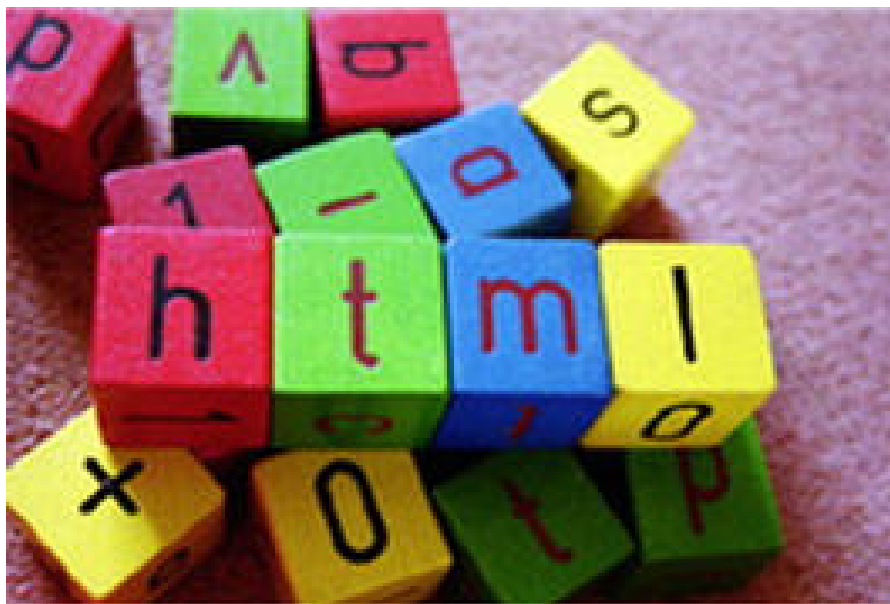


Михаил Русаков

Учебник по Основам HTML



Сайт: <http://MyRusakov.ru>

Оглавление

Об авторе -----	3
Введение -----	5
Создание сайта на HTML -----	6
Ссылки и изображения в HTML -----	9
HTML-теги для текста -----	11
Списки HTML -----	14
Выравнивание элементов в HTML -----	16
Создание форм в HTML -----	18
Создание таблиц в HTML -----	23
Заключение -----	31
Бонус -----	31

Об авторе.



Здравствуйте!

Мне **ОЧЕНЬ** приятно, что Вы решили познакомиться со мной!

Меня зовут Русаков Михаил Юрьевич, и я являюсь создателем и главным администратором сайта **MyRusakov.ru**, который полностью посвящён теме [Как создать свой сайт](#).

Немного о себе. Где я живу, трудно сказать. Родной город - Обнинск, впрочем, на выходных там и сейчас живу. По будням живу в Москве, так как учусь там.

Сам я перешёл на 4-й курс **Московского Энергетического Института (Технического Университета)**. Учусь на факультете **Автоматики и Вычислительной Техники**. Учусь на бюджетной основе, причём вполне успешно. Одновременно получаю второе высшее образование в **Linguae-Centrum при МЭИ (ТУ)** по специальности "**Переводчик в сфере профессиональной коммуникации**". Это уже платное образование. Изучаю, если можно так выразиться, английский язык. А вообще немного знаю ещё и французский язык (учил его в школе).

Программированием увлёкся ещё в **10-ом классе**. Начинал с **Turbo Pascal**, который пошёл у меня настолько успешно, что окружающие были очень удивлены, и я понял, что программирование - это моё призвание!

В конце 11-го класса занялся языком **Java**. Вначале было тяжело (уж, очень я привык к процедурному стилю **Turbo Pascal**, а **Java** чистейшей воды Объектно-Ориентированный язык программирования), но через пару месяцев всё пошло, как по маслу.

На 1-ом курсе в институте у нас был **Turbo Pascal**, где я выполнил все задачи за семестр за 7 часов, и потому получил зачёт и экзамен автоматом уже через пару недель после начала учёбы. Лишний раз понял, что программирование - это то, что получается у меня лучше, чем у других!

Весь 1-й курс, да и половину 2-го занимался **Java**. Одновременно, в этот промежуток познакомился с такими языками, как **C++**, **ActionScript** (язык сценариев **Flash**).

Потом увлёкся созданием сайтов. Изучил **HTML**, **CSS**, **JavaScript**, **PHP**, **SQL** (пользовался **ПО MySQL**).

Летом 2009-го года занимался разработкой корпоративной сети (по просьбе брата), благодаря чему познакомился с **XML**, **XML Schema**, **SOAP**, **WSDL**.

На 3-ем курсе начал заниматься программирование микроконтроллеров на **Assembler**.

Вот и вся информация обо мне.

Полгода назад я решил поделиться своим багажом знаний, который я получал почти **5** лет. И сейчас я вовсю занимаюсь созданием своего **видеокурса**, посвящённого **созданию и раскрутке сайта**. В него войдут **все необходимые знания** для создания **99.9%** сайтов: **HTML, CSS, JavaScript, PHP, MySQL, XML**. Обязательно рассмотрим **CMS Joomla!** Также на этот диск попадёт **вся необходимая информация по раскрутке сайта!** Более того, я планирую (и уже начал) записать создание этого сайта, а потом ещё и его раскрутку. Таким образом, Вы сможете увидеть **в прямом эфире создание и раскрутку сайта** <http://myrusakov.ru>, на котором Вы сейчас и находитесь.

Планирую выпустить свой диск примерно в **сентябре-октябре 2010** года.

Ещё раз огромное спасибо, что решили познакомиться со мной!

С уважением, Михаил Русаков!

P.S. Все свои вопросы (в том числе и по программированию) слать на myrusakov@gmail.com. Либо пишите в **ICQ 600038357** (при авторизации напишите, что пришли с <http://myrusakov.ru>).

С удовольствием, отвечу на любые Ваши вопросы!

Введение.

Первое, что мне хотелось бы сделать, так это поздравить Вас с приобретением этой бесплатной, но очень полезной книги. В данной книге рассказывается о языке гипертекстовой разметки - **HTML** (*Hyper Text Markup Language*), который является необычайно простым и в то же время необычайно важным для любого человека, который имеет хоть какое-то отношение к созданию сайтов.

Я не буду Вас долго мучить ожиданиями, давайте кратко обрисую ситуацию. Во-первых, данная книга предназначена для новичков. Во-вторых, данная книга даёт, пожалуй, все знания **HTML**, которые необходимы при создании сайтов. В-третьих, материал изложен очень кратко и лаконично. Всё разложено по полочкам. Поэтому освоить данный материал сможет даже любой школьник.

Также к учебнику приложен справочник по HTML-тегам! Если справочник отсутствует, то скачать его можно здесь: <http://myrusakov.ru/skachat-uchebnik-osnovy-html.html>.

Ладно, закончу хвалить свою книжку, и пожелаю Вам успехов в её изучении!

Если у Вас возникают какие-либо вопросы при изучении, или Вы хотите оставить отзыв о моей книге, то смело пишите мне на **Е-mail**: myrusakov@gmail.com. На все письма я с удовольствием отвечу!

Напоследок, об **авторских правах** на эту книгу! Авторские права полностью принадлежат её автору – Михаилу Русакову.

Вы **имеете право** распространять данную книгу, но только при следующих условиях:

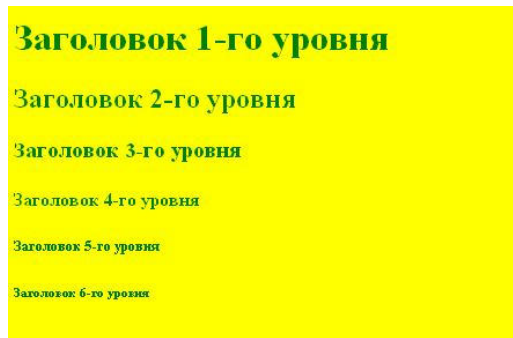
- 1) Вы **НЕ имеете права** продавать эту книгу.
- 2) Вы **НЕ имеете права** каким-либо образом изменять контент настоящего документа – Вы обязаны сохранить данную книгу как есть!

