логики, что одному звуку - одна буква, сочетание "іо" решили заменить буквой "е" получился "Шепоть". Потом родилась буква "ё" потому что народ не знал, как правильно написать слово "ёлка", далее с развитием телеграфа и типографии в целях экономии времени, бумаги, чернил и т.д., пропал твёрдый знак в окончаниях слов заканчивающихся на согласную букву. В настоящее время продвигают правило "Как слышим, так и пишем!" так что слово "Шопот" по нынешним меркам, уже, наверное, тоже правильно написано. Так же допускается писать букву "е" вместо "ё" если это не мешает верному пониманию слова, например "узнаем" вместо слова "узнаём".

Теперь представьте, что Вы пишете диктант и его будет проверять лингвист - старовер, учитель Марья Ивановна советской закалки, и тинэйджер, который привык общаться на сленге. Каким правилам грамматики Вы будите подчинятся чтобы всем угодить? Думаю единственно верным решением будет указать в начале диктанта, что то типа: "При написании диктанта придерживаюсь правил русского языка утверждённых приказом министра образования от такого то числа", тогда все проверяющие будут оценивать Ваш диктант не так как им больше нравится, а согласно правил, на которые Вы сослались.

К чему я это всё?

Язык разметки гипертекстовых документов, точно также как и русский язык живёт во времени и тоже меняется.. появляются новые слова - теги например **<footer>**, **<header>**, **<video>**.., какие-то наоборот умирают, забывается и осуждаются в использовании, например теги: **<center>**, ****, **<frameset>**.., тоже самое происходит с атрибутами, меняются правила синтаксиса..

Так вот чтобы различные браузеры например: "старовер" - Internet Explorer 6.0, "учительница советской закалки" - Opera 8.0, "тинэйджер" - Chrome 5.0 ... не разругались на Ваш документ - отображали его правильно, не запутались в чтении Вашей страницы, необходимо указать в соответствии с какими стандартами он был написан.

Теперь немного о самих стандартах.

Разработкой стандартов HTML языка (и не только HTML), иначе спецификаций, занимается организация **World Wide Web Consortium**, W3C - Консорциум Всемирной паутины, официальный сайт: (www.w3.org). С историй возникновения данной организации её целями и задачами, Вы можете ознакомиться **здесь: (http://ru.wikipedia.org/wiki/Консорциум_Всемирной_паутины)**, моя же задача доложить Вам о том, что эта организация разработала несколько спецификаций HTML, по которым, собственно, мы и учимся.

Вот официальные спецификации:

- **HTML 3.2**- 14 января 1997 года; (<http://www.w3.org/TR/REC-html32>)
- **HTML 4.0** - 18 декабря 1997 года; (<http://www.w3.org/TR/REC-html40-971218>)
- **HTML 4.01** - 24 декабря 1999 года; (<http://www.w3.org/TR/html401>)
- **HTML 5** - в разработке. (официально ещё не объявлена, но уже используется) (<http://www.w3.org/TR/html5>)

Именно этими документами должны руководствоваться как веб-мастера при создании сайтов, так и разработчики браузеров. Эти стандарты созданы для того чтобы не вышло как в Библейской истории про Вавилонскую башню, где все вдруг начали говорить на разных языках и окончательно перестали понимать друг друга.. а такой момент называл в своё время, да сейчас не всё так уж гладко, впрочем это уже совсем другая история.. вернемся к **<!DOCTYPE>**.

Итак, заголовок **<!DOCTYPE>** указывает на тип документа - **DTD** (document type definition - описание типа документа) для правильной его интерпретации браузерами, другими словами указывал браузерам, согласно каким стандартам следует обрабатывать ту или иную страницу.

Ниже перечислены основные варианты **<!DOCTYPE>**

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
```

- Строгий DTD. Используя такой заголовок, веб-страница должна в точности следовать спецификации HTML 4.01 не использовать теги и атрибуты, обозначенные спецификацией как "нежелательные", а также не должна использовать фреймы.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
```

- Переходный синтаксис HTML. При таком заголовке допускаются "вольности" при составлении документа, страница может содержать теги и атрибуты, помеченные спецификацией HTML 4.01 как "нежелательные".

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Frameset//EN"
"http://www.w3.org/TR/html4/frameset.dtd">
```

- Указывает, что в HTML-документе используются фреймы.

```
<!DOCTYPE html>
```

- А такой заголовок обозначает, что используется спецификация HTML 5.

Существуют и другие варианты заголовков **<!DOCTYPE>**, например для XHTML, выше перечислены основные для HTML.

Заголовок **<!DOCTYPE>** принято располагать в самом начале документа перед тегом **<html>**

Как показано на примере:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
  <head>
    <title>&lt;!DOCTYPE&gt;</title>
  </head>
  <body>
    <center>
      <font face="arial" size="+2" color="#ff0000">
```


Данный документ содержит теги и атрибуты осуждаемые спецификацией HTML

4.01

```
</font>
<br><br>
<font face="arial" size="+1" color="#0fff00">Для того чтобы этот документ прошёл
проверку на валидность следует указывать переходный заголовок
<!DOCTYPE></font>
</center>
</body>
</html>
```

Полезные советы:

- Всегда используйте заголовок **<!DOCTYPE>** для всех Ваших страничек.
- Проверяйте Ваш код на валидность - "правильность написания" с помощью специальных сервисов. Например **здесь: (<http://validator.w3.org>)**
- Использование неправильного - несоответствующего HTML документу **<!DOCTYPE>** может привести к плачевным результатам, будьте внимательны и осторожны при выборе нужного заголовка. В большинстве случаев для страниц, не использующих фреймы лучше указывать переходный **<!DOCTYPE>**

Глава 12

И снова таблицы..

Данная глава является продолжением 4 главы **про таблицы**, отложил я её на потом по нескольким причинам. Во-первых, четвёртая глава и без того получилась длинноватой и "неповоротливой".. Во-вторых, не хотел, что бы Вы на тот момент погрязли в изучении таблиц и так сказать топтались на месте.. обучение, на мой взгляд, должно происходить живенько, интересно, а главное результативно! Зато теперь, когда у Вас сложилось общее представление о HTML, Вы разобрались с его базовой частью, сделали свой первый нехитрый сайт, можно наращивать багаж знаний, умений и навыков. Ну и в-третьих, я надеюсь, что Вы к этому времени, параллельно начали **изучать CSS: (<http://www.webremeslo.ru/css/glava0.html>)**.. что разве ещё нет? жаль.. Дело в том, что далее в этом учебнике нет нет будут попадаться вещи, которые сами по себе могут показаться бесполезными и бессмысленными, однако в купе с другими науками они порой незаменимы!

