ТЕМА: Основы HTML

[1. Основные понятия html](#в1)

[2. Структура html документа](#в2)

[3. Форматирование текста](#в3)

[4. Работа с формами](#в4)

[5. Фреймы в html](#в5)

[6. Музыка в html](#в6)

1. Основные понятия HTML.

**HTML (HyperText Markup Language)** - язык разметки гипертекста. Отнести HTML к языкам программирования можно лишь с большой натяжкой, это все-таки язык РАЗМЕТКИ. Из названия становится понятным его предназначение - указывать браузеру, как разместить элементы на странице при ее отображении на экране монитора.

Что такое html элемент? Да все, что мы размещаем: текст, рисунки, анимации и т.д. С помощью HTML мы можем указать браузеру где именно их разместить; какой текст сделать большим и жирным, а какой - маленьким курсивом; где начать новую строку, а где сделать переход на другую страницу. И многое другое.

Например,

<b>Что такое html?</b> - этот текст будет отображаться жирным;

<i>Этот текст будет выделен курсивом</i>

В данном случае, символы - делают текст жирным, а символы - курсивом. Такие символы составляют основу html, они называются ТЕГАМИ. Между < и > находится имя тега и его параметры (если они есть). В основном все теги парные, т.е. один открывающий <>, а другой - закрывающий . Такая пара тегов называется контейнером. А все, что находится между ними - содержимым контейнера. Действия тегов распространяются только на их содержимое. Бывают также теги одиночные, т.е. не требующие закрытия. Такие теги просто совершают определенное действие, например, перевод строки или вставку рисунка. Теги могут иметь параметры (или атрибуты),

например,

<body bgcolor="red">Здесьсодержимое</body>

Здесь содержимое В данном случае тег <body>имеет параметр bgcolor, значение которого равно "red". Переводя на человеческий язык - сделать фон документа красного цвета. По сути, HTML - это набор тегов и их параметров.

2. Структура html документа

**Структура html**-документа состоит из трех пар тегов:

<html>

<head>

Заголовок документа

</head>

<body>

Тело документа

</body>

</html>

Теги *<html></html>* являются контейнером для всех остальных, т.е в них помещаются все остальные. Таким образом, ваш документ должен начинаться с тега *<html>*, а заканчиваться тегом *</html>*.  
  
Сам документ условно разделен на две части - заголовок документа (теги *<head></head>*) и тело документа (теги *<body></body>*).

### Заголовок документа - тег HEAD и его элементы

Заголовок документа содержит служебную информацию и не влияет на внешний вид документа. Его задачей является предоставление браузеру пользователя или серверу информации о том, как отобразить ваш документ.

В заголовке могут быть использованы такие теги как: title, base, link, meta, script.  
  
**Title**  
  
Единственным обязательным элементом заголовка документа являются теги *<title></title>*. Они необходимы, чтобы дать документу название, оно отражается в заголовке окна браузера. Например, если написать следующий код:

<html>

<head>

<title>Заголовок документа</title>

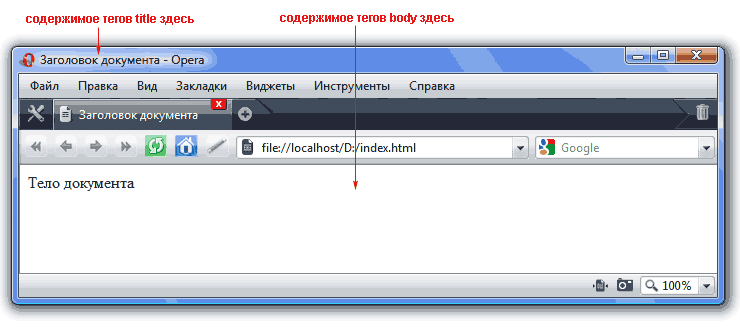
</head>

<body>

Тело документа

</body>

</html>

В окне браузера он будет выглядеть так:  
  
  
Теоретически название документа может иметь неограниченную длину, на практике рекомендуется ограничиться 60 символами.  
  
Не давайте своим документам безликие названия, типа "Первая страница", во-первых название документа должно характеризовать его содержимое, а во-вторых, содержимое тегов *<title></title>* играет не последнюю роль при оптимизации и раскрутке сайта.  
  
**Base**  
Одиночный тег *<base>* служит для указания полного URL-адреса документа. Зачем это нужно? Представьте, что блуждая по интернету, вы сохранили какую-нибудь html-страницу себе на компьютер, с тем, чтобы просмотреть ее позже. Все картинки на этой страницы превратятся в красные крестики. Но если вы не отключены от сети, а на странице присутствует тег *<base>*, то браузер будет знать, где искать необходимый файл, найдет его и загрузит картинки.  
  
У этого тега один атрибут *href*, значением которого является адрес страницы. Примеркода:

<html>

<head>

<title>Структура html</title>

<base href="http://www.my\_site.ru/">

</head>

<body>

Тело документа

</body>

</html>

**Link**  
Одиночный тег *<link>* необходим для подключения внешних файлов. Например, если вы будете использовать каскадную таблицу стилей, то ее удобнее хранить в отдельном файле и подключать этот файл ко всем страницам сайта.   
  
У тега *<link>* несколько атрибутов:  
  
*href* - указывает URL-адрес подключаемого файла.  
  
*rel* - указывает на тип отношения данного документа к внешнему (например: rel="stylesheet" указывает, что внешний файл определяет стиль текущего документа).  
  
