1. Название свойства transition-property

Содержит название CSS-свойств, к которым будет применен эффект перехода. Значение свойства может содержать как одно свойство, так и список свойств через запятую. При создании перехода можно использовать как начальное, так и конечное состояние элемента. Свойство не наследуется.

Создаваемые эффекты должны быть ненавязчивыми. Не все свойства требуют плавного изменения во времени, что связано с пользовательским опытом. Например, при наведении на ссылку мы хотим видеть мгновенную смену цвета ссылки или цвета и стиля подчёркивания. Поэтому переходы следует использовать для тех свойств, к которым действительно нужно привлечь внимание.

|  |  |
| --- | --- |
| **transition-property** | |
| Значения: |  |
| none | Отсутствие свойства для перехода. |
| all | Значение по умолчанию. Применяет эффект перехода ко всем свойствам элемента. |
| свойство | Определяет список CSS-свойств, перечисленных через запятую, участвующих в переходе. |
| initial | Устанавливает значение свойства в значение по умолчанию. |
| inherit | Наследует значение свойства от родительского элемента. |

**Синтаксис**

div {

width: 100px;

transition-property: width;

}

div:hover {

width: 300px;

}

CSS

2. Продолжительность перехода transition-duration

Задаёт промежуток времени, в течение которого должен осуществляться переход. Если разные свойства имеют разные значения для перехода, они указываются через запятую. Если продолжительность перехода не указана, то анимация при смене значений свойств происходить не будет. Свойство не наследуется.

|  |  |
| --- | --- |
| **transition-duration** | |
| Значения: |  |
| время | Время перехода указывается в секундах или миллисекундах, например, 2s или 5ms. |
| initial | Устанавливает значение свойства в значение по умолчанию. |
| inherit | Наследует значение свойства от родительского элемента. |

**Синтаксис**

div {

transition-duration: .2s;

}

CSS

3. Функция перехода transition-timing-function

Свойство задаёт временную функцию, которая описывает скорость перехода объекта от одного значения к другому. Если вы определяете более одного перехода для элемент, например, цвет фона элемента и его положение, вы можете использовать разные функции для каждого свойства. Свойство не наследуется.

|  |  |
| --- | --- |
| **transition-timing-function** | |
| Значения: |  |
| ease | Функция по умолчанию, переход начинается медленно, разгоняется быстро и замедляется в конце. Соответствует cubic-bezier(0.25,0.1,0.25,1). |
| linear | Переход происходит равномерно на протяжении всего времени, без колебаний в скорости. Соответствует cubic-bezier(0,0,1,1). |
| ease-in | Переход начинается медленно, а затем плавно ускоряется в конце. Соответствует cubic-bezier(0.42,0,1,1). |
| ease-out | Переход начинается быстро и плавно замедляется в конце. Соответствует cubic-bezier(0,0,0.58,1). |
| ease-in-out | Переход медленно начинается и медленно заканчивается. Соответствует cubic-bezier(0.42,0,0.58,1). |
| cubic-bezier(x1, y1, x2, y2) | Позволяет вручную установить значения от 0 до 1 для кривой ускорения. [На этом сайте](https://html5book.ru/goto/http:/roblaplaca.com/examples/bezierBuilder/) вы сможете построить любую траекторию перехода. |
| initial | Устанавливает значение свойства в значение по умолчанию. |
| inherit | Наследует значение свойства от родительского элемента. |

**Синтаксис**

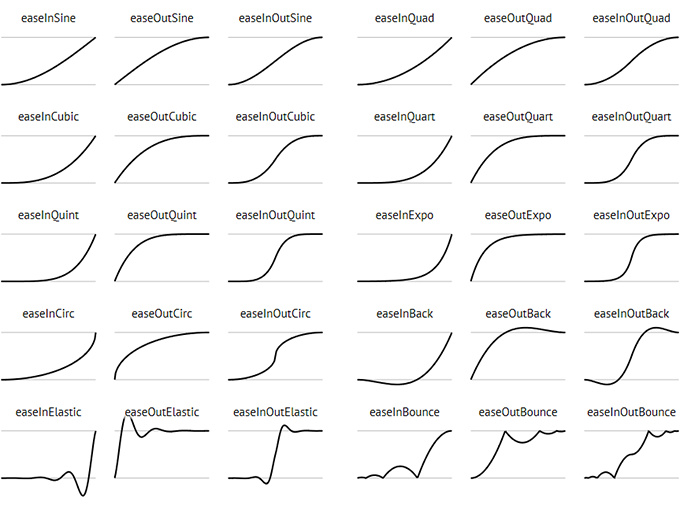
div {

transition-timing-function: linear;

