

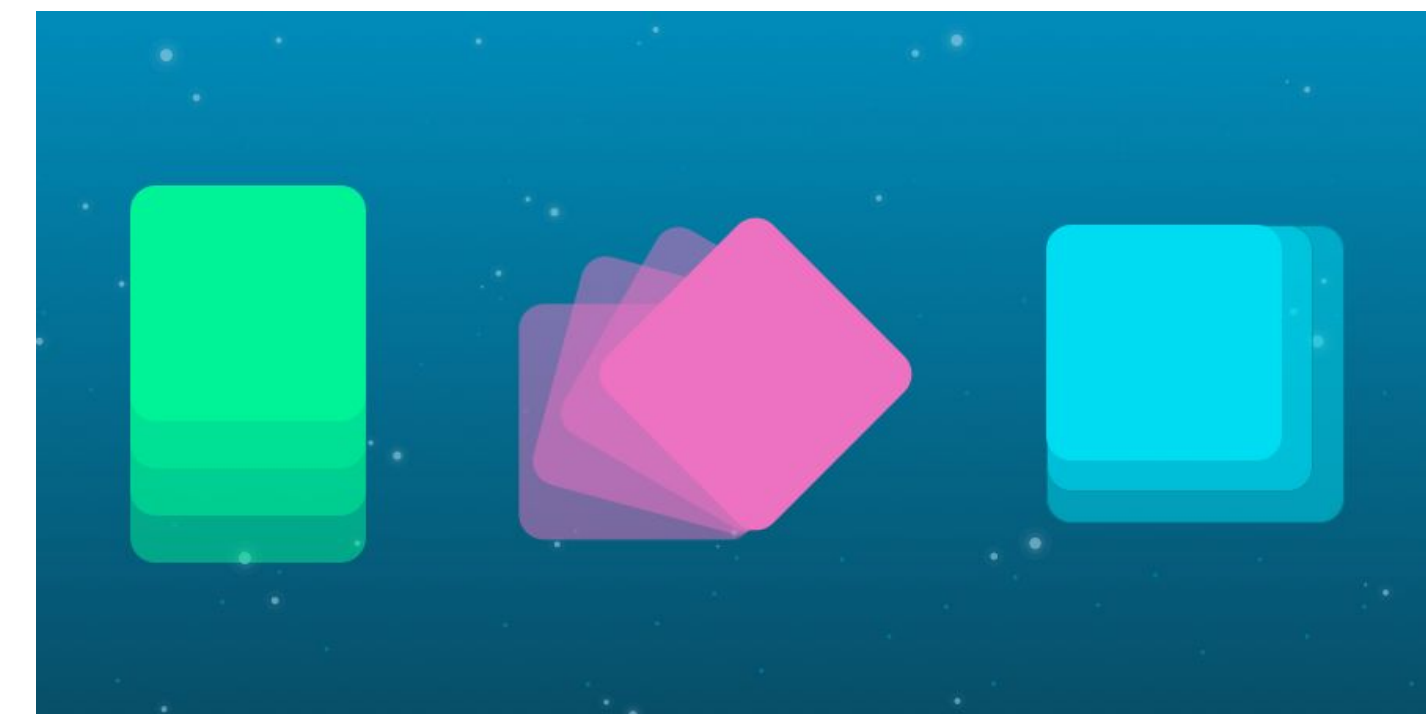
Анимация в CSS3

Трансформации

CSS3-трансформации позволяют сдвигать, поворачивать и масштабировать элементы.

Трансформации преобразовывают элемент, не затрагивая остальные элементы веб-страницы (другие элементы не сдвигаются относительно него).

Трансформации задаются css-свойством `transform` .



Виды CSS3-трансформаций:

- 2D - преобразовывают элементы в двумерном пространстве с помощью 2D-матрицы преобразований (преобразования работают по осям x и y)
- 3D - преобразовывают элементы в трехмерном пространстве (преобразования работают по осям x и y и оси z , которая задает глубину)