При частичном копировании содержимого данного документа указывайте автора (Михаил Русаков) и сайт <http://myrusakov.ru>.

Уверен, что Вы легко справитесь с данными условиями.

С уважением, *Михаил Русаков!*

Создание сайта на HTML



HTML-исходники к этой главе лежат в папке «**glava1**» (Скачать **HTML**-исходники можно здесь: <http://myrusakov.ru/skachat-uchebnik-osnovy-html.html>).

Здравствуйте, меня зовут Михаил Русаков!

В этой статье я расскажу Вам о создании простейшего сайта. Кто знает, может быть, знания, которые Вы получите в этой статье, впоследствии станут началом Вашего успеха.

Я склонен полагать, что у Вас имеется большое желание научиться создавать свои собственные сайты. Разумеется, крайне глупо пытаться познать эту науку с изучения какого-нибудь **PHP**. Гораздо разумнее начать с чего-нибудь попроще, а именно с **HTML** (*HyperText Markup Language* - язык гипертекстовой разметки) - базового языка, на котором и построен любой, даже самый сложный сайт. Без этого языка не было бы ни одного сайта, более того, **HTML** - простейший язык, который может освоить даже школьник (яркий пример - Ваш покорный слуга, начавший осваивать **HTML** ещё в школе, причём весьма успешно).

Ладно, хватит болтовни - пора приступить к созданию нашего с Вами первого сайта, и этот процесс будет называться: "Создание сайта на **HTML**".

Первое, что нужно решить: где создавать сайт? Разумеется, можно и в обычном блокноте. Но это несерьёзно (мы так в школе писали), поэтому давайте лучше будем использовать что-нибудь "потяжелее". Я остановил свой выбор на редакторе "**Notepad++**" - замечательная программа, которая позволяет писать код на огромном количестве языков, в том плане, что это будет весьма и весьма удобно. Не буду сильно её хвалить, потому что для написания кода для серьёзных языков (**Java**, **C++** и т. д.) она не подходит - много лишних проблем возникнет. Но для таких языков, как **HTML**, **CSS**, **JavaScript** и даже **PHP** - очень хороший выбор.

Скачать **Notepad++** можно, пройдя по этой ссылке: <http://notepad-plus.sourceforge.net/ru/site.htm>

Устанавливаем и запускаем редактор. Сразу сохраняем страницу под именем `index.html`. Теперь пишем (пожалуйста, напишите, а не скопируйте!) в окне редактора:

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

</body>

</html>

Написали? Сохранили? Поздравляю Вас, Вы только что создали самую простую (но очень важную!!!) **HTML**-страницу!

Теперь можно нашу страницу открыть в браузере. Так и поступим.

Что-нибудь видите? Я уверен, что нет. И это неудивительно, ведь это самая элементарная страница. Вы можете посмотреть исходник страниц. Для этого в меню "**Вид**" выберите пункт меню "**Просмотр HTML кода**", "**Исходный код страницы**" или что-то в этом духе, в зависимости от Вашего браузера. Также можно просто нажать **CTRL-U** (в большинстве браузеров откроется исходный код страницы), и Вы увидите Ваш код.

А теперь давайте обсудим, что мы здесь написали.

<html>, **<head>**, **<body>** - это открывающие теги. Иногда их ещё называют дескрипторы, но я привык называть их тегами, поэтому далее так и буду писать.

</html>, **</head>**, **</body>** - это закрывающие теги.

Теги бывают парные и одиночные. Парные теги - это теги, которые нуждаются в закрывающем, то есть все три тега, которые мы использовали являются парными, так как они открываются и потом закрываются.

Одиночные теги - это теги, которые не требуют наличие закрывающего тега.

Теперь что касается назначения данных тегов.

Тег **<html>** означает начало **HTML**-документа.

Тег **<head>** означает начало заголовка **HTML**-документа. Сюда могут входить, например, заголовок страницы, различные мета-теги. Обо всём об этом поговорим чуть ниже.

Тег **</head>** означает конец заголовка **HTML**-документа.

Тег **<body>** означает начало самого документа. То есть всё (текст, картинки, ссылки и т. д.), что Вы видите на страницах в Интернете - всё расположено в теге **<body>**.

Тег **</body>** означает конец содержимого документа.

Тег **</html>** означает конец **HTML**-страницы.

Теперь давайте немного добавим в нашу страницу кода, уж, больно она примитивная у нас пока. Внутри тега **<head>** добавьте такой код:

```
<title>Наша страница</title>
```

```
<meta http-equiv="Content-type" content="text/html; charset=windows-1251">
```

Тег **<title>** определяет заголовок страницы, который будет отображён в заголовке окна браузера.

Тег **</title>** сообщает, что здесь заголовок заканчивается.

Что касается тега **<meta>**, то здесь сначала я должен сообщить об атрибутах тегов. У тегов могут быть различные атрибуты, определяющие, например, вид их содержимого. Атрибуты ставятся внутри открывающего тега и имеют следующий вид:

Название = "значение".

Тег **<meta>** у нас определяет тип содержимого документа. Атрибут **http-equiv** и его значение **Content-type** определяет, что сейчас будет описан тип документа. Атрибут **content** и его значение **text/html; charset = windows-1251** сообщают браузеру, что данный документ является HTML-документом (**text/html**), и кодировка этого документа кириллица (**windows-1251**). Это был пример мета-тегов. Я думаю, что пока с ними хватит.

Теперь займёмся содержимым документа. Надеюсь, что Вы уже поняли, что раз содержимое страницы - значит, будем писать что-то внутри тега **<body>**. Давайте напишем следующий код внутри тега **<body>**:

```
<h1>Заголовок 1-го уровня</h1>
<h2>Заголовок 2-го уровня</h2>
<h3>Заголовок 3-го уровня</h3>
<h4>Заголовок 4-го уровня</h4>
<h5>Заголовок 5-го уровня</h5>
<h6>Заголовок 6-го уровня</h6>
```

Теги **<h1>**, **<h2>**, **<h3>**, **<h4>**, **<h5>**, **<h6>** - выводят текст в браузере. Причём этот текст является заголовком, более того, чем больше цифра (от 1-го до 6-ти), тем менее крупным шрифтом будет выведен соответствующий текст. Таким образом, код Вашей страницы имеет вид:

```
<html>
<head>
<title>Наша страница</title>
<meta http-equiv="Content-type" content="text/html; charset=windows-1251">
</head>
<body>
  <h1>Заголовок 1-го уровня</h1>
  <h2>Заголовок 2-го уровня</h2>
  <h3>Заголовок 3-го уровня</h3>
  <h4>Заголовок 4-го уровня</h4>
  <h5>Заголовок 5-го уровня</h5>
  <h6>Заголовок 6-го уровня</h6>
</body>
</html>
```

Сохраните страницу и посмотрите на неё в браузере.

Также можно добавить немного атрибутов и в тег **<body>**. Давайте добавим атрибут **text** со значением **green** (зелёный). И атрибут **bgcolor** со значением **yellow** (жёлтый).

Таким образом, тег **<body>** выглядит так:

```
<body text = "green" bgcolor = "yellow">
  <h1>Заголовок 1-го уровня</h1>
  <h2>Заголовок 2-го уровня</h2>
  <h3>Заголовок 3-го уровня</h3>
```



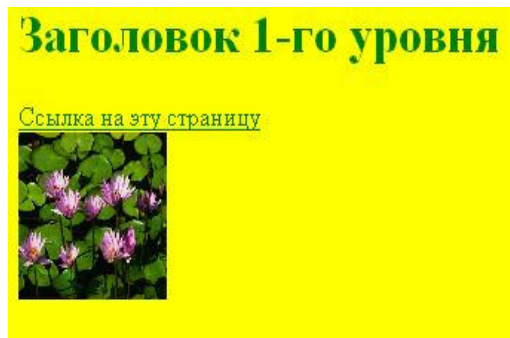
```
<h4>Заголовок 4-го уровня</h4>
<h5>Заголовок 5-го уровня</h5>
<h6>Заголовок 6-го уровня</h6>
</body>
```

Сохраните страницу и посмотрите на неё в браузере. Как видите, фон страницы стал жёлтым, а текст стал зелёным.