}

CSS

Для создания более реалистичных анимаций используйте функцию cubic Bézier:

РИС. 1. ПОЛЬЗОВАТЕЛЬСКИЕ ФУНКЦИИ CUBIC BÉZIER С САЙТА EASINGS.NET

|  |  |
| --- | --- |
| **Пользовательское название** | **Значение функции** |
| easeInSine | cubic-bezier(0.47, 0, 0.745, 0.715) |
| easeOutSine | cubic-bezier(0.39, 0.575, 0.565, 1) |
| easeInOutSine | cubic-bezier(0.445, 0.05, 0.55, 0.95) |
| easeInQuad | cubic-bezier(0.55, 0.085, 0.68, 0.53) |
| easeOutQuad | cubic-bezier(0.25, 0.46, 0.45, 0.94) |
| easeInOutQuad | cubic-bezier(0.455, 0.03, 0.515, 0.955) |
| easeInCubic | cubic-bezier(0.55, 0.055, 0.675, 0.19) |
| easeOutCubic | cubic-bezier(0.215, 0.61, 0.355, 1) |
| easeInOutCubic | cubic-bezier(0.645, 0.045, 0.355, 1) |
| easeInQuart | cubic-bezier(0.895, 0.03, 0.685, 0.22) |
| easeOutQuart | cubic-bezier(0.165, 0.84, 0.44, 1) |
| easeInOutQuart | cubic-bezier(0.77, 0, 0.175, 1) |
| easeInQuint | cubic-bezier(0.755, 0.05, 0.855, 0.06) |
| easeOutQuint | cubic-bezier(0.23, 1, 0.32, 1) |
| easeInOutQuint | cubic-bezier(0.86, 0, 0.07, 1) |
| easeInExpo | cubic-bezier(0.95, 0.05, 0.795, 0.035) |
| easeOutExpo | cubic-bezier(0.19, 1, 0.22, 1) |
| easeInOutExpo | cubic-bezier(1, 0, 0, 1) |
| easeInCirc | cubic-bezier(0.6, 0.04, 0.98, 0.335) |
| easeOutCirc | cubic-bezier(0.075, 0.82, 0.165, 1) |
| easeInOutCirc | cubic-bezier(0.785, 0.135, 0.15, 0.86) |
| easeInBack | cubic-bezier(0.6, -0.28, 0.735, 0.045) |
| easeOutBack | cubic-bezier(0.175, 0.885, 0.32, 1.275) |
| easeInOutBack | cubic-bezier(0.68, -0.55, 0.265, 1.55) |

4. Задержка перехода transition-delay

Необязательное свойство, позволяет сделать так, чтобы изменение свойства происходило не моментально, а с некоторой задержкой. Не наследуется.

|  |  |
| --- | --- |
| **transition-delay** | |
| Значения: |  |
| время | Время задержки перехода указывается в секундах или миллисекундах. |
| initial | Устанавливает значение свойства в значение по умолчанию. |
| inherit | Наследует значение свойства от родительского элемента. |

**Синтаксис**

div {

transition-delay: .5s;

}

CSS

5. Краткая запись перехода

Все свойства, отвечающие за изменение внешнего вида элемента, можно объединить в одно свойство transition  
transition: transition-property transition-duration transition-timing-function transition-delay;

Если воспользоваться значениями по умолчанию, то запись

div {transition: 1s;}

CSS

будет эквивалентна

div {transition: all 1s ease 0s;}

CSS

6. Плавный переход нескольких свойств

Для элемента можно задать несколько последовательных переходов, перечислив их через запятую. Каждый переход можно оформить своей временной функцией.

div {transition: background 0.3s ease, color 0.2s linear;}

CSS

или

div {

transition-property: height, width, background-color;

transition-duration: 3s;

transition-timing-function: ease-in, ease, linear;

}

CSS

### 1. Функции 2D-трансформации transform

Свойство задаёт вид преобразования элемента. Свойство описывается с помощью **функций трансформации**, которые смещают элемент относительно его текущего положения на странице или изменяют его первоначальные размеры и форму. Не наследуется.