Ладно, вернёмся к таблицам..

В качестве "полигона" для изучения нового материала возьмём вот такую жутко скучную таблицу:

Продукт	Белки	Жиры	Углеводы	ккал
Хлеб ржаной	4,7	0,7	49,8	214
Молоко	2,8	3,2	4,7	58
Картофель	2	0,1	19,7	83
Свинина	11,4	49,3	9	489
Итого:	20,9	53,3	83,2	844

Ну что ж, давайте поколдуюем над ней..

Заголовки таблиц.

Любая ячейка таблицы может быть определена как заголовок. Для того чтобы сделать ячейку таблицы заголовком нужно вместо знакомого тега **<td>** использовать тег **<th>**.

Например, вот так:

```
<table>
  <tr>
    <th>Данная ячейка является заголовком</th>
  </tr>
</table>
```

Как правило, текст ячейки определённый в качестве заголовка отображается браузерами жирным шрифтом по центру ячейки.

Ну что давайте попробуем сделать некоторые ячейки нашей таблицы заголовками? смотрим пример:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
```

```

<head>
  <title>Заголовок таблицы</title>
</head>
<body>
  <table border="1" width="400" cellpadding="3" cellspacing="0">
    <tr>
      <th>Продукт</th><th>Белки</th><th>Жиры</th><th>Углеводы</th><th>ккал</th>
    </tr>
    <tr>
      <th>Хлеб ржаной</th><td>4,7</td><td>0,7</td><td>49,8</td><td>214</td>
    </tr>
    <tr>
      <th>Молоко</th><td>2,8</td><td>3,2</td><td>4,7</td><td>58</td>
    </tr>
    <tr>
      <th>Картофель</th><td>2</td><td>0,1</td><td>19,7</td><td>83</td>
    </tr>
    <tr>
      <th>Свинина</th><td>11,4</td><td>49,3</td><td>9</td><td>489</td>
    </tr>
    <tr>
      <th>Итого:</th><td>20,9</td><td>53,3</td><td>83,2</td><td>844</td>
    </tr>
  </table>
</body>
</html>

```

К тегу **<th>** могут применяться точно такие же атрибуты как и к тегу **<td>** о них мы говорили в 4 главе поэтому повторяться не буду, а просто перечислю их:

- **width** - Ширина ячейки в процентах или пикселях.
- **height** - Высота ячейки в процентах или пикселях.
- **align** - Выравнивает текст в ячейке:
 - **left** - по левому краю.
 - **right** - по правому краю.
 - **center** - по центру.
 - **justify** - по левому и правому краю.
- **valign** - Выравнивает текст в ячейке по вертикали:
 - **top** - по верхнему краю.
 - **middle** - по центру.
 - **bottom** - по нижнему краю.
 - **baseline** - по базовой линии.
- **colspan** - количество столбцов занимаемое ячейкой.
- **rowspan** - количество рядов занимаемое ячейкой.
- **bgcolor** - Цвет фона ячейки.
- **background** - Задаёт фоновый рисунок в ячейке.
- **bordercolor** - Цвет бордюра.
- **title** - Всплывающая подсказка
- **nowrap** - Запрещает перенос строк в ячейке.

Думаю понятно, а посему идем дальше..

Название таблицы и её описание

Тег **<caption>** позволяет озаглавить таблицу, подписать её каким либо текстом. Данный тег должен располагаться внутри таблицы сразу после **<table>**

```

<table>
<caption>Название таблицы</caption>
<tr>
<td>ячейка</td>
</tr>
</table>

```

Вот пример:

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<title>Название таблицы</title>
</head>
<body>
<table border="1" width="400" cellpadding="3" cellspacing="0">
  <caption><b>Энергетическая ценность продуктов питания:</b></caption>
  <tr>
    <th>Продукт</th><th>Белки</th><th>Жиры</th><th>Углеводы</th><th>ккал</th>
  </tr>
  <tr>
    <th>Хлеб ржаной</th><td>4,7</td><td>0,7</td><td>49,8</td><td>214</td>
  </tr>
  <tr>
    <th>Молоко</th><td>2,8</td><td>3,2</td><td>4,7</td><td>58</td>
  </tr>
  <tr>
    <th>Картофель</th><td>2</td><td>0,1</td><td>19,7</td><td>83</td>
  </tr>
  <tr>
    <th>Свинина</th><td>11,4</td><td>49,3</td><td>9</td><td>489</td>
  </tr>
  <tr>
    <th>Итого:</th><td>20,9</td><td>53,3</td><td>83,2</td><td>844</td>
  </tr>
</table>
</body>
</html>

```

К тегу **<caption>** может быть применён атрибут **align** - выравнивание названия таблицы по горизонтали с возможными значениями **left**, **right** и **center**, а также атрибут **valign** который говорит о том где должно располагаться название сверху - **top** или снизу - **bottom** таблицы.

Однако хочу отметить, что данные атрибуты в разных браузерах ведут себя по разному, так например, запись:

```
<caption align="left">Текст</caption>
```

- для браузеров IE и Opera разместит название сверху таблицы по её левому краю, а для браузера Firefox это будет значить, что название следует размещать слева от самой таблицы.

Так же название таблицы или её краткое описание можно указывать с помощью атрибута **summary** тега **<table>**

Вот так:

```
<table summary="описание таблицы">
```

Такое описание никак не отображается и не выводится на экран обычными браузерами, однако может быть использовано поисковыми системами, а так же речевыми браузерами.