Думаю, что на сегодня хватит для Вас материала. Очень полезно будет потренироваться в выводе разного текста, разными цветами.

Увидимся в следующей статье!

Ссылки и изображения в HTML



HTML-исходники к этой главе лежат в папке «glava2» (Скачать HTML-исходники можно здесь: <http://myrusakov.ru/skachat-uchebnik-osnovy-html.html>).

В этой статье я расскажу Вам о добавлении изображений и ссылок на Вашу HTML-страницу.

Я думаю, что Вы ни разу не встречали ни одной страницы в Интернете, на которой не было ни одного изображения или ссылки. Поэтому материал данной статьи весьма важен, в то же время он очень прост, благодаря чему его сможет освоить даже школьник.

Первое, что мы сделаем - это добавим **HTML** **ссылку** на нашу страницу.

Для этого напишем в теге **<body>** такой код:

```
<a target = _self href = "index.html" title = "Ссылка">Ссылка на эту страницу</a>
```

Тег HTML **<a>** означает начало ссылки. У этого тега есть различные атрибуты. Давайте начнём с **"target"**.

Атрибут **"target"** может принимать три значения (их чуть больше, но они слишком редко применяются, поэтому не буду их даже упоминать): **"_self"**, **"_new"**, **"_blank"**. Вообще говоря, результат этих атрибутов у разных браузеров зачастую происходит по-разному, поэтому я скажу, что означает каждое значение атрибута **"target"** лишь в теории, а не на практике.

Значение **"_self"** означает, что страница откроется в этом же окне.

Значение **"_new"** означает, что страница откроется в новом окне (некоторые браузеры открывают не новое окно, а создают новой вкладку в браузере).

Значение **"_blank"** означает, что страница будет открыта в новой вкладке.

Я настоятельно рекомендую Вам испробовать все значения этого атрибута в различных браузерах, чтобы окончательно понять, что каждое из них означает.

Следующий атрибут **"href"** означает путь к странице. Так как нам пока ссылаться не на что, то приходится сослаться на ту же страницу. Что касается формы записи, то здесь бывают абсолютные и относительные пути.

Абсолютный путь задаёт полный путь к файлу, например, так:

"http://mysite.ru/images/image1.gif". А относительный путь указывают путь относительно каталога, в котором находится текущий файл. Например, наш файл **"index.html"** лежит в корне сервера, а мы пишем: **"href = 'images/image1.gif'"**. Это означает то же самое, что и если бы мы написали: **"http://mysite.ru/images/image1.gif"**. Это, что касается абсолютных и относительных путей.

Надеюсь, что с атрибутом **"href"** всё стало понятно. Разумеется, ссылаться можно не только на **HTML**-страницы, но и на любые файлы, будь то картинки, будь то фильмы, будь то музыка, будь то архивы, будь то ещё всё, что угодно.

Последний атрибут - это **"title"**. Здесь всё совсем просто. Этот атрибут задаёт текст, который будет виден при наведении мышки на ссылку.

С атрибутами тега **<a>** покончено. Что касается того, что внутри тега, то здесь только Ваша фантазия. В большинстве случаев - это обычный текст, как, например, у нас. Очень часто делают картинку внутри тега **<a>**, тогда эта ссылка будет в виде картинки.

Тег **** означает конец гиперссылки.

Также можно задавать **цвет ссылок**. Опять же будет работать или нет - зависит от браузера. Но всё-таки, давайте я Вас с этим познакомлю. Добавим в тег **<body>** три атрибута : **"link"**, **"vlink"** и **"alink"**. Каждый из этих трёх атрибутов могут принимать значение какого-либо цвета. Атрибут **"link"** означает цвет непосещённой ссылки. Соответственно, **"vlink"** указывает цвет посещённой ссылки. И, наконец, атрибут **"alink"** указывает цвет ссылки в момент непосредственного нажатия. Как видите, здесь тоже ничего сложного нет.

Теперь добавим изображения, но прежде, чем это делать. Сразу после добавления **HTML-ссылки** ставим тег **"
"**. Этот тег также является одиночным, и он означает переход на новую строку. Напишем в теге **<body>** такой код:

```
<img src = "image1.gif" alt = "Картинка" width = "100" height = "100">
```

Тег **** означает, что мы хотим добавить здесь изображение. Этот тег, как видно, является одиночным, то есть у него нет закрывающего тега. Атрибуты тега также очень и очень простые.

Атрибут **"src"** указывает путь к картинке. Опять же можно указывать, как абсолютный путь, так и относительный путь, в зависимости от ситуации.

Атрибут **"alt"** указывает текст, который описывает картинку. Этот же текст будет показываться в случае, если картинка по каким-либо причинам будет не найдена, либо у

пользователя в браузере отключён показ картинок. Очень желательно этот атрибут ставить, так как это помогает оптимизации сайта (впрочем, это уже раскрутка сайта, а не его создание).

Атрибуты **"width"** и **"height"** указывают ширину и высоту картинки соответственно. Если их не указывать, то картинка будет иметь такой размер, какой имеет исходник.

Советую потренироваться, выводя различные изображения, только предварительно разместите файлы картинок вместе с **"index.html"**. Например, создайте в этом же каталоге папку **"images"** и размещайте туда изображения. А в атрибуте **"src"** пишите значение **"images/Имя файла картинки"**.

Таким образом, код Вашей страницы должен быть примерно таким:

```
<html>
<head>
</head>
<body text = "green" bgcolor = "yellow" link = "green" vlink = "red" alink =
"blue">
    <h1>Заголовок 1-го уровня</h1>
    <a target = "_new" href = "index.html" title = "Ссылка">Ссылка на эту страницу
</a>
    <br>
    <img src = "Водяные лилии.jpg" alt = "Картинка" width = "100" height = "100">
</body>
</html>
```

Вот и вся наука. Как видите, ничего сложного нет, как и всё в базовых знаниях по **HTML** и базовых знаниях по созданию сайта.

До встречи в следующей статье.

HTML теги для текста

Заголовок 1-го уровня

Это курсив Это жирное начертание Подчёркнутый текст Жирный курсивный текст Жирный курсивный текст
Это текст с size = "5" Это увеличенный красный текст
Это текст, в котором написано очень многое

HTML-исходники к этой главе лежат в папке **«glava3»** (Скачать **HTML**-исходники можно здесь: <http://myrusakov.ru/skachat-uchebnik-osnovy-html.html>).

В сегодняшней статье я Вас познакомлю с различными способами форматирования текста.

Совершенно очевидно, что было бы очень скучно, если бы везде был однообразный текст: одного размера, одного шрифта, одного цвета, одного вида. Поэтому среди **HTML** тегов имеются различные теги, позволяющие задать определённый внешний вид текста.

Давайте, для начала разберёмся с начальными данными, если можно так выразиться, а именно со страницей, которая у нас имеется. Пусть она будет такой:

```
<html>
<head>
</head>
<body text = "green" bgcolor = "yellow">
  <h1>Заголовок 1-го уровня</h1>
</body>
</html>
```

Собственно, здесь уже имеются элементы форматирования: тег **<h1>**, который делает текст внутри тега крупным, более того, это идёт, как заголовок, поэтому дальнейшие элементы располагаются уже под ним. Также мы здесь задали **цвет текста (green)** в атрибуте **text** тега **<body>**.

Теперь поговорим о других возможностях по форматированию текста.

Первое, что можно сделать - это узнать, как сделать различный вид текста: *курсив*, подчёркнутый, **жирный** и различные комбинации.

Начнём с курсива. Текст будет курсивным, если он находится внутри тега **<i>**. Поэтому давайте напишем так:

```
<i>Это курсив</i>
```

Если Вы запустите браузер, то Вы сможете увидеть текст в виде курсива.

