



[ СЛОВАРЬ ]



## Теги

Это простейшие элементы разметки вебстраницы, чтобы браузер при ее загрузке понимал, как именно отображать тот или иной элемент (текст, картинку, видео).

Все тэги имеют одинаковый формат: они начинаются знаком "<" и заканчиваются знаком ">". Обычно имеются два тэга - открывающий: <p> и закрывающий: </p>. Различие в том, что в закрывающем имеется слэш "/". Всё содержимое, помещенное между открывающим и закрывающим тэгами, является содержимым тэга.

Тег <p></p> определяет текстовый абзац

Пример тега:

```
<p>Сайт – страница в интернете с определенной информацией</p>
```

## Атрибут тега

Дополнительные свойства которые расширяют возможности тега и дают возможность форматировать тег и влиять на его отображение в браузере.

Атрибут записываются в виде сочетания: **имя атрибута="значения"**, причём текстовые значения заключаются в кавычки.

Пример:

```
<p class="text">Сайт – страница в интернете с определенной информацией</p>
```

имя атрибута — **class**;

значение атрибута — **text**;

## Emmet

Подключаемое расширение практически для всех редакторов, применяемых для веб-разработки для ускорения написания кода.

**Emmet** очень простой, но в то же время невероятно мощный и экономит много времени. Он позволяет автодополнять/автозавершать теги и атрибуты по аббревиатуре.

Принцип работы **Emmet** очень прост — пишется не весь тег, а только его имя или аббревиатура, после чего нажимается клавиша **TAB**. В результате редактор сам создает из названия весь тег.

[Шпаргалка по Emmet](#)



## Класс (class)

Специальный атрибут тега с помощью которого мы можем присвоить свое имя этому тегу и потом выбрать один или более элемент с таким именем класса, размещенный в разных местах страницы.

Чтобы выбрать тег с определенным классом в CSS перед именем класса ставиться точка .

Пример:

```
<p class="text">Сайт – страница в интернете с определенной информацией</p>
```

Для тега **p** добавили клас “text” и теперь можем выбрать этот класс в CSS и оформить его по своему желанию.

```
.text {  
  font-size: 10px; /*размер текста*/  
  font-weight: bold; /*жирный текст */  
}
```

Поставили точку и написали название класса.

## /\*комментарий\*/

Это пояснение к коду для лучшего его понимания. Смотрите картинку выше.

Начните свой комментарий, добавив /\*. Закройте комментарий, добавив \*/.

```
/* Это комментарий */  
/* Это более длинный комментарий,  
который занимает более одной строки */
```

## div

Универсальный блочный тег предназначен для группирования элементов документа. В большинстве ситуаций используется как блок-обертка для логически связанного контента.

```
<div>  
  <h1>Заголовок статьи</h1>  
  <p>Lorem ipsum dolor sit, amet consectetur adipisicing elit. Ipsam rerum  
  pariatur nihil sunt atque totam tenetur culpa neque, a repellat ut, doloremque  
  recusandae saepe, adipisci porro sit eum consectetur omnis!</p>  
</div>
```

В примере, **div** оборачивает заголовок и текст, таким образом выделяя статью.

## <body></body>

Тег **body** определяет тело документа и предназначен для хранения содержания веб-страницы, отображаемого в окне браузера. Контент, который должен отображаться на странице, следует располагать именно внутри **body**. К такому контенту относятся текст, изображения, таблицы, списки и др.

## Хедер («шапка сайта»)

Это блок в верхней страницы сайта, в котором, как правило, размещается логотип и слоган сайта, краткая контактная информация, основное горизонтальное меню и другие элементы, которые считаются наиболее важными в зависимости от специфики ресурса. Тег **<header>** содержит вводное или навигационное содержимое для раздела документа.

## <br>

Тег **<br>** устанавливает перевод строки в том месте html кода, где он находится.

## <h1>...<h6>

Эта группа тегов определяет текстовые заголовки разного уровня, которые показывают относительную важность секции, расположенной после заголовка. Обратите внимание, что тег **h1** используется только один раз, как основной заголовок страницы. **h2** до **h6**, однако, могут использоваться сколько угодно, но их всегда следует использовать по порядку, указывая важность и иерархию заголовков контента. Например, **h4** должен быть подзаголовком **h3**.

## <footer>

Футер, он же подвал сайта — это блок в нижней части страницы, куда выносят полезную, но не первостепенную информацию.

