

# Введение в основы HTML

*Ты никогда не решишь проблему, если будешь думать так же, как и те, кто её создал.*

*Альберт Эйнштейн.*

---

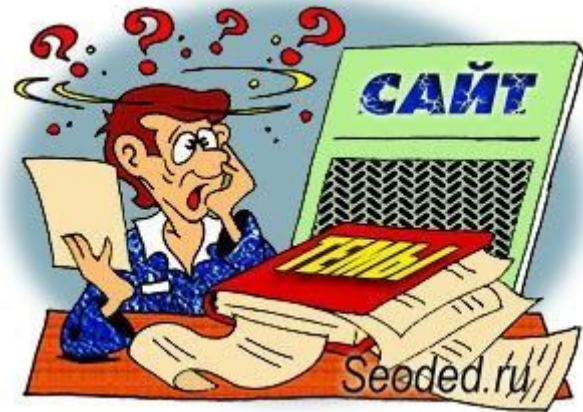
Итак, **введение в HTML**. Здесь, наверное, должны идти слова о том, что в этом разделе каждый найдёт что-нибудь полезное для себя, как обычно это делают авторы практически всех книг и статей. Но хочу сразу расставить все точки, запятые и прочие знаки препинания над и под буквами алфавита: полезной информация из этого раздела будет **новичку**, причем тому новичку, который хочет узнать как создать сайт, но не знает с чего начать.

Я не знаю, какой сайт ты задумал создать. Об этом мы поговорим чуть позже. Но спешу разочаровать тех, кто думает, что создав ресурс из двух страниц, одна из которых отведена под фотографию себя любимого на белом коне, а на второй красуется надпись «Это я — самый крутой чувак нашего района», он сразу же станет «как Бритни Спирс». Только работа, в которую вкладывается душа, труд и старание в итоге станет интересна людям. Не создавай красивые баночки без внутреннего содержимого. Качественная вещь и отличается от дешёвого многообразия неброским внешним видом и полезным, а самое главное нужным наполнением. Знания и умения со временем придут, но если внутри нет стремления к созданию качественного продукта, то ты так навсегда и останешься генератором дорвеев.

Всё, на этом вводная страшилка закончена, можешь идти дальше по пути к светлому будущему.

# Выбор темы для сайта

*Сначала Бог создал мужчину, потом он создал женщину.  
Потом Богу стало жалко мужчину, и он дал ему табак.  
Марк Твен.*



**П**равильный **выбор темы для сайта** и воплощение в жизнь проекта под названием «создание собственного сайта» является одновременно и очень простым, и очень сложным делом. Но остаётся при этом чертовски увлекательным занятием. По крайней мере, скучать тебе точно не придётся.

**С**ложность этого дела зависит от того, что ты хочешь получить в итоге. Если это простая личная страничка о себе любимом, состоящая из двух-трёх «листочков», то особо сложного тут ничего нет, и создать её можно новичку за пару часов. Но если речь идёт о чём-то большем, то для того, чтобы облегчить себе задачу, нужно с самого начала разбить весь процесс на этапы.

**В**ыглядит это примерно вот так:

1. Определи **для чего предназначен** твой будущий сайт (продажа чего-либо, обмен информацией о чем-либо и т.д.);
2. Продумай, что должен **включать** в себя твой сайт для выполнения этой задачи;
3. Создай **предварительный набросок** главной страницы на бумаге (схематично,

не надо рисовать шедевр для Эрмитажа). Эту страницу ты можешь потом использовать в качестве шаблона для остальных, таким образом у сайта будет единый стиль и похожесть страниц друг на друга;

**4.** После всего этого останется перевести твой «красивый картина» в код на языке HTML.

Самое важное и самое сложное на первоначальном этапе — это **выбор темы для будущего сайта**. Приходилось много раз видеть на различных форумах фразы типа: «Хочу заработать на своём сайте! Продавать там чего-нибудь или рекламу поставить!» Если именно так сформулировать тему для создания сайта, то слово «хочу» так и останется главным до конца жизни. Нам же нужно слово «могу».

Чтобы сайт приносил деньги нужно чётко определить: как он будет это делать, что для этого должны сделать мы и кто будут те люди, которые позволят нам получить доход. Если создать сайт с целью, например, обмена какой-либо информацией или это будет ресурс о твоём хобби, то смысл этих вопросов останется тем же, измениться только их формулировка. В общем, все эти вопросы можно привести к общему виду:

~ Почему ты хочешь создать сайт?

~ Кто будет аудиторией твоего сайта?

~ Чего хотят получить эти люди, прийдя на твой сайт?

Ещё одной распространённой ошибкой начинающих вебмастеров является выбор самой «**жирной**» (популярной) темы для сайта (ты мечтаешь стать королем порноиндустрии?), т. е. темы, в которой создано большое количество сайтов и которой интересуется много пользователей. Этого не стоит делать по нескольким причинам:

**а).** Помимо «порно» и «бальные сиськи» люди ищут в Интернете ещё много других вещей;

б). Вместе с большими деньгами в этих «жирных» темах (к которым кроме «сексуальных» относятся и темы недвижимости, продажи популярных товаров, автомобили и т.д.) находится и немалое число уже сложившихся вебмастеров, которые, в основном, объединены в коллективы. Конкурировать новичку с ними бессмысленно;

в). В большинстве этих тем начинающий слабо разбирается (знание запроса «большие сиськи» ещё не означает, что твой сайт будет в десятке выдачи по этому запросу). Поэтому создать конкурентноспособный ресурс будет очень проблематично;

г). Помимо знания самой темы, необходимо знать тонкости и нюансы раскрутки и продвижения сайта по этой тематике, чего опять-таки по определению не может новичок.

Я сам также наступал на эти «сексуальные» грабли. Я создал даже не сайт, а блог на бесплатном блогговом сервисе. Залил туда фотографии легко одетых тёток, развернул рекламную кампанию по блогам других пользователей этого сервиса и начал прикидывать, на что же я буду тратить деньги, которые вот-вот на меня посыпятся. При этом и посещаемость была (небольшая конечно, примерно 50 человек в день), но все посетители хотели только глазеть на тёток и упорно не желали тратить свои деньги на то, что я им предлагал. Тогда я решил, что это неправильные посетители и у них «неправильный мёд». Я удвоил усилия по рекламе и получил... кукиш я получил. Меня выбросили с этого сервиса с формулировкой «гад ползучий» и только после долгой переписки с администрацией, в которой пришлось им наплести, что я работаю в Мегафоне и что всё это издержки воспитания, мне вернули доступ к блогу. Он так до сих пор и стоит пустой (админы всех голых тёток попросили удалиться).

Поэтому, делая вывод из всего вышесказанного, могу посоветовать выбирать такую тему для создания первого сайта,

~ которая будет интересна тебе самому,

~ в которой ты хоть что-то понимаешь

~ и по которой ты можешь сказать другим чего-нибудь интересного.

