

МИНИ

<!--savme.ru-->

СПРАВОЧНИК ВЕРСТАЛЯ



Автор: Савельев В.А.

*Дорогой друг! Я рад, что тебя заинтересовала данная книга!
В первую очередь, это пособие будет интересно тем людям,
которые хотят научиться создавать, редактировать и
изменять сайты с помощью языков HTML и CSS...
В этом справочнике собраны основные теги и их значения!*

2015

Оглавление

ВВЕДЕНИЕ.....	3
НЕ МНОГО ОБ АВТОРЕ	4
ОСНОВНЫЕ ТЕГИ САЙТА.....	5
ТЕГИ ОФОРМЛЕНИЯ ТЕКСТА.....	10
СОЗДАНИЕ СПИСКОВ	13
ТЕГИ ССЫЛОК	14
ТЕГИ МЕДИА ДАННЫХ.....	15
СКРИПТЫ НА САЙТЕ	18
СОЗДАНИЕ ТАБЛИЦ	18
СОЗДАНИЕ ФОРМ.....	20
ТАБЛИЦА СОБЫТИЙ	24
УНИВЕРСАЛЬНЫЕ АТТРИБУТЫ К ТЕГАМ	25
КОРОТКО О ФАЙЛЕ СТИЛЕЙ CSS	26

*Для перехода к нужному разделу книги - кликните...

ВВЕДЕНИЕ

Дорогой друг, я рад, что тебя заинтересовала данная книга! Меня зовут Владимир Савельев, и я являюсь автором блога savme.ru.

Первым делом я хочу рассказать, о чем эта книга! Данная рукопись не является обучающим материалом, она содержит в себе лишь основные теги html, которые пригодятся при создании сайтов, либо их редактировании!

Я занимаюсь разработкой сайтов уже более 5 лет, создал много проектов на различных cms, но вот запомнить такое количество тегов и держать в памяти, очень сложно, не правда ли?

Именно поэтому и родилась мысль создать справочник! Что бы самые нужные теги, были в одном месте, всегда под рукой, не зависимо от интернета!

Почему я решил создать этот справочник, когда в интернете можно найти подобную информацию, практически на любой тег? Отвечаю - для удобства!

Эта маленькая книжица имеет ряд преимуществ:

- ★ Не нужно ничего искать, что экономит время!
- ★ Не требуется доступ в интернет, все нужное собрано в одном справочнике в формате pdf.
- ★ Простота использования - скопировал, вставил!
- ★ Четкое и понятное описание каждого кода!

Этот справочник абсолютно бесплатный! Пользуйтесь, творите, создавайте валидные сайты! Самую актуальную версию «мини справочника версталя», всегда можно бесплатно скачать по адресу - savme.ru.

Надеюсь, этот справочник еще не раз пригодится Вам, в процессе разработки и редактировании сайта, либо иного проекта!

НЕ МНОГО ОБ АВТОРЕ

Меня зовут Владимир Савельев, 1984 года рождения. Родился и живу по настоящее время в городе Нижнекамск. Женат, воспитываю троих детей! Очень люблю свою жену Надежду, которая поддерживает меня во всем!

Свою деятельность в интернете веду уже более 5 лет. За это время накопил большой опыт и знания в сфере создания и разработки сайтов, а так же в получении дохода через интернет.

Получаю знания и реализую их на практике по настоящее время, так как Интернет не стоит на месте, а динамично развивается, особенно в последние годы!

В первую очередь мой ресурс будет интересен людям, которые хотят научиться создавать сайты и зарабатывать на них, а так же для тех, у кого есть сайт, но они не довольны результатом!

Кроме этого, я устраиваю конкурсы и игры с ценными призами, помогаю экономить на покупке софта или обучающих курсов, раздаю бесплатные консультации!

Чтобы быть в курсе жизни блога - подпишись на мою [бесплатную рассылку](#), я ценю своих читателей и уважаю их!

Более подробно обо мне, можно почитать на странице блога «[об авторе](#)».

До встречи на блоге, с кликом по жизни...



С уважением Владимир Савельев

Личный блог автора: savme.ru.

ОСНОВНЫЕ ТЕГИ САЙТА

<!DOCTYPE>

Этот элемент указывает на то, в каком формате был создан сайт, так как существуют различные версии HTML разметки.

Обычно его размещают в самом начале документа и браузер сразу понимает, в каком формате был разработан сайт и правильно его считывает и выводит.

Если неправильно прописать этот элемент, сайт может отобразиться - криво!

DOCTYPE	Описание
HTML 4.01	
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">	Строгий синтаксис HTML.
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">	Переходный синтаксис HTML.
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Frameset//EN" "http://www.w3.org/TR/html4/frameset.dtd">	В HTML-документе применяются фреймы.
HTML 5	
<!DOCTYPE html>	Для всех документов.

Тег <html>

Этот тег заключает в себе весь сайт и все прочие теги. Как правило, тег <html> идет вторым элементом после значения <!DOCTYPE> и самым последним, закрывающим </html>.

Например:

```
<html> ... </html>
```

Атрибуты:

title - Добавляет всплывающую подсказку на веб-страницу.

manifest - Указывает файл манифеста, необходимый для создания оффлайн-приложения.

