

«Основы HTML и CSS»

продвинутый CSS

УРОК 24. ПЕРЕХОДЫ И АНИМАЦИИ

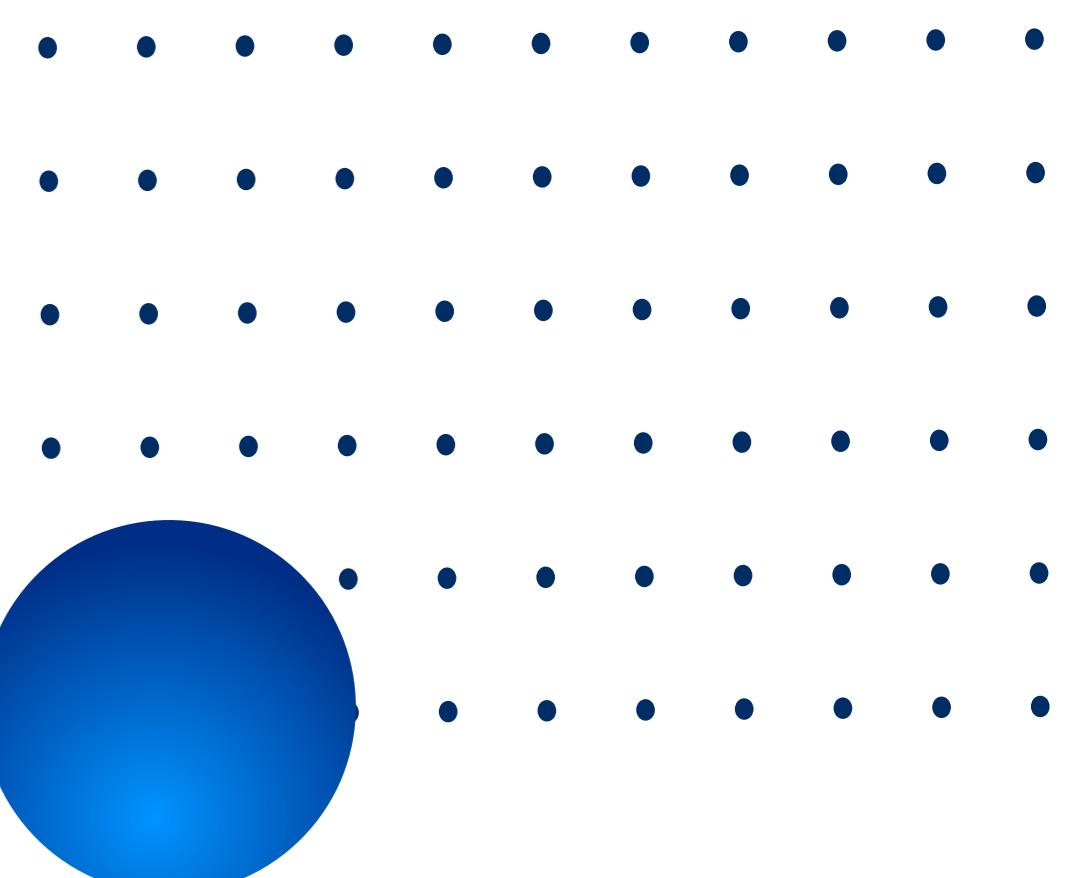


Спикер: Радимов Игорь – Frontend разработчик

ПЕРЕХОДЫ ►►►

transition: имя свойства | длительность | временная функция | задержка;

```
.button {  
    color:#fff;  
    background: linear-gradient(265.71deg, rgba(161, 69, 189, 0.8) -1.44%,  
                                rgba(55, 0, 89, 0.8) 77.83%), rgba(55, 0, 89, 1);  
    transition: color .4s, background-color 1s, box-shadow 1s;  
}  
.button:hover {  
    color:rgb(161, 69, 189);  
    background: #fff;  
    box-shadow: 0px 4px 33px #FFFFFF;  
}
```



