



CSS 3. Переходы

Что такое переходы

- **переходы** — это просто способ **анимации** стилевых свойств от *исходного* до *конечного* состояния.

Переходы в CSS являются *специфическим* видом анимации, где:

- есть только два состояния: начало и конец;
- анимация не зациклена;
- промежуточные состояния управляются только функцией времени.

Селекторы для задания переходов

- transition-property: какие **свойства** анимируются;
- transition-duration: **как долго** длится анимация;
- transition-timing-function: как вычисляются **промежуточные состояния**;
- transition-delay: задержку перед выполнением перехода (или анимация начинается **через** некоторое время).

Можно установить каждое свойство CSS по отдельности или использовать сокращённую версию: transition. В этом случае, только transition-duration является **обязательным**.

Пример

<https://codepen.io/gavorilka/pen/ExEvqzd>

```
.animate{  
  transition: transform 3s linear; /*переход  
свойства трансформ*/  
  transform: scale(1,0);  
  transform-origin: 100% 0;  
}  
.wrapper:hover .animate{  
  transform: scale(1,1);  
  transition: transform 5s easeInElastic; /*переход  
свойства трансформ*/  
}
```

transition-duration

- **transition-duration** является единственным стилевым свойством, необходимым для создания перехода. Оно может быть установлено либо в **секундах** (2s), либо в **миллисекундах** (100ms).
- Если вы хотите, чтобы ваш переход длился **полсекунды**, то можете написать 0.5s или 500ms. В зависимости от того, насколько быстро вы хотите, чтобы длился переход, может быть проще и/или быстрее писать одну единицу.

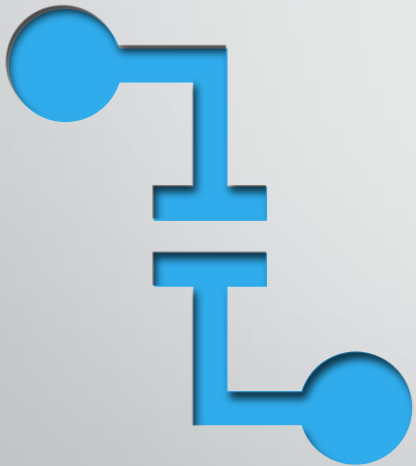
Пример

```
a {  
  background: lightgrey;  
  color: grey;  
}  
a:hover {  
  background: yellow;  
  color: red; }  
a.with-fast-transition {  
  transition: 0.5s;  
}  
a.with-slow-transition {  
  transition: 3s;  
}
```

<https://codepen.io/gavorilka/pen/zYWEOKe>

transition-property

- Только **треть** свойств CSS может быть анимирована. На сайте Mozilla есть [полный список](#) свойств.
- По умолчанию у свойства transition-property значение all, это просто означает, что будут анимироваться все возможные свойства.
- Вы можете разрешить анимацию только одного или нескольких свойств.



Пример

```
1 a {  
2   background: lightgrey;  
3   color: grey;  
4 }  
5 a:hover {  
6   background: yellow;  
7   border: 5px solid blue;  
8   color: green;  
9   text-decoration-color:red;  
10 }  
11 a.with-background-transition {  
12   transition-duration: 2s;  
13   transition-property: background;  
14 }
```


<https://codepen.io/gavorilka/pen/BarwBJw>

transition-timing-function

- Функция времени определяет, как вычисляется **значение** каждого свойства **во время перехода**.
- По умолчанию переход равен ease: он ускоряется в начале и замедляется в конце.
- Вы можете решить, что переход будет происходить с **постоянной скоростью**. Функция времени может **ускорить** или **замедлить** переход.
- Самый простой способ визуализации функций времени заключается в изменении **свойств позиционирования**.
- Временные функции могут работать с [шагами](#).