*type* - указывает тип и параметры присоединенной таблицы стилей.  
  
Примеркода:

<html>

<head>

<title>Структура html</title>

<base href="http://www.my\_site.ru/">

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

</head>

<body>

Тело документа

</body>

</html>

**Meta**  
Информация в этом теге не имеет никакого отношения к HTML, однако ее использование очень удобно для решения ряда задач:  
  
- указание кодировки страницы, например, для русского текста в кодировке Windows:

<meta http-equiv="Content-Type"

content="text/html;charset=windows-1251">

- указание ключевых слов страницы (используется при оптимизации страниц):

<meta http-equiv="KEYWORDS"

content="тег, структура html, заголовок страницы">

- указание краткого описания страницы:

<meta http-equiv="DESCRIPTION"

content="Описание структура html и элементов заголовка">

Возможны и другие варианты. Рассмотрим параметры тега *<meta>*:   
  
*http-equiv* - определяет свойство тега (тип контента, ключевые слова, описание и т.д.)  
  
*name* - используется для дополнительного описания тега, если отсутствует, то считаеся эквивалентным параметру *http-equiv*.  
  
*content* - значение параметра *http-equiv*.  
  
Примeркода:

<html>

<head>

<title>Структура html</title>

<base href="http://www.my\_site.ru/">

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

<meta http-equiv="Content-Type"

content="text/html;charset=windows-1251">

<meta http-equiv="KEYWORDS"

content="тег, html, заголовок страницы">

<meta http-equiv="DESCRIPTION"

content="Описание элементов заголовка">

</head>

<body>

Тело документа

</body>

</html>

\* Пишите теги в одну строчку, здесь они разбиты на две по причине ограничения ширины страницы. \*  
  
**Script**  
Теги *<script></script>*используются для подключения внешних файлов скриптов. Это позволяет оптимизировать код страниц. Например, если вы используете функции java script для большинства своих страниц, то поместив эти функции на отдельную страницу - function.js, с помощью тега *<script>* можно указать путь к этой страницы.  
  
Это повышает читабельность кода и ускоряет загрузку страниц. У этого тега несколько параметров:  
  
*language* - указывает язык написания скрипта, в спецификации HTML 4.0 данный параметр не рекомендуется к употреблению. Вместо него следует указывать параметр *type*.  
  
*type* - указывает тип MIME для языка.  
  
*src* - указывает путь к внешнему файлу со скриптами.  
  
Примeр кода с относительным адресом скрипта:

<html>

<head>

<title>Структураhtml</title>

<base href="http://www.my\_site.ru/">

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

<meta http-equiv="Content-Type"

content="text/html;charset=windows-1251">

<meta http-equiv="KEYWORDS"

content="тег, html, заголовок страницы">

<meta http-equiv="DESCRIPTION"

content="Описание элементов заголовка">

<script type="text/javascript" src="function.js">

</script>

</head>

<body>

Телодокумента

</body>

</html>

### Тело документа - тег BODY

Все, что отображается на web-странице, находится в тегах *<body></body>*. Это текст, картинки и исполняющиеся скрипты, а также теги для оформления всего этого.  
  
Обязательных параметров у тега *<body>*нет, да и применение необязательных параметров тоже не приветствуется. Тем не менее, большинство параметров до сих пор поддерживается разными браузерами. Рассмотрим те, которые пока поддерживаются всеми браузерами:  
  
*alink* - устанавливает цвет активной ссылки. Текущий цвет ссылки меняется на активный при нажатии на нее.   
  
*vlink* - устанавливает цвет посещенной ссылки, т.е. той, по которой уже щелкали.   
  
*background* - указывает на изображение, которое будет использоваться в качестве фонового рисунка. Этот рисунок заполняет собой все видимое пространство окна. Если рисунок меньше окна браузера, то он повторяется, образуя мозаику из одинаковых картинок. На стыке этих картинок возникают видимые переходы. Поэтому к подбору фоновых рисунков следует подходить с большим вниманием.  
  
*bgcolor* - указывает фоновый цвет документа.  
  
*leftmargin* - определяет отступ от левого края окна браузера до контента страницы.  
  
*rightmargin* - определяет отступ от правого края окна браузера до контента страницы.  
  
*topmargin* - определяет отступ от верхнего края окна браузера до контента страницы.  
  
*bottommargin* - определяет отступ от нижнего края окна браузера до контента страницы.  
  
*text* - устанавливает цвет текста для всего документа.  
  
Примeр кода:

<html>

<head>

<title>Тег body в html</title>

<base href="http://www.my\_site.ru/">

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

<meta http-equiv="Content-Type"

content="text/html;charset=windows-1251">

<meta http-equiv="KEYWORDS"

content="тег, html, заголовок страницы">

<meta http-equiv="DESCRIPTION"

content="Описание элементов заголовка">

<script type="text/javascript" src="function.js">

</script>

</head>

<body bgcolor="khaki" leftmargin="100" topmargin="50"

rightmargin="50" bottommargin="50" text="gray"

alink="red" vlink="green">

Простотекст

<br>

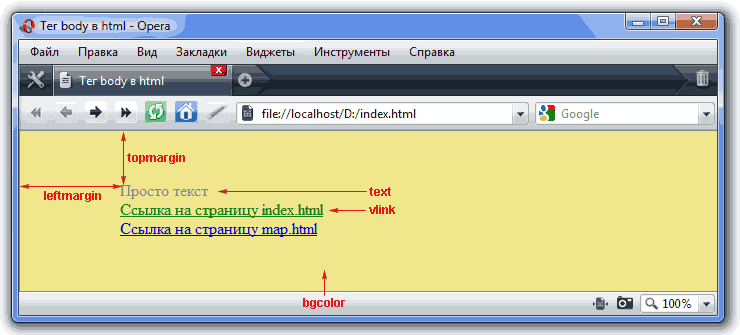
<a href="index.html">Ссылканастраницу index.html</a>

<br>

<a href="map.html">Ссылканастраницу map.html</a>

</body>

</html>

В окне браузера он будет выглядеть так:  


На этом второй закончен, мы рассмотрели основную структуру html-документа.  
  