transition-timing-function



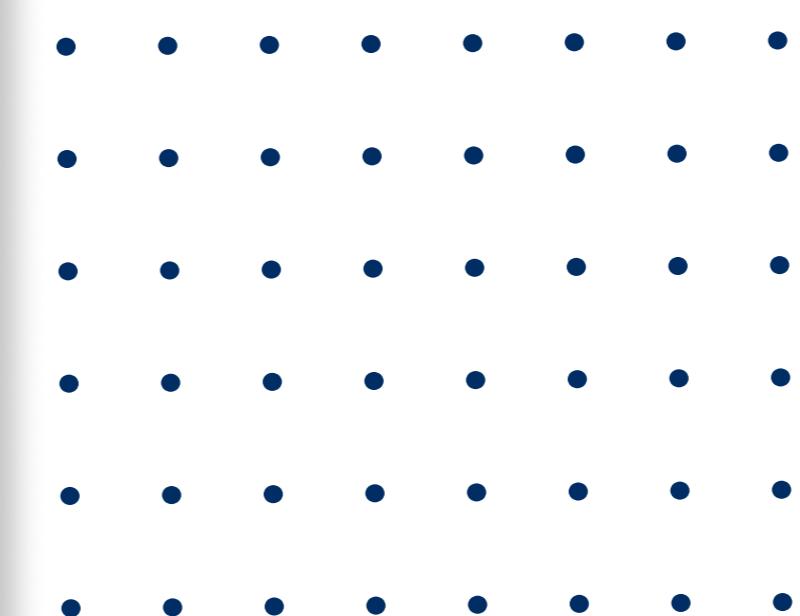
transition-timing-function: ease; /* по умолчанию */

transition-timing-function: ease-in-out;

transition-timing-function: ease-in;

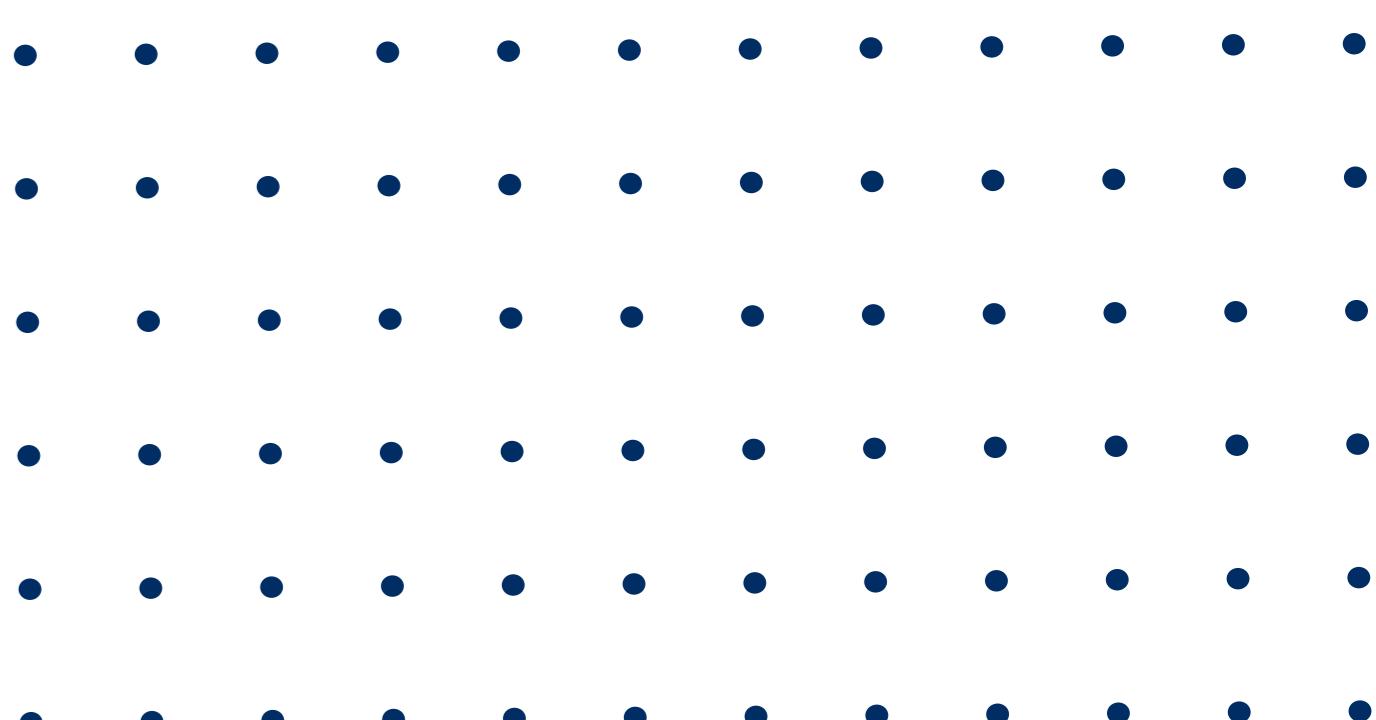
transition-timing-function: linear;

transition-timing-function: ease-out;