Пример:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
  <head>
    <title>Описание таблицы</title>
  </head>
  <body>
    <table border="1" width="400" cellpadding="3" cellspacing="0"
summary="Энергетическая ценность продуктов питания">
      <tr>
        <th>Продукт</th><th>Белки</th><th>Жиры</th><th>Углеводы</th><th>ккал</th>
      </tr>
      <tr>
        <th>Хлеб ржаной</th><td>4,7</td><td>0,7</td><td>49,8</td><td>214</td>
      </tr>
      <tr>
        <th>Молоко</th><td>2,8</td><td>3,2</td><td>4,7</td><td>58</td>
      </tr>
      <tr>
        <th>Картофель</th><td>2</td><td>0,1</td><td>19,7</td><td>83</td>
      </tr>
      <tr>
        <th>Свинина</th><td>11,4</td><td>49,3</td><td>9</td><td>489</td>
      </tr>
      <tr>
        <th>Итого:</th><td>20,9</td><td>53,3</td><td>83,2</td><td>844</td>
      </tr>
    </table>
  </body>
</html>
```

Группировка строк.

Знакомимся с тегами: **<thead>**, **<tbody>** и **<tfoot>**. Данные теги являют собой емкость для строк таблицы (тег **<tr>**) позволяя тем самым группировать и логически ими управлять.

Так:

- Тег **<thead>** - (От Англ. table head - "голова" таблицы) Предназначен для группировки одной или нескольких верхних строк в таблице, делая тем самым "шапку" таблицы.
- Тег **<tbody>** - (От Англ. table body - "тело" таблицы), Как правило этим тегом группируют центральные строки в таблице с основным содержанием.
- Тег **<tfoot>** - (От Англ. table footer - нижний колонтитул таблицы) Определяет "подвал" таблицы, состоящий из одной или нескольких нижних строк.

Следует отметить несколько правил использования данных тегов:

Данные теги могут быть расположены только внутри таблицы - контейнера **<table>**.

Теги **<thead>** и **<tfoot>** допускается использовать только один раз для одной таблицы, на тег **<tbody>** данное правило не распространяется.

Структура таблицы при использовании всех трёх данных тегов должна выглядеть вот так:

```
<table>
  <thead>
    <tr>
      <td>голова</td>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td> тело</td>
    </tr>
  </tbody>
  <tfoot>
    <tr>
      <td>подвал</td>
    </tr>
  </tfoot>
</table>
```

То есть сверху всегда должен быть контейнер **<thead>** потом **<tbody>** и **<tfoot>**, данные теги не могут пересекаться и быть вложенными друг в друга. А вот тег **<tfoot>** можно поменять местами с **<tbody>**, и что примечательно строки взятые в данный тег всё равно браузерами будут отображаться внизу таблицы.

К данным тегам допускается применять атрибуты выравнивания текста в ячейках **align** и **valign**, указывать цвет фона строк взятых в группу с помощью **b bgcolor**, делать всплывающую подсказку с помощью **title**, однако на практике работать во всех браузерах будет только **align** и **title**.

Просто ради информации.. есть ещё атрибуты:

- **char** - Выравнивание относительно заданного символа.
- **charoff** - Смещение относительно указанного символа.

Но эти товарищи вообще не прижились ни в одном из известных браузеров! Толку с них как с козла молока.

Так зачем спросите Вы вообще нужны теги группирования строк если с помощью них практически ничего не сделать? Отвечу чуть ниже, а пока смотрим пример:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
  <head>
    <title>Группировка строк</title>
  </head>
  <body>
    <table border="1" width="400" cellpadding="3" cellspacing="0">
      <caption><b>Энергетическая ценность продуктов питания:</b></caption>
      <thead bgcolor="#deb887" title="Шапка">
        <tr>
          <th>Продукт</th><th>Белки</th><th>Жиры</th><th>Углеводы</th><th>ккал</th>
        </tr>
      </thead>
      <tbody align="left" bgcolor="#faebd7" title="Энергетическая ценность
продуктов питания">
        <tr>
          <th>Хлеб ржаной</th><td>4,7</td><td>0,7</td><td>49,8</td><td>214</td>
```

```

</tr>
<tr>
  <th>Молоко</th><td>2,8</td><td>3,2</td><td>4,7</td><td>58</td>
</tr>
<tr>
  <th>Картофель</th><td>2</td><td>0,1</td><td>19,7</td><td>83</td>
</tr>
<tr>
  <th>Свинина</th><td>11,4</td><td>49,3</td><td>9</td><td>489</td>
</tr>
</tbody>
<tfoot align="left" bgcolor="#f5f5dc" title="Итого:">
  <tr>
    <th>Итого:</th><td>20,9</td><td>53,3</td><td>83,2</td><td>844</td>
  </tr>
</tfoot>
</table>
</body>
</html>

```

Теперь отвечу, зачем вообще нужны **<thead>**, **<tbody>** и **<tfoot>**..

Основное предназначение данных тегов это логическая группировка строк в группы, а это в свою очередь облегчает работу с таблицей, например через стили или скрипты, теперь не нужно определять тот же стиль для каждой строчки, а достаточно указать его для всей группы.. Я сказал стиль? Ах да я же забыл, что Вы еще незнакомы с CSS..

Выходя за рамки темы данной главы, хочу отметить, что практически к любому тегу применим замечательный атрибут **style**, с помощью которого можно сделать практически все, что касается дизайна того или иного элемента.. А еще есть не менее замечательный тег **<style>** который внедряет в страницу стили CSS.. но это всё материал из другого **учебника**: (<http://www.webremeslo.ru/css/glava0.html>), поэтому здесь мы говорить об этом не будем.

Просто покажу пока что малопонятный для Вас пример:

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
  <head>
    <title>Группировка строк и CSS</title>
    <style type="text/css">
      table {width: 600px; margin: 30px; border: #b8860b 6px ridge; border-collapse: collapse;}
      td,th,tr {border: #b8860b 2px outset; padding: 3px;}
      caption{font: italic bold 16px Arial; padding: 10px;}
      thead {background-color: #deb887; text-align: center;}
      tbody {background-color: #faebd7; text-align: left;}
      tfoot {background-color: #f5f5dc; text-align: left; font: italic 14px Arial;}
      tr:hover {background-color:#b8860b; color: #ffffff;}
    </style>
  </head>
  <body>
    <table>
      <caption>Энергетическая ценность продуктов питания:</caption>
      <thead title="Шапка">
        <tr>
          <th>Продукт</th><th>Белки</th><th>Жиры</th><th>Углеводы</th><th>ккал</th>
        </tr>
      </thead>
      <tbody title="Энергетическая ценность продуктов питания">

```