Пример

<https://codepen.io/gavorilka/pen/gOeGYQx>

transition-timing-function 

gavorilka

HTML

```
1 * <main>
2 * <div class="ease">
3 *   <strong>ease</strong>: медленно начинается,
      быстро в середине, медленно завершается
4 * </div>
5 * <div class="linear">
6 *   <strong>linear</strong>: постоянная скорость
7 * </div>
8 * <div class="ease-in">
9 *   <strong>ease-in</strong>: медленно начинается,
```

CSS

```
1 * div {
2 *   left: 0; position: relative; transition: 1s;
3 * }
4 * main:hover div {
5 *   left: 1000px;
6 * }
7 * .ease {
8 *   transition-timing-function: ease;
9 * }
10 * /* Поведение по умолчанию */
```

ease: медленно начинается, быстро в середине, медленно завершается
linear: постоянная скорость
ease-in: медленно начинается, быстро завершается
ease-out: быстро начинается, медленно завершается
ease-in-out: похоже на ease, но с более выраженным ускорением/замедлением

cubic-bezier



Полный список готовых функций времени: <http://easings.net/ru>



Если все эти **готовые** функции времени не подходят, вы можете написать свою собственную, используя **кривые Безье**.



Сайт cubic-bezier.com — это простой инструмент для визуального написания своих собственных кривых.