Трансформации

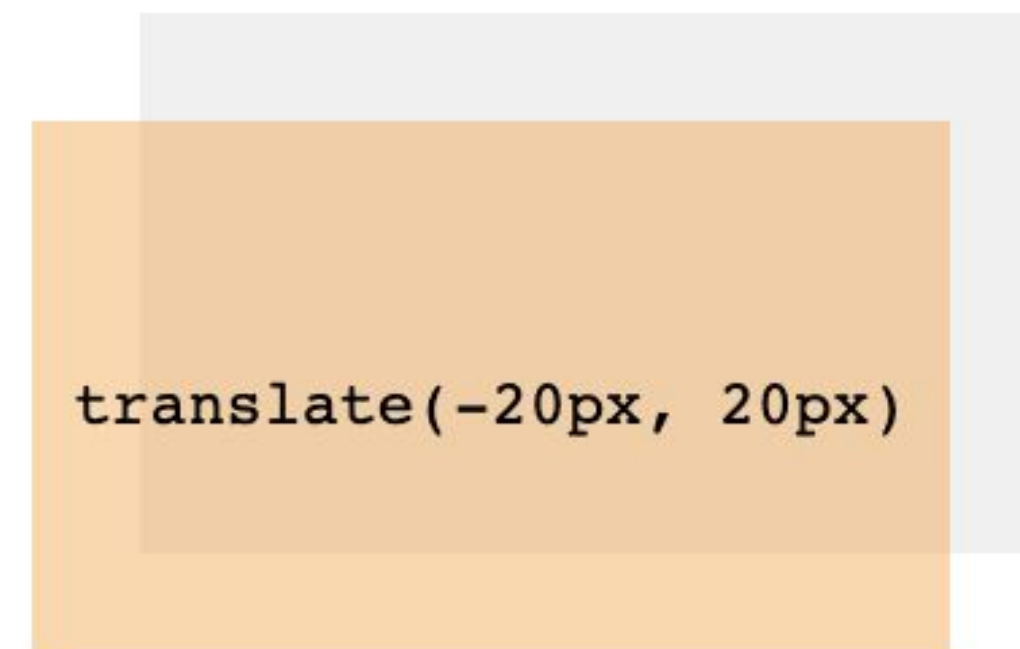
Виды 2D-трансформаций (функции, которые передаются в значении свойства transform):

- `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.

Значения расстояния - основные единицами длины (часто применяются px и %).

Если нужно сдвинуть элемент влево или вверх, то нужно использовать отрицательные значения.

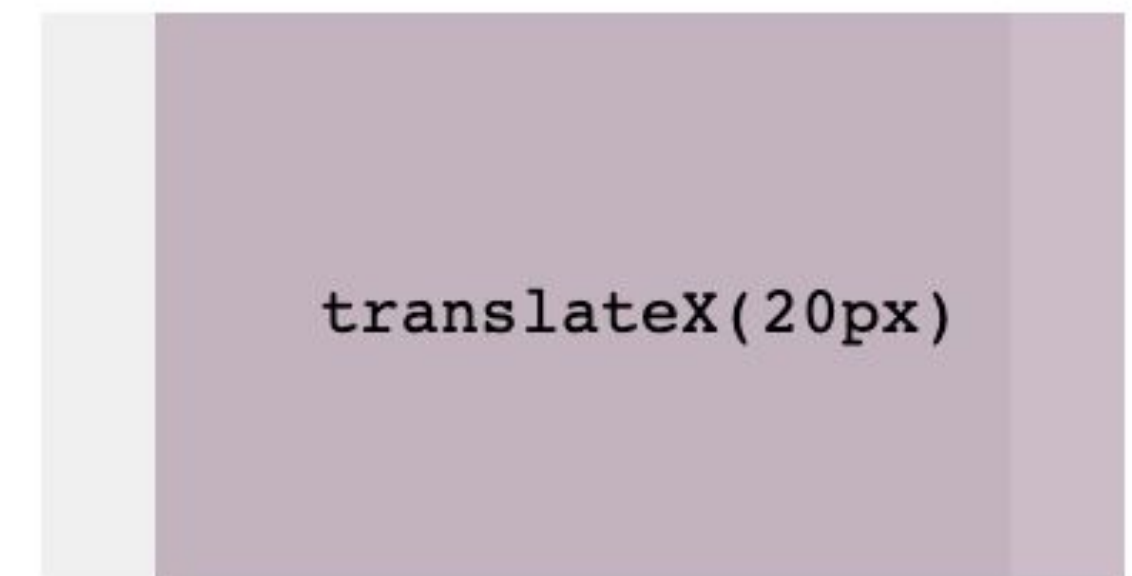


Трансформации

Виды 2D-трансформаций (функции, которые передаются в значении свойства transform):