```

<tr>
  <th>Хлеб ржаной</th><td>4,7</td><td>0,7</td><td>49,8</td><td>214</td>
</tr>
<tr>
  <th>Молоко</th><td>2,8</td><td>3,2</td><td>4,7</td><td>58</td>
</tr>
<tr>
  <th>Картофель</th><td>2</td><td>0,1</td><td>19,7</td><td>83</td>
</tr>
<tr>
  <th>Свинина</th><td>11,4</td><td>49,3</td><td>9</td><td>489</td>
</tr>
</tbody>
<tfoot title="Итого:">
  <tr>
    <th>Итого:</th><td>20,9</td><td>53,3</td><td>83,2</td><td>844</td>
  </tr>
</tfoot>
</table>
</body>
</html>

```

Курсивом в данном примере выделено стилевое описание тегов написанное с помощью средств CSS.

Группировка столбцов.

Группировать можно не только строки, но и столбцы таблицы с помощью тегов **<col>** и/или **<colgroup>** данные теги позволяют определить основные параметры столбцов таблицы с помощью следующих атрибутов:

- **width** - Ширина столбца в процентах или пикселях.
- **align** - Выравнивает текст в ячейке:
 - **left** - по левому краю.
 - **right** - по правому краю.
 - **center** - по центру.
 - **justify** - по левому и правому краю.
- **valign** - Выравнивает текст в ячейке по вертикали:
 - **top** - по верхнему краю.
 - **middle** - по центру.
 - **bottom** - по нижнему краю.
 - **baseline** - по базовой линии.
- **bgcolor** - Цвет фона ячейки.
- **span** - Количество столбцов, к которым следует применять параметры.

Давайте сгруппируем колонки нашей пробной таблицы про энергетическую ценность продуктов питания, она у нас состоит из пяти столбцов, значит, после тега **<table>** следует написать, что-то типа:

```

<col width="140" bgcolor="#deb887">
<col width="100" bgcolor="#faebd7">
<col width="100" bgcolor="#faebd7">
<col width="100" bgcolor="#faebd7">
<col width="60" bgcolor="#f5f5dc">

```

Для каждой колонки свой тег **<col>** с указанными параметрами, где первый тег это описание левой колонки таблицы. второй тег - группирует вторую колонку и далее по порядку..

Обратите внимание, что в нашей таблице вторая, третья и четвертая колонки имеют одинаковые параметры **width="100" bgcolor="#faebd7"** так вот чтобы не прописывать каждую одинаковую колонку отдельно можно воспользоваться атрибутом **span**, который указывает какое количество последующих колонок имеет идентичные параметры. В нашем случае запись приобретёт следующий вид:

```
<col width="140" bgcolor="#deb887">
<col span="3" width="100" bgcolor="#faebd7">
<col width="60" bgcolor="#f5f5dc">
```

Таким образом мы указали, что три колонки, начиная со второй, имеют одинаковую ширину и цвет фона. Представьте, если бы таких колонок было штук пятьдесят? как бы мы облегчили код.

Ну а вот собственно рабочий пример наших стараний:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
  <head>
    <title>Группировка столбцов</title>
  </head>
  <body>
    <table width="500" border="1" cellpadding="3" cellspacing="0">
      <caption><b>Энергетическая ценность продуктов питания:</b></caption>
      <col width="140" bgcolor="#deb887">
      <col span="3" width="100" bgcolor="#faebd7">
      <col width="60" bgcolor="#f5f5dc">
      <thead title="Шапка">
        <tr>
          <th>Продукт</th><th>Белки</th><th>Жиры</th><th>Углеводы</th><th>ккал</th>
        </tr>
      </thead>
      <tbody align="left" title="Энергетическая ценность продуктов питания">
        <tr>
          <th>Хлеб ржаной</th><td>4,7</td><td>0,7</td><td>49,8</td><td>214</td>
        </tr>
        <tr>
          <th>Молоко</th><td>2,8</td><td>3,2</td><td>4,7</td><td>58</td>
        </tr>
        <tr>
          <th>Картофель</th><td>2</td><td>0,1</td><td>19,7</td><td>83</td>
        </tr>
        <tr>
          <th>Свинина</th><td>11,4</td><td>49,3</td><td>9</td><td>489</td>
        </tr>
      </tbody>
      <tfoot align="left" title="Итого:">
        <tr>
          <th>Итого:</th><td>20,9</td><td>53,3</td><td>83,2</td><td>844</td>
        </tr>
      </tfoot>
    </table>
  </body>
</html>
```

Тег **<colgroup>** практически идентичен тегу **<col>** и имеет точно такие же атрибуты, так наш выше изложенный пример мог бы иметь такой вид:

```
<colgroup width="140" bgcolor="#deb887">
<colgroup span="3" width="100" bgcolor="#faebd7">
<colgroup width="60" bgcolor="#f5f5dc">
```

Никакой заметной разницы мы бы не увидели. Однако, некоторые отличия всё же есть.. Во-первых, **<colgroup>** может служить емкостью для тегов **<col>**, конструкция такая:

```
<colgroup>
<col>
</colgroup>
```

Во-вторых, если бы мы к тегу **<table>** применили атрибут **rules** со значением **groups** (об этом атрибуте речь пойдёт ниже) эффект будет достигнут только при использовании **<colgroup>**.

Больше никакой разницы между этими тегами нет.