Так ты и сайт не бросишь на полпути, и создашь нормальный ресурс, в который вложена душа и труд. А это наше единственное и главное преимущество перед всеми этими «гуру», у которых во главу угла поставлены деньги, причем быстрые. Качество же проекта даже не на втором, а на предпоследнем месте. И использовать с умом это преимущество наша святая обязанность. Поэтому только конкретно определившись с выбором темы для сайта и с механизмом взаимодействия ресурса с пользователями, можно переходить непосредственно к воплощению задуманного в жизнь, т.е. в Сеть. Так что определяйся и переходи к следующему разделу, в котором я расскажу тебе об основах языка HTML.

# Основные теги HTML

*Человек не может жить одними словами,  
в каком бы количестве он их ни проглатывал.  
Эдлай Стивенсон.*

---



Вообще-то, язык HTML не является языком программирования. Это язык разметки гипертекста (HyperText Markup Language). Т. е. называть его «языком HTML» не верно.

Если говорить проще, то HTML — это **список тегов** (управляющих слов), которые позволяют представить обычный текст в форматированном виде. Например, выделить его **жирным**, или *курсивом* и т. д. Но представить так текст можно только в специальных программах — браузерах (программы, с помощью которых ты бродишь по Интернету).

Они отображают только отформатированный текст и скрывают **теги HTML**, которые применялись для форматирования. Чтобы тебе было понятней, о чем я говорю, щелкни правой кнопкой мыши по этой странице и в открывшемся меню выбери пункт «Исходный код» или что-то подобное. Перед тобой откроется эта страница в своём настоящем виде.

Создавать код HTML можно как в обычных текстовых редакторах, так и использовать специальные программы, так называемые визуальные редакторы. Визуальные редакторы отличаются от текстовых тем, что заточены под набор кода HTML, CSS, JavaScript, PHP и др., т. е. под web-программирование. Кроме того, они позволяют сразу же просмотреть плоды своих трудов во встроенном браузере и подсвечивают некоторые ошибки в набранном коде.

Настоятельно рекомендую набирать весь код руками (для улучшения прямоты этих самых рук), визуальные редакторы же значительно облегчают этот процесс. Лично я пользуюсь Macromedia Dreamweaver 8. Это далеко не самая последняя версия, но её возможностей мне хватает с избытком.

Я поступаю так: набираю код в Dreamweaver, затем сохраняю и нажимаю кнопку «Просмотр в браузере» (список браузеров в этом меню ты редактируешь сам), после возвращаюсь в Dreamweaver и продолжаю редактирование.

Начнём со строения html-страницы, вернее с её основных тегов.

## Обязательные теги HTML-страницы

К обязательным (основным) тегам HTML, которые используются в каждом html-документе, относятся следующие:

< HTML >

< HEAD>< /HEAD>

< BODY>< /BODY>

< /HTML >



**Примечание.** Для того, чтобы браузер выводил код HTML как текст (не

интерпретировал его в код), после каждой открывающей угловой скобки «<» я оставляю пробел. Ты, когда будешь набирать код, пробелы **НЕ ОСТАВЛЯЙ**.

Как видишь, **все теги парные** (есть открывающий тег и закрывающий), в HTML почти все теги такие. Отличаются открывающий тег от закрывающего тем, что перед закрывающим стоит косая черта «/». Такие теги ещё называют **контейнерными**, потому что между ними можно вставить другие теги, т.е. поместить в контейнер. Ты и сам можешь видеть, что между тегами < html> и < /html> находятся другие теги.

Набирать имена тегов можно как заглавными буквами, так и прописными, разницы нет никакой. Т. е. записи вида < HTML> и < html> для браузера различий не имеют и интерпретируются одинаково. Теперь поговорим подробнее о том, что означают эти теги.

Так, основные теги < HTML > и < /HTML > показывают браузерам и другим программам для просмотра гипертекстовых страниц, что они как раз и имеют дело с

гипертекстовым документом. Любой html-документ должен начинаться с < HTML> и заканчиваться < /HTML>. Т. е. между этими тегами находится **ВЕСЬ** код html-страницы.

Между тегами < **HEAD**> и < /**HEAD**> находятся мета-теги (заголовок страницы, описание, ключевые слова и т.д.). В них хранится информация об html-странице и технические сведения. Вообще, данный тег не является обязательными, т. е. без него html-документ будет нормально отображаться браузерами. Но даже, если ты не планируешь использовать область заголовка, то напиши хотя бы просто < HEAD> < /HEAD>. Это нужно для совместимости между различными версиями программ.

Все сведения, находящиеся между тегами «HEAD», никак не отображаются браузером (за исключением информации между тегами < TITLE> < /TITLE>, где расположено название html-документа, выводимое в верхней панели браузера). Но могут оказывать большое влияние на вид html-страницы, её жизнь в Интернете и ранжирование поисковыми системами. Подробнее о тегах, которые располагаются в заголовке гипертекстового документа, мы поговорим позже.

Основное содержимое html-документа располагается между тегами < **BODY**> и < /**BODY**>. Здесь находится всё, что мы видим, когда открываем html-страницу: текст, графика, меню, кнопки и т. д. Это основное «тело» страницы.

Все остальные теги, использующиеся при создании html-документа, находятся между (т. е. внутри) этими обязательными тегами. Практически все теги в HTML имеют различные атрибуты, свойства и параметры, которые позволяют отобразить информацию именно так, как задумал разработчик. Например, если ты напишешь:

```
< BODY bgcolor="#FF0000">< /BODY>
```

, то фон всей страницы будет красным.

Здесь тег < **BODY**> — это, собственно, тег; **bgcolor** — его атрибут; "#FF0000" — значение атрибута (у тега может быть несколько атрибутов).



**Примечание.** Все атрибуты тега и их значения указываются в **открывающем** теге (тот, который **без** косой черты), но ни в коем случае не в **закрывающем**.

Итак, мы разобрались с основными тегами html-документа. Теперь настало время более подробно изучить остальные. Здесь я хочу сделать небольшое отступление.

## Как я предлагаю изучать HTML



Применяя на практике знания, полученные из [различных руководств по основам HTML](#), я понял, какую ошибку допустило большинство авторов этих книг и самоучителей. Все эти учебники, во-первых, предназначены **новичкам**, во-вторых, изучение основ HTML идёт в них по порядку.

Т. е. сначала рассматриваются теги заголовка html-документа (< HEAD>< /HEAD>, а потом теги «тела» (< BODY>< /BODY>).

Основываясь на собственном опыте, смею утверждать, что такой порядок изучения основ HTML для новичка категорически неприемлем. Новичок предпринимает попытку изучить все эти многочисленные и малопонятные ему теги заголовка, во-первых, не видя результатов своего обучения (напомню, теги заголовка браузером не отображаются). А во-вторых, даже не подозревая, что большинство этих тегов для создания задуманного им сайта не понадобятся (по крайней мере, в начале работы).

