**CSS3-трансформации**

CSS3-трансформации позволяют сдвигать, поворачивать и масштабировать элементы. Трансформации преобразовывают элемент, не затрагивая остальные элементы веб-страницы, т.е. другие элементы не сдвигаются относительно него.

К элементам, которые могут быть трансформированы, относятся элементы с display: *block;* *и display: inline-block;*, а также элементы, значение свойства display которых вычисляется как *table-row, table-row-group, table-header-group, table-footer-group, table-cell или table-caption.* Трансформированным считается элемент с любым установленным значением свойства transform, отличным от none.

**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 {

transform: rotate(360deg);

}

**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 {

transform-origin: 20% 40%;

}

**Множественные трансформации**

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

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

**CSS transitions** предоставляют способ контролировать скорость анимации, при изменении CSS свойств. Вместо того, чтобы свойство применилось сразу, вы можете сделать это действие происходящим в течение какого-то момента времени. Например, если вы смените цвет элемента с белого на чёрный, изменение произойдёт моментально, а вот с CSS transitions, измения произойдут за временные интервалы, следующих кривой ускорения, все из которых могут быть настроены.

Анимации, составляющие переход между двумя состояниями, часто называют *неявными переходами*, так как промежуточные состояния в период между начальным и конечным состоянием неявно определяются браузером.

CSS переходы позволяют вам решить какие свойства нужно анимировать (перечислением их явно), когда анимация начнётся (установкой задержки), как долго переход будет выполняться (установкой продолжительности), а также как она будет запущена (определением функции по времени, например линейно или быстро в начале, медленно в конце).

**Замечание:** CSS переходы могут использоваться без префиксов, однако спецификация только недавно достигла стабильности, префиксы по-прежнему необходимы для браузеров на WebKit. Они также необходимы для поддержки старых версий браузеров (таких как: Firefox 15, Opera 12 и более ранних). Таблица совместимости приведена внизу страницы с ещё большим количеством информации.

## [Какие CSS свойства анимируются?](https://developer.mozilla.org/ru/docs/Web/CSS/CSS_Transitions/Using_CSS_transitions#%D0%BA%D0%B0%D0%BA%D0%B8%D0%B5_css_%D1%81%D0%B2%D0%BE%D0%B9%D1%81%D1%82%D0%B2%D0%B0_%D0%B0%D0%BD%D0%B8%D0%BC%D0%B8%D1%80%D1%83%D1%8E%D1%82%D1%81%D1%8F)

Разработчик может определить какое свойство и как анимировать. Это позволяет создавать сложные переходы. Так как некоторые свойства не имеет смысла анимировать, перечень [доступных для анимирования свойств](https://developer.mozilla.org/en-US/docs/CSS/CSS_animated_properties) ограничен определенным набором.

Замечание: Перечень анимируемых свойств может меняться по мере изменения спецификации.

Значение свойства auto является сложным случаем. Спецификация не рекомендует анимировать в значение auto и из значения auto. Браузеры, основанные на Gecko, исполняют это требование в точности, а основанные на WebKit не так строго. Использование переходов с auto следует избегать, так как это может привести к непредсказуемым результатам, в зависимости от браузера и его версии.

Необходимо также соблюдать осторожность при использовании переходов сразу после добавления элемента в DOM с помощью .appendChild() или удаления его display: none; свойства. Это выглядит, как будто никогда не происходило начальное состояние, а элемент всегда был в конечном состоянии. Самый простой способ преодолеть это ограничение - применить window.setTimeout() c некоторым количеством миллисекунд до изменения CSS свойства, которое вы собираетесь анимировать.

### [Пример анимирования нескольких свойств](https://developer.mozilla.org/ru/docs/Web/CSS/CSS_Transitions/Using_CSS_transitions#%D0%BF%D1%80%D0%B8%D0%BC%D0%B5%D1%80_%D0%B0%D0%BD%D0%B8%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F_%D0%BD%D0%B5%D1%81%D0%BA%D0%BE%D0%BB%D1%8C%D0%BA%D0%B8%D1%85_%D1%81%D0%B2%D0%BE%D0%B9%D1%81%D1%82%D0%B2)

#### HTML

<body>

<p>К блоку ниже применятся анимации для свойств: width, height, background-color, transform. Наведите на него, чтоб увидеть, как они анимируются</p>

<div class="box"></div>

</body>

#### CSS

.box {

border-style: solid;

border-width: 1px;

display: block;

width: 100px;

height: 100px;

background-color: #0000FF;

-webkit-transition: width 2s, height 2s, background-color 2s, -webkit-transform 2s;

transition: width 2s, height 2s, background-color 2s, transform 2s;

}

.box:hover {

background-color: #FFCCCC;

width: 200px;

height: 200px;