Пример:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<title>colgroup</title>
</head>
<body>
<table border="1">
<colgroup bgcolor="#deb887" align="left">
<col span="3" width="20">
</colgroup>
<colgroup bgcolor="#faebd7" align="center">
<col span="2" width="40">
<col width="60">
</colgroup>
<colgroup bgcolor="#f5f5dc" align="right">
<col width="20">
<col width="30">
<col width="50">
</colgroup>
<tr>
<td>20</td><td>20</td><td>20</td><td>40</td><td>40</td><td>60</td><td>20</td>
<td>30</td><td>50</td>
</tr>
<tr>
<td>20</td><td>20</td><td>20</td><td>40</td><td>40</td><td>60</td><td>20</td>
<td>30</td><td>50</td>
</tr>
<tr>
<td>20</td><td>20</td><td>20</td><td>40</td><td>40</td><td>60</td><td>20</td>
<td>30</td><td>50</td>
</tr>
<tr>
<td>20</td><td>20</td><td>20</td><td>40</td><td>40</td><td>60</td><td>20</td>
<td>30</td><td>50</td>
</tr>
</table>
</body>
</html>
```

Обратите внимание, что в этом примере колонки таблицы поделены на три группы с помощью **<colgroup>**, в этих группах мы указали цвет фона и способ выравнивания, а ширину колонок мы задаём с помощью тегов **<col>**, которые находятся внутри контейнеров **<colgroup>**.

Как и в случае с группировкой строк, колонки группируются с основной целью дальнейшего воздействия на них через стили или скрипты.

Границы таблицы.

Ранее в главе 4 "Таблицы" мы с Вами познакомились с атрибутами **border** - ширина границы и **bordercolor** - цвет границы, но это не весь набор инструментов применимых к границам таблицы.. Есть ещё два атрибута: **frame** и **rules** о них мы и поговорим ниже.

Начнём с атрибута **frame** - рамка, данный атрибут применяется к тегу **<table>** и указывает на то, как следует отображать внешние границы таблицы, т.е. то есть вокруг самой таблицы, а не между её отдельными ячейками

Атрибут **frame** может принимать следующие значения:

- **void** - не отображать границы вокруг таблицы.
- **border** - отображать границы вокруг таблицы (по умолчанию).
- **hsides** - отображать только горизонтальные границы сверху и снизу.
- **vsides** - отображать только вертикальные границы слева и справа.
- **above** - отображать только верхнюю границу таблицы.
- **below** - отображать только нижнюю границу таблицы.
- **lhs** - отображать только левую границу.
- **rhs** - отображать только правую границу.

Пример:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
  <head>
    <title>frame</title>
  </head>
  <body>
    <table border="8" frame="hsides" width="400" cellpadding="3" cellspacing="0">
      <caption><b>Энергетическая ценность продуктов питания:</b></caption>
      <thead bgcolor="#deb887" title="Шапка">
        <tr>
          <th>Продукт</th><th>Белки</th><th>Жиры</th><th>Углеводы</th><th>ккал</th>
        </tr>
      </thead>
      <tbody align="left" bgcolor="#faebd7" title="Энергетическая ценность продуктов
питания">
        <tr>
          <th>Хлеб ржаной</th><td>4,7</td><td>0,7</td><td>49,8</td><td>214</td>
        </tr>
        <tr>
          <th>Молоко</th><td>2,8</td><td>3,2</td><td>4,7</td><td>58</td>
        </tr>
        <tr>
          <th>Картофель</th><td>2</td><td>0,1</td><td>19,7</td><td>83</td>
        </tr>
        <tr>
          <th>Свинина</th><td>11,4</td><td>49,3</td><td>9</td><td>489</td>
        </tr>
      </tbody>
    </table>
```

```

</tbody>
<tfoot align="left" bgcolor="#f5f5dc" title="Итого:">
  <tr>
    <th>Итого:</th><td>20,9</td><td>53,3</td><td>83,2</td><td>844</td>
  </tr>
</tfoot>
</table>
</body>
</html>

```

А вот атрибут **rules** в отличие от **frame** наоборот указывает на то, как следует рисовать границы между ячейками таблицы.

Атрибут **rules** может принимать следующие значения:

- **all** - отображать все границы между ячейками.
- **none** - не отображать границы между ячейками.
- **rows** - отображать границы только между строками.
- **cols** - отображать границы только между столбцами.
- **groups** - отображать границы только между группами строк и группами столбцов образованными с помощью тегов: **<thead>**, **<tbody>**, **<tfoot>** и **<colgroup>**.

Пример:

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
  <head>
    <title>rules</title>
  </head>
  <body>
    <table border="1" rules="groups" width="500" cellpadding="3" cellspacing="0">
      <caption><b>Энергетическая ценность продуктов питания:</b></caption>
      <thead bgcolor="#deb887" title="Шапка">
        <tr>
          <th>Продукт</th><th>Белки</th><th>Жиры</th><th>Углеводы</th><th>ккал</th>
        </tr>
      </thead>
      <tbody align="left" bgcolor="#faebd7" title="Энергетическая ценность продуктов
питания">
        <tr>
          <th>Хлеб ржаной</th><td>4,7</td><td>0,7</td><td>49,8</td><td>214</td>
        </tr>
        <tr>
          <th>Молоко</th><td>2,8</td><td>3,2</td><td>4,7</td><td>58</td>
        </tr>
        <tr>
          <th>Картофель</th><td>2</td><td>0,1</td><td>19,7</td><td>83</td>
        </tr>
        <tr>
          <th>Свинина</th><td>11,4</td><td>49,3</td><td>9</td><td>489</td>
        </tr>
      </tbody>
      <tfoot align="left" bgcolor="#f5f5dc" title="Итого:">
        <tr>
          <th>Итого:</th><td>20,9</td><td>53,3</td><td>83,2</td><td>844</td>
        </tr>
      </tfoot>
    </table>
  </body>
</html>

```

```
</table>
</body>
</html>
```

Атрибут **rules** напрочь игнорирует браузер Опера, но как я уже не раз намекал такие проблемы можно решить с помощью средств CSS.

Полезные советы:

- Тег **<col>** помимо своей прямой задачи, решает еще одну полезную штуку.. При загрузке таблицы браузеры не показывают её содержание до тех пор пока таблица полностью не загрузилась. Так вот при использовании тега **<col>**, браузер зная количество и будущие размеры колонок, сразу начинает показывать содержание, не дожидаясь полной загрузки.. что есть гуд! особенно при медленном соединении Интернета. Сайты свёрстанные табличным способом перестают "плясать" во время загрузки.
- По наблюдениям, на тег **<th>** обращают внимание поисковые системы, и слова в заголовке таблицы имеют больший "вес", в глазах поисковиков, нежели чем текст в обычных ячейках, что в свою очередь может повлиять на результаты выдачи по тому или иному поисковому запросу.
- Всё что написано в этой главе, по большей своей части, ориентированно на логику таблиц, что облегчает жизнь всем!:
- Веб-мастерам, для более простого управления таблицей через стили или скрипты.
- Браузерам, и речевым браузерам для более корректного отображения или чтения таблиц.
- Алгоритмам поисковых систем, которые стараются найти эту самую логику на той или иной странице, чтобы выдать пользователю наиболее релевантные результаты.

Так что не стоит пренебрегать тегами: **<thead>**, **<tbody>**, **<tfoot>**, **<th>**, **<col>**, **<caption>**, **<colgroup>** - они весьма полезны!

Глава 13

Фоновая музыка.

.. или то чего лучше не делать.

Раскрывая все возможности HTML языка, в тринадцатой несчастливой главе, вынужден Вам рассказать о теге **<bgsound>** который позволяет внедрять в страницу сайта фоновую музыку. Другими словами когда пользователь зайдёт на Ваш сайт он услышит звучание музыки указанной с помощью вышеупомянутого тега, ну естественно если у него подключены колонки или наушники.

Итак, тег **<bgsound>** первое, что необходимо сказать об этом теге то, что он не входит в официальную спецификацию HTML, а является инициативой компании Microsoft для браузера Internet Explorer, так что большинство других браузеров игнорируют данный тег.

Для того чтобы вставить аудио файл в какую либо страницу сайта, в качестве фоновой музыки, нужно в теге **<bgsound>** с помощью атрибута **src** указать путь к аудио файлу

Вот так:

<bgsound src="music.mid"> - закрывающий тег необязателен.

Ну и собственно вставить эту конструкцию в html документ, обычно тег **<bgsound>** располагают в "голове" документа между тегами **<head>** **</head>**.

Сам аудио файл должен быть в формате: **WAV**, **AU** или **MIDI**.

Пример:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
  <head>
    <title>Фоновая музыка</title>
    <bgsound src="mammoth.mid">
  </head>
  <body>
    <table border="0" align="center" width="800" cellspacing="0" cellpadding="15">
      <tr>
        <td valign="top"></td>
        <td>
          <h2>Песенка мамонтенка.</h2>
          <pre>
По синему морю, к зелёной земле
Плыву я на белом своём корабле.
На белом своём корабле,
На белом своём корабле.
... ..
<b>Музыка: В. Шаинский </b>
<b>Слова: Д. Непомнящий</b>
          </pre>
        </td>
      </tr>
```

```
</table>
</body>
</html>
```

