

# CSS

---



# Свойства границы

---

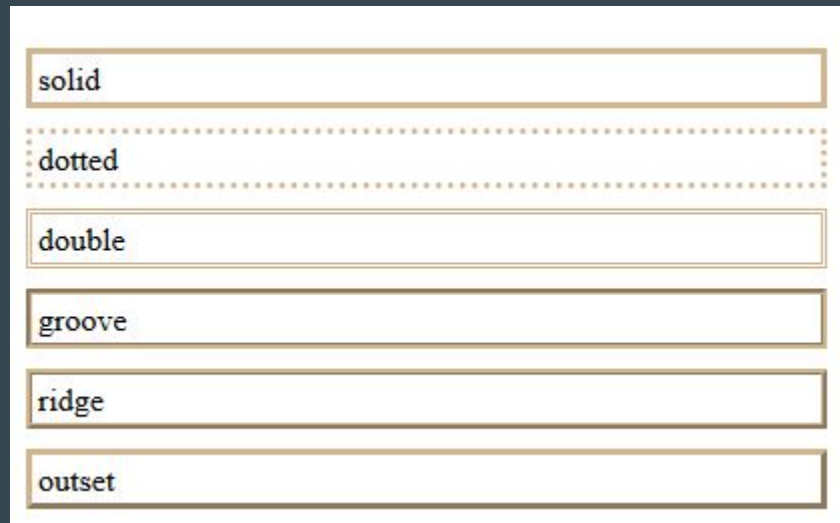
**CSS-рамка** - дает возможность задавать границы для окружающих содержимое элемента.

Рамка задается с помощью краткого свойства **border**. Стиль рамки задается с помощью трех свойств: **стиль, цвет и ширина**.

# Свойства границы: border-style

По умолчанию рамки всегда отрисовываются поверх фона элемента, фон распространяется до внешнего края элемента. Стиль рамки определяет ее отображение, без этого свойства рамки не будут видны вообще.

`border-style` - задает стиль рамки вокруг элемента



# Свойства границы: Цвет рамки border-color

Свойство задает цвет рамок всех сторон одновременно.

С помощью уточняющих свойств можно установить свой цвет для рамки каждой стороны элемента. Если для рамки цвет не задан, то он будет таким же, как и цвет текста элемента. Если в элементе нет текста, то цвет рамки будет таким же, как и цвет текста родительского элемента.

- **transparent** - Устанавливает прозрачный цвет для рамки. При этом ширина рамки остается.
- **border-color: #cacd58** - устанавливает цвет границы
- **border-color: #cacd58 #5faf8a #b9cea5 #aab238** - Одновременное перечисление четырех разных цветов для рамок элемента, только для свойства **border-color**

## Свойства границы: Ширина рамки `border-width`

---

Ширина рамки задается с помощью единиц измерения длины или ключевых слов. Если для свойства `border-style` задано значение `none`, и для рамки элемента установлена какая-то ширина, то в данном случае ширина рамки приравнивается к нулю.

`border-width: 5px;` - размер в px

`border-width: 5px 10px 15px 3px;` - возможность одновременного задания четырех разных ширин для рамок элемента, только для свойства `border-width`

# Свойства границы: Задание рамки одним свойством

---

Свойство `border` позволяет объединить в себе следующие свойства:

`border-width`, `border-style`, `border-color`, например:

```
border: 2px solid grey;
```

## Свойства границы: Граница для одной стороны элемента

---

В случае, когда необходимо задать разный стиль границ элемента, можно воспользоваться краткой записью для соответствующей границы.

Перечисленные ниже свойства объединяют в одно объявление следующие свойства: `border-width`, `border-style`, `border-color`. Перечень свойств указывается в заданном порядке, при этом одно или два значения могут быть пропущены, в этом случае их значения примут значения по умолчанию.

Стиль верхней границы задается с помощью свойства `border-top`, нижней — `border-bottom`, левой — `border-left`, правой — `border-right`.

# Свойства границы: Закругление углов

---

Свойство позволяет закруглить углы строчных и блочных элементов. Кривая для каждого угла определяется с помощью одного или двух радиусов, определяющих его форму — круга или эллипса. Радиус распространяется на весь фон, даже если элемент не имеет границ.