Таким образом, он только теряет время и желание продолжать изучение HTML. При этом утверждаясь во мнении, что [создать сайт](#) могут только «гуру» за хорошие деньги. Я бы тоже бросил это занятие, если бы не наплевал на то, что мне пока непонятно и не пошёл дальше.

Поэтому, я предлагаю тебе сначала рассмотреть теги, относящиеся к «телу» html-документа (< BODY>< /BODY>), а уж затем переходить к остальным. Тем более, что в «теле» html-страницы и содержится информация, за которой будут приходить посетители сайта. А создать или найти её гораздо сложнее, чем написать код.

# Атрибуты тега BODY

*Небывалая толщина этого отчёта  
защищала его от опасности быть прочитанным.  
Уинстон Черчилль.*

---



Все атрибуты тега **BODY** оказывают своё влияние на объекты, находящиеся между тегами **< BODY>** и **< /BODY>**. Т. е. на все элементы html-страницы. Это означает, что, по-умолчанию, браузер будет выводить html-страницу так, как это прописано или не прописано в теге **BODY**.

Если в каком-либо месте html-документа нужно отобразить текст, фон или другой элемент иначе, то это нужно прописывать в значениях атрибутов соответствующего тега. Т.е. значения тега **BODY** являются общими для всех элементов документа. И для изменения оформления какого-то одного достаточно указать это в его атрибутах.

Я пока не рассматриваю каскадные таблицы стилей (CSS), с помощью которых также можно задать отображение html-страницы. Кроме того, в этом разделе я не буду рассказывать об абсолютно всех атрибутах тега **BODY**. Я расскажу только об основных.

Итак, наиболее часто используемые атрибуты тега **BODY**:

**bgcolor** — цвет фона html-страницы. По-умолчанию, браузер заполняет фон сплошным цветом (в настройках по умолчанию — белым) или в зависимости от настроек пользователя этого браузера. Допустим, если вам нужен html-документ с чёрным фоном, то нужно написать так:

```
< BODY bgcolor="black">< /BODY>
```

или так:

< BODY bgcolor="#000000">< /BODY>

Здесь нужно коротко сказать о способах представления цвета в HTML. Цвета в HTML можно задать либо именем цвета из предопределенного списка, либо шестнадцатеричным (hex) значением выбранного цвета. В предопределенном списке содержится 140 цветов и все они соответствуют так называемой Web-палитре. Использование цветов из этой палитры гарантирует их одинаковое воспроизведение всеми браузерами.

Список цветов из Web-палитры ты найдёшь на странице Приложения. Там же ты найдёшь и шестнадцатеричные (hex) значения этих цветов (при использовании шестнадцатеричных (hex) значений перед числом нужно ставить символ #). В hex-значении цвета используется модель RGB: первые две цифры задают яркость красной (Red), вторые две цифры — зелёной (Green), а последние две цифры — синей (Blue) составляющей цвета.

Для того, чтобы использовать цвета, не входящие в Web-палитру, нужно знать их hex-значение. Я для этого использую программу, которую можно взять с моего сайта: ColorT Final.

С представлением цвета в HTML разобрались, теперь продолжим разговор об атрибутах тега BODY.

**background** — позволяет задать в качестве фона html-страницы картинку.

Допустим, есть картинка logo.jpg, расположенная по адресу <http://www.seoded.ru/images/logo.jpg>. Для того, чтобы сделать её фоном html-страницы нужно записать так:

< BODY background="http://www.seoded.ru/images/logo.jpg">< /BODY>

Если html-страница и картинка logo.jpg находятся в одном каталоге (папке), то можно записать так:

< BODY background="logo.jpg">< /BODY>

Если logo.jpg находится в подкаталоге images каталога с html-страницей (т.е. в папке images, которая находится в каталоге с html-страницей), то можно записать так:

< BODY background="images/logo.jpg">< /BODY>

Обычно в качестве фона используют небольшой и простой рисунок, так называемую текстуру, который многократно выводится на экран, заполняя всё окно. Поскольку размер такого изображения небольшой, то создание фона происходит очень быстро. Думаю с этим атрибутом всё понятно, перейдём к следующим.

**text** — устанавливает цвет текста html-документа. Например, сделаем текст зелёным:

```
< BODY text="#008000">< /BODY>
```

Следующие атрибуты относятся к представлению вида ссылок на странице:

**link** — цвет ещё неиспользованной (непросмотренной) ссылки (по умолчанию синий).

**vlink** — цвет уже использованной (просмотренной) ссылки.

**alink** — цвет гиперссылок при наведении на них курсора мыши.

Допустим, нам нужно, чтобы цвет ещё неиспользованных ссылок был красным (#FF0000), а уже нажатых — синим (#0000FF). Сделать это можно так:

```
< BODY link="#FF0000" vlink="#0000FF">< /BODY>
```

Да, я думаю, ты уже заметил, что атрибут в теге (не только BODY, но и в любом другом) может быть как один, так и несколько сразу. А может и ни одного, в этом случае браузер будет использовать настройки по-умолчанию. В следующем примере мы создадим html-страницу с чёрным фоном, зелёным текстом, неиспользованные ссылки — красные, а посещённые — синие. Весь этот ужас можно создать так:

```
< HTML>
```

```
< HEAD>
```

```
< /HEAD>
```

```
< BODY bgcolor="#000000" text="#008000" link="#FF0000" vlink="#0000FF">
```

```
< /BODY>
```

```
< /HTML>
```

Набери этот html-код в любом текстовом редакторе или визуальном. Набирай весь код вручную. Между тегами < BODY> и < /BODY> **вставь текст**, какой захочешь, сохрани этот файл под любым именем (латинскими буквами или цифрами), но **обязательно с**

расширением **html** (т. е., например, `primer.html`). А затем открой его в браузере и наслаждайся открывшейся тебе авангардистской картиной.



**Примечание.** Не забывай — я ставлю после скобки `<` пробел для того, чтобы браузер выводил код HTML, как текст. В самом html-коде этого делать **НЕ НАДО**.

Помимо собственных атрибутов (атрибуты, которые используются только в этом теге) в теге BODY можно применять атрибуты и других тегов (их можно употреблять в разных тегах, они общие для них всех). О них ты узнаешь в следующих частях этого раздела. Пока же настоятельно рекомендую поэкспериментировать с полученными знаниями. Только на практике ты **увидишь результаты** этих знаний и поймёшь, как это работает. И только поняв и запомнив, как атрибуты тега BODY используются при создании html-страницы, переходи к следующей главе, в которой мы поговорим о форматировании текста в HTML.

# Форматирование текста в HTML

*Указую господам сенаторам, чтобы речь держать не по-писаному,  
а своими словами, дабы дурь была видна каждого!*  
*Пётр Первый.*

---



В настоящее время, **текст в HTML** является основным способом подачи информации. Подавляющее большинство сайтов в Интернете состоят именно из текстового контента. Это означает, что создать свой сайт можно при умении, как минимум, читать, а как максимум — писать (в каждой шутке есть доля правды).