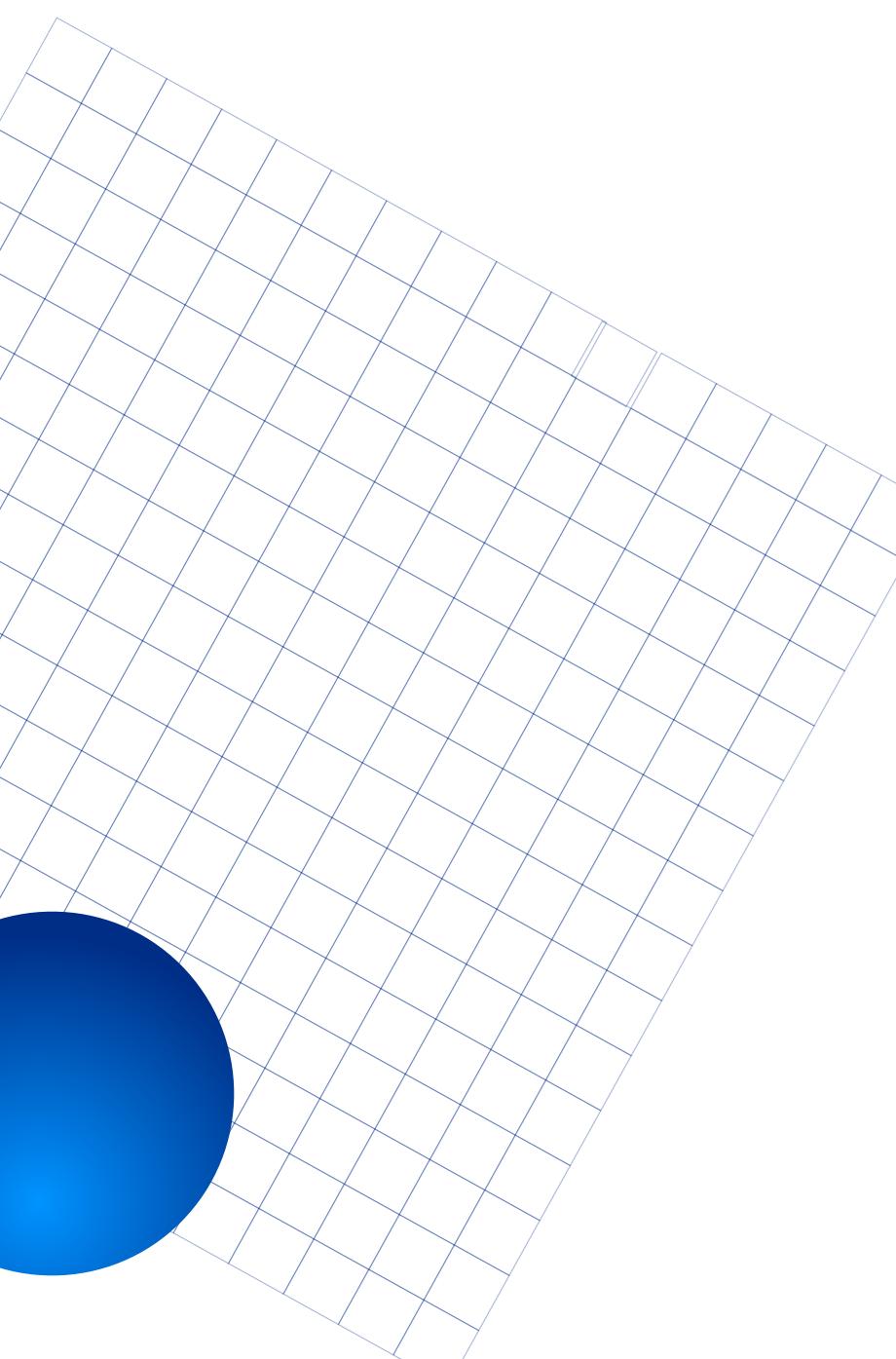


transition-timing-function: cubic-bezier(p1, p2, p3, p4);