Также для этого тега доступны [универсальные атрибуты](#).

Тег <head>

Данный тег предназначен вмещать в себя другие теги, такие как например: стили сайта, метаданные, тайтл (заголовок сайта), скрипты (но я рекомендую по возможности размещать скрипты перед тегом </body>) и другие.

Цель тега <head> помочь браузеру и Поисковым системам лучше обработать и понять Ваш сайт.

Содержимое тега <head> не отображается на сайте, за исключением тега <title> устанавливающего заголовок окна веб-страницы.

Внутри <head> можно размещать такие теги как: <base>, <basefont>, <bgsound>, <link>, <meta>, <script>, <style>, <title>.

Пример написания:

```
<head> ... </head>
```

Тег <title>

Определяет заголовок сайта, либо отдельной его страницы! Прописывается в контейнере <head>. Заголовок играет большую роль в продвижении сайта.

Пример написания:

```
<head><title>Заголовок</title></head>
```

Тег <meta>

Метатеги предназначены в основном для браузера и поисковых систем. В них мы можем например указать описание сайта, ключевые слова и фразы, индексацию, кодировку сайта, и прочее...

Все метатеги размещаются в контейнере <head>.

Пример написания:

```
<head><meta name="..." content="..." /></head>
```

Атрибуты

charset - Задаёт кодировку документа.

content - Устанавливает значение атрибута, заданного с помощью name или http-equiv.

http-equiv - Предназначен для конвертирования метатега в заголовок HTTP.

name - Имя метатега, также косвенно устанавливает его предназначение.

Тег <link>

Устанавливает связь с внешним документом например файлы со стилями или шрифтами. Тег <link> размещается всегда внутри контейнера <head> и не создает ссылку.

Пример написания:

```
<head><link атрибуты></head>
```

Атрибуты

charset - Кодировка связываемого документа.

href - Путь к связываемому файлу.

media - Определяет устройство, для которого следует применять стилевое оформление.

rel - Определяет отношения между текущим документом и файлом, на который делается ссылка.

sizes - Указывает размер иконок для визуального отображения.

type - MIME-тип данных подключаемого файла.

Также для этого тега доступны [универсальные атрибуты](#).

Тег <style>

Применяется для определения стилей сайта или его элементов. Этот тег необходимо использовать внутри контейнера <head>. Можно задавать более чем один тег <style>. Я рекомендую по возможности размещать все стили в один файл, дабы уменьшить вес сайта.

Пример написания:

```
<style type="text/css">  
  h1 {font-size: 28px;font-family:Arial;color: #333;}  
</style>
```

Тег <body>

Этот тег включает в себя весь контент и другие теги. Например текст, картинки, видео и любые другие элементы, которые будут показаны браузером. Обычно прописывается сразу после тега </head>.

Пример написания:

```
<body> ... </body>
```

Атрибуты

alink - Устанавливает цвет активной ссылки.

background - Задаёт фоновый рисунок на веб-странице.

bgcolor - Цвет фона веб-страницы.

bgproperties - Определяет, прокручивать фон совместно с текстом или нет.

bottommargin - Отступ от нижнего края окна браузера до контента.

leftmargin - Отступ по горизонтали от левого края окна браузера до контента.

link - Цвет ссылок на веб-странице.

rightmargin - Отступ от правого края окна браузера до контента.

scroll - Устанавливает, отображать полосы прокрутки или нет.

text - Цвет текста в документе.

topmargin - Отступ от верхнего края окна браузера до контента.

vlink - Цвет посещенных ссылок.

Тег <header>

Тег <header> задает «шапку» сайта или раздела, в которой обычно располагается заголовок.

Пример написания:

```
<header> ... </header>
```

Тег <nav>

Тег <nav> обычно используется при создании меню. Заключает в себе ссылки. Допускается использовать несколько таких тегов.

Пример написания:

```
<nav>ссылки</nav>
```

Тег <footer>

Тег <footer> называется еще подвал сайта. В нем может быть все что угодно, но обычно прописывают Авторские права и кнопки статистики.

Пример написания:

```
<footer> ... </footer>
```

Тег <!-- -->

Тег добавляет комментарий в код документа. Закомментированный элемент не выводится в браузере. Можно использовать в нем другие теги. Вложенные комментарии друг в друга - недопустимы.

Пример написания:

<!-- пример закомментированного элемента -->

Тег <div>

Элемент <div> является блочным элементом. Блок можно визуальным образом изменять с помощью стилей. Для этого создаем уникальное имя блока и добавляем его в атрибут class или id. Блок <div> всегда начинается с новой строки, все элементы после него отображаются на следующей строке.

Пример написания:

<div>...</div> или <div class="Blok"></div> или <div id="Blok"></div>

Атрибуты

align - Задаёт выравнивание содержимого тега <div>.

title - Добавляет всплывающую подсказку к содержимому.

Также для этого тега доступны [универсальные атрибуты](#).