-webkit-transform: rotate(180deg);

transform: rotate(180deg);

}

## [CSS свойства, опредяющие переходы](https://developer.mozilla.org/ru/docs/Web/CSS/CSS_Transitions/Using_CSS_transitions#css_%D1%81%D0%B2%D0%BE%D0%B9%D1%81%D1%82%D0%B2%D0%B0_%D0%BE%D0%BF%D1%80%D0%B5%D0%B4%D1%8F%D1%8E%D1%89%D0%B8%D0%B5_%D0%BF%D0%B5%D1%80%D0%B5%D1%85%D0%BE%D0%B4%D1%8B)

CSS переходы контролируются свойством[transition](https://developer.mozilla.org/ru/docs/Web/CSS/transition). Это лучший способ их настроить, позволяет избежать длинных свойств и их несоответствий, которые могут сделать отладку CSS долгой.

Можно контролировать определенные параметры перехода следующими подсвойствами:

**(Заметьте, что циклы переходов бесконечны только для наших примеров; CSS transitions только анимируют смену свойства. Если вам нужно визуализировать этот цикл, обратите внимание на свойство** [**animation**](https://developer.mozilla.org/en-US/docs/CSS/animation)**.)**

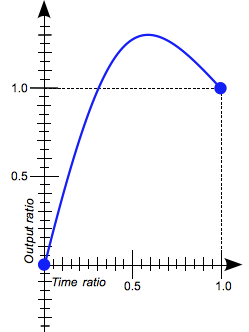
[**transition-property**](https://developer.mozilla.org/ru/docs/Web/CSS/transition-property)

Указывает имя или имена свойств, чьи переходы должны анимироваться. Только свойства, указанные здесь, анимируются в переходах; изменение других свойств будет происходить обычным образом.

[**transition-duration**](https://developer.mozilla.org/ru/docs/Web/CSS/transition-duration)

Определяет время происхождения перехода. Можно указать время анимирования всех свойств перехода сразу или для каждого свойства в отдельности.

[**transition-timing-function**](https://developer.mozilla.org/ru/docs/Web/CSS/transition-timing-function)

****Определяет функцию, указывающую, как вычисляются промежуточные значения свойств. Вы также можете выбрать ослабления от функции замедления Шпаргалка.Большинство [функций времени](https://developer.mozilla.org/en-US/docs/CSS/timing-function) может быть определено графиком соответствующей функции, образующей четырьмя точками кривую Безье. Также можете выбрать функцию отсюда [Easing Functions Cheat Sheet](http://easings.net/).

transition-timing-function: ease

linear

step-end

steps(4, end)

[**transition-delay**](https://developer.mozilla.org/ru/docs/Web/CSS/transition-delay)

Определяет как много должно пройти времени, перед тем как начнётся переход.

Короткая запись синтаксиса:

div {

transition: <property> <duration> <timing-function> <delay>;

}

## [Определение завершения перехода](https://developer.mozilla.org/ru/docs/Web/CSS/CSS_Transitions/Using_CSS_transitions#%D0%BE%D0%BF%D1%80%D0%B5%D0%B4%D0%B5%D0%BB%D0%B5%D0%BD%D0%B8%D0%B5_%D0%B7%D0%B0%D0%B2%D0%B5%D1%80%D1%88%D0%B5%D0%BD%D0%B8%D1%8F_%D0%BF%D0%B5%D1%80%D0%B5%D1%85%D0%BE%D0%B4%D0%B0)

Есть событие, срабатывающее, когда переход завершен. Во всех браузерах, совместимых со стандартами, есть событие transitionend, в WebKit есть webkitTransitionEnd. Смотрите таблицу поддержки браузерами для дополнительной информации. У transitionend есть 2 свойства:

**propertyName**

Строка, показывающая изменение какого свойства завершено.

**elapsedTime**

Число с точкой, какое количество времени переход выполнялся, до того как запустилось событие. Это значение не равно свойству [transition-delay](https://developer.mozilla.org/ru/docs/Web/CSS/transition-delay).

Как обычно, используйте метод [element.addEventListener()](https://developer.mozilla.org/ru/docs/Web/API/Element/addEventListener), чтобы следить за этим событием:

el.addEventListener("transitionend", updateTransition, true);

**Замечание:** Событие transitionend не произойдёт, когда переход был прерван до его завершения, например, если установили [display](https://developer.mozilla.org/ru/docs/Web/CSS/display): none или значение анимируемого свойства изменилось.