Запомните, любой ваш html-документ должен содержать следующие теги и именно в том порядке, как они указаны:

<html>

<head>

<title></title>

</head>

<body></body>

</html>

Все остальные теги рассмотренные в этом е на этом этапе вам не нужны, будете добавлять их позже по мере необходимости.  
  
Все элементы, которые мы будем рассматривать на следующих ах, будут помещаться внутрь тегов *<body></body>*и их порядок уже не будет иметь принципиального значения.

**3.Форматирование текста**

Для форматирования текста существует много тегов. Одни их них используются часто (и их вы быстро запомните), другие - редко (их и не надо запоминать).  
  
Здесь мы рассмотрим те, которые используются часто.  
  
Теги, делающие текст заголовками  
  
*<h1></h1>*  
*<h2></h2>*  
*<h3></h3>*  
*<h4></h4>*  
*<h5></h5>*  
*<h6></h6>*  
Эти теги выделяют текст в виде заголовков. Т.е. каждый заголовок начинается с новой строки, выделен полужирным шрифтом и имеет свой размер (заголовок первого уровня самый большой, шестого - самый маленький).  
  
Эти теги могут использоваться с параметром горизонтального выравнивания *align*. Возможные значения этого параметра:

* *left* - слева,
* *right* - справа,
* *center* - по центру,
* *jastify* - по ширине.

Пример кода:

# Это заголовок первого уровня

## Это заголовок второго уровня

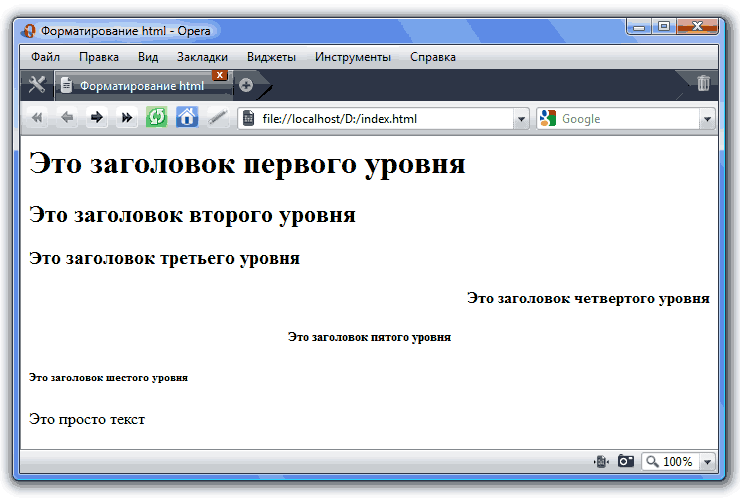
### Это заголовок третьего уровня

#### заголовок четвертого уровня

##### Это заголовок пятого уровня

Это заголовок шестого уровня

Это просто текст

В окне браузера это будет выглядеть так:  
  


Теги разделения на абзацы и переноса строки  
  
Тег *<br>* - тег принудительного перевода строки. Текст, после этого тега начинается с новой строки.  
Теги *<p></p>* разделяют текст на абзацы. Перед началом каждого абзаца следует поместить тег *<p>*, закрывающий тег не обязателен. В отличии от тега *<br>* абзацы отделяются друг от друга пустой строкой.  
  
У тега *<p>* есть параметр *align*, который задает способ выравнивания текста внутри параграфа. Возможные значения этого параметра:

* *left* - слева,
* *right* - справа,
* *center* - по центру,
* *jastify* - по ширине.

Пример кода:

Это абзац, он отделен от всего текста

пустыми строками сверху и снизу и выровнен

по левому краю.

Это абзац, он отделен от всего текста

пустыми строками сверху и снизу и выровнен

по правому краю.

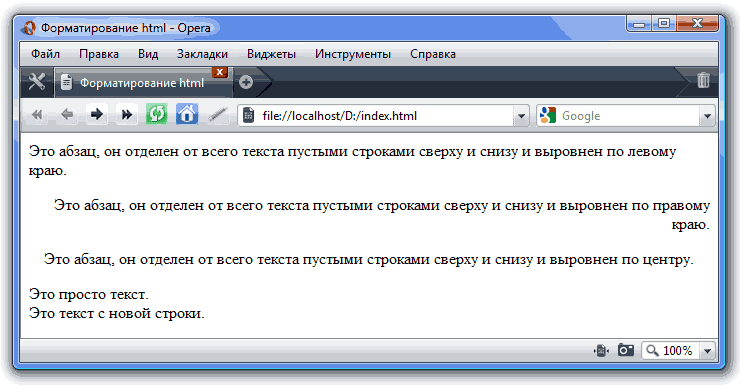
Это абзац, он отделен от всего текста

пустыми строками сверху и снизу и выровнен

по центру.