Музыка нааас сближала.. тайною нааашей стала.. :)

Атрибуты тега <bgsound>

К тегу <bgsound> могут быть применены следующие атрибуты:

- **src** - адрес звукового файла
- **loop** - число повторов звукового файла
- **balance** - стереобаланс
- **volume** - громкость

Теперь поподробнее:

src - адрес звукового файла.. ну думаю понятно указывает путь к аудио файлу, например такая запись:
<bgsound src="music/mammoth.mid"> говорит, что рядом с документом есть папка **music**, а в ней лежит музыкальный файл **mammoth.mid** который следует воспроизвести в качестве фоновой музыки.

loop - данный атрибут указывает сколько раз подряд будет проигрываться аудио файл, значения 0 и 1 проигрывает файл один раз, любое целое положительное число проигрывает звуковой файл заданное число раз, а вот отрицательное значение -1 говорит браузеру о том что данный файл необходимо проигрывать бесконечно. По умолчанию значение равно 1.

balance - стереобаланс между правой и левой колонками компьютера, возможный диапазон значений от -10000 до 10000. Так, например значение -10000 переведёт весь звук на левую колонку компьютера при этом из правой колонки будет доноситься лишь тишина.. Значение по умолчанию равно 0 т.е. распределять звук равномерно между левой и правой колонками.

volume - громкость можно определить значениями в диапазоне от -10000 до 0, где ноль это максимальная громкость, а -10000 минимальная. Значение 0 стоит по умолчанию.

Пример:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
  <head>
    <title>Фоновая музыка</title>
    <bgsound src="music/mammoth.mid" loop="-1" volume="-800" balance
volume="1500">
  </head>
  <body>
    <table border="0" align="center" width="800" cellspacing="0" cellpadding="15">
      <tr>
        <td valign="top"></td>
        <td>
          <h2>Песенка мамонтенка.</h2>
          <pre>
По синему морю, к зелёной земле
Плыву я на белом своём корабле.
На белом своём корабле,
На белом своём корабле.
```

```
... ..  
<b>Музыка: В. Шаинский </b>  
<b>Слова: Д. Непомнящий</b>  
    </pre>  
    </td>  
  </tr>  
</table>  
</body>  
</html>
```

Вот собственно и всё.. а теперь внимательно читаем полезные советы..

Полезные советы:

- Использование фоновой музыки, это не есть хорошо по нескольким причинам:

Во-первых, мало кому из пользователей нравится, когда против его воли из динамиков начинает доноситься то, что он не заказывал! К тому же лазая в Интернете, у него может играть собственная музыка открытая с помощью какого ни будь плеера, может быть ему не нравится именно Ваше музыкальное сопровождение и он предпочитает классику вместо тяжёлого рока, может быть вокруг все спят да и просто он хочет поработать в тишине..

Во-вторых, пользователю не предоставляется возможности управлять данной музыкой, проще говоря, нет кнопок "Стоп" "Пауза" "Плей" и т.д.

В-третьих, музыкальные записи в форматах **WAV, AU** и **MIDI**, как правило, низкого качества, кому нравится слушать некачественную музыку?

В-четвёртых, несмотря на то, что указанные выше форматы, как правило, сравнительно небольшого объёма памяти - это всё же трафик, а любой лишний трафик это плохо.

Ну и в-пятых, как уже говорилось ранее тег **<bgsound>** это придумка для браузера Internet Explorer так что в большинстве других браузеров музыка играть не будет! Кроме того, так как данный тег не входит в спецификацию HTML, то документ с этим тегом будет не валидным.

- Если уж Вам действительно так необходима музыка на сайте, то внедряйте её с помощью тегов **<embed>** и/или **<object>**.. о данных тегах я постараюсь Вам рассказать в пока что ещё ненаписанной главе "объекты".

Глава 14

Текст - логика и физика..

В этой главе мы снова будем работать с текстом, и в отличие от материала, изложенного во второй главе: "Редактируем текст" посмотрим на некоторые вещи под другим углом, а так же поговорим о ранее не вошедших в этот учебник тегах, с помощью которых так же можно форматировать текст на страницах сайта.

Дело в том что все теги предназначенные для работы с текстом можно разделить на две группы: "теги физического форматирования текста" и "теги логического форматирования текста"

Теперь по порядку..

Теги физического форматирования текста.

О тегах физического форматирования текста речь в этом учебнике шла ранее, поэтому повторятся, не стану, а просто перечислю их:

****, **<i>**, **<u>**, **<strike>**, **<s>**, **<tt>**, **<small>**, **<big>**, **<sup>**, **<sub>**, **<p>**, ****, **<pre>**...

- всё это теги физического форматирования текста.

Почему их так называли? Да все просто.. потому что данные теги способны тем или иным способом физически воздействовать на текст.. делать его жирным, курсивом, маленьким, большим.. и т. д. Заключив текст, в какой либо тег из вышеперечисленных, мы получаем ожидаемый результат, который описан в спецификации HTML и в различных браузерах отображается практически одинаково.

На всякий случай вновь покажу старый пример:

```
<html>
<head>
  <title>Стили текста</title>
</head>
<body>
  <big><b><u>Научная статья.</u></b></big>
  <br>
  <br>
  Если разбавить дистиллированную воду Н<sub>2</sub>О сорока процентами
  этилового спирта С<sub>2</sub>Н<sub>5</sub>ОН то получится жидкость в
  40<sup>о</sup> более известную широкой публике под названием <i>- водка.</i>
  <br>
  Впервые данную пропорцию <s>придложил</s> предложил <tt>Дмитрий Иванович
  МЕНДЕЛЕЕВ.</tt>
  <br>
  <br>
  <br>
  <small>Распитие спиртных напитков вредит вашему здоровью.</small>
</body>
</html>
```

Основная задача тегов физического форматирования текста это выполнение сугубо декоративных функций.

Однако если Вы изучали **справочник HTML:** (<http://www.webremeslo.ru/spravka/spravka.html>), то наверняка обратили внимание что выше перечисленные теги это далеко не все инструменты для работы с текстом! А если Вы изучали справочник скрупулезно и пристально, то наверное заметили что некоторые теги выполняют одни и те же действия над текстом..

Например, чем отличается тег **** от тега **<i>**? или какая разница между **** и ****? ведь результат и там и там одинаков!

Так да не так.. отличия есть.. и они достаточно существенные!!..

Логическая разметка текста.

Прежде чем рассказывать о новых тегах расскажу, о том, что следует подразумевать под фразой: "логическая разметка документа".

Под логической разметкой документа, прежде всего, принято понимать разметку текста тегами направленную на удобство работы поисковых систем и других программ, которые работают с данным документом, а так же придавать тексту в документе некий смысл. Так вот программам абсолютно по барабану как выглядит Ваш документ, красивый он или нет, какой стиль шрифта и текста используется и.т.д. программы в первую очередь пытаются найти логику на Вашей страничке, что бы в дальнейшем использовать её по своему усмотрению, например для более релевантной поисковой выдачи или для автоматического составления содержания Вашего документа...

Теперь давайте опять вернемся к вопросу выше: "Чем отличается тег **** от тега **<i>**?" - казалось бы какая разница ведь текст и там и там будет отображаться курсивом .. всё дело в том что тег **<i>** - (курсив) - является тегом физического форматирования текста и предназначен в первую очередь для людей, а тег **** (особенно важный текст) - является тегом логического формирования текста и предназначен для машин! А то, что текст, взятый в тег ****, обычно в браузерах отображается курсивом - это всего лишь, скажем так "мода" и придумка браузеров.

Спецификация HTML не говорит о том, что теги логического форматирования текста должны определённым образом отображаться браузерами, что вот, например тот же тег **** должен быть отображен именно курсивом и не как иначе, а говорит лишь о том, что такой текст должен отличаться от основного и особым образом выделяться в документе..

Так какие теги лучше использовать? Ну раз уж мы уже заочно познакомились с тегами **** и **** покажу пример, комментарии будут ниже..

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
  <head>
    <title>Как создать сайт</title>
  </head>
  <body>
    <p><strong>Как создать сайт?</strong></p>
    <p>Для того, что бы <em>самостоятельно создать сайт</em> для начала в блокноте
нужно написать следующие теги:</p>
    <b>&lt;html&gt;</b> - <i>начало документа</i><br>
    <b>&lt;head&gt;</b> - <i>начало головы</i><br>
    <b>&lt;/head&gt;</b> - <i>заккрытие головы</i><br>
    <b>&lt;body&gt;</b> - <i>начало тела</i><br>
    <b>&lt;/body&gt;</b> - <i>заккрытие тела</i><br>
    <b>&lt;/html&gt;</b> - <i>конец документа</i><br>
  </body>
</html>
```

Теперь объясню, почему на страницах данного сайта, как и в примере выше, в некоторых случаях используются теги **** и ****, а в других случаях **** и **<i>**.

Всё дело в том, что мне как создателю данного сайта очень хочется, чтобы люди набрав в том же гугле фразу: "**Как создать сайт?**" или "**Самостоятельное создание сайта**" видели в первых строчках поисковой выдачи гугла ссылку: **Как создать сайт?:** "**(<http://webremeslo.ru>)**" и переходили по ней на главную страницу моего ресурса :) . Так вот пометив нужные мне куски текста с помощью тегов **** и ****, я тем самым даю понять гуглу что этот текст важный и что на него следует обратить особенное внимание! - конечно, это не залог успеха, тем более для столь популярных (высокочастотных) пользовательских запросов, но при прочих равных условиях, это большой плюс в глазах поисковых систем.

И противоположенный случай..

Как человек старающийся сделать сайт удобным для чтения и восприятия другими людьми, я выделяю некоторые участки текста курсивом (юмор, примечания и прочую отсебятину) и жирным шрифтом (изучаемые теги в примерах), но я не хочу, чтобы поисковые системы обращали сильное внимание на фразы типа: "начало головы" "закрытие тела" - так как сайт написан для начинающих веб-ремесленников, а не для начинающих патологоанатомов, поэтому размечаю такой текст тегами **** и **<i>** - они дают нужный результат, собственно полужирный шрифт и курсив, но не являются "красной тряпкой" для программ.

Теги логического форматирования текста.

Разобравшись с сутью и предназначением логической разметки текста настало время познакомить Вас с тегами которые способны разбавить сухой текст неким смыслом.

Буду перечислять..

Тег **<abbr>**

Выделяет в тексте аббревиатуру. Браузерами обычно подчеркивается пунктирной линией.

<abbr>CSS</abbr>

Практически ко всем тегам применим атрибут **title** - всплывающая подсказка, так вот если Вы в тексте выделяете некую аббревиатуру, старайтесь давать к ней расшифровку, используя данный атрибут.

<abbr title="Cascading Style Sheets">CSS</abbr> позволит Вам без труда изменить стиль любого тега логического форматирования текста!

Тег **<acronym>**

Данный тег выделяет в тексте акроним. Акроним это почти то же самое что и аббревиатура, только образованная из начальных букв, слов или словосочетаний, произносимая как единое слово, а не побуквенно.

Вступай в **<acronym title="коммунистический союз молодёжи">комсомол</acronym>!!**

Так же как и с аббревиатурой для расшифровки текста желательно использовать атрибут **title**.

Тег **<address>**

Указывает автора документа и его адрес. Обычно отображается курсивом. Предназначен для поисковых систем для сбора информации об авторе его адресе и прочей информации владельца/цев сайта.