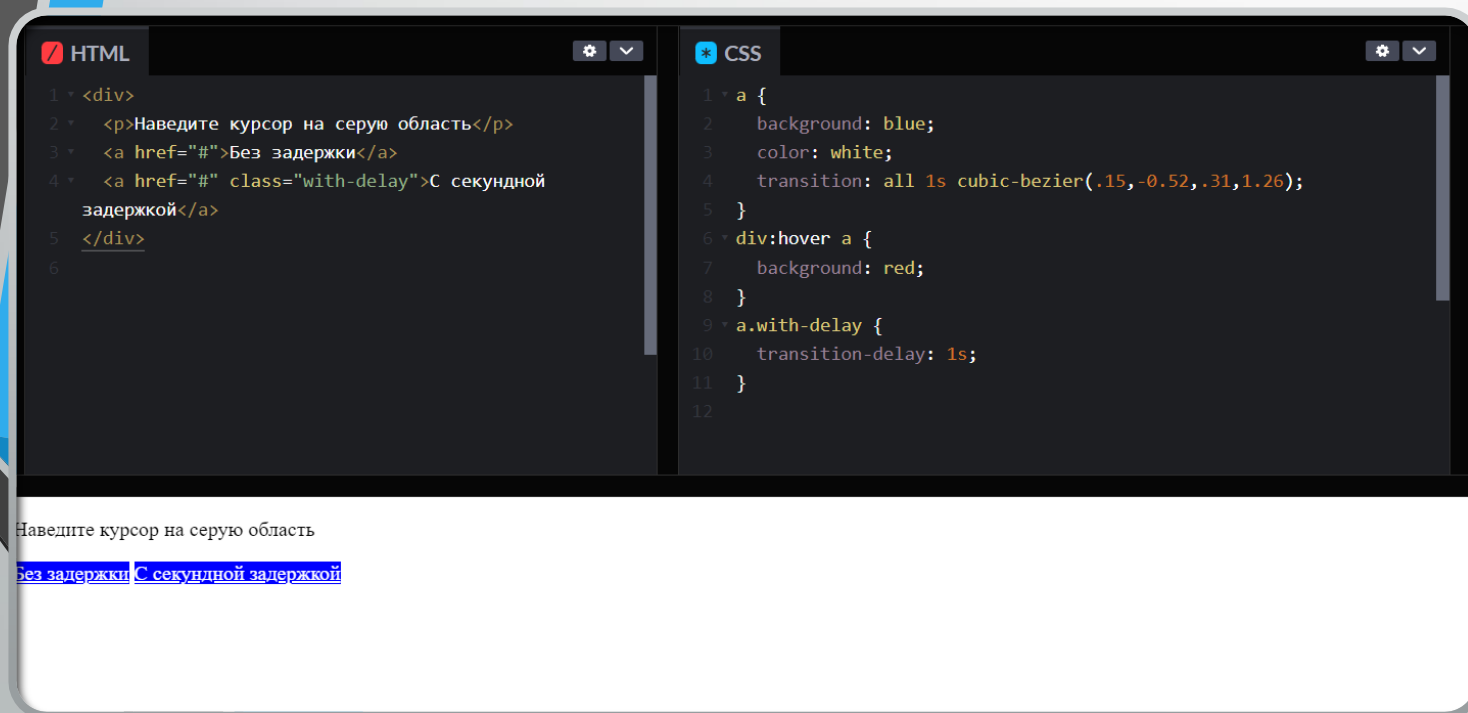
transition-delay

- transition-delay определяет задержку, т. е. как долго переход должен **ждать**, *прежде* чем начнётся на самом деле.
- Как и с transition-duration вы можете использовать секунды (s) или миллисекунды (ms).



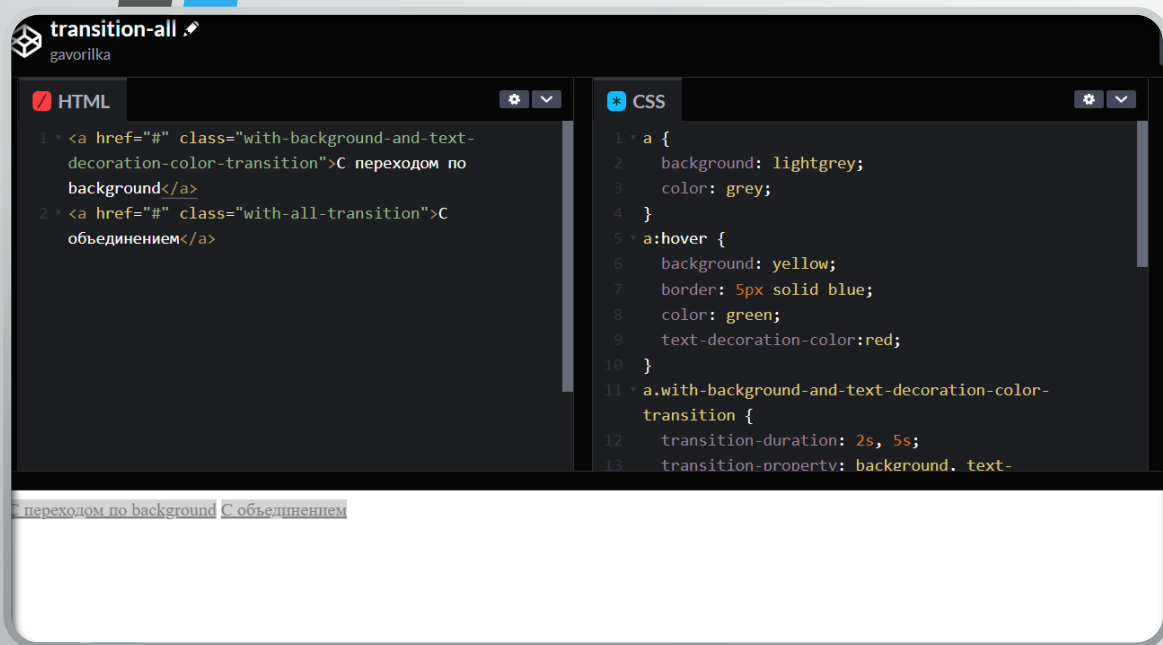
Пример

<https://codepen.io/gavorilka/pen/vYReBMm>



Плавный переход нескольких свойств

<https://codepen.io/gavorilka/pen/jOzGOVR>



Дополнительные ссылки

- <https://html5book.ru/css3-transition>
- <https://webref.ru/layout/advanced-html-css/transitions-animations>

Задачи

<http://old.code.mu/tasks/css/rabota-s-plavnymi-perehodami-transition-na-css.html>