

CSS и Анимация, часть 2



Web.Dev.
Courses

www.courses.dp.ua

Анимация средствами CSS

Два инструмента анимации в CSS

1. Плавный переход между состояниями некоторых свойств - базируется на свойстве **transition**

2. Заранее определенный сценарий, который проигрывается и плавно меняет значения заданных свойств - базируется на свойстве **animation**

Анимация на веб-странице это, как правило, изменение во времени того или иного стилевого (CSS) свойства элемента во времени.

На самом деле **CSS** не позволяет сделать полноценную интерактивную анимацию (в частности, игры), но **CSS** позволяет сделать анимированный переход между значениями стилевого свойства. Другими словами: сделать плавный переход между состояниями.

<http://jsbin.com/eRiMeDE/2/edit?html,css,output>

Анимация «по сценарию»

```
16 @keyframes my-script{
17     0%{
18         background: yellow;
19         transform: rotate(0deg);
20     }
21
22     70% {
23         background: blue;
24         transform: rotate(45deg);
25     }
26
27     90%{
28         background: orange;
29         transform: rotate(-15deg);
30     }
31
32     100%{
33         background: yellow;
34         transform: rotate(0deg);
35     }
36 }
```

Анимация «по сценарию»

Анимация «по сценарию» состоит из двух шагов. **Первый шаг:** создается «сценарий» при помощи директивы **@keyframes** задаются «опорные кадры» т.е. состояния через которые наша анимация должна проходить. Кадры нумеруются в **процентах** (для анимации между двумя состояниями возможно использовать **from - to**), за счёт этого не только определяется последовательность кадров но и рассчитываются время необходимое на проигрывание перехода между кадрами. У сценария должно быть указано имя.

<https://webref.ru/css/keyframes>

Анимация «по сценарию»

Анимация «по сценарию» состоит из двух шагов. **Второй шаг** включает проигрывание анимации для тега. Для этого с помощью с помощью соответствующих свойств указать: название сценария (**animation-name**), продолжительность анимации (**animation-duration**), функция распределения времени (**animation-timing-function**), задержка начала анимации (**animation-delay**), количество повторений анимации (**animation-iteration-count** – от **1** до бесконечности **infinite**), направление анимации (**animation-direction**), и какие стили применять по завершению анимации (**animation-fill-mode**). Или же можно использовать общее свойство **animation** позволяющее одной строкой задать все параметры анимации.

```
3  div{
4      animation-name: my-script;
5      animation-duration: 3s;
6      animation-timing-function: linear;
7      animation-delay: 1s;
8      animation-iteration-count: infinite;
9      animation-direction: alternate;
10     animation-fill-mode: forwards;
11
12     /* Или же */
13
14     animation: my-script 3s linear 1s infinite alternate forwards;
15 }
```