Если ты в предыдущем уроке последовал моим советам и поэкспериментировал с *цветом текста*, фона и т. д., то уже должен знать, что простой, неотформатированный текст можно добавить в html-страницу, просто поместив его между тегами **< BODY>** и **< /BODY>**.

Но такой текст не всегда удовлетворяет задумке создателя сайта и возникает необходимость какую-то часть текста как-то выделить или применить форматирование текста для подчеркивания его смыслового наполнения. Самое главное тут — соблюдать меру и следить за тем, чтобы всё было уместно и по делу. Даже стихами заговорил.

Ну что же, рассмотрим наиболее часто используемые **теги для форматирования текста в HTML** и их атрибуты.

## § 1. Теория

Здесь нужно коротко упомянуть о том, что такое текст в «электронном» виде. Я не буду подробно вдаваться в то, как он представляется в различных кодировках и системах. Скажу лишь о том, как связаны форматирование текста в HTML и его редактирование в текстовых редакторах.

Допустим, ты набрал текст в редакторе, произвел его форматирование (выделил какие-то слова **жирным** или *курсивом*), либо сделал с ним что-то ещё. Затем скопировал и вставил текст в свой html-код. Открываешь html-страницу в браузере, а всё твоё форматирование куда-то исчезло. Происходит это потому, что текстовый редактор для форматирования текста применяет свои спецсимволы (по сути те же теги, только используются в текстовых редакторах), которые ты по умолчанию не видишь.

В окне редактора ты видишь только текст. А на самом деле там есть ещё управляющие спецсимволы, с помощью которых редактор «понимает», как ему отображать тот или иной кусок текста. Т. е., повторюсь, это то же самое, что и теги в HTML, только «заточенные» под использование в текстовых редакторах.

Но, не смотря на это, существует способ перенести текст из редактора и вставить его в html-код в исходном виде. И раз уж об этом зашла речь, то форматирование текста и начнем изучать с этого тега. Итак, приступим.

## § 2. Сами теги для форматирования

Предварительно отформатированный текст можно вставить в html-страницу с помощью тегов `<pre>` и `</pre>`. Браузер покажет текст, заключенный между этими тегами в том виде, в каком он был в текстовом редакторе. Я, если честно, ни разу не использовал этот тег на практике, поэтому что-то ещё про него сказать не могу.

Самыми популярными тегами для форматирования текста являются:

`<B></B>` — используется для выделения текста **полужирным**.



**Примечание.** Во многих последних (по времени выпуска) учебниках и самоучителях вместо него рекомендуется использовать тег `<strong></strong>`. Я не знаю, почему это так, но для «жирного» выделения использую рекомендуемый

тег **strong**.

`< I>< /I>` — используется для получения *курсивного* текста. Вместо этого тега рекомендуется использовать тег `< em>< /em>`.

`< U>< /U>` — позволит подчеркнуть текст. Тут главное, чтобы пользователь не перепутал подчеркнутый текст с ссылкой.

`< STRIKE>< /STRIKE>` — делает текст ~~перечёркнутым~~.

`< SUB>< /SUB>` — предназначен для выделения подстрочного текста (нижний индекс). Этот тег может быть полезен, например, для записи химических формул.

Вот этот код

**H<sub>2</sub>O**

даст нам вот такую формулу

**H<sub>2</sub>O**.

`< SUP>< /SUP>` — нужен для выделения надстрочного текста (верхнего индекса). Например, для записи возведения в степень: пишем

**(a+b)<sup>2</sup>**,

получаем

**(a+b)<sup>2</sup>**.

Все эти теги для форматирования текста можно использовать как по одному, так и несколько вместе. Достигается это путем вложения тегов друг в друга. Например, следующий html-код:

**`< strong>H2O< /strong>` — это `< strong>< em>формула воды.< /em>< /strong>`**

при просмотре в браузере будет выглядеть так:

**H<sub>2</sub>O — это *формула воды*.**



## § 3. Абзацы

В HTML абзацы создаются с помощью тега **< P>**. Использование этого тега приводит к вставке одной пустой строки и размещению находящегося за ним видимого элемента html-страницы в следующей строке. Например, если мы напишем:

**< p>Это первый абзац.< /p> < p>Это второй абзац.< /p>**

То выглядеть это будет так:

**Это первый абзац.**

**Это второй абзац.**

Использовать закрывающий тег **< /P>** не обязательно, т. к. браузер считает, что начало следующего абзаца означает конец предыдущего, но для избежания путаницы желательно.

## § 4. Выравнивание текста

Для того, чтобы выровнять текст используется атрибут **align** с возможными значениями **center**, **left**, **right** и **justify**. Он, соответственно, выравнивает текст по центру, по левому краю, по правому краю и одновременно по двум краям сразу. Например, html-код:

**< p align="center">Текст по центру< /p>**

выравнивает текст по центру:

**Текст по центру**

А этот код:

**< p align="right">Выравнивание текста по правому краю< /p>**

выравнивает текст по правому краю

**Выравнивание текста по правому краю**

Если атрибут **align** не используется, то текст будет выровнен по левому краю. Это происходит по умолчанию. Пример, код:

**< p>По умолчанию текст выравнивается по левому краю< /p>**

выровнит текст по левому краю

**По умолчанию текст выравнивается по левому краю**

## § 5. Перенос строки и горизонтальная черта

Если нужно обеспечить переход на новую строку без вставки пустой строки (это происходит, когда ты в текстовом редакторе нажимаешь Enter), то используется тег **< BR>**. Например, html-код

**1. Первая строка. < br> 2. Вторая строка.**

в браузере будет выглядеть так:

**1. Первая строка.**

**2. Вторая строка.**

У **< BR>** нет закрывающего тега. Ты уже наверное догадался, что для вставки нескольких пустых строк, нужно записать тег **< BR>** несколько раз подряд.

Визуально разделить части текста можно ещё при помощи тега **< HR>**. Этот тег создает на странице разделительную полосу (например, как зелёная в начале этой статьи). Тег **< HR>** имеет следующие атрибуты:

**size** — толщина полосы;

**width** — ширина полосы;

**align** — выравнивание;

**color** — цвет полосы;

**noshade** — атрибут, не имеющий значений. Если указан, то создается сплошная чёрная полоса без тени.

Например, html-код:

< hr align="center" size="5" width="50%" color="#3399ff">

в браузере примет вид

---

Ширину и толщину можно указывать как в пикселах (тогда величина будет постоянной), так и в процентах (тогда величина будет меняться в зависимости от разрешения экрана).