<https://cubic-bezier.com/#.25,.1,.25,1>



ПРИМЕНЕНИЕ transition >>>



- :hover
- :active
- :checked
- :valid
- :invalid

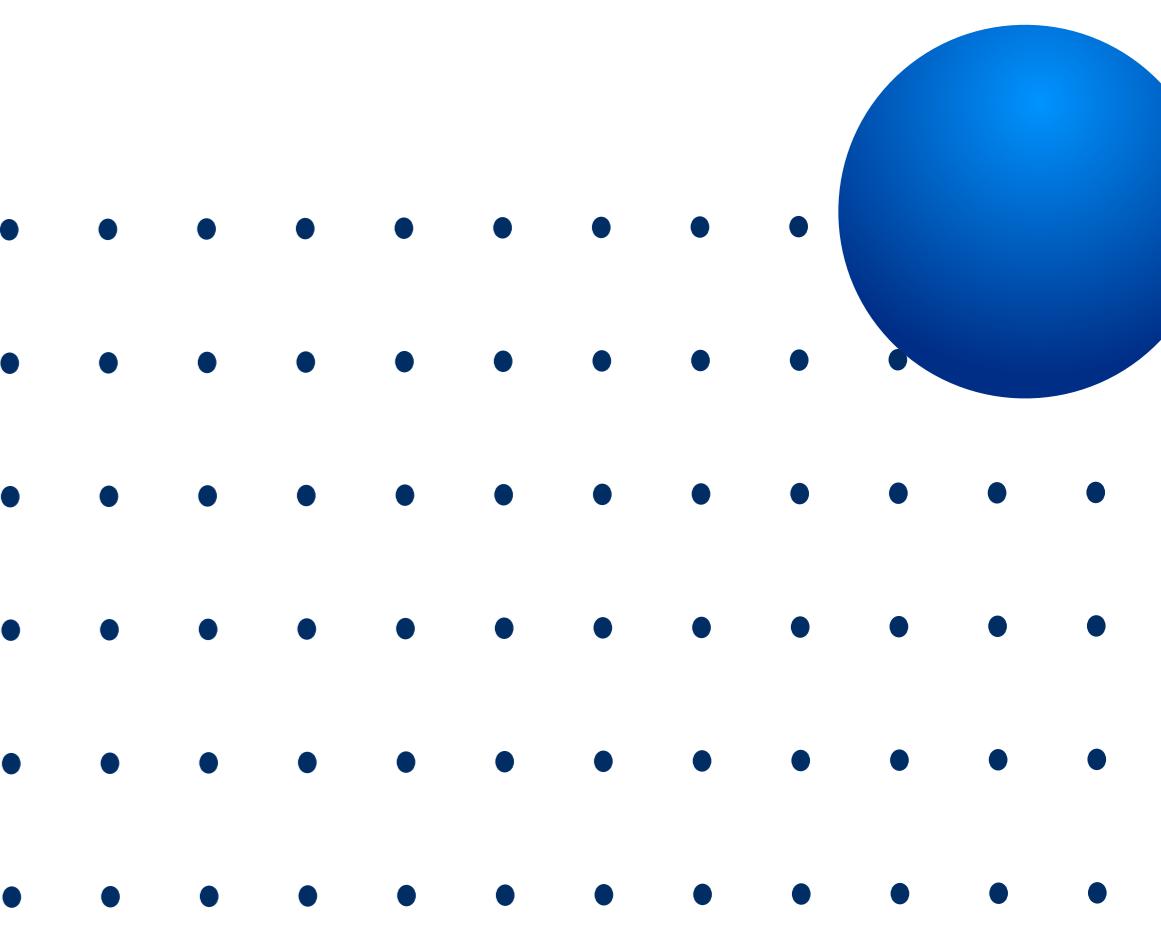
или

Дополнительный класс
элементу при помощи JS

ПРИМЕНЕНИЕ transition >>>

01 Большинство свойств с числовыми значениями

margin, padding, width, height, border, box-shadow, font-size, top, right, bottom, left и т.д.