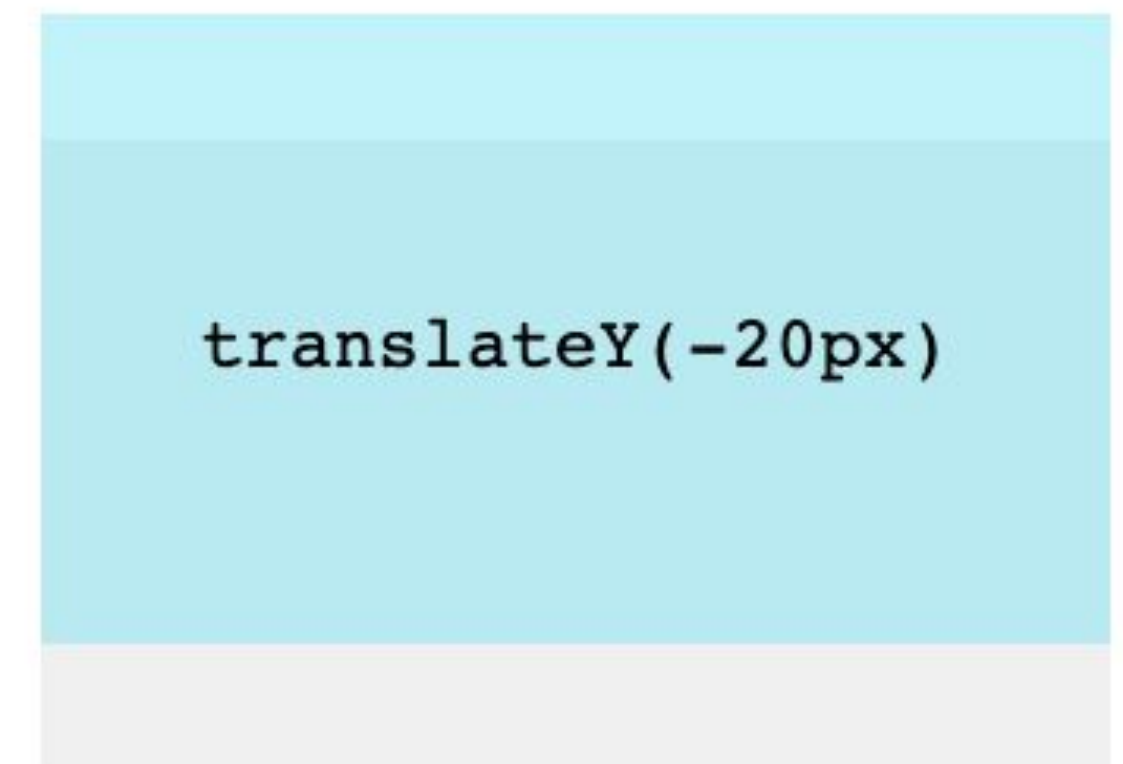
- `translateX(x)`

Сдвигает элемент относительно его обычного положения по оси X.



- `translateY(y)`

Сдвигает элемент относительно его обычного положения по оси Y.



Трансформации

Виды 2D-трансформаций (функции, которые передаются в значении свойства transform):

- `scale(x, y)`

Масштабирует элементы, делая их больше или меньше. Значения от 0 до 1 уменьшают элемент, больше 1 - увеличивают. Первое значение масштабирует элемент по ширине, второе — по высоте. Отрицательные значения отображают элемент зеркально.



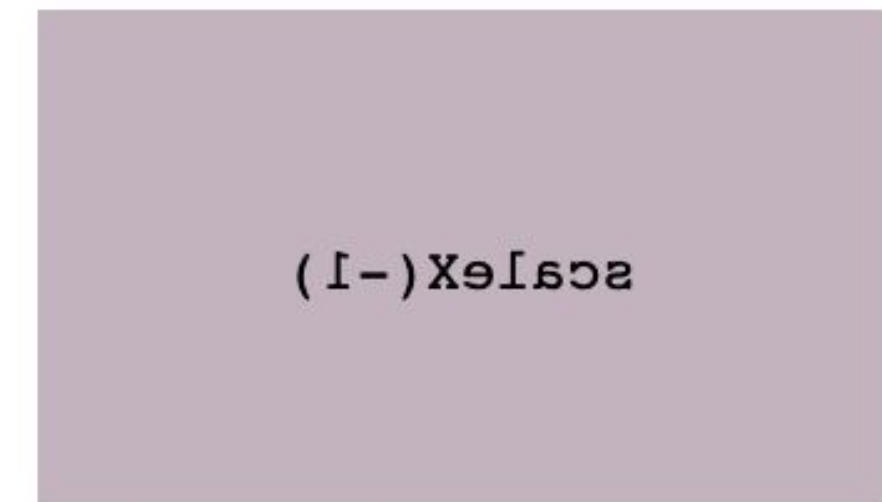
```
scale(1.2, 0.5)
```

Трансформации

Виды 2D-трансформаций (функции, которые передаются в значении свойства transform):