## § 6. Заголовки

Для задания названий разделов и подразделов в HTML используются **теги заголовков**. Существуют шесть уровней заголовков и обозначаются они так:

< H1>< /H1> — заголовок первого уровня

< H2>< /H2> — заголовок второго уровня

< H3>< /H3> — заголовок третьего уровня

< H4>< /H4> — заголовок четвёртого уровня

< H5>< /H5> — заголовок пятого уровня

< H6>< /H6> — заголовок шестого уровня

Заголовки выделяются полужирным шрифтом и отделяются от остального текста пустой строкой. Самый крупный заголовок < H1>< /H1> (им выделено название этого урока), а самый маленький < H6>< /H6>. Теги заголовков имеют атрибуты:

**align** — выравнивание текста заголовка на странице (значения те же, что и для выравнивания обычного текста);

**title** — всплывающая подсказка, которая появляется при наведении мыши на заголовок.

Например, код HTML:

< h4 align="center" title="Всплывающая подсказка. Чтобы увидеть, наведи мышь на заголовок.">Заголовок 4 уровня< /h4>

в браузере будет выглядеть так:

**Заголовок 4 уровня**

## § 7. Работа со шрифтами

Для форматирования непосредственно шрифтов, в HTML есть тег **< FONT>< /FONT>**. У этого тега существуют следующие атрибуты:

**color** — цвет текста шрифта;

**face** — гарнитура шрифта;

**size** — размер текста шрифта. Может быть задан абсолютным (от 1 до 6) либо относительным значением (от +1 до +6 и от -1 до -6). По-умолчанию, размер текста 3. Если указать **< font size="+1">< /font>**, то текст, размещённый между тегами **< font>** и **< /font>**, будет отображаться увеличенным на 1 по сравнению с другим текстом. Аналогично и с -1.

## § 8. Создание списков в HTML

В HTML списки создаются с помощью тегов **< OL>< /OL>** и **< UL>< /UL>**. В первом случае создается упорядоченный (нумерованный), во втором неупорядоченный (маркированный) списки. Каждый элемент списка выделяется тегом **< LI>< /LI>**. Браузер отображает элементы списка с новой строки в виде столбца с отступом и маркерами. Вид маркера определяется атрибутом **type**.

В нумерованном списке маркеры (значения атрибута **type**) такие:

**1** — нумерация арабскими цифрами (по умолчанию);

**A** — большими латинскими буквами по алфавиту;

**a** — маленькими латинскими буквами;

**I** — большими римскими цифрами;

**i** — маленькими римскими цифрами.

Например, HTML-код

```
< ol type="a">
```

```
< li>Первый элемент.< /li>
```

```
< li>Второй элемент.< /li>
```

**< li>Третий элемент.< /li>**

**< /ol>**

браузер покажет так:

1. **Первый элемент.**
2. **Второй элемент.**
3. **Третий элемент.**

В маркированном списке значения атрибута **type** такие:

**disc** — закрашенный кружок (по-умолчанию);

**circle** — незакрашенный кружок;

**square** — закрашенный квадратик.

Например, HTML-код

**< ul type="square">**

**< li>Первый элемент.< /li>**

**< li>Второй элемент.< /li>**

**< li>Третий элемент.< /li>**

**< /ul>**

в браузере выглядит так:

- **Первый элемент.**
- **Второй элемент.**
- **Третий элемент.**

Как и остальные теги HTML, списки можно вкладывать друг в друга. Получаются многоуровневые вложенные списки. Например, код:

**< ol type="a">**

< li>Первый элемент.< /li>

< li>

< ul type="square">

< li>Второй элемент.< /li>

< li>Третий элемент.< /li>

< li>Четвёртый элемент.< /li>

< /ul>

< /li>

< li>Пятый элемент.< /li>

< /ol>

браузер покажет так:

1. Первый элемент.
  - Второй элемент.
  - Третий элемент.
  - Четвёртый элемент.
2. Пятый элемент.

## § 9. Заключение

Рассказ о том, что нужно для форматирования текста в HTML, был бы не полным без упоминания о **спецсимволах**. Спецсимволы — это символы, которых либо нет на клавиатуре, либо те, которые браузер принимает за управляющий символ и преобразует в код. Полный список этих символов ты найдешь здесь [Спецсимволы HTML](#). Кроме того, самые главные знания для форматирования текста — это [правила грамматики русского языка](#). Грамотно написанный текст уже наполовину отформатирован.

Ну и напоследок, для закрепления всего вышеизложенного, в качестве самостоятельной работы, изучи вот этот HTML-код:

< font color="#FF0000" size="+3">< strong>&nbsp;H< /strong>< /font>

и попробуй сам понять, что означают эти теги и их атрибуты, а также, где на этой странице мог бы использоваться этот код HTML.

**Н**а этом рассказ о **тегах HTML для форматирования текста** объявляется законченным. Раздел получился довольно большим. Не пытайся выучить все новые теги и атрибуты наизусть, всё равно не получится. Они сами постепенно запомнятся во время работы над созданием сайта.

# Картинки в HTML

*Об этом человеке известно только то,  
что он не сидел в тюрьме, но почему не сидел — неизвестно.  
Марк Твен.*

---



Это урок о том, **как вставить картинку в HTML**, как её оформить, как сделать обтекание картинки текстом и т.д. Ведь известно, что изображения делают сайт более привлекательным и отличным от других ресурсов, поэтому умение использовать тег **<IMG>** и его атрибуты весьма пригодится в современном Интернете. Но главное тут — чувство меры!

Переизбыток графики вызовет утяжеление html-страницы и, соответственно, увеличит время её загрузки. Кроме того, наличие большого числа изображений будет отвлекать посетителей от главного содержимого сайта (если, конечно, графика не является главным содержимым сайта). Так что, соблюдай меру и используй **картинки в HTML** только там, где это нужно. И будет тебе счастье!

## § 1. Как вставить картинку

Для вставки картинок в HTML используется тег **IMG** с *обязательным* атрибутом **SRC**. Этот атрибут указывает браузеру путь к файлу изображения. Т.е. для вставки картинки с **названием logo.jpg** в определенное место страницы (при условии, что и страница, и картинка расположены **в одной папке** (каталоге)) нужно в этом месте вставить следующий html-код:

**< img src="logo.jpg" >**

Если картинка и страница расположены в **разных** каталогах (папках), то нужно указать путь к изображению **относительно** страницы. Например, если html-страница лежит в каталоге (папке) site, в этом же каталоге (папке) расположен подкаталог (папка) images, в котором и находится наша картинка logo.jpg, то для её вставки нужно написать так:



< img src="images/logo.jpg">

А можно не мучиться и указать **полный адрес картинки**. Например, так:

< img src="http://www.seoded.ru/image/seoded.png">

В последнем случае браузер отобразит код так:



**!** **Примечание.** Если картинка расположена **на вашем компьютере**, а вставить вы её хотите **на страницу в Интернете**, то ничего не выйдет. Для этого картинку нужно сначала переместить в какое-нибудь место **в Интернете** (например, на файловый хостинг). И указать в коде страницы полный адрес **до этого места с картинкой**.

Помимо обязательного атрибута **SRC** у тега **IMG** есть ещё несколько необязательных атрибутов. Рассмотрим их подробнее.