**Тег **

Тег определяет строчный элемент. В отличие от блочных элементов, с его помощью можно выделить часть информации внутри других тегов и установить для нее свой стиль. Можно использовать атрибуты class или id с уникальным именем, для написания стилей во внешнем документе «style.css».

Пример написания:

... или ...

Тег <p>

Определяет текстовый абзац. Тег <p> является блочным элементом, всегда начинается с новой строки, абзацы текста идущие друг за другом разделяются между собой отбивкой. Ее величину можно прописать в стилях. Если закрывающего тега нет, считается, что конец абзаца совпадает с началом следующего блочного элемента.

Пример написания:

<p>Текст</p>

Атрибуты

align - Определяет выравнивание текста.

Также для этого тега доступны [универсальные атрибуты](#).

Тег <hr>

Создает горизонтальную линию, которая внешне зависит от ее параметров, а также браузера. Тег <hr /> относится к блочным элементам, линия всегда начинается с новой строки, все элементы после нее отображаются на следующей строке.

Пример написания:

```
<hr />
```

Атрибуты

align - Определяет выравнивание линии.

color - Цвет линии.

noshade - Рисует линию без трехмерных эффектов.

size - Толщина линии.

width - Ширина линии.

Также для этого тега доступны [универсальные атрибуты](#).

ТЕГИ ОФОРМЛЕНИЯ ТЕКСТА

Тег <h1> ... <h6>

В HTML коде применяются шесть заголовков разного типа, по умолчанию тег <h1> самый крупный и важный, а заголовок <h6> самый меньший и менее важный по сравнению с другими уровнями. Теги <h1>,...,<h6> это блочные элементы, начинаются они с новой строки, после них прочие элементы, идут на следующей строке.

Пример написания:

```
<h1>Заголовок первого уровня</h1>
```

Атрибуты

align - Определяет выравнивание заголовка.

Также для этого тега доступны [универсальные атрибуты](#).

Тег

Делает шрифт жирным. Можно использовать с другими тегами текста.

Пример написания:

```
<b>текст</b>
```

Тег

Тег `` предназначен для выделения важности текста. Браузеры отображают такой текст жирным начертанием. Обычно им выделяю ключевые фразы, но не увлекайтесь с выделениями...

Пример написания:

```
<strong>Текст</strong>
```

Тег `<i>`

Делает шрифт курсивом. Можно использовать с другими тегами текста.

Пример написания:

```
<i>Текст</i>
```

Тег ``

Тег `` предназначен для выделения важности текста. Браузеры отображают такой текст курсивом. Обычно им выделяю ключевые фразы, но не увлекайтесь с выделениями...

Пример написания:

```
<em>Текст</em>
```

Тег `<u>`

Делает текст подчеркнутым. Этот тег осуждается спецификацией HTML, взамен рекомендуется использовать стили.

Пример написания:

```
<u>Текст</u>
```

Тег `<big>`

Увеличивает размер текста. В HTML размер шрифта определяется либо в единицах от 1 до 7, либо в пикселях. Средний размер текста, по умолчанию, принят 3. Можно использовать несколько вложенных тегов.

Пример написания:

```
<big>Текст</big>
```

Тег `<small>`

Тег `<small>` наоборот, уменьшает размер текста на одну единицу по сравнению с обычным текстом. Можно использовать несколько вложенных тегов.

Пример написания:

```
<small>Текст</small>
```

Тег

Тег
 переводит строку на новый уровень в том месте, где он прописан. В отличие от тега абзаца <p> в том, что тег
 не добавляет пустой отступ перед строкой.

Пример написания:

Абзац
Новая строка

Атрибуты

Clear - сообщает браузеру, как обрабатывать следующую строку, если текст обтекает плавающий элемент.

Тег <pre>

Тег <pre> это блок предварительно форматированного текста. Внутри такого контейнера можно отобразить любые теги, кроме этих: <big>, , <object>, <small>, <sub> и <sup>. Так же учитываются все пробелы.

Пример написания:

<pre>Текст</pre>

Тег <code>

Тег <code> позволяет отобразить на сайте любой программный код. Это имена переменных, ключевые слова, тексты функции и т.д. В отличие от тега <pre> пробелы внутри контейнера <code> не учитываются. Используйте
 или <p>.

Пример написания:

<code>Текст</code>

Тег <cite>

Тег <cite> относится к цитатам или сноскам. Обычно такой текст отображается курсивом.

Пример написания:

<cite>Текст</cite>

Тег <q>

Тег <q> нужен для выделения в тексте цитат. Отображается выделенный текст в кавычках.

Пример написания:

<q>Текст</q>

Тег <sup>

Отображает шрифт в виде верхнего индекса.

Пример написания:

```
<sup>Текст</sup>
```

Тег <sub>

Отображает шрифт в виде нижнего индекса.

Пример написания:

```
<sub>Текст</sub>
```

Тег <blockquote>

Этот тег нужен для выделения длинных цитат внутри документа. Такой текст обычно имеет отступы, слева отступ будет намного больше остальных.

Пример написания:

```
<blockquote>Цитата</blockquote>
```

СОЗДАНИЕ СПИСКОВ

**Тег **

Тег маркированный список. Каждый элемент списка имеет теги .