## [Когда у списков значений свойств разные длины](https://developer.mozilla.org/ru/docs/Web/CSS/CSS_Transitions/Using_CSS_transitions#%D0%BA%D0%BE%D0%B3%D0%B4%D0%B0_%D1%83_%D1%81%D0%BF%D0%B8%D1%81%D0%BA%D0%BE%D0%B2_%D0%B7%D0%BD%D0%B0%D1%87%D0%B5%D0%BD%D0%B8%D0%B9_%D1%81%D0%B2%D0%BE%D0%B9%D1%81%D1%82%D0%B2_%D1%80%D0%B0%D0%B7%D0%BD%D1%8B%D0%B5_%D0%B4%D0%BB%D0%B8%D0%BD%D1%8B)

Если любой список свойств короче, чем другие, его значения повторяюся, чтобы сделать его длину как и у других. Например:

div {

transition-property: opacity, left, top, height;

transition-duration: 3s, 5s;

}

Это рассматривается, как если бы это было:

div {

transition-property: opacity, left, top, height;

transition-duration: 3s, 5s, 3s, 5s;

}

Похожим образом, если какой-то список свойств длиннее, чем у [transition-property](https://developer.mozilla.org/ru/docs/Web/CSS/transition-property), он обрезается:

div {

transition-property: opacity, left;

transition-duration: 3s, 5s, 2s, 1s;

}

Будет интерпретировано как:

div {

transition-property: opacity, left;

transition-duration: 3s, 5s;

}

## [Простой пример](https://developer.mozilla.org/ru/docs/Web/CSS/CSS_Transitions/Using_CSS_transitions#%D0%BF%D1%80%D0%BE%D1%81%D1%82%D0%BE%D0%B9_%D0%BF%D1%80%D0%B8%D0%BC%D0%B5%D1%80)

Пример 4 секунды выполняет плавное измение шрифта за 2 секунды, после того как пользователь навел мышь на элемент:

#delay1 {

position: relative;

transition-property: font-size;

transition-duration: 4s;

transition-delay: 2s;

font-size: 14px;

}

#delay1:hover {

transition-property: font-size;

transition-duration: 4s;

transition-delay: 2s;

font-size: 36px;

}

## [Использование переходов для подсвечивания меню](https://developer.mozilla.org/ru/docs/Web/CSS/CSS_Transitions/Using_CSS_transitions#%D0%B8%D1%81%D0%BF%D0%BE%D0%BB%D1%8C%D0%B7%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5_%D0%BF%D0%B5%D1%80%D0%B5%D1%85%D0%BE%D0%B4%D0%BE%D0%B2_%D0%B4%D0%BB%D1%8F_%D0%BF%D0%BE%D0%B4%D1%81%D0%B2%D0%B5%D1%87%D0%B8%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F_%D0%BC%D0%B5%D0%BD%D1%8E)

Они часто используются для подсвечения элементов в меню при наведении мыши. С помощью переходов легко сделать такой эффект более привлекательным.

Перед тем, как вы посмотрите на код, возможно захотите [посмотреть на живое демо](https://codepen.io/anon/pen/WOEpva) (считаем, что ваш браузер поддерживает переходы). Также можеть [посмотреть на CSS](https://developer.mozilla.org/samples/cssref/transitions/sample2/transitions.css), используемый этим примером.

Сначала зададим меню в HTML:

<div class="sidebar">

<p><a class="menuButton" href="home">Home</a></p>

<p><a class="menuButton" href="about">About</a></p>

<p><a class="menuButton" href="contact">Contact Us</a></p>

<p><a class="menuButton" href="links">Links</a></p>

</div>

Теперь напишем CSS для нашего меню:

.menuButton {

transition-property: background-color, color;

transition-duration: 1s;

transition-timing-function: ease-out;

text-align: left;

background-color: grey;

margin: 5px

line-height: 26px;

color: white;

border-color: black;

font-family: sans-serif;

font-size: 20px;

text-decoration: none;

box-shadow: 2px 2px 1px black;

padding: 2px 4px;

border: solid 1px black;

}

.menuButton:hover {

transition-property: background-color, color;

transition-duration: 1s;

transition-timing-function: ease-out;

background-color:white;

color:black;

}

Этот CSS устанавливает внешний вид меню, с фоном и цветом текста, изменяющимися при наведении ([:hover](https://developer.mozilla.org/ru/docs/Web/CSS/:hover)).

Вместо описания эффекта можете [посмотреть на живой пример](https://developer.mozilla.org/samples/cssref/transitions/sample2), если ваш [браузер поддерживает переходы](https://developer.mozilla.org/ru/docs/Web/CSS/CSS_Transitions/Using_CSS_transitions#Browser_compatibility).

[ТУТ](https://developer.mozilla.org/ru/docs/Web/CSS/CSS_Transitions/Using_CSS_transitions)

Наглядные примеры трансформаций HTML5BOOK - https://html5book.ru/css3-transform/