Это просто текст.

Это текст с новой строки.

В окне браузера это будет выглядеть так:  


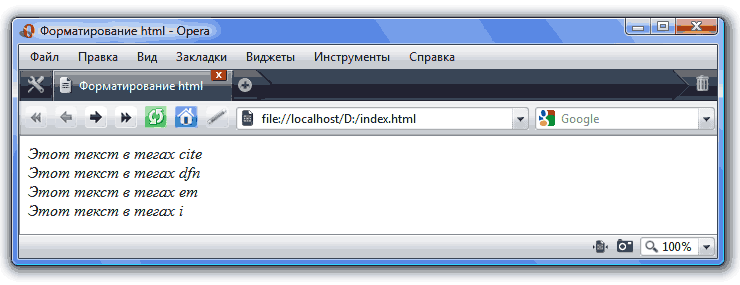
Теги, выделяющие текст курсивом  
*<cite></cite>*  
*<dfn></dfn>*  
*<em></em>*  
*<i></i>*  
Эти теги выделяют текст курсивом, но делают они это по разным причинам.   
Теги *<cite></cite>* используются для логического выделения названий книг, статей и цитат.  
Теги *<dfn></dfn>* используются для выделения определений.  
Тегами *<em></em>* и *<i></i>* выделяют важные фрагменты текста. Последний не рекомендуется к употреблению.  
  
Пример кода:

Этот текст в тегах cite

Этот текст в тегах dfn

Этот текст в тегах em

*Этот текст в тегах i*

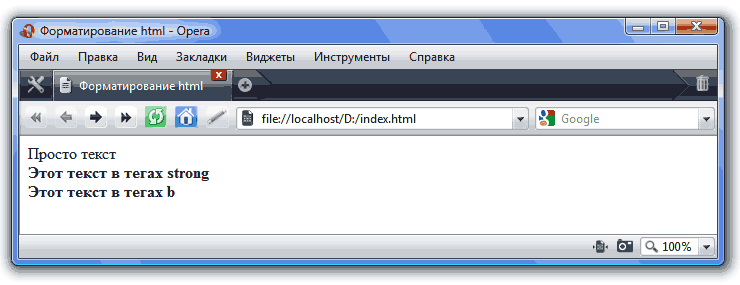
В окне браузера это будет выглядеть так:  


Теги, выделяющие текст полужирным шрифтом  
*<strong></strong>*  
*<b></b>*  
Оба используются для выделения важных фрагментов текста, но предпочтительнее использовать первый.  
Пример кода:

Просто текст

**Этот текст в тегах strong**

**Этот текст в тегах b**

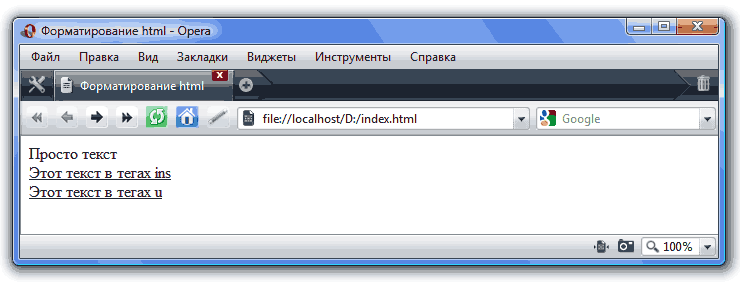
В окне браузера это будет выглядеть так:  


Теги, выделяющие текст подчеркиванием  
  
*<ins></ins>*  
*<u></u>*  
Оба используются для подчеркивания важных фрагментов текста, но предпочтительнее использовать первый.  
Пример кода:

Просто текст

Этот текст в тегах ins

Этот текст в тегах u

В окне браузера это будет выглядеть так:  
  


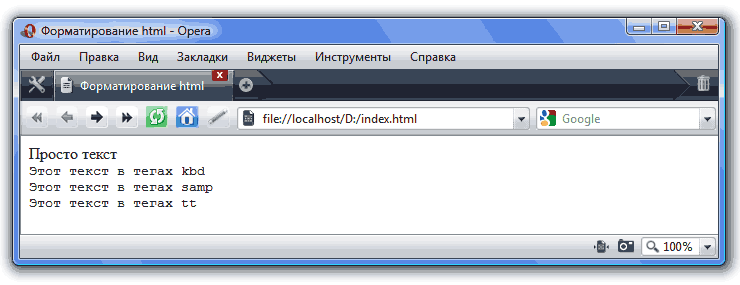
Теги, выводящие текст моноширинным шрифтом  
*<kbd></kbd>*  
*<samp></samp>*  
*<tt></tt>*  
Выводят текст моноширинным шрифтом, но предпочтительнее использовать первый.  
Пример кода:

Просто текст

Этот текст в тегах kbd

Этот текст в тегах samp

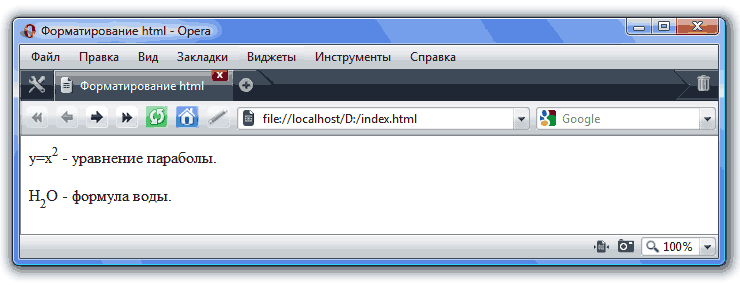
Этот текст в тегах tt

В окне браузера это будет выглядеть так:  


Теги, выводящие текст в верхнем и нижнем индексах  
Теги *<sub></sub>* выводят текст ниже уровня строки шрифтом меньшего размера.   
Теги *<sup></sup>* выводят текст выше уровня строки шрифтом меньшего размера.  
Удобны для вывода математических и химических формул.  
Пример кода:

y=x2 - уравнение параболы.