## § 2. Указание размеров картинки

Начнём с атрибутов, которые позволяют задать **размеры картинки** (точнее, застолбить место под эти размеры на страницы). Вот они:

**width** — ширина картинки в пикселах или процентах;

**height** — высота картинки в пикселах или процентах.

Если используются эти атрибуты, то браузер сначала выделит место под графику, подготовит макет документа, отобразит текст и только затем загрузит изображение. При этом он поместит картинку в прямоугольник выделенного размера, даже если реальные ширина и высота картинки больше (сожмёт) или меньше (растянет). В случае, когда эти атрибуты не используются, браузер будет грузить картинку сразу, а вывод идущих за ней текста и остальных элементов задержится.

Ширину и высоту изображений можно указывать как в пикселах (при этом размер картинки будет постоянным вне зависимости от разрешения экрана), так и в процентах (размер картинки будет зависеть от разрешения экрана пользователя). Например:

```
< img src="logo.jpg" width="50" height="20">
```

или

```
< img src="logo.jpg" width="10%" height="5%">
```

## § 3. Альтернативный текст

В случае, если пользователь в настройках браузера отключил показ изображений, то можно вместо картинки вывести альтернативный текст, который бы объяснил, что за графика здесь должна быть. Достигается это путём применения атрибута **ALT**:

```
< img src="logo.jpg" alt="Логотип сайта www.seoded.ru">
```

В этом случае браузер резервирует место на странице под изображение, но вместо самой картинки покажет текст, который ты напишешь в значении атрибута **ALT**:

Повторю, это произойдет, если пользователь отключил показ графики. Если же нет, то картинка скроет собой альтернативный текст.

## § 4. Выравнивание картинки

С помощью уже знакомого тебе атрибута **align** можно управлять выравниванием картинок относительно других элементов html-страницы. У атрибута **align** есть несколько значений, но нас больше всего на данный момент интересуют два:

**left** — изображение располагается у левого края страницы, а текст обтекает картинку справа;

**right** — изображение располагается у правого края страницы, а текст и другие элементы обтекают картинку слева.

Например, HTML-код

```
< img src="logo.jpg" align="left"> < p>Это мой аватар на форумах и в различных  
сервисах< /p>
```

браузер покажет так



Это мой аватар на форумах и в различных сервисах

А этот HTML-код:

```
 <p>Это мой аватар на форумах и в различных  
сервисах</p>
```

будет выглядеть вот так:



Это мой аватар на форумах и в различных сервисах

Для прекращения обтекания картинки текстом можно использовать тег **BR** (знакомый нам из прошлого раздела про [форматирование текста в HTML](#)). У тега **BR** есть атрибут **clear**, который может принимать три значения:

**left** — прекращение обтекания текстом картинок, выровненных по левому краю;

**right** — прекращение обтекания текстом картинок, выровненных по правому краю;

**all** — прекращение обтекание текстом картинок, выровненных и по левому, и по правому краю.

## § 5. Создание отступов вокруг картинки

По-умолчанию, картинки встраиваются в страницу почти вплотную к тексту и другим элементам. Если такое отображение не устраивает, то можно задать отступы вокруг картинки. Делается это с помощью следующих атрибутов тега IMG:

**vspace** — создает верхний и нижний отступы;

**hspace** — создает боковые отступы (слева и справа).

Значения этих атрибутов указывается в пикселах. Например, если мы немного дополним предыдущий пример:

```
 <p>Это мой аватар на  
форумах и в различных сервисах</p>
```

, то получим следующее:



Сравни это с тем, как вся «композиция» выглядела раньше (выше).

## § 6. Рамка вокруг картинки

Вокруг картинок можно создать чёрную рамку с помощью атрибута **border**. Значение этого атрибута указывается в пикселах и определяет толщину рамки. Например, HTML-код:

```

```

в браузере выглядит так:



## § 7. Заключение

На этом раздел о тегах HTML, которые нужны для вставки картинок на страницы сайта можно считать завершённым. В следующем разделе поговорим о том, как создаются ссылки, и о том, как сделать ссылку-картинку в HTML.

# Гиперссылки в HTML

*В этом браке нас было трое. А я не люблю толпы.*

*Принцесса Диана.*

---



Главным признаком HTML-документа является наличие в нём **гиперссылок** (или просто ссылок) на другие документы, сайты, файлы, картинки и т. д. Именно возможность вставлять в страницы ссылки на объекты вне неё и сделала Интернет столь популярным и удобным для использования. Поэтому при создании своего сайта всегда помни про «магию» ссылок.

В этом уроке мы поговорим о том, **как создать ссылку** на сайт, на его отдельную страницу или файл. Ты узнаешь о том, как изменить текст ссылки, как открыть её в новом окне, как сделать ссылкой картинку, что такое внешние и внутренние ссылки и многое другое. Плюс, у тебя уже есть информация о работе с ссылками, которой мы касались в прошлых уроках (например, о том, как можно поменять цвет ссылки мы говорили в [уроке про тег body](#)).

В общем, этот урок сделает твоё представление о создании ссылок законченным и достаточным. Ты поймёшь, как создать гиперссылку в HTML и зачем. И научишься управлять её свойствами.

## Термины

**Анкор ссылки** (от англ. «*anchor*» — «якорь») — то, что находится в коде гиперссылки между тегами `< A>` и `< /A>`. Например, там может быть:

- Текст
  - Код: `< a href="http://www.seoded.ru/">Анкор ссылки< /a>`
  - Вид в браузере: [Анкор ссылки](#)
- Адрес ссылки
  - Код: `< a href="http://www.seoded.ru/">http://www.seoded.ru/< /a>`

- Вид в браузере: <http://www.seoded.ru/>

И так далее.

## § 1. Ссылка на файл, ссылка на сайт, ссылка на страницу

Многочисленные вопросы о том, чем отличается ссылка на файл от ссылки на сайт или его отдельную страницу, вынудили меня вынести ответ на него в самое начало этого урока. Ответ такой: ничем. Все перечисленные ссылки являются внешними по отношению к **исходной** странице и создаются одним и тем же способом.

Дабы не растекаться мыслью по древу, покажу всё на примере.

Итак, код ссылки на сайт выглядит так:

```
< a href="http://www.seoded.ru/">Ссылка на сайт< /a>
```

В браузере мы увидим вот, что:

[Ссылка на сайт](http://www.seoded.ru/)

Код ссылки на страницу выглядит так:

```
< a href="http://www.seoded.ru/beginner.html">Ссылка на страницу< /a>
```

В браузере мы увидим вот, что:

[Ссылка на страницу](http://www.seoded.ru/beginner.html)

Код ссылки на файл выглядит так:

```
< a href="http://www.seoded.ru/downloads/soft/colortfinal1.5.rar">Ссылка на  
файл< /a>
```

В браузере мы увидим вот, что:

[Ссылка на файл](http://www.seoded.ru/downloads/soft/colortfinal1.5.rar)

Как видишь, все типы ссылок создаются абсолютно одинаково. Разница только в **адресе** объекта, на который нужно сослаться. А теперь перейдём к основной части урока.