Пример написания:

```
<ul><li>элемент маркированного списка</li></ul>
```

Атрибуты

type - Устанавливает вид маркера списка.

Также для этого тега доступны [универсальные атрибуты](#).

**Тег **

Тег нумерованный список. Каждый элемент списка имеет теги .

Пример написания:

```
<ol><li>элемент нумерованного списка</li></ol>
```

Атрибуты

type - Устанавливает вид маркера списка.

reversed - Нумерация в списке становится по убыванию (3,2,1).

start - Задаёт число, с которого будет начинаться нумерованный список.

Также для этого тега доступны [универсальные атрибуты](#).

Тег

Тег это отдельный элемент списка. Внешний тег или устанавливает тип списка — маркированный или нумерованный.

Пример написания:

```
<li>элемент списка</li>
```

Атрибуты

type - Устанавливает вид маркера нумерованного или маркированного списка.

value - Число, с которого будет начинаться нумерованный список.

Также для этого тега доступны [универсальные атрибуты](#).

ТЕГИ ССЫЛОК

Тег <a>

Тег <a> это важный элемент HTML и предназначен для создания ссылок. В зависимости от атрибутов name или href, устанавливает ссылку или якорь. Якорь это закладка внутри страницы, по которой возможен переход к элементу.

Пример написания:

```
<a href="URL">...</a>
```

```
<a name="идентификатор">...</a>
```

Атрибуты

href - Задаёт адрес документа, на который следует перейти.

name - Устанавливает имя якоря внутри документа.

rel - Отношения между ссылаемым и текущим документами.

target - Имя окна или фрейма, куда браузер будет загружать документ.

title - Добавляет всплывающую подсказку к тексту ссылки.

Тег <nav>

Тегом <nav> обозначают навигацию по сайту. Если на странице несколько блоков ссылок, то в <nav> обычно помещают приоритетные ссылки. Допустимо использовать несколько тегов <nav>.

Пример написания:

```
<nav>ссылки</nav>
```

ТЕГИ МЕДИА ДАННЫХ

(Картинки, видео, аудио, объекты)

Тег

Тег нужен для отображения картинки на сайте. Путь до файла с картинкой задаётся через атрибут src. Можно картинку сделать ссылкой, заключив в теги <a>.

Пример написания:

```

```

Атрибуты

align - Определяет как рисунок будет выравниваться по краю и способ обтекания текстом.

alt - Альтернативный текст для изображения.

border - Толщина рамки вокруг изображения.

height - Высота изображения.

hspace - Горизонтальный отступ от изображения до окружающего контента.

ismap - Говорит браузеру, что картинка является серверной картой-изображением.

longdesc - Указывает адрес документа, где содержится аннотация к картинке.

lowsrc - Адрес изображения низкого качества.

src - Путь к графическому файлу.

vspace - Вертикальный отступ от изображения до окружающего контента.

width - Ширина изображения.

usemap - Ссылка на тег <map>, содержащий координаты для клиентской карты-изображения.

Также для этого тега доступны [универсальные атрибуты](#).

Тег <embed>

<embed> нужен для загрузки и отображения объектов (например, видеофайлов, флэш-роликов, музыки и прочее). Рекомендуется тег <embed> помещать внутрь тега <object>.

Пример написания:

```
<object><embed width="..." height="..."></embed></object>
```


Атрибуты

align - Определяет, как объект будет выравниваться на странице и способ его обтекания текстом.

height - Высота объекта.

hidden - Указывает, скрыть объект на странице или нет.

hspace - Горизонтальный отступ от объекта до окружающего контента.

pluginspage - Адрес страницы в Интернете, откуда можно скачать и установить плагин к браузеру.

src - Путь к файлу.

type - MIME-тип объекта.

vspace - Вертикальный отступ от объекта до окружающего контента.

width - Ширина объекта.

Тег <object>

Тег <object> сообщает браузеру, как загружать и отображать объекты, которые исходно браузер не понимает.

Пример написания:

```
<object width="ширина" height="высота"></object>
```

Атрибуты

align - Определяет, как объект будет выравниваться на странице и способ его обтекания текстом.

archive - Устанавливает путь к файлам, необходимым для работы объекта.

classid - Адрес программы (приложения или плагина), которая работает с данным объектом, и будет запускать его.

code - Имя объекта для его выполнения.

codebase - Путь к папке с объектом, который указан атрибутом code или classid.

codetype - Указывает на тип объекта, который задан атрибутом classid.

data - Адрес файла для его отображения в окне браузера.

height - Высота объекта.

hspace - Горизонтальный отступ от объекта до окружающего контента.

tabindex - Определяет порядок перехода между элементами с помощью клавиши Tab.

type - MIME-тип объекта.

vspace - Вертикальный отступ от объекта до окружающего контента.

width - Ширина объекта.

Также для этого тега доступны [универсальные атрибуты](#).

Тег <source>

Вставляет звуковой или видеофайл для тегов <audio> и <video>.

Пример написания:

```
<audio><source src="URL"></audio>
```

```
<video><source src="URL"></video>
```

Атрибуты

media - Определяет устройство, для которого будет воспроизводиться файл.

src - Адрес файла.

type - MIME-тип медийного источника.