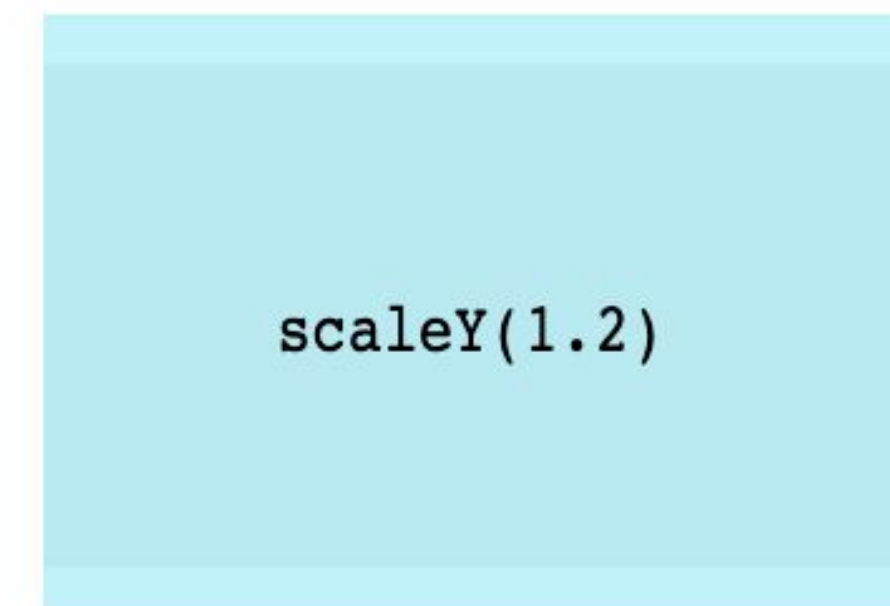
- `scaleX(x)`

Масштабирует элемент по ширине, делая его шире или уже. Отрицательные значения отображают элемент зеркально по горизонтали.



- `scaleY(y)`

Масштабирует элемент по высоте, делая его выше или ниже. Отрицательные значения отображают элемент зеркально по вертикали.

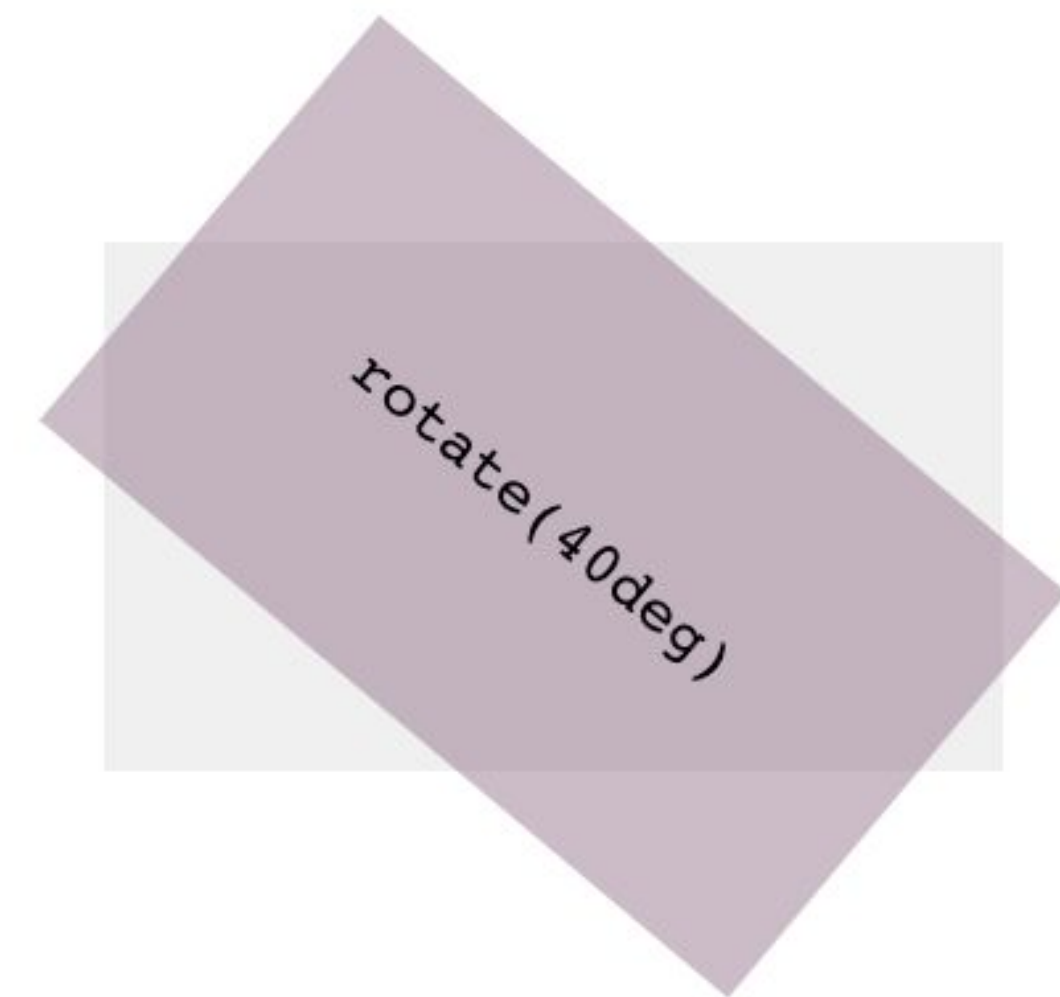


Трансформации

Виды 2D-трансформаций (функции, которые передаются в значении свойства transform):

- 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.