H2O - формула воды.

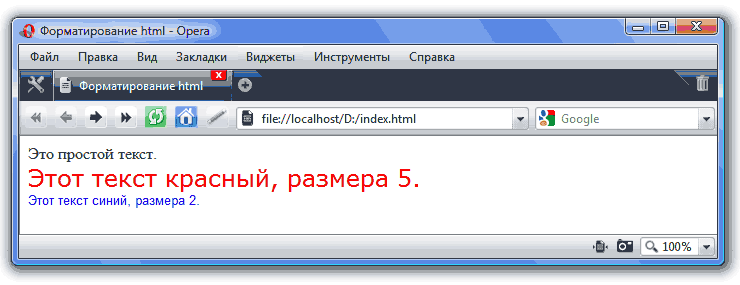
В окне браузера это будет выглядеть так:  
  


Тег font и его параметры  
Теги *<font></font>* указывают параметры шрифта текста:  
*face* - название шрифта. Названий шрифтов можно привести несколько, через запятую. В этом случае, если первый указанный шрифт не будет найден (вы же не знаете, какие шрифты установлены на компьютере пользователя), браузер станет использовать следующий по списку.  
  
*size* - размер шрифта в условных единицах от 1 до 7. По умолчанию размер шрифта равен 3.   
  
*color* - цвет текста (по умолчанию - черный).   
  
Существуют два способа задания цвета: по имени и указанием шестнадцатеричного кода цвета.  
  
С именными цветами (их 156) все просто, смотрим [в соответствующую таблицу](http://www.site-do.ru/graphic/color.php), выбираем понравившийся цвет и пишем его имя в значение параметра (например, color="blue").  
  
Но гораздо больший выбор предоставляет второй способ. Здесь мы можем выбирать из миллиона цветов, указав его шестнадцатеричный код. Этот код представляет собой 6 цифр и начинается с символа "#". Не будем вдаваться в подробности, как формируется код цвета, укажем лишь на то, что получить его можно, например, в программе Photoshop. О том, как это делать читайте на странице [Цвета для web](http://www.site-do.ru/graphic/color_for_web.php).  
  
Пример кода:

Это простой текст.

Этот текст красный, размера 5.

Этот текст синий, размера 2.

В окне браузера это будет выглядеть так:  


Тег center  
Теги *<center></center>* предназначен для выравнивания всех элементов внутри него по центру окна браузера.  
Пример кода:

Это простой текст

Это выровненный по центру текст.

Совместное использование тегов  
Мы рассмотрели основные теги форматирования текста, но что если вам необходимо сделать какое-нибудь слово в тексте жирным красным курсивом? Для этого вам придется применить теги strong, font и em вместе. Здесь главное соблюдать порядок вложенности тегов. Тег, который открыт первым, должен быть закрыт последним.  
Понятнее будет на примере:  
Выделим слово "текст" красным цветом:

Текст

Теперь добавим теги курсива (открывающий - слева, закрывающий - справа):

Текст

А теперь - теги полужирного начертания:

Текст

Каждый раз все предыдущие теги мы помещаем в новые. Это и называется порядком вложенности. Вы можете использовать разные теги для оформления текста, главное не нарушать порядок вложенности.

На этом третий урок закончен, мы рассмотрели основные способы оформления текста, на следующем уроке мы рассмотрим менее применимые теги форматирования.

**4. Работа с формами.**

Формы дают возможность пользователям вводить информацию. Вам наверно, не раз встречались всевозможные тесты, опросы, голосования. Для того, чтобы все это сделать на своих web-страницах и нужны формы.  
  
Здесь следует заметить, что html-формы сами по себе только позволяют вводить информацию, а вот обрабатывать ее HTML не умеет (это все-таки язык разметки, а не программирования). Для обработки информации используются такие языки, как javascript, php и другие.  
  
Но всему свое время, пока мы научимся добавлять html-формы на свои страницы.  
Итак, в html форма задается тегами *<form></form>*.   
У тега *<form>* есть несколько параметров:

* *name* - имя формы. Необходимо, если на странице несколько форм
* *action* - определяет URL-адрес, по которому будет отправлена информация введеная пользователем
* *method* - определяет способ отправки информации
* *target* - указывает имя окна, в котором будут отображаться результаты обработки отправленной формы

Все остальные элементы формы располагаются внутри формы. К ним относятся:  
[текстовое поле](http://www.site-do.ru/html/html11.php#1)

[текстовое поле для ввода пароля](http://www.site-do.ru/html/html11.php#2)

[флажки](http://www.site-do.ru/html/html11.php#3)

[переключатели](http://www.site-do.ru/html/html11.php#4)

[кнопки](http://www.site-do.ru/html/html11.php#5)