##### **Допустимые значения:**

matrix() — любое число  
translate(), translateX(), translateY() — единицы длины (положительные и отрицательные), %  
scale(), scaleX(), scaleY() — любое число  
rotate() — угол (deg, grad, rad или turn)  
skew(), skewX(), skewY() — угол (deg, grad, rad)

|  |  |
| --- | --- |
| **Функция** | **Описание** |
| none | Значение по умолчанию, означает отсутствие трансформации. Также отменяет трансформацию для элемента из группы трансформируемых элементов. |
| matrix(a, c, b, d, x, y) | Смещает элементы и задает способ их трансформации, позволяя объединить несколько функций 2D-трансформаций в одной. В качестве трансформации допустимы поворот, масштабирование, наклон и изменение положения. Значение a изменяет масштаб по горизонтали. Значение от 0 до 1 уменьшает элемент, больше 1 — увеличивает. Значение c деформирует (сдвигает) стороны элемента по оси Y, положительное значение — вверх, отрицательное — вниз. Значение b деформирует (сдвигает) стороны элемента по оси X, положительное значение — влево, отрицательное — вправо. Значение d изменяет масштаб по вертикали. Значение меньше 1 уменьшает элемент, больше 1 — увеличивает. Значение x смещает элемент по оси X, положительное — вправо, отрицательное — влево. Значение y смещает элемент по оси Y, положительное значение — вниз, отрицательное — вверх. |
| translate(x,y) | Сдвигает элемент на новое место, перемещая относительно обычного положения вправо и вниз, используя координаты X и Y, не затрагивая при этом соседние элементы. Если нужно сдвинуть элемент влево или вверх, то нужно использовать отрицательные значения. |
| translateX(n) | Сдвигает элемент относительно его обычного положения по оси X. |
| translateY(n) | Сдвигает элемент относительно его обычного положения по оси Y. |
| scale(x,y) | Масштабирует элементы, делая их больше или меньше. Значения от 0 до 1 уменьшают элемент. Первое значение масштабирует элемент по ширине, второе — по высоте. Отрицательные значения отображают элемент зеркально. |
| scaleX(n) | Функция масштабирует элемент по ширине, делая его шире или уже. Если значение больше единицы, элемент становится шире, если значение находится между единицей и нулем, элемент становится уже. Отрицательные значения отображают элемент зеркально по горизонтали. |
| scaleY(n) | Функция масштабирует элемент по высоте, делая его выше или ниже. Если значение больше единицы, элемент становится выше, если значение находится между единицей и нулем — ниже. Отрицательные значения отображают элемент зеркально по вертикали. |
| rotate(угол) | Поворачивает элементы на заданное количество градусов, отрицательные значения от -1deg до -360deg поворачивают элемент против часовой стрелки, положительные — по часовой стрелке. Значение rotate(720deg) поворачивает элемент на два полных оборота. |
| skew(x-угол,y-угол) | Используется для деформирования (искажения) сторон элемента относительно координатных осей. Если указано одно значение, второе будет определено браузером автоматически. |
| skewX(угол) | Деформирует стороны элемента относительно оси X. |
| skewY(угол) | Деформирует стороны элемента относительно оси Y. |
| initial | Устанавливает значение свойства в значение по умолчанию. |
| inherit | Наследует значение свойства от родительского элемента. |

**Синтаксис**

div {

-webkit-transform: rotate(360deg);

-ms-transform: rotate(360deg);

transform: rotate(360deg);

}

CSS

Наведите курсор мыши на блоки, чтобы посмотреть функции трансформации в действии.