02 Большинство свойств, указывающих цвет

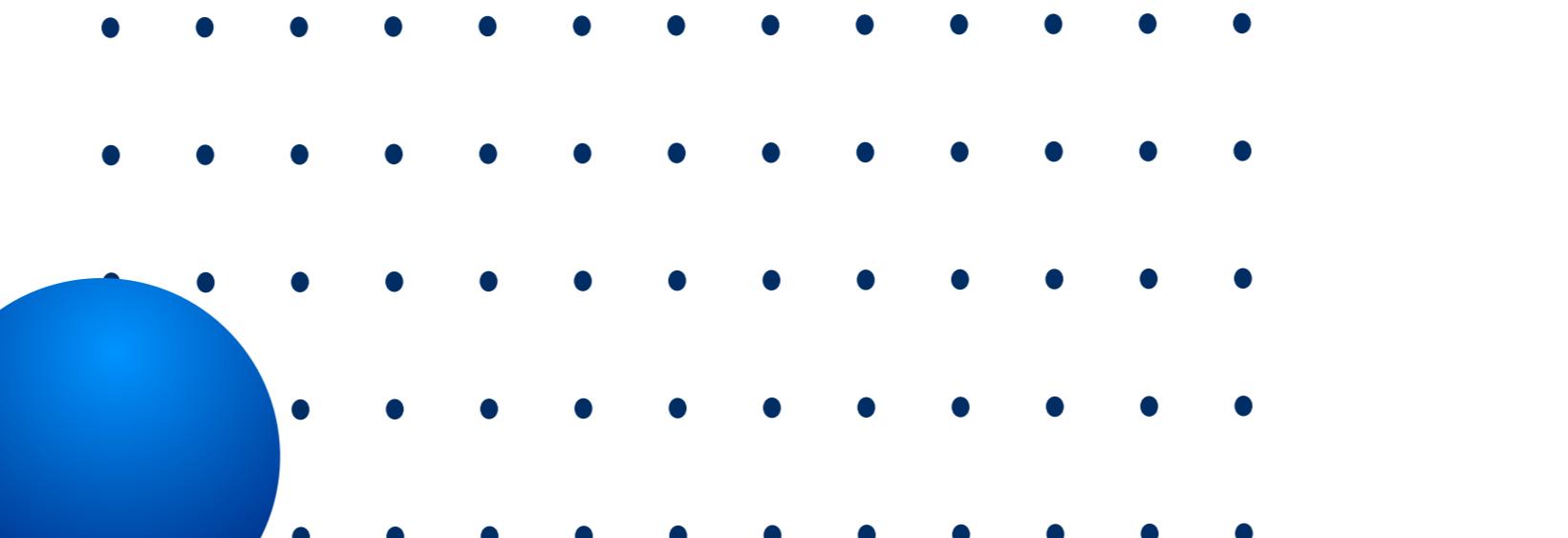
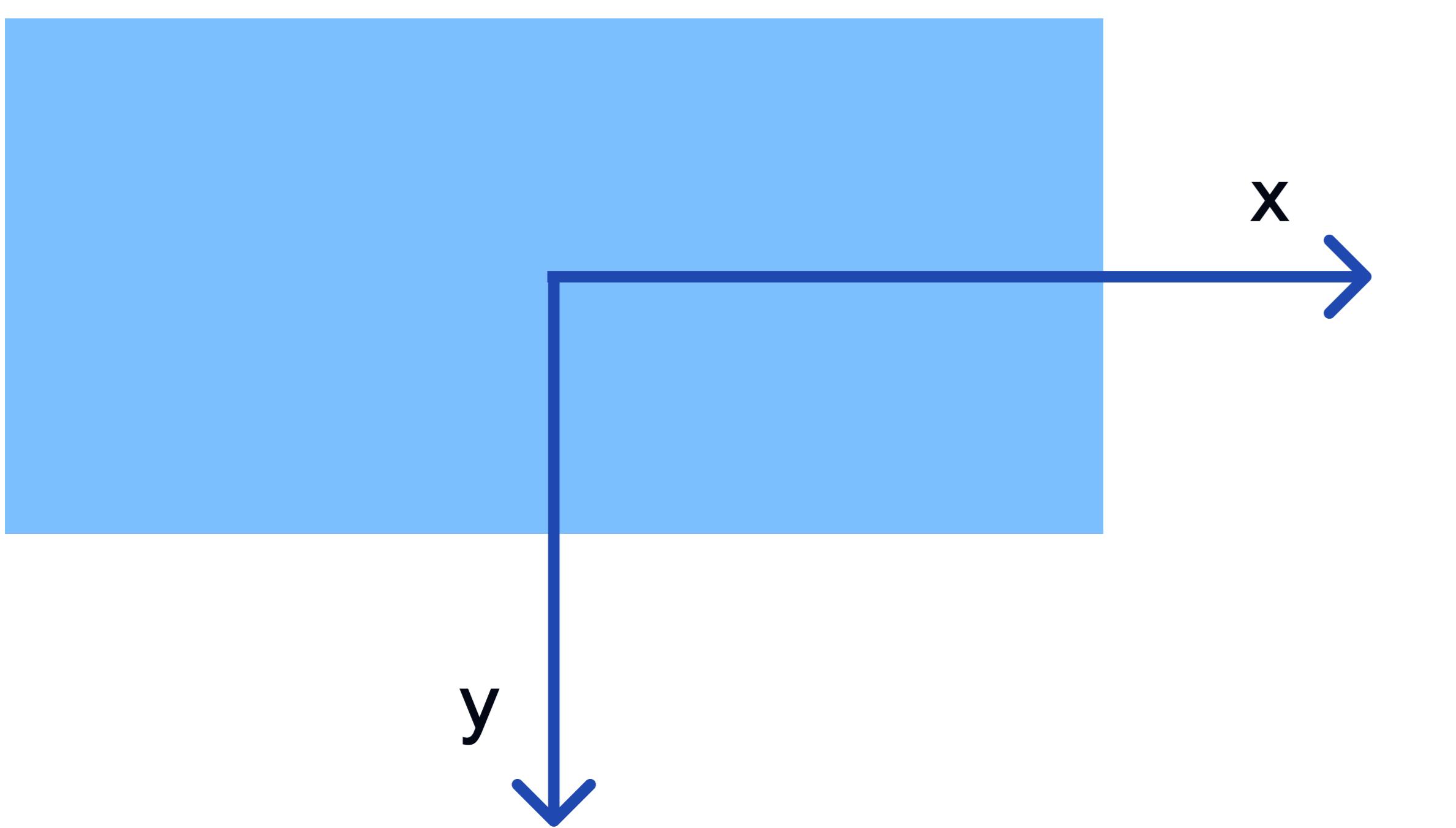
color, background-color и т.д.