[поле для файлов](http://www.site-do.ru/html/html11.php#6)

[многострочное текстовое поле](http://www.site-do.ru/html/html11.php#7)

[раскрывающиеся списки](http://www.site-do.ru/html/html11.php#8)

[надписи](http://www.site-do.ru/html/html11.php#9)

***Текстовое поле***  
  
Простое однострочное текстовое поле, в которое можно вводить и редактировать текст. Задаетсятегом*<input>*

<form name="forma1">

<input type="text" name="text1" size="20"

maxlength="50" value="Введитетекст">

</form>

Результат:

Начало формы

Конец формы

*Параметры:*

* *name - имя элемента,*
* *type - тип элемента (в данном случае - text),*
* *size - размер текстового поля в символах, которые одновременно будут видны, при вводе большего количества символов, они будут прокручиваться,*
* *maxlength - максимальное количество символов, которое можно ввести в поле, если опустить этот параметр, то число символов будет неограниченным,*
* *value - текст, который будет отображаться (его можно стереть), при отсутствии этого параметра поле будет пустым.*

*Возможны еще два параметра:*

* *disabled - блокирует поле от любых изменений,*
* *readonly - делает поле доступным только для чтения.*

*Пример:*

*<form name="forma1">*

*<input type="text" name="text1" size="20"*

*maxlength="50" value="неактивноеполе" disabled>*

*<input type="text" name="text1" size="20"*

*maxlength="50" value="только для чтения" readonly>*

*</form>*

Результат:

Начало формы

Конец формы

***Текстовое поле для ввода пароля***  
*Это такое же текстовое поле, как и предыдущий элемент. Разница только в том, что вводимый текст не отображается, вместо него появляются специальные символы, например звездочки. Чаще всего используется при вводе паролей. Все параметры такие же, как у простого текстового поля, кроме параметра type="password".*  
Пример:

<form name="forma1">

Введитепароль:<br>

<input type="password" name="text1" size="20"

maxlength="50">

</form>

*Результат:*

*Начало формы*

*Введите пароль:*

*Конец формы*

*Попробуйте ввести что-нибудь в этом поле.*  
***Флажки***  
*Вы, конечно, встречали подобный элемент:*

*Начало формы*

*Какими языками вы владеете:  
английский немецкий испанский французский*

*Конец формы*

*Он задается все тем же тегом <input>, причем один тег задает один флажок. Нужно четыре флажка, придется четыре раза писать input.*  
Пример:

<formname="forma1">

Какими языками вы владеете:<br>

<input type="checkbox" name="lan1" value="english"

checked>английский

<input type="checkbox" name="lan2" value="german">

немецкий

<input type="checkbox" name="lan3" value="spanish">

испанский

<input type="checkbox" name="lan4" value="french">

французский

</form>

*Рассмотрим его параметры:*

* *type - тип элемента (в данном случае - checkbox),*
* *name - имя элемента, указывает программе обработчику формы, какой пункт выбрал пользователь,*
* *value - значение элемента, указывает программе обработчику формы значение пункта, который выбрал пользователь. В нашем примере выбран пункт английский, следовательно, программа-обработчик получит: lan1="english",*
* *checked - им обычно помечают наиболее вероятные для выбора пункты, пользователь щелчком мыши может выбрать другие пункты.*

***Переключатели***  
*В отличии от флажков, можно выбрать только один пункт. В связи с этим значения параметра name должны быть одинаковы для всех элементов группы. Параметр type="radio", все остальные такие же, как у флажков.*  
Пример:

<form name="forma1">

Укажитевашпол:<br>

<input type="radio" name="sex" value="man"

checked>мужской

<input type="radio" name="sex" value="woman">

женский

</form>

*Результат:*

*Начало формы*

*Укажите ваш пол:  
мужской женский*Конец формы

***Кнопки***  
  
*Существует четыре вида кнопок:*

* *submit - кнопка отправки содержимого формы web-серверу. Ее параметры:*
  + *type="submit" - тип кнопки,*
  + *name - имя кнопки,*
  + *value - надпись на кнопке.*
* *image - графическая кнопка отправки содержимого формы web-серверу. Для ее использования необходимо подготовить картинку кнопки, а потом использовать ее в виде кнопки. Ее параметры:*
  + *type="image" - тип графической кнопки,*
  + *name - имя кнопки,*
  + *src - адрес картинки для кнопки.*
* *reset - кнопка, позволяющая восстановить все значения по умолчанию в форме. Ее параметры:*
  + *type="reset" - тип кнопки очищения,*
  + *name - имя кнопки,*
  + *value - надпись на кнопке.*
* *button - произвольная кнопка, ее действия назначаются вами, т.е. сама она делать ничего не умеет. Ее параметры:*
  + *type="button" - тип произвольной кнопки,*
  + *name - имя кнопки,*
  + *value - надпись на кнопке.*
  + *onclick - указывает, что делать при щелчке по кнопке. Вообще, у этого типа кнопок есть и другие события (например, двойной щелчок), но здесь мы не будем их рассматривать.*

*Если на форме несколько кнопок, то они должны иметь разные названия.*  
Примеркода:

<form name="forma1">

<input type="submit" name="submit" value="Отправить">

<input type="image" name="but\_img" src="but.gif">

<input type="reset" name="reset" value="Очистить">

<input type="button" name="button" value="Отправить">

</form>

Результат:

