

Anatomy of a CSS Animation

What is a CSS animation?

CSS animations allow you transform elements of your webpages with different effects that before would have been created with Javascript or Flash before that. You can do things like move a shape across a page, fade it in and out, change its color, make it grow in size, and more.

How does it work?

First, you define what happens at different stages of your animation using **@keyframes**. In this example, we will be making text fade in when the page loads, so we will be animating the opacity of that element.

```
@keyframes my-animation {  
  0% {  
    opacity: 0;  
  }  
  100% {  
    opacity: 1;  
  }  
}
```

This is the name of your animation. Call it whatever makes sense to you. You will apply it later to the elements you want to animate.

0% indicates what the element will look like when the animation starts. In this case, because the opacity is 0, it will not be visible.

100% indicates what the element will look like at the end of the animation. In this case, because the opacity is 1, it will be visible.

Since this animation example just has a beginning and an end and a fade in between, you only need to define its appearance at 0% and 100%. But if you were doing something like making text cycle through three colors, you could define them at 0%, 50%, and 100%.

```
@keyframes my-color-animation {  
  0% {  
    color: red;  
  }  
  50% {  
    color: yellow;  
  }  
  100% {  
    color: blue;  
  }  
}
```

Next, you apply the keyframe animation you defined to the element you want to animate.

Reference the name of the animation you created earlier

This is how long you want your animation to take to complete, or how long to from 0% to 100%

This is how many times you want your animation to repeat. You can put a specific number or infinite if you want it to keep repeating.

```
.element {  
  animation-name: my-animation;  
  animation-duration: 5s;  
  animation-iteration-count: 1;  
}
```

This is a basic example, but there are additional animation properties you can define.

| | |
|---------------------------|--|
| animation-timing-function | ease, ease-out, ease-in, ease-in-out, linear, cubic-bezier(x1, y1, x2, y2) (e.g. cubic-bezier(0.5, 0.2, 0.3, 1.0)) |
| animation-duration | Xs or Xms |
| animation-delay | Xs or Xms |
| animation-iteration-count | X |
| animation-fill-mode | forwards, backwards, both, none |
| animation-direction | normal, alternate |
| animation-play-state | paused, running, running |

Source: <http://css-tricks.com/almanac/properties/a/animation/> CSS Tricks is a great resource!

You can also put all of your animation properties together in one line using the shorthand animation property.

```
.element {  
  animation: my-animation 5s 1;  
}
```

Activity

Open <http://go.unl.edu/css-animation> in your Chrome web browser. Use the resources on the page to create your own animated hero graphic!