## § 2. Создание внешних ссылок

Между собой ссылки различаются на **внешние** и **внутренние**, а также на **текстовые** и **графические**. Внешние ссылки ведут за «пределы» html-страницы, внутренние на различные части **этой же** страницы. Текстовые ссылки представляют из себя текст (по-умолчанию, он выделен синим цветом и подчеркнут), а графические, в качестве объекта, по которому нужно щелкнуть для перехода, содержат какую-либо картинку. Все эти разновидности ссылок создаются в HTML с помощью тега `< A>< /A>` (сокращение от англ. anchor — якорь). Рассмотрим его поподробнее.

Для создания внешней ссылки на сайт, страницу или файл служит атрибут тега `< A>` — **href**. В качестве значения этот атрибут принимает **URL-адрес** сайта, страницы или файла (об этом мы говорили выше). Между тегами `< A>` и `< /A>` располагается видимая часть ссылки (анкор ссылки), т. е. то, что мы видим на экране браузера. Анкор ссылки может быть как обычным текстом (текстовая ссылка), так и графическим изображением (ссылка-картинка). Ссылка-картинка создаётся вставкой знакомого нам по прошлому уроку тега `< IMG>` между тегами `< A>` и `< /A>`. В общем, синтаксис создания ссылки выглядит так:

`< a href="URL-адрес или имя файла">текст ссылки (анкор)< /a>`

Например, чтобы создать текстовую ссылку на главную страницу этого сайта, нужно написать следующий HTML-код:

`< a href="http://www.seoded.ru/">Главная страница сайта Seoded.ru< /a>`

В браузере это будет выглядеть так:

[Главная страница сайта Seoded.ru](http://www.seoded.ru/)

Как я писал в самом начале этого урока, цвет текста ссылок (анкора) можно изменить с помощью атрибутов тега BODY. Вообще, к тексту ссылок можно применять все те же варианты форматирования, что и к основному тексту страницы, т. е. выделять жирным, курсивом, использовать заголовки и т. д.

### § 2.1 Графические ссылки (ссылки-картинки)

Как я уже сказал выше, для того, чтобы создать ссылку-картинку, нужно вместо текста использовать изображение. Пример такой ссылки выглядит так:

```
< a href="http://www.seoded.ru/">< img src="logo.jpg">< /a>
```

А браузер покажет:



По-умолчанию, браузер окружает картинку в графической ссылке рамкой. Если это нежелательно, то атрибуту **border** тега **IMG** нужно присвоить значение 0:

```
< a href="http://www.seoded.ru/">< img src="logo.jpg" border="0">< /a>
```

Текстовые и графические ссылки можно комбинировать. Если написать:

```
< a href="http://www.seoded.ru/">< img src="logo.jpg"> Главная страница< /a>
```

, то ссылкой будет как картинка, так и текст «Главная страница»



Главная страница

### § 3. Внутренние ссылки

Для комфортного перемещения по страницам с большим количеством контента используются внутренние ссылки. Именно с их помощью я сделал «Содержание урока» (см. в начале этой страницы). Внутренние ссылки создаются по тому же принципу, что и внешние. Только в значении атрибута **href** указывается «якорь» ссылки. «Якорь» создаётся атрибутом **name**:

```
< a name="имя якоря">текст< /a>
```



Имя «якоря» задаётся произвольно. Тут стоит сказать, что не все браузеры понимают русские имена «якорей», поэтому рекомендую использовать латиницу. Текст между тегами `< A>` `< /A>` для создания «якоря» не обязателен и, чаще всего, не указывается.

«Якорь» располагается в тех местах страницы, в которые должен попасть пользователь после щелчка по ссылке.

Как я уже сказал выше, в атрибуте **href** внутренней ссылки вместо адреса указывается имя нужного «якоря» с обязательным символом решётки (#) перед ним. Разберём на примере.

Я создал «якорь» с именем **zagolovok** и разместил его в коде страницы рядом с заголовком этого урока («Гиперссылки в HTML»). Код «якоря» следующий:

```
< a name="zagolovok">< /a>
```

HTML-код внутренней ссылки будет выглядеть так:

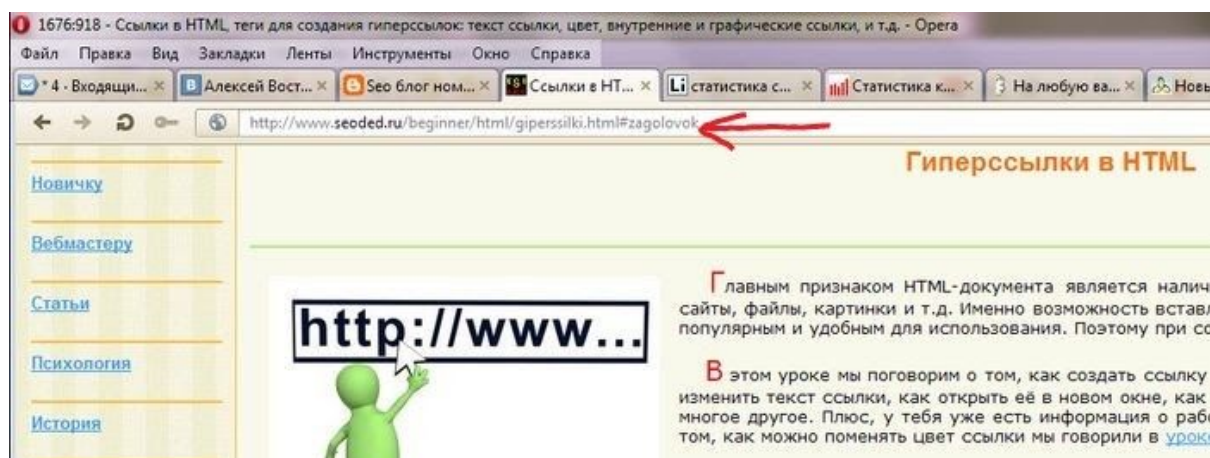
```
< a href="#zagolovok">К заголовку< /a>
```

, а в браузере так:

### К заголовку

После щелчка по этой ссылке, ты попадёшь к заголовку этого урока.

Если ты заметил, то после перехода по внутренней ссылке к заголовку изменился URL-адрес в адресной строке браузера:



К изначальному адресу:

<http://www.seoded.ru/beginner/html/giperssilki.html>

Добавилась внутренняя ссылка:

<http://www.seoded.ru/beginner/html/giperssilki.html#zagolovok>

Используя эту особенность, можно ссылаться на определённое место страницы с любого ресурса в Интернете! Т. е., допустим, ты создал страницу с объёмной статьёй о чём-то (или выложил на странице большое число фотографий) и разметил её внутренними ссылками. Находясь в социальной сети ВКонтакте, тебе потребовалось сослаться не просто на страницу со статьёй (или фотографиями), а на определённое место на ней (или определённую фотографию). Используя вариант с внутренней ссылкой в адресе, ты с лёгкостью добьёшься нужного.