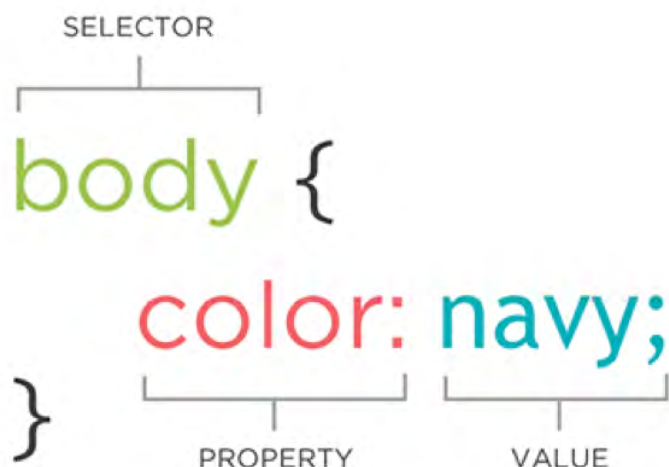
Не стоит перегружать футер лишними данными — большинство пользователей, скорее всего, не обратит на них внимания, а поисковые системы могут расценить это как методы черного продвижения и применить к сайту санкции, убрав его из поисковой выдачи.

## CSS

Это язык, используемый для описания оформления веб-страницы.

Синтаксис CSS очень простой, объявление стиля состоит из двух частей: элемента веб-страницы и команды оформления.

Вначале пишется имя элемента который мы хотим оформить, например `body`, это означает, что все стилевые параметры будут применяться к тегу `body`, затем идут фигурные скобки, в которых записывается стилевое свойство, а его значение указывается после двоеточия.



**Селектор (selector)** — указывая его, мы говорим браузеру, к чему именно хотим применить CSS правило (html-тег, id, класс и т.д.).

**Блок объявлений (declaration block)** — так называется всё содержимое, находящееся между фигурными скобками `{}` после селектора.

**Свойство (property)** — идентификатор действия, которое будет применено к элементу (например, вы хотите изменить начертание шрифта `font-weight`, цвет `color`, размер шрифта `font-size` и т. д.). После имени свойства необходимо ставить двоеточие.

**Значение (value)** — описывает, как именно свойство будет обработано браузером. Каждое свойство имеет набор допустимых значений, определенных формальными правилами, а также семантический смысл, реализованный движком браузера. Записывается после двоеточия.

## px (пиксели)

Это самый маленький элемент изображения (как атом в молекуле). PX также являются одной из множества единиц измерения размеров в CSS.

## <link>

Тег который устанавливает связь с внешним документом вроде файла со стилями или со шрифтами.

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

`link` на картинке имеет 2 атрибута, `rel` — который говорит про взаимосвязь файла `style.css` и нашего документа. `href` — атрибут который показывает где расположен файл оформления страницы.

## reset css

Каждый браузер устанавливает свои значения стилей по умолчанию для различных HTML-элементов. С помощью файла CSS Reset мы можем нивелировать эту разницу.

## height

Css свойство которое устанавливает высоту содержимого элемента. По умолчанию высота определяется автоматически, исходя из содержимого элемента, но если задать фиксированное значение высоты, то она будет установлена, несмотря на объём содержимого.

## background

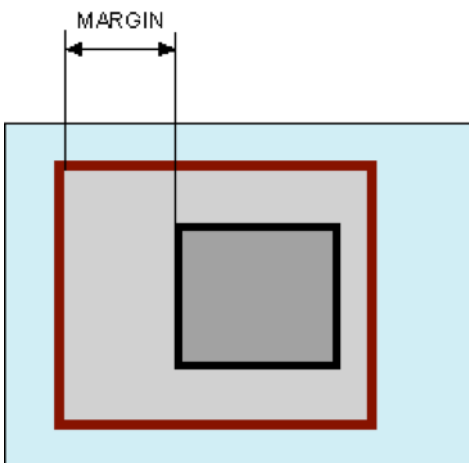
Css свойство которое устанавливает фон для выбраного html элемента.

## width

Css свойство которое устанавливает ширину содержимого элемента. По умолчанию ширина зависит от типа элемента: блочные занимают всю доступную ширину; ширина строчно-блочных равна ширине их содержимому. Свойство **width** позволяет явно задать желаемую ширину элемента, несмотря на его исходное поведение.

## margin

Css свойство которое устанавливает величину внешнего отступа от каждого края элемента. Отступом является пространство от границы текущего элемента до внутренней границы его родительского элемента.



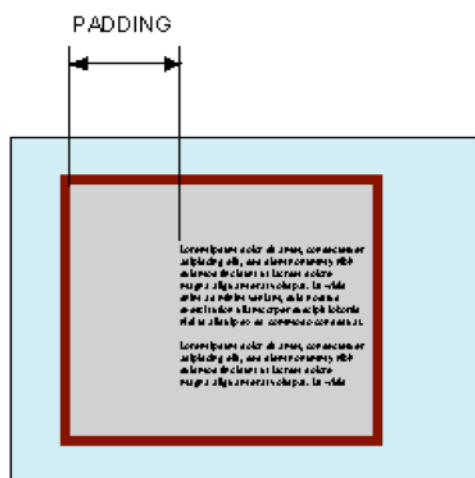
Величину отступов можно указывать в пикселях (px), процентах (%) или других допустимых для CSS единицах. Значение может быть как положительным, так и отрицательным числом.