03 Свойство transform

01110011001000001101001011010100011010000110100101100111101101101011001010000100110010000001110011011101101001000110111010010111001100001101000010010000011100110001101011100001110100001000000110010000011101000110101110000111

transform: translate(X, Y) ➔

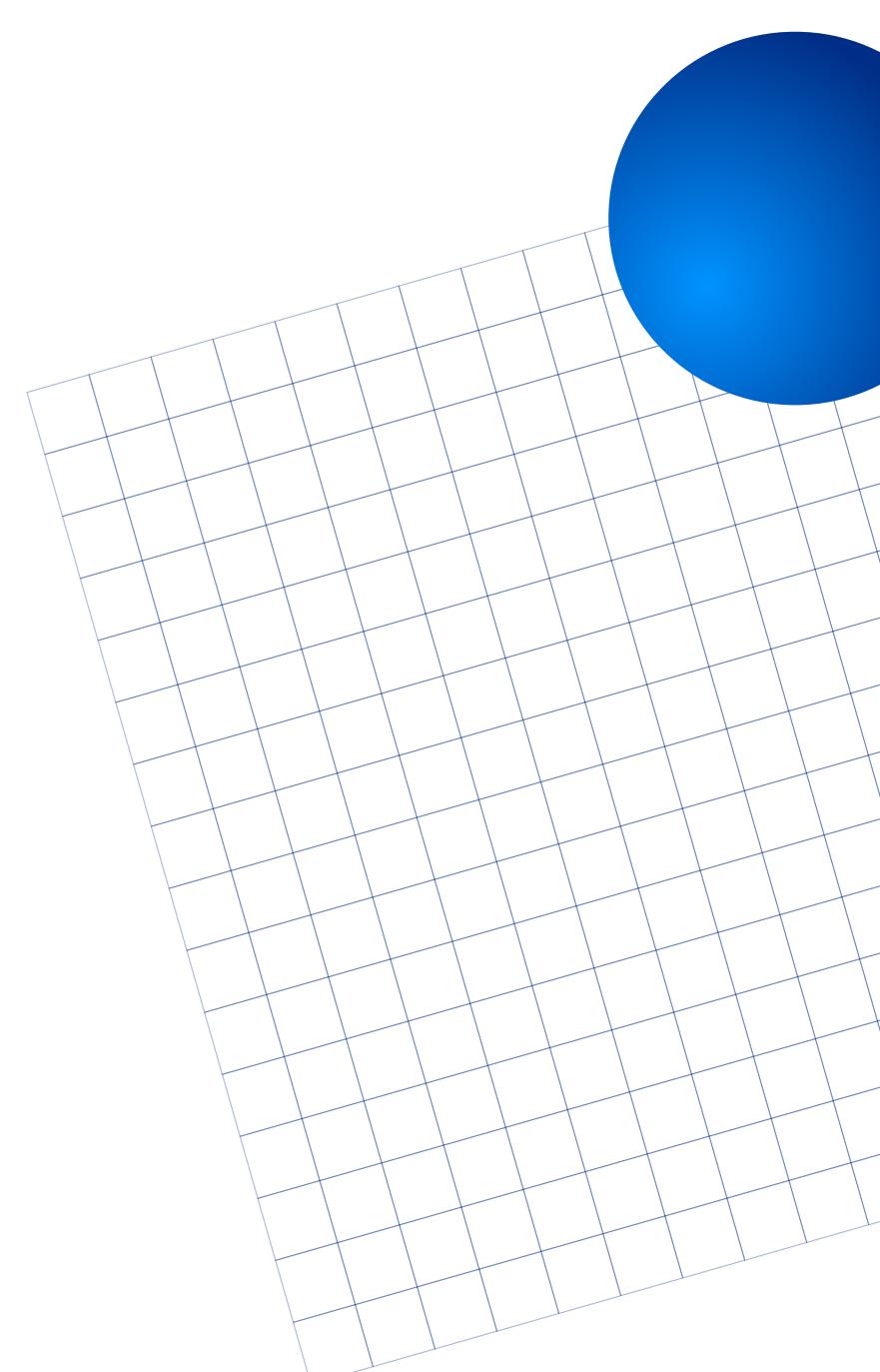
```
.button {  
  transition: transform 2s;  
}  
.button:hover {  
  transform: translate(100px,50px);  
}
```