Также текст может быть жирным. Делается это с помощью тега ****. Например, так:

```
<b>Это жирное начертание</b>
```

Также текст можно сделать подчёркнутым. Для этого необходимо использовать тег **<u>**.

```
<u>Подчёркнутый текст</u>
```

Помимо этого можно сделать текст, например, курсивным и жирным. Делается это следующий образом:

```
<i><b>Жирный курсивный текст</b></i>
```

Также можно записать и так:

```
<b><i>Жирный курсивный текст</i></b>
```

Если Вы посмотрите в браузере, то не увидите никакой разницы. Это и логично, ведь какая разница: сначала сделали текст курсивным, а потом жирным, или, наоборот, сначала жирным, а потом курсивным. Суть, думаю, понятна. Единственное правило, но **ОЧЕНЬ** важное правило - это соблюдать *принцип вложенности*. То есть сначала закрываются все внутренние теги, и только потом закрываются внешние теги. Таким образом, **НЕЛЬЗЯ** писать так:

```
<i><b>Жирный курсивный текст</i></b>
```

Думаю, понятно, что работать это не будет. Ведь мы не закрыли ещё тег ****, а уже закрываем тег **<i>**. Всегда помните о принципе вложенности - это одно из немногих правил HTML, соблюдение которого **обязательно!**

Теперь поговорим о размере шрифта текста. Изменить **размер текста** можно сделать с помощью тега ****, а, точнее, с помощью его атрибута **"size"**, значение которого и означает размер шрифта. Давайте перейдём на следующую строку с помощью тега **
** и напишем такую строчку:

```
<font size = "5">Это текст с size = "5"</font>
```

Обновив страницу в браузере, Вы увидите, что текст стал крупнее, чем раньше. Это и сделал атрибут **"size"** тега ****. Чем больше значение атрибута **"size"**, тем крупнее шрифт, впрочем, думаю, что Вам это очевидно.

Цвет текста можно задавать также с помощью тега ****, так как у этого тега имеется ещё один атрибут - **"color"**. Значение данного атрибута может быть любой цвет. Давайте напишем так:

```
<font size = "5" color = "red">Это увеличенный красный текст</font>
```

Вот это и были основные способы по изменению внешнего вида текста.

Напоследок, напишем строчку, в которой намешано абсолютно всё, что мы сегодня узнали. Сначала, правда, давайте перейдём на следующую строчку, с помощью уже знакомого нам тега **
. Обратите внимание, что необходимо соблюдать принцип вложенности: **пока не закрыт внутренний тег - внешний не закрывается.

```
<font size = "4" color = "navy"><b><i><u>Это текст, в котором намешано очень  
много</u></i></b></font>
```

Таким образом, код нашей страницы стал таким:

```
<html>
<head>
</head>
<body text = "green" bgcolor = "yellow">
  <h1>Заголовок 1-го уровня</h1>
  <i>Это курсив</i>
  <b>Это жирное начертание</b>
  <u>Подчёркнутый текст</u>
  <i><b>Жирный курсивный текст</b></i>
  <b><i>Жирный курсивный текст</i></b>
  <br>
  <font size = "5">Это текст с size = "5"</font>
  <font size = "5" color = "red">Это увеличенный красный текст</font>
  <br>
  <font size = "4" color = "navy"><b><i><u>Это текст, в котором намешано очень
```

```
многое</u></i></b></font>  
</body>  
</html>
```

Вот и вся наука. Как видите, снова ничего сложного не было, и это неудивительно, ведь это основные знания по **HTML**, которыми обладает любой уважающий себя **Web-мастер**. Без этих знаний речи о создании сайтов и идти даже не может.

Всего Вам доброго, увидимся в следующей статье.

Списки HTML

- Первый элемент
 - Второй элемент
-
1. Первый элемент
 2. Второй элемент

HTML-исходники к этой главе лежат в папке «**glava4**» (Скачать **HTML**-исходники можно здесь: <http://myrusakov.ru/skachat-uchebnik-osnovy-html.html>).

В этой статье я Вас познакомлю Вас с **созданием списков в HTML**.

Собственно, как и весь **HTML**, создание **HTML списков** - это дело очень и очень простое.

Давайте с Вами сейчас создадим простую **HTML**-страницу. Постарайтесь это сделать, не смотря вниз статьи. То есть написать основные **HTML**-теги самостоятельно, так как уже пора их запомнить и строчить автоматически.

Теперь поговорим о списках.

Никаких секретов я сейчас не открою, списки - это информация, расположенная в виде:

1. Элемент списка 1.
2. Элемент списка 2.

.....

n. Элемент списка n.

Наличие нумерации зависит от типа списка: нумерованный или нenumерованный список.

Для начала создадим нenumерованный список. Начинается список с тега ****, а заканчивается, разумеется, закрывающим тегом ****. Элементы списка добавляются внутри тега **** с помощью тега ****.

Например, вот такой список:

```
<ul>  
  <li>Первый элемент</li>  
  <li>Второй элемент</li>  
</ul>
```

Тег **** может быть как парным, так и одиночным, то есть следующая запись также является *верной и результат будет тот же*:

```
<ul>
  <li>Первый элемент
  <li>Второй элемент
</ul>
```

Нумерованный список отличается выводом номера элемента справа от самого элемента.

Правила те же самые, что и с нумерованным списком, но только начальный тег - это тег ****.

Закрывающий тег, соответственно, ****. Вот пример нумерованного списка:

```
<ol>
  <li>Первый элемент</li>
  <li>Второй элемент</li>
</ol>
```

Посмотрите и сравните результат в браузере. Напоследок показываю код, который у Вас должен был получиться:

```
<html>
<head>
</head>
<body>
<ul>
  <li>Первый элемент
  <li>Второй элемент
</ul>
<ol>
  <li>Первый элемент
  <li>Второй элемент
</ol>
</body>
</html>
```

Вот и вся наука о **списках в HTML**. Напоследок скажу, что под элементом списка может подразумеваться не только элемент в виде текста, а, вообще говоря, всё, что угодно. Например, вместо текста может быть ссылка или даже изображение. Просто само форматирование будет в виде списка.

В принципе, Вы уже можете создавать сайты на **HTML**, потому что элементарная база по **HTML** у Вас уже имеется, хотя, разумеется, изучать дальше необходимо, просто уже можно потихоньку начинать создавать сайт.

Выравнивание элементов в HTML

Заголовок 1-го уровня, выравненный по центру



Заголовок 1-го уровня, выравненный по правому краю



HTML-исходники к этой главе лежат в папке «**glava5**» (Скачать **HTML**-исходники можно здесь: <http://myrusakov.ru/skachat-uchebnik-osnovy-html.html>).

До сих пор мы с Вами выравнивали элементы только по левому краю. Точнее, мы с Вами вообще этим не занимались, а сам браузер по умолчанию выравнивает элементы по левому краю. Разумеется, было бы слишком скучно выравнивать всё по левому краю. Поэтому существуют различные способы выравнивания по центру и по правому краю.

Выравнивание элементов - это самые основы **HTML**, которые просто необходимо знать при создании интернет сайтов. Первое, что необходимо сделать - это набрать **HTML**-код простейшей страницы.

Ещё когда-то давно появился тег `<center>` я сейчас Вам не советую его применять, ввиду наличия более современных способов, но не упомянуть я о нём не могу. Использовать его очень и очень просто. Всё, что Вам необходимо выравнивать по центру, Вы помещаете внутри этого тега. Вот, например, тут мы выравниваем заголовок 1-го уровня по центру.

```
<center>
  <h1>Заголовок 1-го уровня, выравненный по центру</h1>
</center>
```

Можно добавить картинку, выравненную также по центру, также давайте перейдём на следующую строку с помощью тега `
`:

```
<center>
  <h1>Заголовок 1-го уровня, выравненный по центру</h1>
  <br>
  <img src = "Водяные лилии.jpg" width = "150" height = "150">
</center>
```

Это был тег `<center>`, который уже устарел, вдобавок, вопреки Вашим ожиданиям тегов `<left>` и `<right>` просто не существует. Допустим, по левому краю выравнивается по умолчанию, по центру выравнивается с помощью тега `<center>`, но как же быть с правым?