**margin** можно записать как 4 отдельных свойства:

- `margin-bottom` — отступ снизу
- `margin-left` — отступ слева
- `margin-right` — отступ справа
- `margin-top` — отступ сверху

## padding

Css свойство которое устанавливает расстояние между основным содержимым и его границей (внутренний отступ). Внутренний отступ не может принимать отрицательных значений.



**padding** можно записать как 4 отдельных свойства:

- `padding-bottom` — внутренний отступ снизу
- `padding-left` — внутренний отступ слева
- `padding-right` — внутренний отступ справа
- `padding-top` — внутренний отступ сверху

## font-size

Css свойство которое устанавливает размер шрифта элемента.

## border-radius

Css свойство которое устанавливает радиус скругления уголков рамки. Если рамка не задана, то скругление также происходит и с фоном.

С помощью этого свойства из квадрата можно сделать круг.

Винни-Пух был всегда не прочь немного подкрепиться, в особенности часов в одиннадцать утра, потому что в это время завтрак уже давно окончился, а обед ещё и не думал начинаться. И, конечно, он страшно обрадовался, увидев, что Кролик достаёт чашки и тарелки.

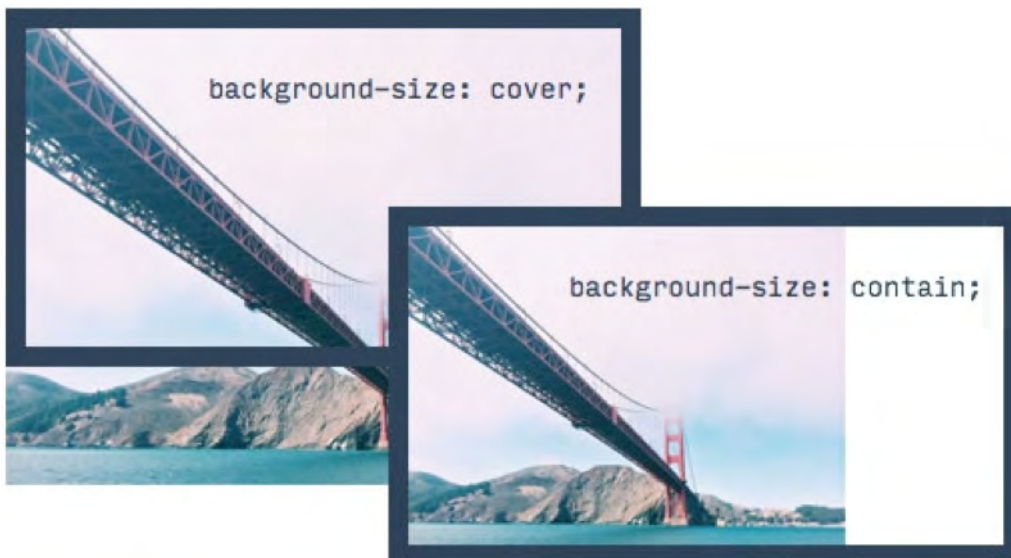


```
div {  
  background: #e4efc7;  
  border: 1px solid #333;  
  padding: 10px;  
  border-radius: 30px;  
}
```

На картинке видно как произошло скругление углов прямоугольника по 30 px с каждой стороны.

## background-size

Css свойство которое масштабирует фоновое изображение, согласно заданным размерам. Значение **cover** — масштабирует изображение с сохранением пропорций так, чтобы его ширина или высота равнялась ширине или высоте блока.



## RGBA

Формат записи цвета в CSS который основывается на сочетании трех цветов R (red — красный), G (green — зеленый), B (blue — синий), и канала прозрачности A (alpha — прозрачность).

Форма записи — `rgba(255, 255, 255, 0.5);`

## vertical-align

Css свойство которое управляет вертикальным выравниванием элементов внутри своих родителей. Оно применяется к строчным элементам и к ячейкам таблиц.



## word-wrap

Css свойство которое указывает, переносить или нет длинные слова, которые не помещаются по ширине в заданную область. Значение wrap — разрешает перенос.

## position

Задаёт режим позиционирования элементов. Значением по умолчанию является **static**, которое означает «обычное позиционирование».

Значение **relative** обозначает «относительное позиционирование». Относительно позиционированный элемент можно перемещать относительно его исходного положения. Добавление свойств left, top, right и bottom изменяет позицию элемента и сдвигает его в ту или иную сторону от первоначального расположения.

## display

Css свойство которое определяет, как элемент должен быть показан в документе.

Значения:

**block** — элемент показывается как блочный. Применение этого значения для строчных элементов, например <span>, заставляет его вести подобно блокам — происходит перенос строк в начале и в конце содержимого.

**inline** — элемент отображается как строчный. Использование блочных элементов, таких, как <div> и <p>, автоматически создаёт перенос и показывает их содержимое с новой строки. Значение **inline** отменяет эту особенность, поэтому содержимое блочных элементов начинается с того места, где окончился предыдущий элемент.

**inline-block** — это значение генерирует блочный элемент, который ведёт себя как строчный.

**GO IT**