transform: scale(X, Y) >>>

Данное свойство отвечает за **увеличение элемента**. В качестве значения можно указать увеличение по разным осям: x и y, однако чаще используется **один аргумент – одинаковое увеличение по обоим осям**. Значение может принимать любое дробное число

```
.button:hover {  
    transform: scale(1.1);  
}
```



transform: skewX(X), skewY(Y) ➔

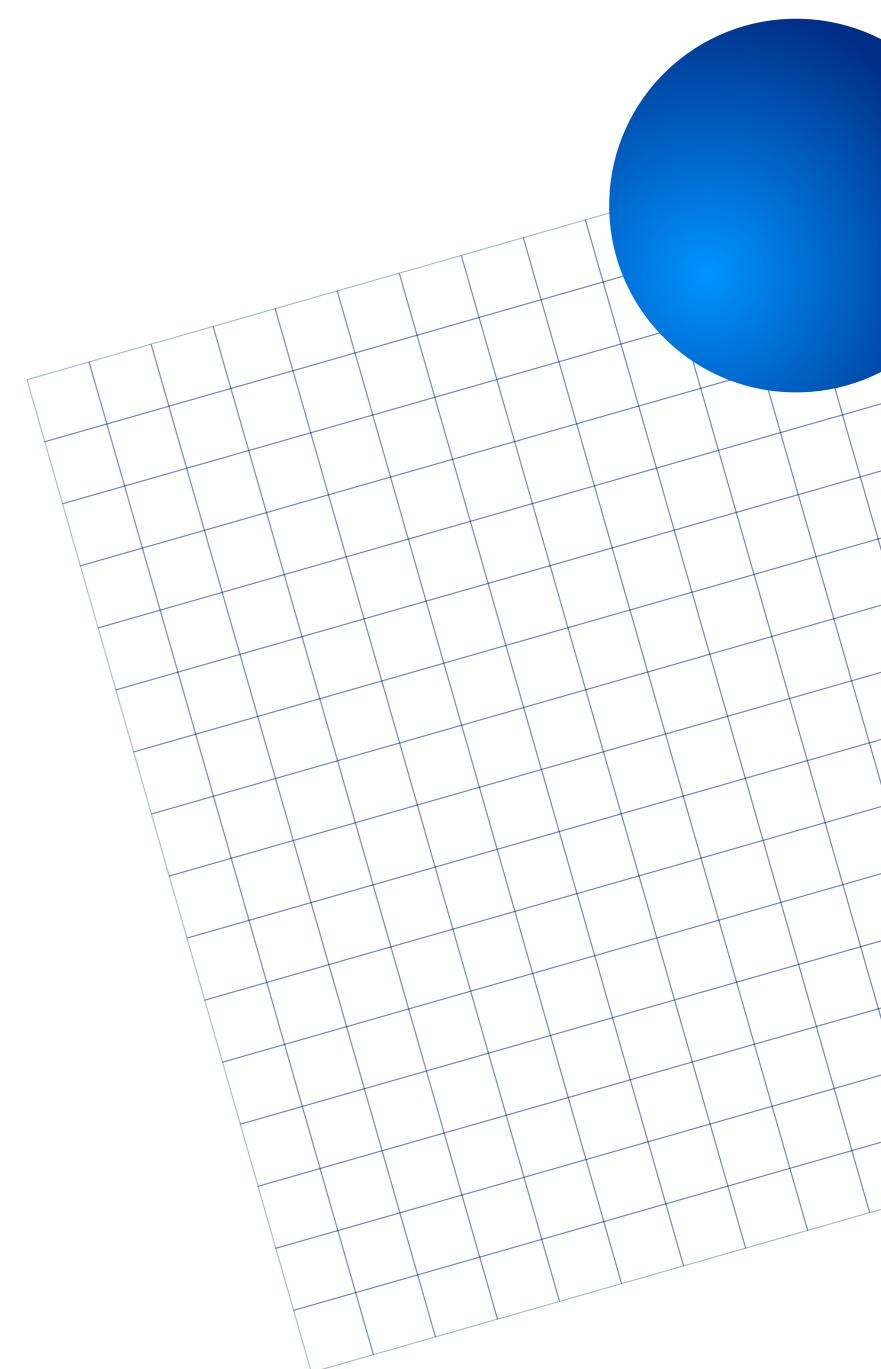
Свойства `skewX(X)` и `skewY(Y)` отвечают за наклон элемента по соответствующим осям. В качестве аргумента указывается градусная мера

```
.button:hover {  
    transform: skewX(-10deg);  
}
```