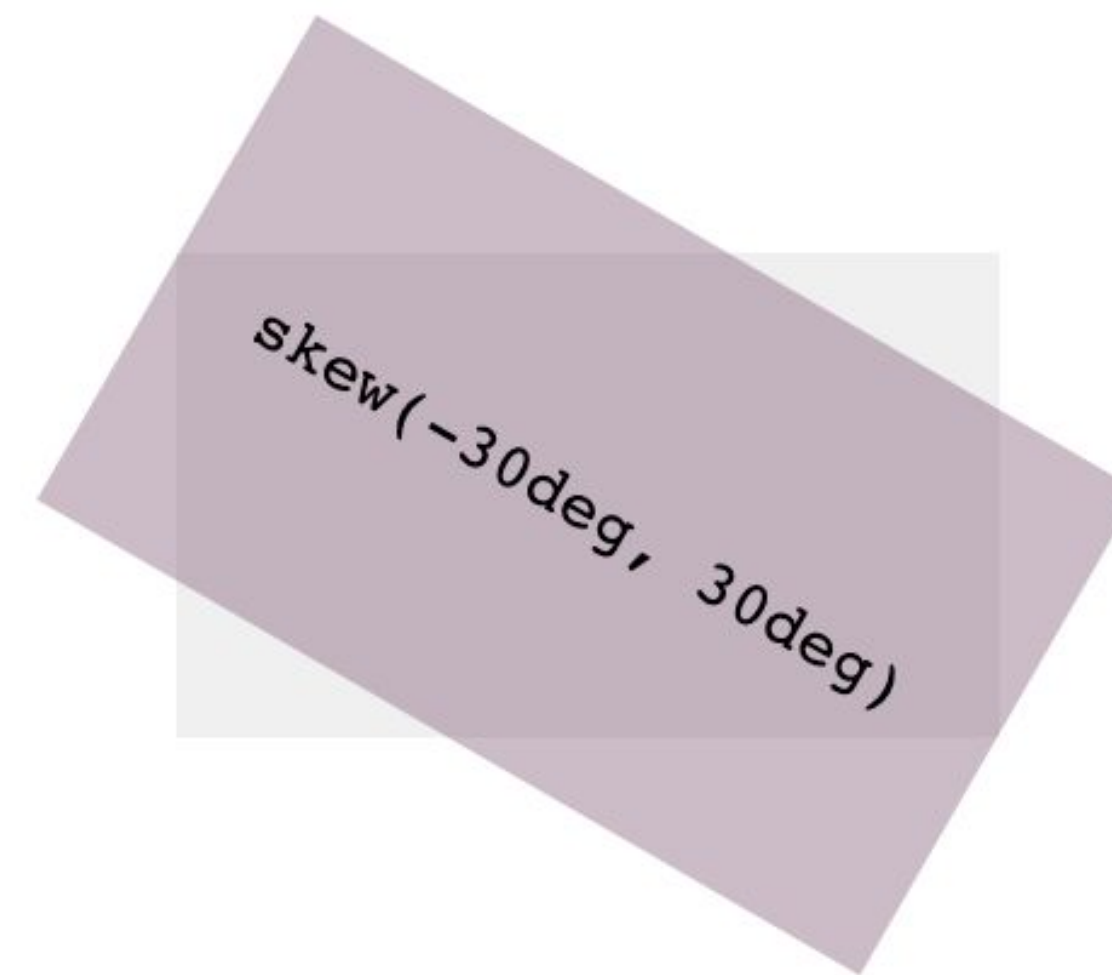


Трансформации

Виды 2D-трансформаций (функции, которые передаются в значении свойства transform):

- skew(х-**угол**, у-**угол**)

Используется для деформирования (искажения) сторон элемента относительно координатных осей. Если указано одно значение, второе будет определено браузером автоматически.



Трансформации

Виды 2D-трансформаций (функции, которые передаются в значении свойства transform):