### 2. Точка трансформации transform-origin

Свойство позволяет сместить центр трансформации, относительно которого происходит изменение положения/размера/формы элемента. Значение по умолчанию — center, или 50% 50%. Задаётся только для трансформированных элементов. Не наследуется.

|  |  |
| --- | --- |
| **transform-origin** | |
| Значения: |  |
| ось Х(left, center, right, длина, %) ось Y(top, center, bottom, длина, %) | Пара значений, заданная с помощью ключевых слов, единиц длины или процентов определяет, относительно какой части элемента будет происходить трансформация. Значения больше 100% увеличивают область трансформации элемента. |
| initial | Устанавливает значение свойства в значение по умолчанию. |
| inherit | Наследует значение свойства от родительского элемента. |

**Синтаксис**

div {

-webkit-transform: rotate(45deg);

-ms-transform: rotate(45deg);

transform: rotate(45deg);

-webkit-transform-origin: 20% 40%;

-ms-transform-origin: 20% 40%;

transform-origin: 20% 40%;

}

CSS

### 3. Множественные трансформации

Можно объединить несколько трансформаций одного элемента, перечислив их через пробел в порядке проявления.

div {transform: scale(1.5) rotate(-10deg);}

### 1. Ключевые кадры

Ключевые кадры используются для указания значений свойств анимации в различных точках анимации. Ключевые кадры определяют поведение одного цикла анимации; анимация может повторяться ноль или более раз.

Ключевые кадры указываются с помощью правила @keyframes, определяемого следующим образом:

@keyframes имя анимации { список правил }

CSS

Создание анимации начинается с установки **ключевых кадров** правила @keyframes. Кадры определяют, какие свойства на каком шаге будут анимированы. Каждый кадр может включать один или более блоков объявления из одного или более пар свойств и значений. Правило @keyframes содержит имя анимации элемента, которое связывает правило и блок объявления элемента.

@keyframes shadow {

from {text-shadow: 0 0 3px black;}

50% {text-shadow: 0 0 30px black;}

to {text-shadow: 0 0 3px black;}

}

CSS

Ключевые кадры создаются с помощью ключевых слов from и to (эквивалентны значениям 0% и 100%) или с помощью процентных пунктов, которых можно задавать сколько угодно. Также можно комбинировать ключевые слова и процентные пункты. Если кадры имеют одинаковые свойства и значения, их можно объединить в одно объявление:

@keyframes move {

from,

to {

top: 0;

left: 0;

}

25%,

75% {top: 100%;}

50% {top: 50%;}

}

CSS

Если 0% или 100% кадры не указаны, то браузер пользователя создает их, используя вычисляемые (первоначально заданные) значения анимируемого свойства.

Если несколько правил @keyframes определены с одним и тем же именем, сработает последнее в порядке документа, а все предыдущие проигнорируются.

После объявления правила @keyframes, мы можем ссылаться на него в свойстве animation:

h1 {

font-size: 3.5em;

color: darkmagenta;

animation: shadow 2s infinite ease-in-out;

}

CSS

Не рекомендуется анимировать нечисловые значения (за редким исключением), так как результат в браузере может быть непредсказуемым. Также не следует создавать ключевые кадры для значений свойств, не имеющих средней точки, например, для значений свойства color: pink и color: #ffffff, width: auto и width: 100px или border-radius: 0 и border-radius: 50% (в этом случае правильно будет указать border-radius: 0%).

#### **1.1. Временная функция для ключевых кадров**

Правило стиля ключевого кадра также может объявлять временную функцию, которая должна использоваться при перемещении анимации к следующему ключевому кадру.

**Пример**

@keyframes bounce {

from {

top: 100px;

animation-timing-function: ease-out;

}

25% {

top: 50px;

animation-timing-function: ease-in;

}

50% {

top: 100px;

animation-timing-function: ease-out;

}

75% {

top: 75px;

animation-timing-function: ease-in;

}

to {

top: 100px;

}

}

CSS

Пять ключевых кадров указаны для анимации с именем «bounce». Между первым и вторым ключевым кадром (то есть между 0% и 25%) используется функция замедления. Между вторым и третьим ключевым кадром (то есть между 25% и 50%) используется функция плавного ускорения. И так далее. Элемент будет перемещаться вверх по страницу на 50px, замедляясь по мере того, как он достигает своей наивысшей точки, а затем ускоряясь, когда он падает до 100px. Вторая половина анимации ведет себя аналогичным образом, но только перемещает элемент на 25px вверх по странице.