Начало формы

Конец формы

**Кнопки можно задавать и по другому, при помощи тегов *<button></button>*.** Возможности у таких кнопок несколько шире, они могут иметь содержимое в виде текста или картинки. *Этот тег имеет следующие параметры:*

* *type - тип кнопки, может принимать значения:* 
  + *reset - кнопка очистки формы,*
  + *submit - кнопка отправки данных,*
  + *button - кнопка произвольного действия.*
* *name - имя кнопки,*
* *value - надпись на кнопке.*

Пример кода:

<formname="forma1">

<button name="submit" type="submit">

<imgsrc="mail.gif" align="absmiddle">

<font size="4">Отправить</font>

</button>

</form>

Результат:

Начало формы

http://www.site-do.ru/images/mail.gif Отправить

Конец формы

Начало формы

Конец формы

***Поле для файлов***

*Поле для ввода имени файла, сопровождаеется кнопкой Browse (Обзор), при щелчке по которой открывается окно просмотра дерева папок компьютера, где можно выбрать нужный файл. Выбранный файл присоединяется к содержимому формы при отправке на сервер.*  
  
Пример:

<form name="forma1">

<input type="file" name="load" size="50">

</form>

Результат: Начало формы

Конец формы

***Многострочное текстовое поле***  
  
*Для объемных текстов, например для почтовых сообщений, удобно использовать именно этот элемент. Он создается тегами <textarea></textarea> и имеет следующие параметры:*

* *name - имя поля,*
* *cols - ширина поля в символах,*
* *rows - количество строк текста, видимых на экране,*
* *wrap - способ переноса слов:*
  + *off - переноса не происходит,*
  + *virtual - перенос отображается, но на сервер поступает неделимая строка,*
  + *physical - перенос и на экране и при поступлении на сервер.*
* *disabled - неактивное поле,*
* *readonly - разрешено только чтение.*

Пример:

<formname="forma1">

<textarea cols="20" rows="3" wrap="off"></textarea><br>

<textarea cols="35" rows="5" wrap="virtual"></textarea><br>

<textarea cols="50" rows="7" wrap="physical"></textarea>

</form>

Результат:

Начало формы

Конец формы

*Попробуйте ввести текст и посмотрите на разницу в значениях параметра wrap.*  
  
***Раскрывающиеся списки***  
*Списки бывают с возможностью выбора одного элемента и с множественным выбором. Задются и те, и другие с помощью тегов <select></select>, внутри которых располагаются элементы значений, заданных тегом <option>. Рассмотрим параметры этих тегов:*

* *<select>:*
  + *name - имя списка. Каждый выбранный элемент списка при передаче на сервер будет иметь вид: name.value, где значение (value) берется из тега option.*
  + *size - определяет количество видимых элементов в списке: 1 - простой раскрывающийся список, больше 1 - список с полосой прокрутки.*
  + *multiple - разрешает выбор нескольких элементов списка.*
* *<option>:*
  + *selected - им помечают наиболее вероятный для выбора элемент списка, если список со множественным выбором, то можно пометить несколько пунктов.*
  + *value - значение, которое будет отправлено серверу, если пункт выбран.*

Пример:

<formname="forma1">

Какой язык вы хотите изучать:

<select name="language" size="1">

<option selected value="html">html

<option value="php">php

<option value="java">java

</select><br><br>

Какое время вы готовы на это потратить:<br>

<select name="time" size="3">

<option selected value="1">1 месяц

<option value="2">2 месяца

<option value="3">3 месяца

</select><br><br>

Какие дни недели для занятий вас устроят:<br>

(выбирайте с нажатой клавишей ctrl)<br>

<select name="day" size="7" multiple>

<option selected value="mon">понедельник

<option value="tue">вторник

<option value="wen">среда

<option selected value="thu">четверг

<option value="fri">пятница

<option value="sat">суббота

<option value="san">воскресенье

</select>

</form>

*Результат:*

*Начало формы*

*Какой язык вы хотите изучать:   
  
Какое время вы готовы на это потратить:  
  
Какие дни недели для занятий вас устроят:  
(выбирайте с нажатой клавишей ctrl)*

Конец формы

*Существуют еще теги <optgroup></optgroup>, позволяющие группировать элементы списка по каким-либо признакам. Например, мы хотим задать каталог сайтов в виде списка, тогда удобнее разбить его на группы по интересам:*

Начало формы

*Для этого нам и нужны теги <optgroup></optgroup> с одним параметром label, который и задает название группе элементов.*  
  
Примеркода:

<form name="forma1">

Каталогсайтов:<br>

<select name="catalog" size="9">

<optgrouplabel="Компьютеры">

<option value="1">интернет</option>

<option value="2">мобильники</option>

<optionvalue="3">hardware</option>

</optgroup>

<optgrouplabel="Работа">

<option value="4">вакансии</option>

<option value="5">трудоустройство</option>

<optionvalue="6">резюме</option>

</optgroup>

<optgroup label="Дом">

<option value="7">здоровье</option>

<option value="8">красота</option>

<option value="9">дети</option>

</optgroup>

</select>

</form>

Обратите внимание, в данном случае необходимо указывать закрывающие теги *</option>*.  
  