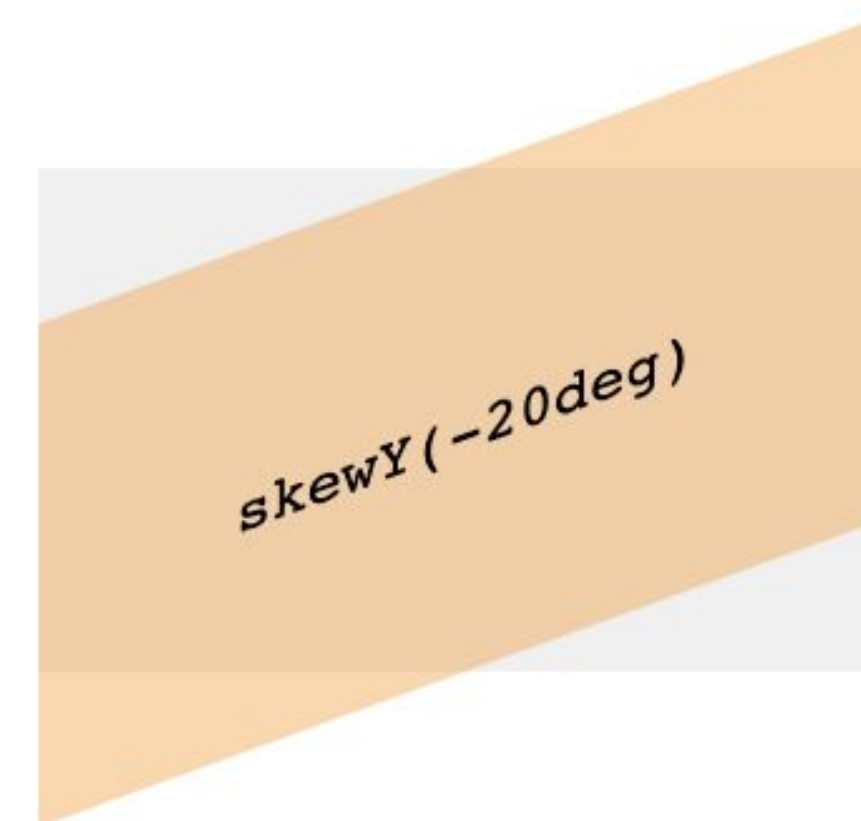
- skewX(**угол**)

Деформирует стороны элемента относительно оси X.



- skewY(**угол**)

Деформирует стороны элемента относительно оси Y.



Трансформации

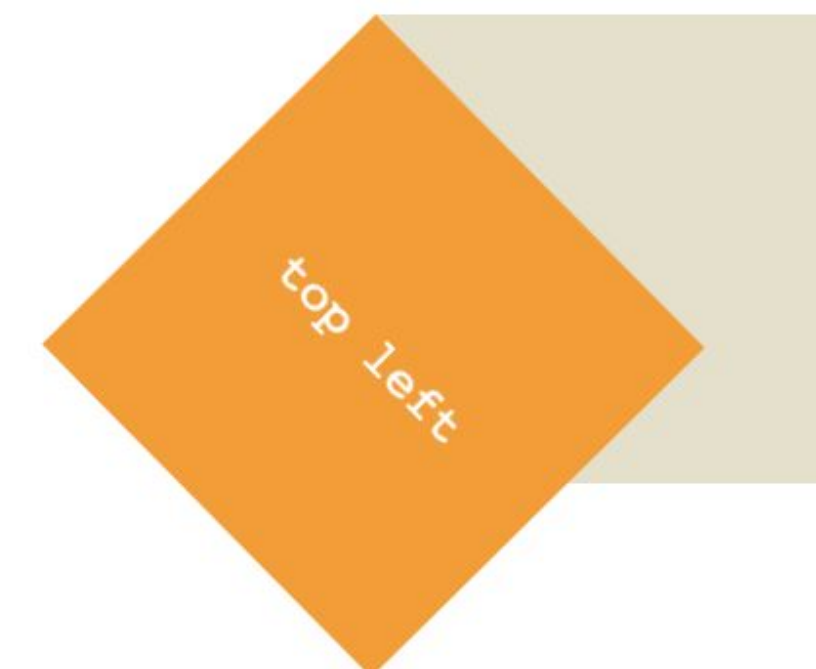
Точка трансформации - точка, относительно которой происходит изменение положения/размера/формы элемента. Задается свойством transform-origin.

Значения - пара значений, заданная с помощью ключевых слов, единиц длины или процентов определяет, относительно какой части элемента будет происходить трансформация.

по оси X - left, right, center, длина, %

по оси Y - top, bottom, center, длина, %

По умолчанию transform-origin: 50% 50%



transform-origin: top left

Трансформации

Множественные трансформации

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

```
transform: scale(1.2, 0.5) rotate(40deg);
```



Трансформации

3D-трансформации (добавляется ось Z):

- `translate3d(x,y,z)`

Функция задает перемещение элемента в 3D-пространстве. Движение происходит по вектору `[tx, ty, tz]`, где `tx` перемещение вдоль оси X, `ty` — перемещение вдоль оси Y, а `tz` — вдоль оси Z. Значения могут задаваться в единицах длины или в %. Отрицательные значения будут перемещать элемент в противоположном направлении.



```
translate3d(20px,20px,20px)
```

- `translateZ(z)`

Функция задает перемещение элемента на заданное расстояние в направлении оси Z.

Трансформации

3D-трансформации (добавляется ось Z):

- `scale3d(x,y,z)`

Функция задает операцию трехмерного масштабирования по вектору масштабирования $[sx, sy, sz]$, описываемому тремя параметрами. Отрицательные значения отображают элемент зеркально вдоль трех осей.