Временная функция, указанная в ключевом кадре to или 100%, игнорируется.

### 2. Название анимации: свойство animation-name

Свойство animation-name определяет список применяемых к элементу анимаций. Каждое имя используется для выбора ключевого кадра в правиле, которое предоставляет значения свойств для анимации. Если имя не соответствует ни одному ключевому кадру в правиле, нет свойств для анимации, отсутствует имя анимации, анимация не будет выполняться.

Если несколько анимаций пытаются изменить одно и то же свойство, то выполнится анимация, ближайшая к концу списка имен.

Имя анимации чувствительно к регистру, не допускается использование ключевого слова none. Рекомендуется использовать название, отражающее суть анимации, при этом можно использовать одно или несколько слов, перечисленных через дефис - или символ нижнего подчеркивания \_.

Свойство не наследуется.

|  |  |
| --- | --- |
| **animation-name** | |
| Значения: |  |
| none | Означает отсутствие анимации. Также используется, чтобы отменить анимацию элемента из группы элементов, для которых задана анимация. Значение по умолчанию. |
| имя анимации | Имя анимации, которое связывает правило @keyframes с селектором. |
| initial | Устанавливает значение свойства в значение по умолчанию. |
| inherit | Наследует значение свойства от родительского элемента. |

**Синтаксис**

animation-name: none;

animation-name: test-01;

animation-name: -sliding;

animation-name: moving-vertically;

animation-name: test2;

animation-name: test3, move4;

animation-name: initial;

animation-name: inherit;

CSS

### 3. Продолжительность анимации: свойство animation-duration

Свойство animation-duration определяет продолжительность одного цикла анимации. Задаётся в секундах s или миллисекундах ms. Если для элемента задано более одной анимации, то можно установить разное время для каждой, перечислив значения через запятую.

Свойство не наследуется.

|  |  |
| --- | --- |
| **animation-duration** | |
| Значения: |  |
| время | Указывает время, которое анимация занимает для завершения одного цикла. Отрицательные значения недействительны. Если время равно 0s, ключевые кадры анимации не действуют, но сама анимация происходит мгновенно. Значение по умолчанию 0s. |

**Синтаксис**

animation-duration: .5s;

animation-duration: 200ms;

animation-duration: 2s, 10s;

animation-duration: 15s, 30s, 200ms;

CSS

### 4. Временная функция: свойство animation-timing-function

Свойство animation-timing-function описывает, как будет развиваться анимация между каждой парой ключевых кадров. Во время задержки анимации временные функции не применяются.

Свойство не наследуется.

|  |  |
| --- | --- |
| **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. Соответствует cubic-bezier(0.25,0.1,0.25,1). |
| ease-in | Анимация начинается медленно, а затем плавно ускоряется в конце. Соответствует cubic-bezier(0.42,0,1,1). |
| ease-out | Анимация начинается быстро и плавно замедляется в конце. Соответствует cubic-bezier(0,0,0.58,1). |
| ease-in-out | Анимация медленно начинается и медленно заканчивается. Соответствует cubic-bezier(0.42,0,0.58,1). |
| cubic-bezier(x1, y1, x2, y2) | Позволяет вручную установить значения от 0 до 1. [На этом сайте](https://html5book.ru/goto/http:/roblaplaca.com/examples/bezierBuilder/) вы сможете построить любую траекторию скорости изменения анимации. |
| **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). |
| steps(количество шагов,положение шага) | Ступенчатая временная функция, которая принимает два параметра. Первый параметр указывает количество интервалов в функции. Это должно быть положительное целое число больше 0, если вторым параметром не является jump-none — в этом случае оно должно быть положительным целым числом больше 1. Второй параметр, который является необязательным, указывает позицию шага — момент, в котором начинается анимация, используя одно из следующих значений:   * jump-start — первый шаг происходит при значении 0 * jump-end — последний шаг происходит при значении 1 * jump-none — все шаги происходят в пределах диапазона (0, 1) * jump-both — первый шаг происходит при значении 0, последний — при значении 1 * start — ведет себя как jump-start * end — ведет себя как jump-end   Со значением start анимация начинается в начале каждого шага, со значением end — в конце каждого шага с задержкой. Задержка вычисляется как результат деления времени анимации на количество шагов. Если второй параметр не указан, используется значение по умолчанию end. |
| initial | Устанавливает значение свойства в значение по умолчанию. |
| inherit | Наследует значение свойства от родительского элемента. |