<address> Вася Пупкин г. Урюпинск Макаaronная фабрика 2010г. **</address>**

Тег <cite>

Выделяет в тексте цитату или сноску на другой документ. Обычно браузеры отображают её курсивом.

Как сказал Остап Бендер **<cite>**Лед тронулся! господа присяжные заседатели! **</cite>**

Тег <code>

Отмечает текст как некий программный код. Обычно отображается моноширинным шрифтом.

<code>function() **</code>** - так обозначается функция в PHP

Тег

Выделяет удалённый текст в новой версии документа. Выделенный текст станет перечёркнутым.

Напоминает теги физического форматирования **<strike>** и сокращённый **<s>** - перечёркнутый текст.

Использование тега ****, более предпочтительно, так как помимо того, что он разбавляет текст логикой еще и заменяет собой теги **<strike>** и **<s>**, которые помечены спецификацией 4.01 как нежелательные.

****Старая цена 1000р. **** - Новая 999 р.!!!

Тег <dfn>

Выделяет текст как определение. Как правило, когда в тексте встречается новый термин, авторы выделяют его курсивом и дают его определение, собственно для этого и нужен тег **<dfn>**

<dfn>Акроним **</dfn>** - аббревиатура, образованная из начальных букв, слов или словосочетаний, произносимая как единое слово, а не побуквенно.

Тег <ins>

Выделяет новый текст в новой версии документа. Выделенный текст в большинстве браузеров станет подчёркнутым.

Противоположен по значению тегу **** с ним же в паре обычно он и используется.

****Старая цена 1000р. **** - **<ins>**Новая 999 р. **</ins>**!!!

Тег

Выделяет особенно важный фрагмент текста. Обычно отображается курсивом.

****Как создать сайт? **** - узнай на Вебремесло.ру!!

Тег <kbd>

От английского keyboard - клавиатура. Указывает текст вводимый с клавиатуры, или для названия клавиш. Обычно отображается моноширинным шрифтом.

Нажмите `<kbd>Ctrl + Z</kbd>` для того чтобы отменить последнее действие.

Тег `<q>`

Выделяет в тексте цитату. В отличие от тега `<cite>` цитата обозначенная тегом `<q>` автоматически берётся браузерами в кавычки.

Как сказал Остап Бендер `<q>Лед тронулся! господа присяжные заседатели!</q>`

Кстати, есть еще и тег физического форматирования текста `<blockquote>` данный тег тоже помечает текст как цитату и при этом выделяет его отступами и переносами строк от остального текста.

Пример:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
  <head>
    <title>Цитаты</title>
  </head>
  <body>
    <p>Эта цитата создана с помощью тега <strong>логического</strong>
форматирования текста <code><b>&lt;q>&gt;</b></code>: <q>Лед тронулся! господа
присяжные заседатели..</q> - браузеры автоматически берут такую цитату в кавычки.
Кстати выходя за рамки темы, видом кавычек легко управлять с помощью <abbr
title="Cascading Style Sheets">CSS</abbr> псевдокласса <code>lang</code> и его
значения <code>quotes</code> - кавычки.</p>
    <p>Эта цитата создана с помощью тега <strong>логического</strong>
форматирования текста <code><b>&lt;cite>&gt;</b></code>: <cite>Лед тронулся!
господа присяжные заседатели.. <a href="#">Остап Бендер. 12 стульев.</a></cite> -
обычно браузерами она отображается курсивом. Второе призвание такого вида цитаты
это выполнение роли сноски на другой документ.</p>
    <p>А эта цитата создана с помощью тега <strong>физического</strong>
форматирования текста <code><b>&lt;blockquote>&gt;</b></code>: <blockquote>Лед
тронулся! господа присяжные заседатели..</blockquote> - в основном тексте она
выделяется отступами и переносами строк, как правило, такой цитатой помечают
длинные куски текста взятые из других источников.</p>
  </body>
</html>
```

Такие вот различия..

Тег `<samp>`

Обозначает текст, который выводится на экран в результате работы какой-либо программы. Обычно браузерами отображается моноширинным шрифтом.

В процессе решения задачи 2+2 программа выдаст ответ: `<samp>4</samp>`

Тег ``

Выделяет особенно важный фрагмент текста. Обычно отображается полужирным.

Тег <var>

Выделяет текст, как переменную в работе какой либо программы. Обычно браузерами отображается курсивом.

Данная функция использует переменную <var>\$count</var>

Ну вот собственно и все теги созданные для логической разметки текста. Примеры показывать не стану, думаю и так всё понятно, а вот несколько полезных советов по разметке текста в документе дам.

Полезные советы:

- Стиль всех тегов как логического, так и физического форматирования текста, великолепно настраивается с помощью средств CSS! Мой Вам совет **учите CSS:** (<http://www.webremeslo.ru/css/glava0.html>), если до этого момента ещё незнакомы с ним. При использовании свойств CSS в работе с текстом Вы забудете про проблему кроссбраузерности т. е. Ваш текст будет отображаться во всех браузерах именно так как нужно Вам, а не так как вздумается браузеру. Кроме того CSS открывает куда более широкие возможности связанные как со стилем текста, так и с дизайном сайта в целом.

- Старайтесь правильно размечать текст на страницах Вашего сайта.. а именно:

Для начала, разбейте весь Ваш текст на параграфы с помощью тега <p> - это облегчит чтение текста как для людей так и для программ, кроме того, позволит Вам в дальнейшем, легко управлять текстом с помощью CSS. Вообще старайтесь не оставлять "голые" слова и предложения, не взятые в какой либо тег форматирования текста.

Соблюдайте значимость заголовков <h1> - <h6> т. е. допустим заголовок страницы берите в тег <h1> подзаголовки в тег <h2> еще менее значимые подзаголовочки в тег <h3> и т. д. В идеале на странице должен быть всего один заголовок <h1>, потом текст разбивается несколькими <h2> и далее по значимости заголовков.

Используйте теги логической разметки лишь там где они действительно необходимы т. е. не стоит например выделять аббревиатуру с помощью цитаты и наоборот.. на то она и логика что бы всё было логично!!

Используйте и в тех случаях, если Вы хотите акцентировать внимание на тексте не только людей, но поисковых систем, в иных ситуациях используйте и <i>.

- Когда выучите CSS забудьте и выкиньте из головы тег и все его атрибуты!! Им уже давно никто не пользуется, ну за редким исключением конечно..