<https://webref.ru/css/animation>

<https://webref.ru/css/animation-name>

<https://webref.ru/css/animation-duration>

<https://webref.ru/css/animation-timing-function>

<https://webref.ru/css/animation-delay>

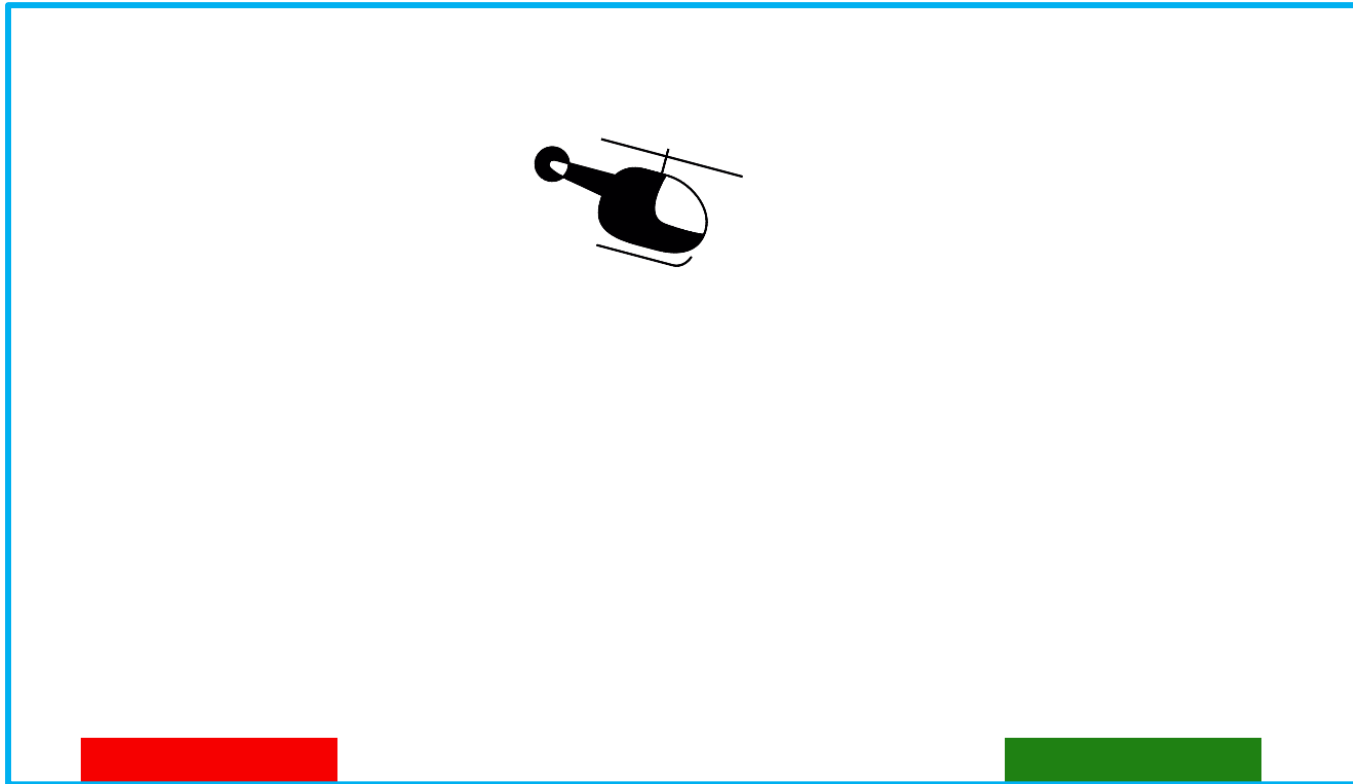
<https://webref.ru/css/animation-iteration-count>

<https://webref.ru/css/animation-direction>

<https://webref.ru/css/animation-fill-mode>

Немного практики

Анимация по «сценарию»



Состоит из двух шагов:

Первый шаг: создается «сценарий» при помощи директивы **@keyframes** задаются «опорные кадры» т.е. состояния через которые наша анимация должна проходить.

Второй шаг: при помощи свойства **animation** «сценарий» крепится к какому-либо элементу.

Воспользуйтесь разметкой по адресу:

<https://files.courses.dp.ua/web/06/ex06.html>

```
32 @keyframes my-script {
33     0% {
34         bottom: 8px;
35         left: 115px;
36     }
37     20%{
38         bottom: 500px;
39         left: 50px;
40         transform: rotate(0deg) ;
41     }
42     25%{
43         transform: rotate(20deg) ;
44     }
45     70%{
46         bottom: 450px;
47         left: 1050px;
48     }
49     100% {
50         bottom: 0px;
51         left: 1050px;
52     }
53 }
54
55 #heli {
56     animation-name: my-script;
57     animation-duration: 4s;
58     animation-iteration-count: infinite;
59     animation-direction: alternate;
60     animation-delay: 5s;
61     animation-timing-function: linear;
62 }
```

Анимация по «сценарию»

Браузер проигрывает сценарий делая плавные переходы между состояниями для опорных кадров, сценарий может быть проигран однократно, или многократно, при многократном повторении необходимо позаботиться о «склейке» финального и начального кадра.

Font Awesome

Вспомогательные классы с анимацией

Анимация в Font Awesome



```
<div class="fa-3x">  
  <i class="fas fa-spinner fa-spin"></i>  
  <i class="fas fa-circle-notch fa-spin"></i>  
  <i class="fas fa-sync fa-spin"></i>  
  <i class="fas fa-cog fa-spin"></i>  
  <i class="fas fa-spinner fa-pulse"></i>  
  <i class="fas fa-stroopwafel fa-spin"></i>  
</div>
```

Библиотека **Font Awesome** содержит ряд эффектов которые могут быть полезным, в первую очередь для анимация для загрузчиков.

<https://fontawesome.com/how-to-use/on-the-web/styling/animating-icons>

<https://cdnjs.com/libraries/font-awesome>

Домашнее задание

Домашнее задание



У вас есть изображение дороги и автомобиля. Создайте CSS-мультфильм о том как автомобиль едет по дороге. *П.С. можно добавить еще автомобилей...*
П.П.С. Адаптивность не требуется, ориентируемся только на десктоп.

<http://files.courses.dp.ua/web/css-animation/road.png>
<http://files.courses.dp.ua/web/css-animation/car.png>