- `scaleZ(z)`

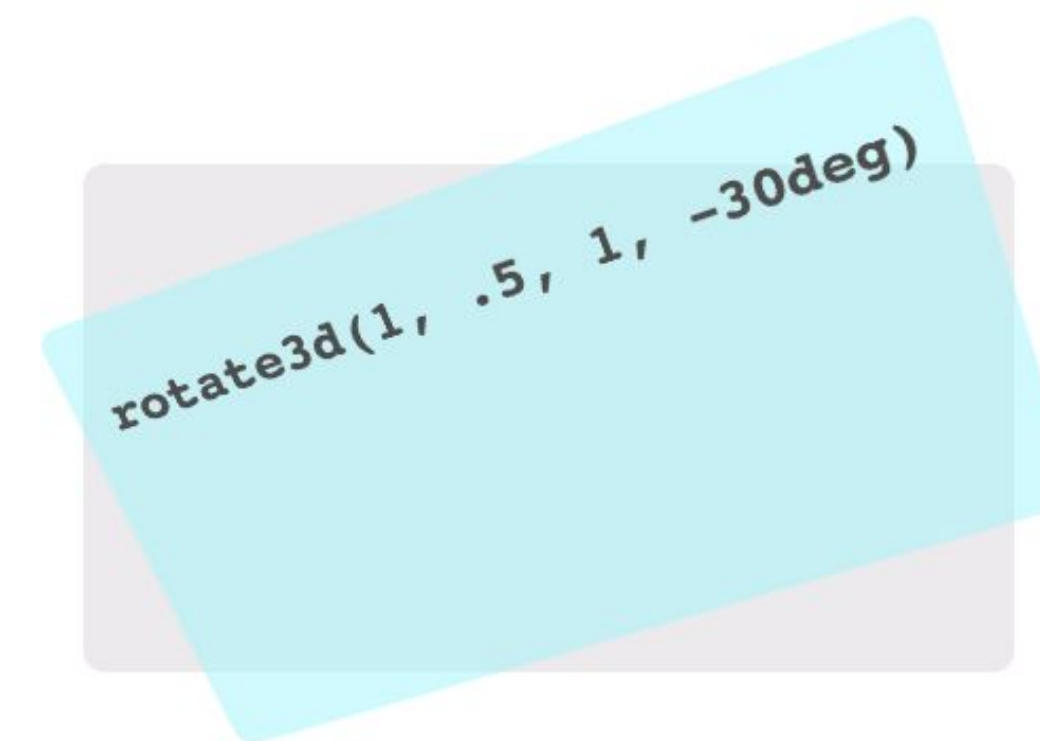
Функция масштабирует элемент в направлении оси Z, делая его больше или меньше.

Трансформации

3D-трансформации (добавляется ось Z):

- `rotate3d(x,y,z,угол)`

Функция вращает элемент по часовой стрелке относительно трех осей. Элемент поворачивается под углом, задаваемым последним параметром относительно вектора направления $[x,y,z]$. Отрицательные значения поворачивают элемент против часовой стрелки.



Трансформации

3D-трансформации (добавляется ось Z):

- rotateX(**угол**)

Функция задает поворот по часовой стрелке под заданным углом относительно оси X.

- rotateY(**угол**)

Функция задает поворот по часовой стрелке под заданным углом относительно оси Y.

- rotateZ(**угол**)

Функция задает поворот по часовой стрелке под заданным углом относительно оси Z.

Трансформации

3D-трансформации (добавляется ось Z):

- perspective(n)

Функция меняет перспективу обзора элемента, создавая иллюзию глубины. Чем больше значение функции перспективы, тем дальше от смотрящего расположен элемент. Значение должно быть больше нуля.

Переходы

CSS3-переходы позволяют анимировать исходное значение CSS-свойства на новое значение с течением времени, управляя скоростью смены значений свойств.

Смена свойств происходит при наступлении определенного события, которое описывается соответствующим псевдоклассом (часто используется :hover)

Переходы применяются ко всем элементам, а также к псевдоэлементам :before и :after

Переходы

Свойства для задания перехода:

- Свойство `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. Значения:
 - `all` - применяет эффект перехода ко всем свойствам элемента (значение по умолчанию)
 - `свойство` - определяет список CSS-свойств, перечисленных через запятую, участвующих в переходе.

Пример `transition-property: width`

Переходы

Свойства для задания перехода:

- Свойство `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). Значение:
 - время перехода (секунды(s) или миллисекунды(ms))

Пример `transition-duration: .2s`

Переходы

Свойства для задания перехода:

- Свойство `transition-timing-function` - задаёт временную функцию, которая описывает скорость перехода объекта от одного значения к другому. Значения:
 - `ease` - переход начинается медленно, разгоняется быстро и замедляется в конце (по умолчанию)
 - `linear` - переход происходит равномерно на протяжении всего времени, без колебаний в скорости
 - `ease-in` - переход начинается медленно, а затем плавно ускоряется в конце
 - `ease-out` - переход начинается быстро и плавно замедляется в конце
 - `ease-in-out` - переход медленно начинается и медленно заканчивается
 - `cubic-bezier(x1, y1, x2, y2)` - позволяет вручную установить значения от 0 до 1 для кривой ускорения (на сайте <http://roblaplaca.com/examples/bezierBuilder/> можно построить любую кривую)