Тег <video>

Добавляет, воспроизводит и управляет настройками видеоролика на веб-странице. Путь к файлу задается через атрибут src или вложенный тег <source>.

Пример написания:

```
<video><source src="URL"></video>
```

Атрибуты

autoplay - Видео начинает воспроизводиться автоматически после загрузки страницы.

controls - Добавляет панель управления к видеоролику.

height - Задаёт высоту области для воспроизведения видеоролика.

loop - Повторяет воспроизведение видео с начала после его завершения.

poster - Указывает адрес картинки, которая будет отображаться, пока видео не доступно или не воспроизводится.

preload - Используется для загрузки видео вместе с загрузкой веб-страницы.

src - Указывает путь к воспроизводимому видеоролику.

width - Задаёт ширину области для воспроизведения видеоролика.

Тег <audio>

Добавляет, воспроизводит и управляет настройками аудиозаписи на веб-странице. Путь к файлу задается через атрибут `src` или вложенный тег `<source>`.

Пример написания:

```
<audio src="URL"></audio>
```

```
<audio><source src="URL"></audio>
```

Атрибуты

`autoplay` - Звук начинает играть сразу после загрузки страницы.

`controls` - Добавляет панель управления к аудиофайлу.

`loop` - Повторяет воспроизведение звука с начала после его завершения.

`preload` - Используется для загрузки файла вместе с загрузкой веб-страницы.

`src` - Указывает путь к воспроизводимому файлу.

СКРИПТЫ НА САЙТЕ

Тег `<script>`

Тег `<script>` предназначен для обработки скриптов. Скрипты могут располагаться во внешнем файле и связываться с HTML.

Скрипты желательно располагать в конце документа BODY, а особо важные в HEAD, так как будут загружаться после контента на странице.

Пример написания:

```
<script type="тип"> ... </script>
```

```
<script type="тип" src="URL"> ... </script>
```

Атрибуты

`async` - Загружает скрипт асинхронно.

`defer` - Откладывает выполнение скрипта до тех пор, пока вся страница не будет загружена полностью.

`language` - Устанавливает язык программирования на котором написан скрипт.

`src` - Адрес скрипта из внешнего файла для импорта в текущий документ.

`type` - Определяет тип содержимого тега `<script>`.

СОЗДАНИЕ ТАБЛИЦ

Тег <table>

Элемент <table> это таблица. Она содержит в себе элементы, которые определяют ее содержимое. Любая таблица состоит из строк и ячеек, которые задаются с помощью тегов <tr> и <td>. Внутри <table> допустимо использовать следующие элементы: <caption>, <col>, <colgroup>, <tbody>, <td>, <tfoot>, <th>, <thead> и <tr>.

<tr> - строка, <td> - столбец, <th> - строка заголовка.

Пример написания:

```
<table><tbody><tr><th></th></tr><tr><td>...</td></tr></tbody></table>
```

Атрибуты

align - Определяет выравнивание таблицы.

background - Задаёт фоновый рисунок в таблице.

bgcolor - Цвет фона таблицы.

border - Толщина рамки в пикселах.

bordercolor - Цвет рамки.

cellpadding - Отступ от рамки до содержимого ячейки.

cellspacing - Расстояние между ячейками.

cols - Число колонок в таблице.

frame - Сообщает браузеру, как отображать границы вокруг таблицы.

height - Высота таблицы.

rules - Сообщает браузеру, где отображать границы между ячейками.

summary - Краткое описание таблицы.

width - Ширина таблицы.

Также для этого тега доступны [универсальные атрибуты](#).

СОЗДАНИЕ ФОРМ

Тег <form>

Тег <form> создает форму на сайте. Эта форма взаимодействует с посетителем и сервером. Область применения форм практически не ограничена.

Пример написания:

```
<form action="URL"> ... </form>
```

Атрибуты

accept-charset - Устанавливает кодировку, в которой сервер может принимать и обрабатывать данные.

action - Адрес программы или документа, который обрабатывает данные формы.

autocomplete - Включает автозаполнение полей формы.

enctype - Способ кодирования данных формы.

method - Метод протокола HTTP.

name - Имя формы.

novalidate - Отменяет встроенную проверку данных формы на корректность ввода.

target - Имя окна или фрейма, куда обработчик будет загружать возвращаемый результат.

Также для этого тега доступны [универсальные атрибуты](#).

Тег <input>

Тег <input> один из элементов формы. Основная задача <input> это создание текстовых полей, различных кнопок, переключателей и флажков. Элемент <input> не требуется размещать внутри <form>, но если какие-либо данные должны отправляться на сервер, то указывать <form> обязательно.

Основной атрибут <input>, это - type. Он может задавать такие элементы формы как: текстовое поле (text), переключатель (radio), флажок (checkbox), скрытое поле (hidden), кнопка (button), кнопка для отправки формы (submit), кнопка для очистки формы (reset), поле для отправки файла (file) и кнопка с изображением (image). Для каждого элемента существует свой список атрибутов, которые определяют его вид и характеристики.