**Синтаксис**

animation-timing-function: ease;

animation-timing-function: ease-in;

animation-timing-function: ease-out;

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

animation-timing-function: linear;

animation-timing-function: step-start;

animation-timing-function: step-end;

animation-timing-function: cubic-bezier(0.1, 0.7, 1.0, 0.1);

animation-timing-function: steps(4, end);

animation-timing-function: ease, step-start, cubic-bezier(0.1, 0.7, 1.0, 0.1);

animation-timing-function: initial;

animation-timing-function: inherit;

CSS

С помощью пошаговой анимации можно создавать интересные эффекты, например, печатающийся текст или индикатор загрузки.

### 5. Повтор анимации: свойство animation-iteration-count

Свойство animation-iteration-count указывает, сколько раз проигрывается цикл анимации. Начальное значение 1 означает, что анимация будет воспроизводиться от начала до конца один раз. Это свойство часто используется в сочетании со значением alternate свойства animation-direction, которое заставляет анимацию воспроизводиться в обратном порядке в альтернативных циклах.

Свойство не наследуется.

|  |  |
| --- | --- |
| **animation-iteration-count** | |
| Значения: |  |
| infinite | Анимация проигрывается бесконечно. |
| число | Анимация будет повторяться указанное количество раз. Если число не является целым числом, анимация закончится в середине последнего цикла. Отрицательные числа недействительны. Значение 0 вызывает мгновенное срабатывание анимации. |

**Синтаксис**

animation-iteration-count: infinite;

animation-iteration-count: 3;

animation-iteration-count: 2.5;

animation-iteration-count: 2, 0, infinite;

CSS

### 6. Направление анимации: свойство animation-direction

Свойство animation-direction определяет, должна ли анимация воспроизводиться в обратном порядке в некоторых или во всех циклах. Когда анимация воспроизводится в обратном порядке, временные функции также меняются местами. Например, при воспроизведении в обратном порядке функция ease-in будет вести себя как ease-out.

Свойство не наследуется.

|  |  |
| --- | --- |
| **animation-direction** | |
| Значения: |  |
| normal | Все повторы анимации воспроизводятся так, как указано. Значение по умолчанию. |
| reverse | Все повторы анимации воспроизводятся в обратном направлении от того, как они были определены. |
| alternate | Каждый нечетный повтор цикла анимации воспроизводятся в нормальном направлении, каждый четный повтор воспроизводится в обратном направлении. |
| alternate-reverse | Каждый нечетный повтор цикла анимации воспроизводятся в обратном направлении, каждый четный повтор воспроизводится в нормальном направлении. |
| initial | Устанавливает значение свойства в значение по умолчанию. |
| inherit | Наследует значение свойства от родительского элемента. |

Чтобы определить, является ли повтор цикла анимации четной или нечетной — количество повторов начинается с 1.

**Синтаксис**

animation-direction: normal;

animation-direction: reverse;

animation-direction: alternate;

animation-direction: alternate-reverse;

animation-direction: normal, reverse;

animation-direction: alternate, reverse, normal;

animation-direction: initial;

animation-direction: inherit;

CSS

### 7. Проигрывание анимации: свойство animation-play-state

Свойство animation-play-state определяет, будет ли анимация запущена или приостановлена. Остановка анимации внутри цикла возможна через использование этого свойства в скрипте JavaScript. Также можно останавливать анимацию при наведении курсора мыши на объект — состояние :hover.