Чтобы решить эту проблему разработчики придумали универсальный способ выравнивания элементов **HTML**. Способ заключается в использовании так называемых контейнеров, которые создаются с помощью тега **<div>**. То есть всё, что необходимо поместить в определённый контейнер помещается внутри тега **<div>**. А уже у этого тега есть атрибут **"align"**, значение которого и определяет положение данного контейнера. Бывают три значения: **"left"**, **"center"**, **"right"**. По умолчанию, стоит **"left"**, впрочем, думаю, что Вас это не удивляет.

Давайте сейчас напишем тот же **HTML**-код, но с использованием контейнеров, вдобавок, давайте выравниваем не по центру, а по правому краю.

```
<div align = "right">  
  <h1>Заголовок 1-го уровня, выравненный по правому краю</h1>  
  <br>  
  <img src = "Водяные лилии.jpg" width = "150" height = "150">  
</div>
```

Как видите, всё работает. Советую Вам также поменять значения атрибута **"align"**, чтобы посмотреть на другие виды выравнивания контейнеров.

Другой способ выравнивания элементов **HTML** - это таблицы, но эта тема заслуживает отдельного разговора, поэтому поговорим о ней в одной из следующих статей.

А пока Ваша страница должна выглядеть так:

```
<html>  
<head>  
</head>  
<body>  
  <center>  
    <h1>Заголовок 1-го уровня, выравненный по центру</h1>  
    <br>  
    <img src = "Водяные лилии.jpg" width = "150" height = "150">  
  </center>  
  <div align = "right">  
    <h1>Заголовок 1-го уровня, выравненный по правому краю</h1>  
    <br>  
    <img src = "Водяные лилии.jpg" width = "150" height = "150">  
  </div>  
</body>  
</html>
```

Создание форм в HTML

Ваше имя:

Ваш пароль:

Выберите вариант:

Напишите что-нибудь:

Сообщение

Выберите что-нибудь одно: ☐ Вариант 1 ☐ Вариант 2 ☐ Вариант 3

Вы согласны с нашими правилами: ☐

Выберите файл для загрузки:

HTML-исходники к этой главе лежат в папке «**glava6**» (Скачать **HTML**-исходники можно здесь: <http://myrusakov.ru/skachat-uchebnik-osnovy-html.html>).

Я абсолютно уверен, что, бегая по Интернету, Вы ни раз встречали различные формы, например, форму регистрации, форму входа в систему, форму обратной связи и многие-многие другие **HTML** формы.

Собственно, форма состоит из различных элементов ввода: текстовых полей, текстовых областей, радио-кнопок, переключателей, кнопок и так далее.

И в этой статье Вы научитесь создавать абсолютно любые формы в **HTML**.

Сначала создайте простейшую **HTML**-страницу, в которую добавьте контейнер (тег **<div>**), выравнивающий своё содержимое по центру. Надеюсь, что Вы с этим без труда справитесь.

А теперь займёмся формой. Форма начинается с тега **<form>**, соответственно, заканчивается закрывающим тегом **</form>**. У этого тега есть несколько атрибутов, которые очень желательно заполнять. Но для начала давайте создадим простейшую форму с атрибутами тега **<form>**, чтобы Вам было понятнее назначение каждого атрибута:

```
<form name = "form1" action = "script/request.php" method = "post">
</form>
```

Здесь у нас начинается описание формы, у которой есть следующие атрибуты:

1) Атрибут **"name"**. Значение этого атрибута означает имя **HTML** формы. Встаёт вопрос, зачем это надо? Ответ очень простой: если Вы будете использовать не одну форму, а несколько, то чтобы отличить одну форму от другой, необходимо задавать различные имена. А сами имена нужны для того, чтобы обращаться к формам, например, через **JavaScript**. Я Вам открою маленький секрет, который я Вам не советую использовать, но честности ради, скажу о нём. На самом-то деле, формы можно легко различать и без имён, поэтому, вообще говоря, имя формы

совсем необязательно. Но я **НАСТОЯТЕЛЬНО** рекомендую давать формам имена, ведь Вам самим так будет проще понять, за что каждая из форм отвечает.

2) Атрибут **"action"**. Значение этого атрибута отвечает за путь к файлу скрипта, который будет обрабатывать форму. То есть мало ввести данные в форму, их же ведь ещё нужно соответствующим образом обработать, и вот как раз путь к этому *файлу-обработчику* и находится в значении атрибута **"action"**.

3) Атрибут **"method"**. У него может быть одно из двух очень популярных значения: **"post"** и **"get"**. Этот атрибут определяет способ отправки. Не буду вдаваться в подробности, просто скажу, что первый способ является скрытой отправкой данных, а второй открытой. Чтобы стало ещё понятнее, то давайте рассмотрим два адреса перехода:

a) <http://mysite.ru/script/request.php>

б) <http://mysite.ru/script/request.php?a=7&b=Michael>

В первом случае, пользователь не видит, что отправляет (метод **"post"**), а во втором он реально видит имена переменных и их значения (метод **"get"**). Вы, в конце статьи, можете попытаться отправить форму с помощью двух разных методов и убедиться в их различии. Но пока что скажу, что **ГОРАЗДО** чаще используют метод **"post"**, то есть скрытую отправку.

Это было описание самой формы, а теперь же можно заняться добавлением элементов в форму.

Первое, что мы добавим - это текстовое поле. Добавляется текстовое поле с помощью тега **<input>**, а точнее с помощью атрибута этого тега **"type"** со значением **"text"**. Также перед созданием текстового поля рекомендуется написать, что это за поле, например, *"Ваше имя"*. Внутри тега **<form>** напишите такую строчку:

```
Ваше имя: <input type = "text" name = "firstname" value = "Тут Ваше имя">
```

Опять же, давайте разберём атрибуты:

1) Атрибут **"type"** отвечает за тип элемента ввода. В данном случае, мы указали, что это обычное текстовое поле. В следующих элементах мы будем менять значение этого атрибута.

2) Атрибут **"name"** отвечает за имя элемента. Здесь мы указали, что имя данного поля **"firstname"**.

3) Атрибут **"value"** отвечает за значение этого поля по умолчанию.

Как говорится, лучше один раз увидеть, чем сто раз услышать, поэтому лучше Вы наберите (**причём, именно наберите, а не скопируете!**) этот текст в редакторе, а потом посмотрите результат в браузере.

Теперь создадим ещё одно похожее поле, но только не для ввода обычных символов, а для ввода пароля. То есть должно быть всё то же самое, но только текст должен быть скрыт за *звёздочками*. Поэтому, перейдя на следующую строку (тег **
**), напомним такой код:

```
Ваш пароль: <input type = "password" name = "pass" value = "Тут Ваш пароль">
```

Как видите, теперь значением атрибута **"type"** является **"password"**. Остальные атрибуты те же, что и для обычного текстового поля.

Следующий элемент - это выпадающий список. Создаётся он чуть посложнее, так как тут необходимо не только объявить создание выпадающего списка, но ещё и добавить элементы в этот список. Напишем, перейдя на следующую строку (далее, упоминать об этом не буду) следующий **HTML**-код.

Выберите вариант:

```
<select name = "choice">
  <option value = "1">Вариант 1
  <option value = "2">Вариант 2
  <option value = "3">Вариант 3
</select>
```

Тег **<select>** означает начало выпадающего списка. Заканчивается список тегом **</select>**. Атрибут **"name"** ясен, ибо уже объяснял его несколько раз. Элементы списка создаются с помощью тега **<option>**. Значение атрибута **"value"** означает, какое значение будет у переменной **choice** (например, в **JavaScript**), то есть либо **choice = 1**, либо **choice = 2**, либо **choice = 3**. Сразу после окончания описания тега **<option>** ставится то, что увидит пользователь в выпадающем списке. Опять же, лучше посмотрите в браузере, и Вам всё сразу станет понятно.