Пример написания:

```
<input атрибуты>
```

Атрибуты

accept - Устанавливает фильтр на типы файлов, которые вы можете отправить через поле загрузки файлов.

`accesskey` - Переход к элементу с помощью комбинации клавиш.

`align` - Определяет выравнивание изображения.

`alt` - Альтернативный текст для кнопки с изображением.

`autocomplete` - Включает или отключает автозаполнение.

`autofocus` - Устанавливает фокус в поле формы.

`border` - Толщина рамки вокруг изображения.

`checked` - Предварительно активированный переключатель или флажок.

`disabled` - Блокирует доступ и изменение элемента.

`form` - Связывает поле с формой по её идентификатору.

`formaction` - Определяет адрес обработчика формы.

`formenctype` - Устанавливает способ кодирования данных формы при их отправке на сервер.

`formmethod` - Сообщает браузеру каким методом следует передавать данные формы на сервер.

`formnovalidate` - Отменяет встроенную проверку данных на корректность.

`formtarget` - Определяет окно или фрейм в которое будет загружаться результат, возвращаемый обработчиком формы.

`list` - Указывает на список вариантов, которые можно выбирать при вводе текста.

`max` - Верхнее значение для ввода числа или даты.

`maxlength` - Максимальное количество символов разрешенных в тексте.

`min` - Нижнее значение для ввода числа или даты.

`multiple` - Позволяет загрузить несколько файлов одновременно.

`name` - Имя поля, предназначено для того, чтобы обработчик формы мог его идентифицировать.

`pattern` - Устанавливает шаблон ввода.

`placeholder` - Выводит подсказывающий текст.

`readonly` - Устанавливает, что поле не может изменяться пользователем.

`required` - Обязательное для заполнения поле.

`size` - Ширина текстового поля.

`src` - Адрес графического файла для поля с изображением.

`step` - Шаг приращения для числовых полей.

`tabindex` - Определяет порядок перехода между элементами с помощью клавиши Tab.

`type` - Сообщает браузеру, к какому типу относится элемент формы.

value - Значение элемента.

Также для этого тега доступны [универсальные атрибуты](#).

Тег <label>

Тег <label> устанавливает связь между определенной меткой, например текст, и элементом формы (<input>, <select>, <textarea>). Эта связь нужна для того чтобы менять значения элементов формы. Например, кликом курсора мышки.

Пример написания:

```
<input id="идентификатор"><label for="идентификатор">Текст</label>  
<label><input type="..."> Текст</label>
```

Атрибуты

accesskey - Доступ к элементам формы с помощью горячих клавиш.

for - Идентификатор элемента, с которым следует установить связь.

Тег <select>

Тег <select> нужен для создания раскрывающихся списков формы.

Пример написания:

```
<select><option>Пункт 1</option><option>Пункт 2</option></select>
```

Атрибуты

accesskey - Позволяет перейти к списку с помощью некоторого сочетания клавиш.

autofocus - Устанавливает, что список получает фокус после загрузки страницы.

disabled - Блокирует доступ и изменение элемента.

form - Связывает список с формой.

multiple - Позволяет одновременно выбирать сразу несколько элементов списка.

name - Имя элемента для отправки на сервер или обращения через скрипты.

required - Список обязателен для выбора перед отправкой формы.

size - Количество отображаемых строк списка.

tabindex - Определяет последовательность перехода между элементами при нажатии на клавишу Tab

Также для этого тега доступны [универсальные атрибуты](#).

Тег <textarea>

Поле <textarea> требуется для ввода текста, например сообщения. В отличие от тега <input> в текстовом поле допустимо делать переносы строк, они сохраняются при отправке данных на сервер.

Между тегами <textarea> и </textarea> можно поместить любой текст, который будет отображаться внутри поля.

Пример написания:

```
<textarea атрибуты> текст</textarea>
```

Атрибуты

accesskey - Переход к полю с помощью сочетания клавиш.

autofocus - Автоматическое получение фокуса.

cols - Ширина поля в символах.

disabled - Блокирует доступ и изменение элемента.

form - Связывает текстовое поле с формой по её идентификатору.

maxlength - Максимальное число введенных символов.

name - Имя поля, предназначено для того, чтобы обработчик формы мог его идентифицировать.

placeholder - Указывает замещающийся текст.

readonly - Устанавливает, что поле не может изменяться пользователем.

required - Обязательное для заполнения поле.

rows - Высота поля в строках текста.

tabindex - Порядок перехода между элементами при нажатии на клавишу Tab.

wrap - Параметры переноса строк.

Также для этого тега доступны [универсальные атрибуты](#).

Тег <button>

Тег <button> нужен для создания различных кнопок на сайте. Используя стили можно менять как внешний вид кнопки, так и физические размеры. Помещать <button> в контейнер <form> обязательно.

Пример написания:

```
<form><button>...</button></form>
```

Атрибуты

accesskey - Доступ к элементам формы с помощью горячих клавиш.

`autofocus` - Устанавливает, что кнопка получает фокус после загрузки страницы.

`disabled` - Блокирует доступ и изменение элемента.

`form` - Связывает между собой форму и кнопку.