transform: rotate(z) >>>

В качестве аргумента также записываем градусную меру или аналогичные ей величины

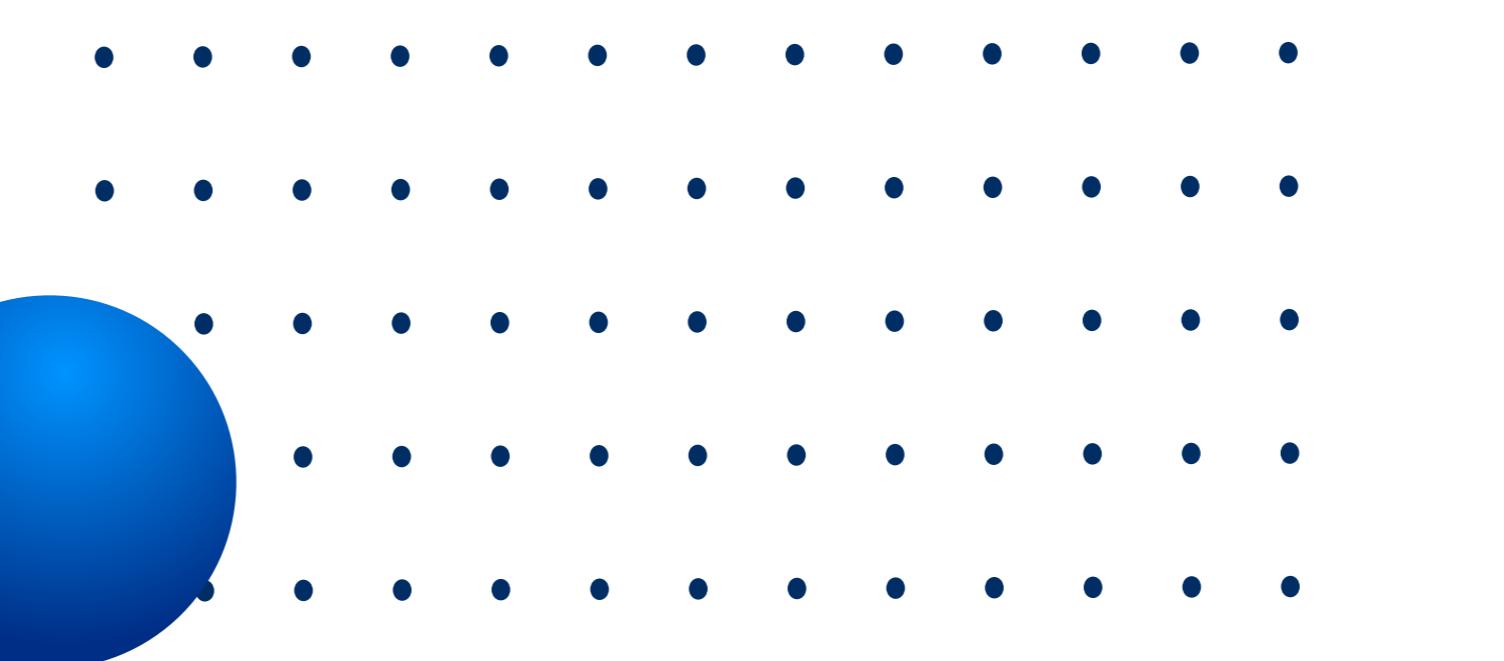
```
.button:hover {  
    transform: rotate(-360deg);  
}
```



АНИМАЦИЯ >>>

Для описания ключевых кадров используется специальная директива keyframes

```
@keyframes super-animation {  
    50% {  
        width: 50px;  
        height: 50px;  
        background-color: #2100df;  
    }  
    100% {  
        width: 200px;  
        height: 200px;  
        border-radius: 200px;  
        background-color: #ff2e2e;  
    }  
}
```



АНИМАЦИЯ >>>

```
.box1 {  
    width: 100px;  
    height: 100px;  
    background-color: #000;  
    animation-name: super-animation;  
    animation-duration: 5s; ←  
    animation-iteration-count: infinite; ←  
    animation-direction: alternate; ←  
}
```

Длительность анимации

Количество итераций

Направление анимации