Свойство `border-radius` позволяет закруглить все углы одновременно, а с помощью свойств

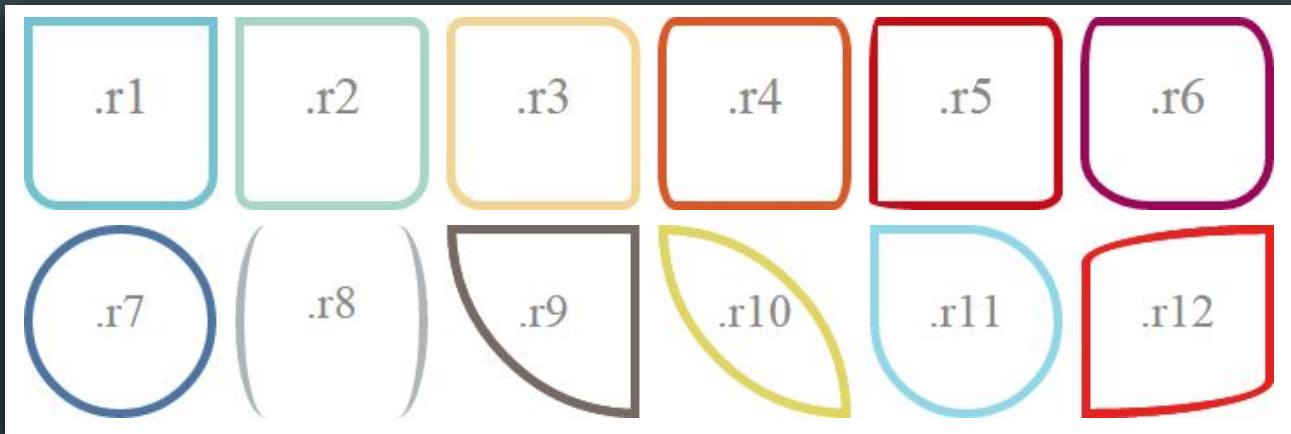
- `border-top-left-radius`,
- `border-top-right-radius`,
- `border-bottom-right-radius`,
- `border-bottom-left-radius`

можно закруглить каждый угол отдельно.



# Свойства границы:

Если задать два значения для свойства `border-radius`, то первое значение закруглить верхний левый и нижний правый угол, а второе — верхний правый и нижний левый.



# Свойства границы: Рамки-изображения `border-image`

---

Свойство позволяет устанавливать изображение в качестве рамки элемента. Основное требование, предъявляемое к изображению — оно должно быть симметричным.

Свойство включает в себя следующие значения:

`border-image: width source slice repeat outset;`

# Свойства границы:

---

Ширина рамки-изображения `border-image-width`

Свойство задает ширину изображения для границы элемента. Если ширина не задана, то по умолчанию она равна 1.

Ресурс рамки-изображения `border-image-source`

Свойство задает путь к изображению, которое будет использоваться для оформления границ блока.

- `url(url)` Относительный или абсолютный путь к изображению.

Элементы рамки-изображения `border-image-slice`

Свойство определяет размер частей изображения, используемых для оформления границ элемента и делит изображение на девять частей: четыре угла, четыре края между углами и центральную часть.

# Свойства границы: Градиентная рамка

---

Значением `border-image` может выступать не только изображение, но и градиентная заливка.

В качестве одного из цветов выступает `transparent`. Таким способом можно задавать границы сразу для всех сторон элемента или по отдельности для каждой стороны. Толщина рамки регулируется свойством `border-width`.

```
border-image : linear-gradient(to right, transparent 0%, #ADF2F7 100%);
```

# Свойства границы: outline

---

Универсальное свойство, одновременно устанавливающее цвет, стиль и толщину внешней границы на всех четырех сторонах элемента. В отличие от линии, задаваемой через `border`, свойство `outline` не влияет на положение блока и его ширину. Также нельзя задать параметры линии на отдельных сторонах элемента, `outline` применяется сразу ко всем четырём сторонам.

Отличие от `Border` в том что можно рамку отодвинуть или углубить в элемент `outline-offset` - положительное или отрицательное значение дает возможность контролировать положение рамки

# Свойства границы:

---

**default** - Зависимый от платформы курсор по умолчанию. Часто отображается как стрелка.

**none** - не отображается на элементе

**help** - Справка доступна для объекта под курсором. Часто отображается как вопросительный знак или воздушный шар.

**pointer** - Курсор-указатель, указывает на ссылку.

**text** - Указывает текст, который может быть выбран. Часто отрисовывается как вертикальная двутавровая балка.

**zoom-in** - Указывает, что что-то можно увеличивать или уменьшать, и часто визуализировать как лупа с + или - в центре для zoom-in или zoom-out соответственно.

# Цвет каретки вставки: свойство `caret-color`

---

Символ каретки является видимым индикатором точки вставки в элементе, в который пользователь вставляет текст (и, возможно, другой контент). Свойство `caret-color` контролирует цвет этого видимого индикатора.

- `caret-color: auto;`
- `caret-color: transparent;`
- `caret-color: currentColor;`
- `caret-color: red;`