`formaction` - Задаёт адрес, на который пересылаются данные формы при нажатии на кнопку.

`formenctype` - Способ кодирования данных формы.

`formmethod` - Указывает метод пересылки данных формы.

`formnovalidate` - Отменяет проверку формы на корректность.

`formtarget` - Открывает результат отправки формы в новом окне или фрейме.

`name` - Определяет уникальное имя кнопки.

`type` - Тип кнопки — обычная, для отправки данных формы на сервер или для очистки формы.

`value` - Значение кнопки, которое будет отправлено на сервер или прочитано с помощью скриптов.

ТАБЛИЦА СОБЫТИЙ

<code>onblur</code>	Потеря фокуса
<code>onchange</code>	Изменение значения элемента формы
<code>onclick</code>	Щелчок левой кнопкой мыши на элементе.
<code>ondblclick</code>	Двойной щелчок левой кнопкой мыши на элементе.
<code>onfocus</code>	Получение фокуса
<code>onkeydown</code>	Клавиша нажата, но не отпущена.
<code>onkeypress</code>	Клавиша нажата и отпущена.
<code>onkeyup</code>	Клавиша отпущена.
<code>onload</code>	Документ загружен.
<code>onmousedown</code>	Нажата левая кнопка мыши.
<code>onmousemove</code>	Перемещение курсора мыши.
<code>onmouseout</code>	Курсор покидает элемент.
<code>onmouseover</code>	Курсор наводится на элемент.
<code>onmouseup</code>	Левая кнопка мыши отпущена.
<code>onreset</code>	Форма очищена.
<code>onselect</code>	Выделен текст в поле формы.
<code>onsubmit</code>	Форма отправлена
<code>onunload</code>	Заккрытие окна.

УНИВЕРСАЛЬНЫЕ АТТРИБУТЫ К ТЕГАМ

Универсальные атрибуты применяются практически ко всем тегам, поэтому выделены в отдельную группу, чтобы не повторять их для всех тегов.

accesskey	Позволяет получить доступ к элементу с помощью заданного сочетания клавиш.
class	Определяет имя класса, которое позволяет связать тег со стилевым оформлением.
contenteditable	Сообщает, что элемент доступен для редактирования пользователем
contextmenu	Устанавливает контекстное меню для элемента
dir	Задаёт направление и отображение текста — слева направо или справа налево.
hidden	Скрывает содержимое элемента от просмотра
id	Указывает имя стилового идентификатора
lang	Браузер использует значение параметра для правильного отображения некоторых национальных символов.
spellcheck	Указывает браузеру проверять или нет правописание и грамматику в тексте.
style	Применяется для определения стиля элемента с помощью правил CSS.
tabindex	Устанавливает порядок получения фокуса при переходе между элементами с помощью клавиши Tab.
title	Описывает содержимое элемента в виде всплывающей подсказки.

КОРОТКО О ФАЙЛЕ СТИЛЕЙ CSS

.css - это файл стилевого оформления сайта и его элементов.

Основная задача файла css уменьшить объем сайта и ускорить его загрузку, а так же он обладает большими возможностями оформления внешнего вида сайта.

Для того чтобы подключить внешние стили сайту, достаточно создать файл с разрешением .css и прописать код и путь до этого файла в области <HEAD>.

Например:

```
<link rel='stylesheet' href='/style.css' type='text/css' media='all' />
```

В самом документе HTML указываем на стиль элемента так:

```
<div id="menu"> ... </div> или <div class="menu"> ... </div>
```

А в документе style.css, прописываем эти классы так:

```
#menu {параметры и атрибуты стиля элемента}
```

или

```
.menu {параметры и атрибуты стиля элемента}
```

Основные параметры и атрибуты стилей:

margin – внешние отступы элемента, которые имеют пиксельные значения, либо в процентном соотношении. Пишутся значения начиная с верха по часовой стрелке. Допустим margin: 1px 2px 4px 6px – 1px будет верх элемента, 2px право, 4px низ и 6px лево.

Примеры:

margin:10px; - означает равные значения элемента со всех сторон. Если отступы будут иметь разную величину то пишем так: margin: 10px 5px 8px 13px; Если же отступы сверху и снизу, а также слева и справа имеют одинаковые значения, их можно объединять - margin: 10px 8px; то есть сверху и снизу будет 10px, а с права и слева по 8px.

Так же если нам нужно указать внешний отступ только одной стороны, то можно писать так – margin-top:10px; margin-right:10px; margin-bottom:10px; margin-left:10px;

Top – это верх, right – это право, bottom – это низ, left – это лево элемента.

padding – аналогично тегу margin, разница лишь в том, что этот тег указывает на внутренние отступы элемента, а не на внешние.

display – определяет как элемент должен быть показан на сайте.

Значения:

block – элемент отображается как блок, происходит перенос строк в начале и в конце содержимого.

Inline – содержимое блочных элементов не переносит строки, а начинается с того места, где окончился предыдущий элемент.

None – элемент становится невидимым, а место которое он занимал не учитывается, как будто его и нет.

Table – блочный элемент определяется как таблица.

List-item – блочный элемент с добавлением маркера.