Теперь добавим текстовую область, используя тег **<textarea>**:

Напишите что-нибудь:

```
<br>
<textarea name = "message" rows = "10" cols = "15">Сообщение</textarea>
```

Здесь мы с Вами создаём текстовую область высотой в 10 строк (значение атрибута **"rows"**) и шириной в 15 символов (значение атрибута **"cols"**).

Внутри этого парного тега указывается текст в текстовой области по умолчанию. Собственно, здесь больше нечего сказать. Двигаемся дальше.

Следующий элемент - это радио-кнопка. Радио-кнопка - это набор таких *"кружочков"*, из которых можно выбрать только один. Создаются радио-кнопки с помощью **HTML**-тега **<input>**, а, точнее, с помощью значения **"radio"** атрибута **"type"**. Напишем такой **HTML**-код:

Выберите что-нибудь одно:

```
<input type = "radio" name = "choiceradio" value = "1">Вариант 1
<input type = "radio" name = "choiceradio" value = "2">Вариант 2
<input type = "radio" name = "choiceradio" value = "3">Вариант 3
```

Тут я остановлюсь на атрибуте **"name"**, потому что помимо задания имени у него есть ещё одна очень важная особенность. Если Вы **НЕ** сделаете одинаковыми имена у этих трёх радио-кнопок, то они станут независимыми, а, следовательно, можно будет выбрать сразу несколько вариантов. Чтобы в этом убедиться, поменяйте имена, а потом попытайтесь выбрать сразу несколько вариантов, и Вы сразу убедитесь в моих словах. Поэтому одна группа радиокнопок должна иметь одно и то же значение атрибута **"name"**. Атрибут **"value"** означает значение

переменной **"choiceradio"** (опять же, например, в **JavaScript**). Сразу после описания тега **<input>** идёт текст, который увидит пользователь рядом с соответствующей радио-кнопкой.

Ещё одним элементом формы являются переключатели (**checkbox**). Создаются они опять же с помощью тега **<input>**. Напишем следующие строчки:

Вы согласны с нашими правилами:

```
<input type = "checkbox" name = "terms" value = "yes">
```

Я думаю, что здесь всё прозрачно, поэтому объяснять не буду. Советую лишь просто посмотреть, как это выглядит, в браузере. И скажу, что если флажок будет стоять, то значение переменной **"terms"** будет **"yes"**, если не будет стоять, то значение этой переменной будет **"",** то есть пустой строкой.

Ещё одним элементом является поле для выбора файла. Наверняка, Вы когда-нибудь загружали файлы на сервер и Вам часто приходилось использовать для этого поле вставки имени файла. Вот такое поле создаётся с помощью уже надоевшего тега **<input>**. Напишем так:

Выберите файл для загрузки:

```
<input type = "file" name = "path">
```

Объяснять здесь особо нечего, единственное, что обязательно посмотрите, как это выглядит, чтобы у Вас не осталось вопросов.

Теперь займёмся двумя последними элементами - это кнопки. Их две: обычная кнопка и кнопка **"submit"**. Начнём с обычной кнопки. Создаётся кнопка с помощью тега **<input>**.

Например, так:

```
<input type = "button" name = "start" value = "Начать">
```

Значение атрибута **"type"** уже **"button"**, что сообщает браузеру, что это кнопка. Атрибут **"name"** - это имя кнопки, а значение атрибута **"value"** отвечает за текст на кнопке.

Обрабатывать эту кнопку мы не будем, так как мы проходим не **JavaScript**, а основы **HTML**, но в будущем обязательно это сделаем.

И последний элемент - это кнопка **"submit"**, при нажатии на которую форма отправляется на обработку к файлу, указанному в атрибуте **"action"** тега **<form>**. Добавляется это кнопка так:

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

Как видите, создать кнопку **"submit"** очень просто: для этого необходимо поставить значение **"submit"** атрибута **"type"** тега **<input>**. А атрибут **"value"** содержит текст, который пользователь увидит на кнопке.

Теперь, напоследок, советую посмотреть разницу между значениями **"post"** и **"get"** атрибута **"method"** тега **<form>**. Разницу можно увидеть в адресной строке при нажатии на кнопку **"submit"**.

После того, как наиграетесь с формой, то результат должен быть таким:

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

```
<body>
<div align = "center">
  <form name = "form1" action = "script/request.php" method = "get">
    Ваше имя: <input type = "text" name = "firstname" value = "Тут Ваше имя">
    <br>
    Ваш пароль: <input type = "password" name = "pass" value = "Тут Ваш
    пароль">
    <br>
    Выберите вариант:
    <select name = "choice">
      <option value = "1">Вариант 1
      <option value = "2">Вариант 2
      <option value = "3">Вариант 3
    </select>
    <br>
    Напишите что-нибудь:
    <br>
    <textarea name = "message" rows = "10" cols = "15">Сообщение</textarea>
    <br>
    Выберите что-нибудь одно:
    <input type = "radio" name = "choiceradio" value = "1">Вариант 1
    <input type = "radio" name = "choiceradio" value = "2">Вариант 2
    <input type = "radio" name = "choiceradio" value = "3">Вариант 3
    <br>
    Вы согласны с нашими правилами:
    <input type = "checkbox" name = "terms" value = "yes">
    <br>
    Выберите файл для загрузки:
    <input type = "file" name = "path">
    <br>
    <input type = "button" name = "start" value = "Начать">
    <br>
    <input type = "submit" value = "Отправить">
  </form>
</div>
</body>
</html>
```

Как видите, опять ничего сложного. Как видите, создать свой сайт - это совсем не сложно. Единственная проблема, как это всё запомнить? Ответ очень прост: практика, практика и ещё раз практика!!!

Всего доброго!

Создание таблиц в HTML

Ваше имя:	Тут Ваше имя
Ваш пароль:
Выберите вариант:	Вариант 1
Напишите что-нибудь:	
Сообщение	
Выберите что-нибудь одно: <input type="radio"/> Вариант 1 <input type="radio"/> Вариант 2 <input type="radio"/> Вариант 3	
Вы согласны с нашими правилами: <input type="checkbox"/>	
Выберите файл для загрузки:	Обзор...
Начать	
Отправить	

(1, 1)		
(2, 1)		(1, 2)
	(2, 2)	

HTML-исходники к этой главе лежат в папке «**glava7**» (Скачать **HTML**-исходники можно здесь: <http://myrusakov.ru/skachat-uchebnik-osnovy-html.html>).

В этой статье Вы познакомитесь с важнейшим моментом при создании своего сайта - **таблицами!**

Я думаю, что Вы обратили внимание на то, что в предыдущих статьях мы не занимались выравниванием, если, конечно, не считать выравнивание по левому краю, по центру и по правому краю браузера. Собственно, этого крайне недостаточно для красивого оформления страницы. Яркий тому пример - наша форма, которую мы создали в предыдущей статье. Как можно видеть, форма **ОЧЕНЬ НЕРОВНАЯ**. Вот, как раз для того, чтобы выравнивать элементы страницы более точно, чем просто по краям браузера, и были придуманы **html таблицы**.

Давайте для начала создадим простую **html таблицу**.

Но прежде, чем это делать, давайте восстановим код, который у нас остался после создания формы, то есть вот этот:

```
<html>
<head>
</head>
<body>
<div align = "center">
    <form name = "form1" action = "script/request.php" method = "get">
```

```
Ваше имя: <input type = "text" name = "firstname" value = "Тут Ваше имя">
<br>
Ваш пароль: <input type = "password" name = "pass" value = "Тут Ваш пароль">
<br>
Выберите вариант:
<select name = "choice">
    <option value = "1">Вариант 1
    <option value = "2">Вариант 2
    <option value = "3">Вариант 3
</select>
<br>
Напишите что-нибудь:
<br>
<textarea name = "message" rows = "10" cols = "15">Сообщение</textarea>
<br>
Выберите что-нибудь одно:
<input type = "radio" name = "choiceradio" value = "1">Вариант 1
<input type = "radio" name = "choiceradio" value = "2">Вариант 2
<input type = "radio" name = "choiceradio" value = "3">Вариант 3
<br>
Вы согласны с нашими правилами:
<input type = "checkbox" name = "terms" value = "yes">
<br>
Выберите файл для загрузки:
<input type = "file" name = "path">
<br>
<input type = "button" name = "start" value = "Начать">
<br>
<input type = "submit" value = "Отправить">
</form>
</div>
</body>
</html>
```

Принцип создания таблиц в html таков: создаётся таблица, потом создаётся строка, потом все столбцы данной строки, потом очередная строка, снова все очередные столбцы данной строки и так далее.