Пример `transition-timing-function: linear`

Переходы

Свойства для задания перехода:

- Свойство `transition-delay` - позволяет сделать так, чтобы изменение свойства происходило не моментально, а с некоторой задержкой. Значение
 - Время задержки перехода (s или ms)

Пример `transition-delay: .5s`

Переходы

Свойства для задания перехода:

- Свойство `transition` - краткая запись перехода (все свойства, отвечающие за изменение внешнего вида элемента)

`transition: transition-property transition-duration transition-timing-function transition-delay`

Значение по умолчанию `transition: all 1s ease 0s`

Переходы

Плавный переход нескольких свойств:

Для элемента можно задать несколько последовательных переходов, перечислив их через запятую.

Каждый переход можно оформить своей временной функцией.

Пример **transition**: background 0.3s ease, color 0.2s linear

Анимация

Создание анимации базируется на на ключевых кадрах, которые позволяют автоматически воспроизводить и повторять эффекты на протяжении заданного времени, а также останавливать анимацию внутри цикла.

CSS3-анимация может применяться практически для всех html-элементов, а также для псевдоэлементов :before и :after.

Анимация

Создание анимации:

Задание ключевых кадров правилом @keyframes

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

@keyframes имя анимации { список правил }

Анимация

Ключевые кадры:

- from (0%), to (100%)
- процентные пункты

```
@keyframes shadow {  
  from {text-shadow: 0 0 3px black;}  
  50% {text-shadow: 0 0 30px black;}  
  to {text-shadow: 0 0 3px black;}  
}
```

После объявления правила @keyframes можно сослаться на него в свойстве animation

Анимация

Свойства анимации:

- animation-name (название анимации) - определяет список применяемых к элементу анимаций. Каждое имя используется для выбора ключевого кадра в правиле, которое предоставляет значения свойств для анимации. Значение - Имя анимации, которое связывает правило @keyframes с селектором (имя анимации чувствительно к регистру).

Анимация

Свойства анимации:

- `animation-duration` (продолжительность анимации) - определяет продолжительность одного цикла анимации. Значение - время в `s` или `ms`, которое анимация занимает для завершения одного цикла (отрицательные значения недействительны)

Анимация

Свойства анимации:

- animation-timing-function (временная функция) - описывает, как будет развиваться анимация между каждой парой ключевых кадров. Во время задержки анимации временные функции не применяются. Значения:
 - linear - линейная функция, анимация происходит равномерно на протяжении всего времени, без колебаний в скорости
 - ease - функция по умолчанию, анимация начинается медленно, разгоняется быстро и замедляется в конце

Анимация

- 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) - задание собственной кривой Безье
- step-start - задаёт пошаговую анимацию, разбивая анимацию на отрезки, изменения происходят в начале каждого шага
- step-end - пошаговая анимация, изменения происходят в конце каждого шага

Анимация

- `steps`(количество шагов, положение шага) - Ступенчатая временная функция, которая принимает два параметра. Первый параметр указывает количество интервалов в функции (положительное целое число > 0). Вторым параметром (необязательным) указывается 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

Анимация

Свойства анимации:

- animation-iteration-count (повтор анимации) - указывает, сколько раз проигрывается цикл анимации. 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

Анимация

Свойства анимации:

- animation-direction (направление анимации) - 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.

Значения:

- normal - все повторы анимации воспроизводятся так, как указано (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

Анимация

Свойства анимации:

- `animation-play-state` (проигрывание анимации) - определяет, будет ли анимация запущена или приостановлена. Остановка анимации внутри цикла возможна через использование этого свойства в скрипте JavaScript. Значения:
 - `running` - анимация выполняется (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

Анимация

Свойства анимации:

- `animation-delay` (задержка анимации) - определяет, когда анимация начнется.

Значение - время (s или ms), которое 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.

Анимация

Свойства анимации:

- animation-fill-mode (состояние элемента до и после воспроизведения анимации) - определяет, какие значения применяются анимацией вне времени ее выполнения. Когда анимация завершается, элемент возвращается к своим исходным стилям. Значения:
 - forwards - после того, как анимация заканчивается, анимация будет применять значения свойств к моменту окончания анимации.
 - backwards - в течение периода, определенного с помощью animation-delay анимация будет применять значения свойств, определенные в ключевом кадре, которые начнут первую итерацию анимации.
 - both - позволяет оставлять элемент в первом ключевом кадре до начала анимации (игнорируя положительное значение задержки) и задерживать на последнем кадре до конца последней анимации.

Анимация

Краткая запись анимации:

Свойство animation

animation: animation-name animation-duration animation-timing-function animation-delay
animation-iteration-count animation-direction

Чаще всего достаточно указать animation: animation-name animation-duration

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;}
```