Другие значения - inline-block, inline-table – сами элементы блочные, но их обтекают другие элементы.

position – позиционирование элемента относительно окна браузера или других элементов.

Значения:

absolute – абсолютное позиционирование элемента, другие элементы ведут себя так будто его и нет.

fixed – похоже на абсолютное позиционирование, разница лишь в том, что фиксированный элемент не меняет свое положение при прокрутке страницы.

relative – по умолчанию, относительно своего исходного положения.

float – элемент выравнивается по определенной стороне, а другие элементы будут обтекать его.

Значения:

Left – выравнивание по левому краю, остальные элементы обтекают с права.

Right – выравнивание по правому краю, остальные элементы обтекают слева.

None – обтекание элемента не задано.

width – задает ширину элемента и измеряется в пикселах (px), дюймах (in), пунктах (pt) или в процентном соотношении (%).

Значения:

Auto – автоматически вычисляет одну из сторон, зависит от типа и содержания.

height – тоже самое что и width (ширина), только это уже высота элемента. Ведет себя точно так же.

Автор: Савельев Владимир

<http://savme.ru>

Значения:

Auto – автоматически вычисляет одну из сторон, зависит от типа и содержания.

max-width и min-width – максимальная и минимальная ширина элемента. Допустим есть фотография 400px на 100px. Так вот, задаем ей ширину width:100% и дополнительно задаем max-width:400px;

Теперь наша фотография будет резиновой, то есть уменьшаться и увеличиваться, в зависимости от содержимого. Но как только изображение достигнет 400px, увеличение прекратиться. Подобное происходит и с min-width, как только изображение достигнет минимальной ширины, то уменьшение прекратиться.

Широко применяется в адаптивной верстке.

background – фон объекта, элемента, либо страницы;

Пример: background: url(http://sait.ru/images/icons.png) no-repeat #000;

Мы указали путь до картинки, а так же отменили любое ее повторение, цвет фона указали черный. Фон можно указать и без картинки, например Background-color:#000333;

Значения:

no-repeat – нет повторений изображения;

repeat x – повторяет изображение по горизонтали;

repeat y – повторяет изображение по вертикали

repeat – повторяет изображение и по вертикали и по горизонтали.

#000333...#ffffff – задает цвет фона.

Transparent – делает фон прозрачным.

color – устанавливает цвет текста. Например - color:#000444;

box-shadow – делает тень элементу. Например - box-shadow:0 0 4px #181818;

font-family – установка шрифта. Например font-family: Arial;

font-size - размер шрифта указывается в пикселах, пунктах, процентах. Например font-size:14px;

font-weight – делает шрифт жирным и имеет следующие значения - bold | bolder | lighter | normal | 100 | 200 | 300 | 400 | 500 | 600 | 700 | 800 | 900.

font-style – имеет следующие значения. normal - Обычное начертание текста, italic – курсивное, oblique – наклонное начертание.

text-decoration – добавляет следующие эффекты тексту. line-through - перечеркнутый текст, overline - линия над текстом, underline - подчеркнутый текст, none - отменяет все эффекты.

text-transform – заглавные или прописные. Capitalize - Первый символ каждого слова в предложении будет заглавным, lowercase - все символы строчные, uppercase – все символы заглавные, none - не меняет символы.

text-align – выравнивание текста по левому краю (left), середине (center), правому (right), по ширине (justify).

line-height – устанавливает межстрочный интервал. Можно использовать пиксели, дюймы, пункты, проценты. 100% это начальный интервал. Так же можно использовать auto – рассчитывается автоматически.

overflow – как отображать содержания блочного элемента, если оно целиком не помещается и выходит за область? Visible - Отображается все содержание элемента, даже за пределами, hidden - Отображается только область внутри элемента, остальное будет скрыто, scroll - добавляются полосы прокрутки, auto - полосы прокрутки добавляются только при необходимости.

border – границы, рамка блока. Например - border:1px solid #002200; Если нужна граница одной стороны, то border-top, border-bottom, border-left, border-right.

Значения: Solid – одна линия, double – двойная, dashed – пунктиром, dotted – точками.

border-radius – округление уголков рамки. Например border-radius:10px;

cursor – вид курсора мышки при наведении на элемент. Default, crosshair, help, move, pointer, progress, text, wait.

z-index - позиционированные элементы на странице могут накладываться друг на друга в определенном порядке – выше или ниже друг друга. В качестве значения используются целые числа - положительные, отрицательные и ноль. Чем больше число тем выше будет элемент. Если вместо цифр указать auto, то элементы будут выравниваться автоматически. Например – z-index: 5; или z-index:-10;

!important – если имеется несколько классов с разными значениями, то этот код указывает на приоритет конкретного значения.

Например есть несколько одинаковых классов, но они имеют разные значения. Так вот, будет использоваться то значение, у которого есть атрибут !important. Например - margin:10px!important;

Для создания Вашего первого сайта, Вам этих тегов должно вполне хватить! С приходом опыта, Вы еще узнаете много других тегов и их свойств, которых здесь нет! Возможно какие-то теги я и забыл упомянуть, но не обессудьте уж меня \$)