Таблица создаётся с помощью тега **<table>**, а заканчивается тегом **</table>**. Это, думаю, понятно. У таблицы есть строки и столбцы, поэтому их необходимо создать. Теперь согласно принципу создания таблиц, необходимо создать строку. Строка создаётся с помощью тега **<tr>**. Соответственно, сигналом к окончанию строки является закрывающий тег **</tr>**. Внутри тега **<tr>** необходимо создавать столбцы, которые создаются с помощью тега **<td>**. И уже внутри этого тега находятся те элементы, которые должны быть расположены внутри данной ячейки.

После того, как все элементы уложили, то можно закрывать столбец с помощью тега `</td>`.

Дальше открывается новый тег `<td>`. В него снова помещаются элементы, после этого закрывается `</td>`. Это сигнал к концу второго столбца. И так далее, столько столбцов, сколько Вам нужно. В конце закрывается строка тегом `</tr>`. Затем следующая строка и так далее. А заканчивается всё закрывающим тегом `</table>`.

Чтобы стало понятнее, давайте сразу после формы напишем следующие строки:

```
<table>
  <tr>
    <td>(1, 1)</td>
    <td>(1, 2)</td>
  </tr>
  <tr>
    <td>(2, 1)</td>
    <td>(2, 2)</td>
  </tr>
</table>
```

Здесь мы создали таблицу, состоящую из двух строк (два тега `<tr>`) и двух столбцов (по два тега `<td>` в каждом теге `<tr>`). В качестве элемента я решил взять обычный текст, в виде координат ячейки. Соответственно, первая цифра означает номер строки, а вторая - номер столбца. Это был яркий пример таблицы.

Теперь поговорим об атрибутах. Сначала атрибуты тега `<table>`.

1) Атрибут **"border"**, значение которого задаёт толщину рамку таблицы в пикселях. По умолчанию, рамки вообще нет. Давайте поставим значение этого атрибута в **"2"**.

2) Атрибуты **"width"** и **"height"** задают ширину и высоту таблицы соответственно. Размер может быть указан, как в абсолютных единицах (пиксели, **px**), так и в относительных (проценты, **%**). Относительный размер, чем хорош, так это тем, что он всегда подстроится под любое разрешение монитора пользователя и любой браузер. А абсолютные тем хороши, что при любых браузерах и любых разрешениях монитора не будет сюрпризов с дизайном, связанные, например, с растягиванием элементов (если монитор широкоэкранный, к примеру). Какой выбрать: решать Вам, но я, лично, предпочитаю, в основном, относительные размеры, изредка разбавляя абсолютными, в зависимости от ситуации. Давайте поставим значение атрибутов **"width"** и **"height"** по **"30%"**.

Думаю, что атрибутов Вам дал достаточно. Таким образом, тег `<table>` выглядит так:

```
<table border = "2" width = "30%" height = "30%">
```

Посмотрите в браузере, как выглядит теперь наша таблица.

Теперь поговорим об атрибутах тега `<tr>`:

1) Атрибут **"height"**. Заметьте, что у тега `<tr>` нет атрибута **"width"**, впрочем, это логично, ведь тег `<tr>` отвечает за строку, а, следовательно, за высоту. А за ширину отвечают столбцы.

Собственно, даже атрибут **"height"** не особо-то и используется, поэтому можно сказать, что атрибутов у тега **<tr>** и вовсе нету.

Наиболее богатым по количеству атрибутов является тег **<td>**, отвечающий за ячейку таблицы:

1) Атрибут **"width"**. Объяснение то же, что и для атрибута тега **<tr>**. Соответственно, атрибута **"height"** нет.

2) Атрибут **"colspan"**. Значение этого атрибута означает количество столбцов, которое занимает данная ячейка. Поставьте этот атрибут в значение **"2"**, где у нас координаты **"(1, 1)"**. Сразу посмотрите на результат.

3) Атрибут **"rowspan"**. Значение этого атрибута означает количество строк, которое занимает данная ячейка. Поставьте этот атрибут в значение **"2"**, где координаты **"(1, 2)"**. Сразу посмотрите на результат.

4) Атрибут **"align"**. Значение этого атрибута означает выравнивание элемента внутри ячейки по горизонтали. Бывают три значения: **"left"** (по левому краю), **"center"** (по центру), **"right"** (по правому краю). По умолчанию стоит выравнивание по левому краю. Давайте выравниваем координаты **"(1, 1)"** по центру, а координаты **"(2, 1)"** по правому краю. Посмотрите результат.

5) Атрибут **"valign"**. Значение этого атрибута означает выравнивание элемента внутри ячейки по вертикали. Снова имеются только три значения: **"top"** (по верху), **"middle"** (по середине), **"bottom"** (по низу). По умолчанию стоит значение **"middle"**. Давайте поменяем значение этого атрибута у элемента **"(2, 1)"** на **"top"**. А также у элемента **"(2, 2)"** поставим значение **"bottom"**. Сразу посмотрите на результат.

Таким образом, Ваша таблица имеет такой **HTML**-код:

```
<table border = "2" width = "30%" height = "30%">
  <tr>
    <td colspan = "2" align = "center">(1, 1)</td>
    <td rowspan = "2">(1, 2)</td>
  </tr>
  <tr>
    <td align = "right" valign = "top" >(2, 1)</td>
    <td valign = "bottom">(2, 2)</td>
  </tr>
</table>
```

В принципе, с теорией всё. А вот на практике необходимо научиться применять эти знания. И в качестве практики, мы сейчас выравниваем нашу форму. Для начала сотрите все теги **
, так как теперь на новую строку у нас будет переходить тег **<tr>.

Теперь помещаем всё содержимое формы в таблицу, то есть сразу после тега **<form>** пишем тег **<table>**, а сразу перед тегом **</form>** закрываем таблицу с помощью тега **</table>** (не забываем про *принцип вложенности*!). Рекомендую поставить временно рамку у таблицы,

чтобы видеть, как всё выравнивается, и как делится общее пространство между элементами таблицы.

Теперь каждую строку формы помещаем в отдельный тег `<tr>`. И делаем два столбца: первый для описания элемента, а второй для самого элемента формы. И так делаем вплоть до `<textarea>`.

С текстовой область поступаем так: во-первых, пояснение (далее буду называть *меткой*) запишем в одну строку, причём ширину делаем в два столбца. Текстовую область на следующей строке таблицы, и также шириной в два столбца. Более того, выравниваем и метку, и текстовую область по центру.

