

# Анимация



# АНИМАЦИЯ В CSS

- `@keyframes shadow {`
  - `from {text-shadow: 0 0 3px black;}`
  - `50% {text-shadow: 0 0 30px black;}`
  - `to {text-shadow: 0 0 3px black;}`
  - `}`
- 
- Создаём ключевые кадры при помощи `@keyframes`. `shadow` – имя анимации, даём самостоятельно. Ключевые кадры записываются либо с помощью `from`, `to`, либо с помощью `%`

- @keyframes shadow {
- from {text-shadow: 0 0 3px black;}
- 50% {text-shadow: 0 0 30px black;}
- to {text-shadow: 0 0 3px black;}
- }
- from и to – то же самое 0% и 100%

# АНИМАЦИЯ В CSS

- `animation: shadow 2s infinite ease-in-out;`
- Свойство `animation` – это упрощенная запись нескольких свойств.
- `animation-name`, `animation-duration`, `animation-iteration-count`,  
`animation-timing-function`

# ANIMATION-NAME

- Имя анимации. То, что вы пишете после @keyframes
  
- animation-name: shadow;

# ANIMATION-DURATION

- Длительность проигрывания анимации. Задаётся в секундах.
  
- `animation-duration: 1s;`

# ANIMATION-ITERATION-COUNT

- Количество итераций анимации. Задаётся числом или словом infinite(бесконечное)
  
- `animation-iteration-count: 5;`

# ANIMATION-TIMING-FUNCTION

- Способ проигрывания анимации



# ANIMATION-TIMING-FUNCTION

## animation-timing-function

Значения:

`linear` Линейная функция, анимация происходит равномерно на протяжении всего времени, без колебаний в скорости.

### функции Безье

`ease` Функция по умолчанию, анимация начинается медленно, разгоняется быстро и замедляется в конце.

`ease-in` Анимация начинается медленно, а затем плавно ускоряется в конце.

`ease-out` Анимация начинается быстро и плавно замедляется в конце.

`ease-in-out` Анимация медленно начинается и медленно заканчивается.

`cubic-bezier(x1, y1, x2, y2)` Позволяет вручную установить значения от 0 до 1. [На этом сайте](https://cubic-bezier.com/) вы сможете построить любую траекторию скорости изменения анимации.

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

# ANIMATION-TIMING-FUNCTION

## пошаговые функции

step-start

Задаёт пошаговую анимацию, разбивая анимацию на отрезки, изменения происходят в начале каждого шага. Вычисляется в `steps(1, start)`.

step-end

Пошаговая анимация, изменения происходят в конце каждого шага. Вычисляется в `steps(1, end)`.

# ANIMATION-PLAY-STATE

- `running` Анимация выполняется.
- `paused` Анимация приостанавливается.
- Можно использовать с `:hover` и в Javascript

# ANIMATION-DELAY

- Задержка перед выполнением анимации. Задаётся в секундах и миллисекундах
- При задании `animation-delay` в сокращённом виде в `animation`, необходимо указывать его после `animation-duration`

# transition

- Свойство transition состоит из задания аналогичных свойств  
transition: transition-property || transition-duration || transition-timing-function ||  
transition-delay

# Трансформация

- `transform: rotate(90deg)`

Повернет элемент на указанное количество градусов по часовой стрелке.

Отрицательные значения повернут против часовой.

- `transform: translate(20px)`

Сдвинет на 20px вправо. При указании второго значения сдвиг будет проходить вправо и вниз. Отрицательные значения сдвинут в обратную сторону.

Значения в % сдвинут на процент от размеров элемента

Трансформация не влияет на поток