## § 4. Абсолютные и относительные ссылки

Также ссылки бывают **абсолютными** и **относительными**. Абсолютная ссылка — это обычная ссылка вида:

```
< a href="http://www.seoded.ru/">Главная страница< /a>
```

А вот с относительными ссылками немного посложнее. В таких ссылках адрес указывается либо **относительно** корневой папки сайта (той, в которой лежит главная страница), либо относительно исходной страницы. Такие ссылки нужны, например, в том случае, если сайт лежит на домашнем компьютере. Или это не сайт, а страница-указатель на другие документы.

Разберём сначала создание ссылки **относительно корневой папки** сайта.

Допустим нам нужно сослаться на страницу **klienty.html**, которая лежит **в одной папке** с главной страницей сайта. Тогда код относительной ссылки примет вид:

```
< a href="/klienty.html">Клиенты< /a>
```

А теперь предположим, что в одной папке с главной страницей лежит **папка zakazy** и уже **в ней** лежит страница klienty.html Тогда код относительной ссылки станет таким:

```
< a href="/zakazy/klienty.html">Клиенты< /a>
```

Т. е. при формировании ссылки относительно корневой папки сайта мы опускаем начальную часть адреса **http://www.seoded.ru** и оставляем всё остальное. Использование слеша «/» в начале **обязательно!**

Теперь рассмотрим создание гиперссылок **относительно исходной страницы**. Допустим, у нас есть страница **price.html** (исходная страница) и с неё нужно сослаться на страницу **klienty.html** Тут бывают следующие типовые варианты:

**1.** Страницы price.html и klienty.html находятся **в одной папке**.

Тогда код ссылки будет таким:

```
< a href="klienty.html">Клиенты< /a>
```

**2.** Страница klienty.html и папка zakazy находятся **в корневой папке сайта**, страница price.html **лежит в папке zakazy** (т. е. страница klienty.html относительно исходной страницы price.html лежит **на один уровень выше**).

Код станет таким:

```
< a href="../klienty.html">Клиенты< /a>
```

Две точки показывают, что нужно выйти из текущей папки на уровень выше.

**3.** Страница klienty.html и папка zakazy находятся **в корневой папке сайта**, папка mebel **лежит в папке zakazy**, страница price.html **лежит в папке mebel** (т. е. страница klienty.html относительно исходной страницы price.html лежит **на два уровня выше**).

Код ссылки примет вид:

```
< a href="../../klienty.html">Клиенты< /a>
```

Т. е. каждый уровень обозначается двумя точками и слешем «/».

**4.** Страница price.html (исходная страница) и папка zakazy находятся **в корневой папке сайта**, страница klienty.html **лежит в папке zakazy** (т. е. теперь страница klienty.html относительно исходной страницы price.html лежит **на один уровень ниже**).

Теперь код ссылки будет таким:

```
< a href="zakazy/klienty.html">Клиенты< /a>
```

В этом случае точки и слеша не ставятся.

**5.** Страница price.html (исходная страница) и папка zakazy находятся **в корневой папке сайта**, папка mebel **лежит в папке zakazy**, страница klienty.html **лежит в папке mebel** (т. е. теперь страница klienty.html относительно исходной страницы price.html **лежит на два уровня ниже**).

Код ссылки такой:

`< a href="zakazy/mebel/klienty.html">Клиенты< /a>`

**6.** В корневой папке сайта **лежат две папки**: zakazy и oplata. Страница klienty.html **лежит в папке zakazy**, исходная страница price.html **лежит в папке oplata** (т. е. обе страницы лежат **в разных папках на уровень ниже** от корневой папки сайта).

Код ссылки станет следующим:

`< a href="../zakazy/klienty.html">Клиенты< /a>`

По аналогии создаются относительные ссылки и в том случае, если страницы лежат на разных уровнях относительно корневой папки. Будет меняться лишь количество указателей уровня (две точки и слеш) и адрес страницы, на которую ставится ссылка.

## § 5. Ссылка на электронную почту

Для того, чтобы **создать ссылку на электронную почту**, нужно вместо URL-адреса в значении атрибута href написать адрес электронной почты с указанием протокола (**mailto:**). И тогда при щелчке по такой ссылке откроется окно почтовой программы с вписанным в поле «Кому» адресом электронной почты. В HTML-коде это выглядит так:

`< a href="mailto:aleksei.seoded@gmail.com">Моя почта< /a>`

, а в браузере так:

**Моя почта**

## § 6. Атрибуты тега «А»

У тега `< A>`, как и у остальных тегов в HTML, есть свои атрибуты. Рассмотрим некоторые из них.

## § 6.1 Атрибут target

По-умолчанию, браузер при переходе по ссылке открывает страницу в этом же окне. Чтобы изменить это, используется атрибут тега **A target**. Он имеет следующие значения:

**\_blank** — открывает ссылку в новом окне (в современных браузерах в новой вкладке).

**\_parent** — загружает ссылку в родительском окне.

**\_self** — открывает ссылку в этом же окне. Это значение стоит у всех ссылок по-умолчанию (т. е. если вообще не указывать атрибут target, то сработает именно это значение).

Это не все значения атрибута **target**, но это самые основные. Пример использования этого атрибута:

```
< a href="http://www.seoded.ru/" target="_blank">Главная страница< /a>
```

Главная страница

## § 6.2 Атрибут title

В теге `< A>` можно также использовать атрибут **title**. С его помощью создаётся всплывающая подсказка, которая появляется при наведении мыши на ссылку.

```
< a href="http://www.seoded.ru/" title="Перейти на главную страницу">Главная  
страница< /a>
```

Главная страница

## § 6.3 Атрибут rel="nofollow"

У атрибута **rel** есть несколько значений, но самым используемым является значение «**nofollow**». Это значение указывает роботам поисковых систем, что эту ссылку не нужно использовать в поисковом ранжировании. Подробнее — в разделе о продвижении сайтов.

Исходный код:

< a href="http://www.seoded.ru/" rel="nofollow">Анкор ссылки< /a>

В браузере:

Анкор ссылки

## § 7. Заключение

Это основные сведения о **создании ссылок в HTML**. На этом данный урок можно считать законченным. А сей факт говорит о том, что большую часть основ HTML ты уже освоил.

Рекомендую тебе сейчас ещё раз пробежаться по предыдущим разделам, а затем создать полноценную HTML-страницу, посвященную, например, тебе любимому. По твоему дизайну и сообразно твоим желаниям. После чего выложить её на какой-нибудь бесплатный хостинг (конечно, если у тебя есть платный хостинг, то на него) и уже полноценно полюбоваться на свой первый сайт.