Далее всё помещаем в одну строку и одну ячейку, но шириной эта ячейка будет в два столбца. Вот и всё. Таким образом, Ваша форма должна иметь такой **HTML**-код:

```
<form name = "form1" action = "script/request.php" method = "get">
  <table border = "1">
    <tr>
      <td>Ваше имя:</td>
      <td>
        <input type = "text" name = "firstname" value = "Тут Ваше имя">
      </td>
    </tr>
    <tr>
      <td>Ваш пароль:</td>
      <td>
        <input type = "password" name = "pass" value = "Тут Ваш пароль">
      </td>
    </tr>
    <tr>
      <td>Выберите вариант:</td>
      <td>
        <select name = "choice">
          <option value = "1">Вариант 1
          <option value = "2">Вариант 2
          <option value = "3">Вариант 3
        </select>
      </td>
    </tr>
    <tr>
      <td align = "center" colspan = "2">Напишите что-нибудь:</td>
    </tr>
    <tr>
      <td colspan = "2" align = "center">
        <textarea name = "message" rows = "10" cols = "15">Сообщение</textarea>
      </td>
    </tr>
  </table>
</form>
```

```

</tr>
<tr>
  <td colspan = "2">
    Выберите что-нибудь одно:
    <input type = "radio" name = "choiceradio" value = "1">Вариант 1
    <input type = "radio" name = "choiceradio" value = "2">Вариант 2
    <input type = "radio" name = "choiceradio" value = "3">Вариант 3
  </td>
</tr>
<tr>
  <td colspan = "2">
    Вы согласны с нашими правилами:
    <input type = "checkbox" name = "terms" value = "yes">
  </td>
</tr>
<tr>
  <td colspan = "2">
    Выберите файл для загрузки:
    <input type = "file" name = "path">
  </td>
</tr>
<tr>
  <td align = "center" colspan = "2">
    <input type = "button" name = "start" value = "Начать">
  </td>
</tr>
<tr>
  <td align = "center" colspan = "2">
    <input type = "submit" value = "Отправить">
  </td>
</tr>
</table>
</form>

```

Таким образом, полный код Вашей страницы выглядит так:

```

<html>
<head>
</head>
<body>
<div align = "center">
<form name = "form1" action = "script/request.php" method = "get">
  <table border = "1">
    <tr>
      <td>Ваше имя:</td>
      <td>

```

```

        <input type = "text" name = "firstname" value = "Тут Ваше имя">
    </td>
</tr>
<tr>
    <td>Ваш пароль:</td>
    <td>
        <input type = "password" name = "pass" value = "Тут Ваш пароль">
    </td>
</tr>
<tr>
    <td>Выберите вариант:</td>
    <td>
        <select name = "choice">
            <option value = "1">Вариант 1
            <option value = "2">Вариант 2
            <option value = "3">Вариант 3
        </select>
    </td>
</tr>
<tr>
    <td align = "center" colspan = "2">Напишите что-нибудь:</td>
</tr>
<tr>
    <td colspan = "2" align = "center">
        <textarea name = "message" rows = "10" cols = "15">Сообщение</textarea>
    </td>
</tr>
<tr>
    <td colspan = "2">
        Выберите что-нибудь одно:
        <input type = "radio" name = "choiceradio" value = "1">Вариант 1
        <input type = "radio" name = "choiceradio" value = "2">Вариант 2
        <input type = "radio" name = "choiceradio" value = "3">Вариант 3
    </td>
</tr>
<tr>
    <td colspan = "2">
        Вы согласны с нашими правилами:
        <input type = "checkbox" name = "terms" value = "yes">
    </td>
</tr>
<tr>
    <td colspan = "2">
        Выберите файл для загрузки:

```

```

        <input type = "file" name = "path">
    </td>
</tr>
<tr>
    <td align = "center" colspan = "2">
        <input type = "button" name = "start" value = "Начать">
    </td>
</tr>
<tr>
    <td align = "center" colspan = "2">
        <input type = "submit" value = "Отправить">
    </td>
</tr>
</table>
</form>
<table border = "2" width = "30%" height = "30%">
    <tr>
        <td colspan = "2" align = "center">(1, 1)</td>
        <td rowspan = "2">(1, 2)</td>
    </tr>
    <tr>
        <td align = "right" valign = "top" >(2, 1)</td>
        <td valign = "bottom">(2, 2)</td>
    </tr>
</table>
</div>
</body>
</html>

```

Вот и всё, что необходимо знать для успешного использования таблиц, а дальше лишь надо набивать руку. Я Вам скажу так: таблица наиболее часто используемая возможность при создании **HTML**-страницы, и без них не обходится почти ни один сайт. Правда, таблицы и наиболее сложная тема в основных знаниях по **HTML**, для усвоения которой необходимо потренироваться, а также избавиться от стереотипа о том, что элементом таблицы может быть только текст (уж, так нас научили в школе, учебниках и вообще в жизни). И когда Вы избавитесь от этого стереотипа и поймёте, что под элементом может подразумеваться не только текст, но и картинки, ссылки, формы, элементы формы, и даже целые таблицы, тогда можно считать, что дело сделано.

С уважением, *Михаил Русаков*.

Заключение.

Вот Вы и закончили изучать язык **HTML**. Я надеюсь, что Вам всё стало ясно и понятно, и Вы теперь можете создавать простые Web-сайты и уже потихоньку начинать зарабатывать. Разумеется, это далеко не весь материал даже по HTML, не говоря уже по теме создания сайтов. Однако, и этого уже достаточно, чтобы начать делать первые шаги.

Для тех же, кто готов продолжать учиться дальше (а впереди ещё очень много, чего предстоит узнать), прошу посетить мой сайт: <http://myrusakov.ru>. Там содержится куча полезного материала по CSS (для создания дизайна страница), по JavaScript (много для чего используется, но чтобы было сейчас понятнее, скажу, что используется для создания динамических страниц – **DHTML**), по PHP (язык, без которого немыслим практически любой серьёзный сайт), по MySQL (необходим для работы с базой данных, без которой также немыслим ни один хоть немного приличный сайт). И многое-многое другое.

Ладно, не буду у Вас более отнимать время на пустую болтовню. Скажу лишь напоследок, что создание сайтов – это, как правило, огромная работа, которую без знаний просто невозможно осилить. И сейчас у Вас имеется самая минимальная крупица знаний, которая является основой при создании любого сайта, а чтобы стать действительно профессионалом, Вам предстоит ещё многое узнать. И следующий шаг – это познакомиться с материалами моего сайта.

Спасибо за внимание!

С уважением, *Михаил Русаков!*

Внимание Бонус!

Всем, кто скачал данную книгу, **я дарю бонус – 50% скидку** на мой видеокурс: «**Создание и раскрутка сайта от А до Я**».

Данный курс выйдет в сентябре 2010-го года, однако, Вы уже сейчас можете записаться и получить скидку в 50%.

Теперь немного о самом видеокурсе.

Видеокурс посвящён созданию и раскрутке сайта. В него войдут **более 50-ти (!!!) часов** авторского обучающего видео по **HTML, CSS, JavaScript, PHP, MySQL, XML**. Обязательно рассмотрим **CMS Joomla**, процесс **размещения сайта в Интернете** и, разумеется, **раскрутку сайта**.

Внимание! На диск войдёт запись создания моего сайта <http://MyRusakov.ru>. Вы в режиме реального времени увидите, как **создавался сайт от начала и до конца**. Затем Вы увидите его размещение в Интернете и потом ещё и его раскрутку. И это всё в режиме реального времени.

Подведём итог, что Вы получите, приобретя данный диск:

1) Научитесь верстать страницы.

- 2) **Научитесь создавать дизайн сайта.**
- 3) Научитесь писать скрипты на языке JavaScript.
- 4) **Научитесь писать скрипты на языке PHP.**
- 5) Научитесь работать с базами данных.
- 6) **Научитесь использовать XML.**
- 7) Научитесь мгновенно создавать сайты на Joomla.
- 8) **Увидите процесс создания сайта <http://MyRusakov.ru>**
- 9) Увидите процесс размещения сайта в Интернете.
- 10) **Увидите процесс раскрутки сайта <http://MyRusakov.ru> в Интернете.**

Напоследок скажу, что знаний, которые даются в данном видеокурсе, хватит на **создание 99% сайтов**, которые Вы видите в Интернете!

Более подробное описание Вы увидите, когда диск выйдет в продажу.

Что нужно для получения скидки в 50%?

Всё, что от Вас требуется – это заполнить простую форму вот здесь:

<http://myrusakov.ru/bonuskurs.html>.

Разумеется, никто Вас не обязывает потом ничего покупать, а скидка у Вас **останется навсегда! Вы ничем не рискуете**: Вы просто получаете скидку, которой потом при желании сможете воспользоваться!

Думаю, что выбор очевиден в любом случае: <http://myrusakov.ru/bonuskurs.html>.

С Уважением, *Михаил Русаков!*

Срок действия Бонуса Ограничен (Вдобавок, он может меняться)! Более точно о сроках действия Бонуса Вы узнаете, пройдя по той же ссылке: <http://myrusakov.ru/bonuskurs.html>

По любым вопросам пишите мне на e-mail: myrusakov@gmail.com.