

# CSS-АНИМАЦИЯ

*В. В. Ащепкова*

CSS-анимация — переход объекта из одного состояния в другое.

Рассмотрим ряд свойств, с помощью которых ее можно реализовать на веб-странице. Первое — transition. Оно позволяет задать переход между объектами для определенного события, например, focus или hover. Свод параметров для контроля данного эффекта:

- transition-property указывает имена свойств CSS, для которых используется переходный эффект (по умолчанию применяется для всех);
- transition-duration задает время перехода в секундах или миллисекундах;
- transition-timing-function определяет кривую скорости анимации;
- transition-delay добавляет задержку начала эффекта.

*Листинг 1. Объявление перехода*

```
letter-spacing: 10px;  
top: 89%;  
background-color: #483D8B;  
box-shadow: 0 10px 0px #4B0082;  
transition-duration: 500ms;
```

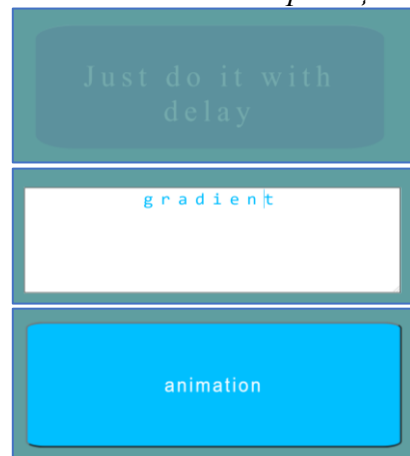
*Листинг 2. Изменяемые свойства*

```
#doIt:hover{  
  opacity: 0%;  
  border-radius: 5% 5% 5% 5%;  
  z-index: 1;  
}#nextpage:hover{  
  letter-spacing: 3px;  
  top: 91%;  
  background-color:#00BFFF;  
  box-shadow: 0 0 0px #2F4F4F;;  
}  
#writeDown:focus{  
  color: #00BFFF;  
}
```

*Рис. 1. Состояние страницы до переходов*



*Рис. 2. Состояние страницы после*



Второе свойство — animation. С его помощью можно задать поведение анимации. Многие параметры перехода повторяют те, что были представлены в transition: animation-duration, animation-timing-function, animation-delay.

Отличительными свойствами являются:

- `animation-name` указывает имя анимации, применяемой к выбранному элементу;
- `animation-iteration-count` задает количество воспроизведений;
- `animation-direction` определяет направление анимации после достижения конечной точки;
- `animation-play-state` добавляет паузу в анимацию при определенном событии;
- `animation-fill-mode` описывает состояние анимации при ее окончании или задержке.

*Листинг 3. Вызов анимации*

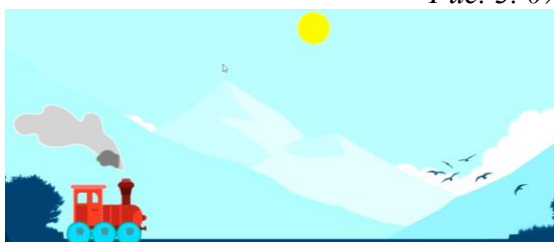
```
animation: train 19s linear infinite;
```

*Листинг 4. Задание keyframes*

```
@keyframes train {  
  50% {  
    opacity: 1;  
    left: 76%;  
  }  
  51% {  
    opacity: 0;  
  }  
  100% {  
    opacity: 0;  
    left: 26%;  
  }  
}
```

С помощью правила `keyframes` можно задать ключевые кадры анимации (см. листинг 4). Разбиение задается в процентах, но существует также запись с `from` и `to`, которые эквивалентны значениям `0%` и `100%` соответственно. Внутри `keyframes` свойства могут изменяться вне зависимости друг от друга.

*Рис. 3. 0%*



*Рис. 4. 50%*



*Рис. 5. 100%*



Третье свойство — `transform`. Оно, как и предыдущие, пока что является экспериментальным: можно использовать только в CSS3. Набор функций представляет из себя:

- `rotate(angle)` — поворот объекта;
- `scale(x, y)` — масштабирование;

- skew(x-angle, y-angle) — изменение перспективы относительно Oх и Oу;
- translate(x, y) — перенос опорной точки.

Листинг 5. Реализация transform

```
.train .smoke img, .trainback > img{
  transform: scale(-1,1);
}
.small{
  width: 100px;
  height: 100px;
  transform: translate(-50px,-50px);
  position: absolute;
}
@keyframes arc1 {
  0%{transform: rotate(0deg);}
  100%{transform: rotate(180deg);}
}
```

Листинг 6. Формулы для matrix

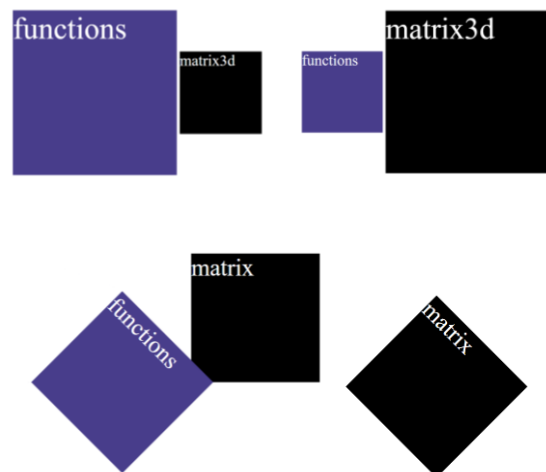
```
matrix (ScaleX/1, 0, 0, ScaleY/1,
0/TranslateX, 0/TranslateY)
matrix3d (ScaleX/1, 0, 0, 0, 0, ScaleY/1, 0,
0, 0, 0, ScaleZ/1, 0, 0/TranslateX,
0/TranslateY, 0/TranslateZ, 1)
```

Данные функции представляются браузером как умножение матрицы (3×3) или (4×4) на вектор (x, y, 1) или (x, y, z, 1), после чего на странице будут загружены элементы с измененными параметрами. Таким образом, можно сделать вывод, что преобразования в Web тесно связаны с линейной алгеброй. Было решено опробовать их самостоятельно. С помощью выведенной формулы (см. листинг 6) для matrix (n,n,n,n,n,n), matrix3d (n,n,n,n,n,n,n,n,n,n,n,n,n,n,n,n) было показано, что специальные функции действительно можно заменить на простые матрицы (см. рис. 6).

Листинг 7. Использование формул

```
#matrix3d{
  transform: matrix3d(1, 0, 0, 0, 0, 1, 0, 0, 0, 0,
1, 0, -50, -50, 50, 1);
} # matrix3d:hover{
  transform:matrix3d(2, 0, 0, 0, 0, 2, 0, 0, 0, 0, 2,
0, -50, -50, -50, 1);
}
#funct1 {
  transform: translate(-50px,-50px);
} #funct1:hover{
  transform: translate(-50px,-50px) scale(2);
}
#matrix {
  transform: matrix(1, 0, 0, 1, -50, -50);
} #matrix:hover{
  transform: matrix(0.7, 0.7, -0.7, 0.7, 0, 0);
}
#funct2 {
  transform: translate(-50px,-50px);
} #funct2:hover{
  transform: rotate(45deg);
}
```

Рис. 6. Реализация matrix



## Литература

1. CSS-Animations Level 1 [Электронный ресурс]. – URL: <https://www.w3.org/TR/css-animations-1/> (дата обращения: 30.04.2021).
2. CSS-animations [Электронный ресурс]. – URL: <https://javascript.info/css-animations/> (дата обращения: 30.04.2021).