Свойство не наследуется.

|  |  |
| --- | --- |
| **animation-play-state** | |
| Значения: |  |
| running | Анимация выполняется. Значение по умолчанию. |
| paused | Анимация приостанавливается. При перезапуске анимация начинается с того места, где она была остановлена, как если бы «часы», управляющие анимацией, остановились и снова запустились. Если анимация остановлена во время задержки, при повторном воспроизведении время задержки также возобновляется. |
| initial | Устанавливает значение свойства в значение по умолчанию. |
| inherit | Наследует значение свойства от родительского элемента. |

**Синтаксис**

animation-play-state: running;

animation-play-state: paused;

animation-play-state: paused, running, running;

animation-play-state: initial;

animation-play-state: inherit;

CSS

### 8. Задержка анимации: свойство animation-delay

Свойство animation-delay определяет, когда анимация начнется. Задается в секундах s или миллисекундах ms.

Свойство не наследуется.

|  |  |
| --- | --- |
| **animation-delay** | |
| Значения: |  |
| время | Время определяет длительность задержки между началом анимации (когда анимация применяется к элементу через свойства) и когда она начинает выполняться. Отрицательные значения разрешены, такая задержка начинает анимацию с определенного момента внутри её цикла, т.е. со времени, указанного в задержке. Это позволяет применять анимацию к нескольким элементам со сдвигом фазы, изменяя лишь время задержки. Чтобы анимация началась с середины, нужно задать отрицательную задержку, равную половине времени, установленном в animation-duration. Значение по умолчанию 0s. |

**Синтаксис**

animation-delay: 5s;

animation-delay: 3s, 10ms;

CSS

### 9. Состояние элемента до и после воспроизведения анимации: свойство animation-fill-mode

Свойство animation-fill-mode определяет, какие значения применяются анимацией вне времени ее выполнения. Когда анимация завершается, элемент возвращается к своих исходным стилям. По умолчанию анимация не влияет на значения свойств, когда анимация применяется к элементу — animation-name и animation-delay. Кроме того, по умолчанию анимация не влияет на значения свойств после ее завершения — animation-duration и animation-iteration-count. Свойство animation-fill-mode может переопределить это поведение.

Свойство не наследуется.

|  |  |
| --- | --- |
| **animation-fill-mode** | |
| Значения: |  |
| none | Значение по умолчанию. Состояние элемента не меняется до или после воспроизведения анимации. |
| forwards | После того, как анимация заканчивается (как определено значением animation-iteration-count), анимация будет применять значения свойств к моменту окончания анимации. Если animation-iteration-count больше нуля, применяются значения для конца последней завершенной итерации анимации (а не значения для начала итерации, которое будет следующим). Если значение animation-iteration-count равно нулю, применяемыми значениями будут те, которые начнут первую итерацию (так же, как и в режиме animation-fill-mode: backwards;). |
| backwards | В течение периода, определенного с помощью animation-delay, анимация будет применять значения свойств, определенные в ключевом кадре, которые начнут первую итерацию анимации. Это либо значения ключевого кадра from (когда animation-direction: normal или animation-direction: alternate), либо значения ключевого кадра to (когда animation-direction: reverse или animation-direction: alternate). |
| both | Позволяет оставлять элемент в первом ключевом кадре до начала анимации (игнорируя положительное значение задержки) и задерживать на последнем кадре до конца последней анимации. |

**Синтаксис**

animation-fill-mode: none;

animation-fill-mode: forwards;

animation-fill-mode: backwards;

animation-fill-mode: both;

animation-fill-mode: none, backwards;

animation-fill-mode: both, forwards, none;

CSS

### 10. Краткая запись анимации: свойство animation

Все параметры воспроизведения анимации можно объединить в одном свойстве — animation, перечислив их через пробел:  
animation: animation-name animation-duration animation-timing-function animation-delay animation-iteration-count animation-direction;

Для воспроизведения анимации достаточно указать только два свойства — animation-name и animation-duration, остальные свойства примут значения по умолчанию. Порядок перечисления свойств не имеет значения, единственное, время выполнения анимации animation-duration обязательно должно стоять перед задержкой animation-delay.

### 11. Множественные анимации

Для одного элемента можно задавать несколько анимаций, перечислив их названия через запятую:

div {animation: shadow 1s ease-in-out 0.5s alternate, move 5s linear 2s;}