



# CSS3



Цвета

# Задаем цвет

color: \_\_\_\_\_;

**Имена:** red, green, gray

**HEX:** #FF0000, #0F0, #808080

**RGB:** rgb(0..255, 0..255, 0..255) *возможен вариант RGB(50%, 34%, 90%)*

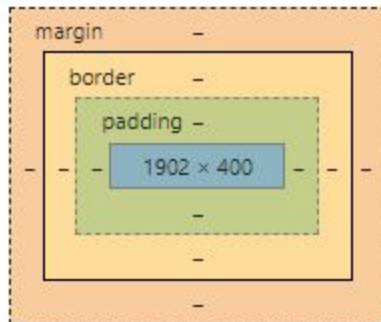
**RGBA:** rgba(0..255, 0..255, 0..255, 0..1)

**HSL:** hsl(0..360, 0..100%, 0..100%);

**HSLA:** hsla(0..360, 0..100%, 0..100%, 0..1);

# Блочная модель

# “Коробка”



# Размеры

**width** длина | проценты | *auto*

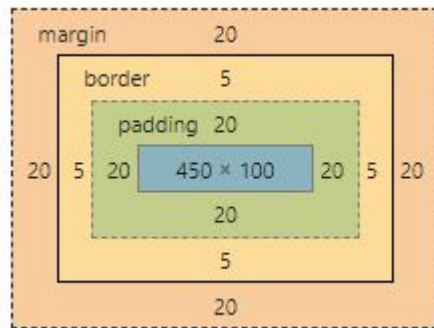
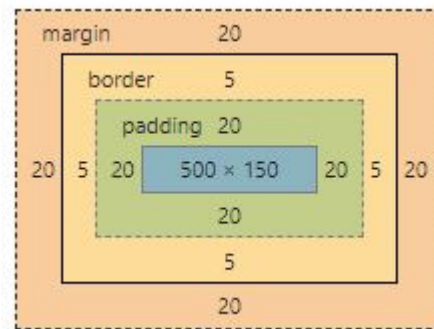
**height** длина | проценты | *auto*

**box-sizing** *content-box* | border-box

# content-box vs border-box

```
div {  
  background: #f2f5d5;  
  width: 500px;  
  height: 150px;  
  padding: 20px;  
  border: 5px solid gray;  
  margin: 20px;  
}
```

если использовать box-sizing: border-box;



# Что делать с содержимым

**overflow** *visible* | hidden | scroll | auto



# Переходы

# Что необходимо решить

- Какое свойство CSS будет меняться (transition-property) (обязательно)
- Как долго это будет происходить (transition-duration) (обязательно)
- Способ ускорения перехода (transition-timing-function)
- Будет ли пауза перед началом перехода (transition-delay)

# Пример

```
.smooth {  
    background-color: mediumblue;  
    transition-property: background-color;  
    transition-duration: 0.2s;  
}  
.smooth:hover, .smooth:focus {  
    background-color: red;  
}
```

# transition-timing-function

ease | linear | ease-in | ease-out | ease-in-out | step-start | step-end  
| steps | cubic-bezier(##,##,##,##)

<https://cubic-bezier.com/>

# Свойство transition

```
.smooth {  
    ...  
    transition: background-color 0.3s ease-in-out 0.2s;  
}
```

# Если несколько переходов

```
.smooth {  
    ...  
    transition-property: background-color, color, letter-spacing;  
    transition-duration: 0.3s, 2s, 0.3s;  
    transition-timing-function: ease-out, ease-in, ease-out;  
}  
  
.smooth:hover, .smooth:focus {  
    background-color: red;  
    letter-spacing: 3px;  
    color: black;  
}
```

# Преобразования

# Преобразования

rotate() | rotateX() | rotateY() | rotateZ() | rotate3d() |

translate() | translateX() | translateY() | scale() | scaleX() |

scaleY() | skew() | skewX() | skewY() | none



# Полезная ссылка

<https://css-transform.moro.es/>

# Анимации

# Две части анимации

1. Задание ключевых кадров с помощью правила `@keyframes`.
2. Добавление свойств анимации для элемента, который будет анимирован.

# Примеры

```
@keyframes colors {  
  0% { background-color: red; }  
  20% { background-color: orange; }  
  40% { background-color: yellow; }  
  60% { background-color: green; }  
  80% { background-color: blue; }  
  100% { background-color: purple; }  
}
```

```
@keyframe slide {  
  from { margin-left: 100% }  
  to { margin-left: 0%; }  
}
```

# Что необходимо решить

- Какую анимацию использовать (animation-name) (обязательно).
- Как долго она должна длиться (animation-duration) (обязательно).
- Способ ускорения анимации (animation-timing-function).
- Будет ли пауза перед началом анимации (animation-delay).

# Дополнительные свойства

animation-iteration-count: (целое число или infinite)

animation-direction: (normal, reverse, alternate, alternate-reverse)

animation-fill-mode: (forwards, backwards, both)

animation-play-state: (running, paused)

# Пример

```
#magic {  
  ...  
  animation-name: colors;  
  animation-duration: 5s;  
  animation-timing-function: linear;  
  animation-iteration-count: infinite;  
  animation-direction: alternate;  
}
```

```
#magic {  
  animation: colors 5s linear infinite  
  alternate;  
}
```

# Отладка в браузере

Chrome Animation Inspector

<https://developers.google.com/web/tools/chrome-devtools/inspect-styles/animations>

Firefox Animation Inspector

[https://developer.mozilla.org/en-US/docs/Tools/Page Inspector/How to/Work with animations](https://developer.mozilla.org/en-US/docs/Tools/Page_Inspector/How_to/Work_with_animations)