***Надписи***  
*Все элементы формы можно связать с их надписями при помощи элемента <label> и его параметра for, значением которого является имя элемента, с которым связываем надпись. Например:*

<form name="forma1">

<label for="load">Выбирайтефайл: </label>

<input type="file" name="load" size="30">

</form>

*Результат:*

*Начало формы*

*Выбирайте файл:* Конец формы

**5.Фреймы в html**

*Пример фреймовой структуры*  
  
Окно браузера может быть разделено на фреймы, т.е. на области расположенные рядом друг с другом. В каждую из этих областей можно загружать своиhtml-страницы.

Пример;

<html>

<head>

<title>Фреймыв html</title>

</head>

<frameset rows="30%, 10%, 60%" >

<frame src="top.html">

<frame src="menu.html">

<frame src="content.html" name="content">

</frameset>

</html>

используются теги *<frameset></frameset>*, которые и указывают браузеру разбить окно на несколько областей - фреймов.   
  
Сколько будет таких областей и как они будут расположены зависит от двух параметров тега *<frameset></frameset>*:

* *rows* - указывает разбить окно на горизонтальные области. Причем, сколько значений у этого параметра, столько и областей. В нашем примере *rows="30%, 10%, 60%"* , т.е. три горизонтальные области: ширина первой - 30% от ширины экрана, ширина второй - 10%, а третьей - 60%.
* *cols* - указывает разбить окно на вертикальные области.

Внутри тегов *<frameset></frameset>* располагаются одиночные теги *<frame>*, причем их должно быть столько, сколько задано областей. Без параметров эти теги бесполезны.  
  
Параметры тега *<frame>*.

* *src* - самый главный параметр, он определяет адрес той html-страницы или картинки, которая будет загружаться во фрейм.
* *name* - имя фрейма, которое будет использоваться для ссылки к данному фрейму.

*marginwidth* - определяет отступ во фрейме от левого края до содержимого.

*marginheight* - определяет отступ во фрейме от верхнего края до содержимого.

*scrolling* - если содержимое фрейма не помещается в окно, то будут появляться полосы прокрутки, иногда это нарушает дизайн. Этот параметр помогает управлять отображением полос прокрутки. Может принимать значения: yes, no, auto.

* *noresize* - если установить курсор мыши на рамки фрейма, то можно выполнить его перемещение. Для предотвращения этой возможности и используется этот параметр.

*Задача открыть ссылку вфрейме*

*У тега <a> имеется параметр target - указывает где открывать указанную в нем ссылку.*

Например:

<a href="other.html" target="content">меню</a>

*iframe*

Пример:

<iframesrc="banner.html" width="468" height="60" align="left">

**6.Фоновая музыка в html**

Фоновая музыка

На web-страницу можно интегрировать фоновую музыку. Когда пользователь будет попадать на эту страницу, будет проигрываться мелодия, конечно, если у пользователя есть колонки или наушники.

Для этого нуженаудифайл в одном из форматов: WAV, AU, MIDI.

Теперь в код любой страницы в теги bodyможно добавить следующий тег:

*<bgsoundsrc="music\_fon.mid" loop="1"></bgsound>*

Теперь при просмотре этой страницы всегда будет присутствовать фоновая музыка. Рассмотрим параметры тега <bgsound>:

src - адрес звукового файла

loop - число повторов мелодии (если равно -1, то повторяется бесконечно)

balance - стереобаланс (значения от -10000 до 10000)

volume - громкость, максимальное значение = 0 (возможные значения от -10000 до 0)

Подключаемые модули

*Подключаемые модули позволяют управлять проигрыванием звуковых файлов.*

*Здесь тоже не все просто. Когда мы встраиваем в html-страницу какой-либо объект, браузер пытается отобразить его самостоятельно. Если это простой объект (текст или рисунок), то браузер это сделает без посторонней помощи. Если объект незнакомый, то браузер попробует прибегнуть к помощи внешней программы или встроенного в браузер программного модуля (плагина), которые и отобразят этот объект. Если же нужного приложения браузер не найдет, то появится сообщение о том, где его можно скачать и как установить (в большинстве случаев, но не всегда).*

*А по статистике лишь 5% людей станут устанавливать себе дополнительные модули, остальные 95% просто уйдут со страницы.*

Для примера всего вышеизложенного создадим html-страницу со следующим кодом:

<embed src="muzic.mid" height="60" width=144>

*Здесь тег <embed> указывает браузеру вставить на web-страницу модуль управления звуковым файлом, а его параметры задают адрес музыкального файла, а также ширину и высоту консоли органов управления.*

**Встраиваем flash-анимацию**  
Рассмотрим встраивание flash-анимации в html-страницу. Для этого нам понадобится flash-player*(скачать можно по адресу* [*www.macromedia.com*](http://www.macromedia.com)*)* и сама flash-анимация (файл с расширением .swf). Сам файл можно и не скачивать, а лишь узнать ссылку на него. Например, мне очень понравился какая-то анимация:

Эта анимация находится по адресу http://dump.nnov.ru/flash/blackcat.swf. Для того, чтобы вставить ее в страницу добавим следующий код:

<object width="480" height="200" align="center">

<param name="movie"

value="http://dump.nnov.ru/flash/blackcat.swf">

<param name="quality" value="high">

<param name="bgcolor" value="#FFFFFF">

<embed src="http://dump.nnov.ru/flash/blackcat.swf"

quality="high"

bgcolor="#FFFFFF"

width="480"

height="200"

align="center"

type="application/x-shockwave-flash"

pluginspage="http://www.macromedia.com/go

/getflashplayer">

</object>

Теги *<param>* позволяют задать объекту начальные значения.