



Transitions & Animations

Summary

In this episode...

- CSS3 Transitions
- Differences between transitions and animations
- CSS3 Animations
- Animations Keyframes



CSS3 Transitions

With CSS3 transitions you have the potential to alter the appearance and behavior of an element whenever a state change occurs, such as when it is hovered over, focused on, active, or targeted.



Differences between transitions and animations

Triggering

- CSS **transitions** need to be triggered. They are reactive and kick in when you interact with the site.
- CSS **animations** are created to start running when the site is loaded.

Differences between transitions and animations

Looping

- **Transitions** don't loop. They might do it but they're created to run once.
- **Animations** loop.



Differences between transitions and animations

Intermediate Points

- **Transitions** only do one thing. The item has a starting value and an ending value. Points between them are interpolated.
- **Animations** can have several intermediate points. They allow you to define keyframes and you have more control of what the animation does at any given time.

Differences between transitions and animations

JavaScript Interaction

- CSS **Transitions** are very JavaScript friendly. You can easily manipulate their behaviour.
- Friends don't let friends write JavaScript to alter an **animation**. The syntax for manipulating a frame is hopelessly complex.

Choosing between animation and transition

- If you need the flexibility provided by having multiple keyframes and looping, then go with an **animation**.
- If you're looking for a simple from/to animation, use a **transition**.
- If you want to manipulate the property values that you wish to animate using JavaScript, try a **transition**.

Transitions

Let's code

- Let's add some transitions to our IronHack button. You will find it by downloading [this](#)




Transitions

Transform: translate

The CSS **transform** property lets you modify the coordinate space of the CSS visual formatting model. Using it, elements can be **translated**, rotated, scaled, and skewed.

```
transform: translate(0px, 5px);
```

*This rule changes
object's position 5px
vertically from its
initial position*



Transitions

Border:

To make the effect more realistic, we will change the border of our bottom. If we make it thinner, it will look like if we press it, so we should change the border from 5px to 1px:

```
border-bottom: 1px solid;
```

Transitions

For a **transition** to take place, an element must have a change in state, and ***different styles*** must be identified for each state.

The easiest way for determining styles for different states is by using the : **hover**, **:focus**, **:active** and **:target pseudo-classes**.

Transitions

Let's code

This rule will be applied when the button status is active

```
.action-button:active
{
  transform: translate(0px,5px);
  -webkit-transform: translate(0px,5px);
  border-bottom: 1px solid;
}
```

Transitions

Transitions have 4 properties:

It defines the duration in which a transition takes place

- transition-property
- transition-duration
- transition-timing-function
- transition-delay

Determines exactly what properties will be altered in conjunction with the other transitional properties

Sets a time value to specify how long a transition should wait before executing

It is used to set the speed in which a transition will move

Not all of them are required to build a transition

Transitions

Let's code

We will create a new transition for our button when we hover on it. Let's be creative and use transitions properties to change the background. First we will add the transition in the button style

```
.action-button
{
    transition-property: background;
    transition-duration: .6s;
    transition-timing-function: linear;
    transition-delay: 0s;
}
```

Transitions

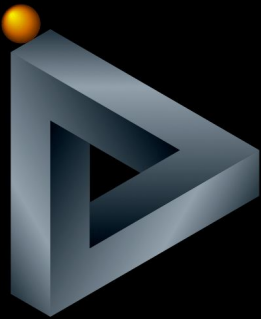
Let's code

Now we will add the transition for the hover state

```
.action-button:hover  
{  
    background: #FF7B29;  
}
```


Animations





Animations

When more control is required and we need to have multiples states, transitions are not enough.

Animations consist of **two** components, a **style describing the CSS animation** and a **set of keyframes** that indicate the start and end states of the animation's style, as well as possible intermediate waypoints.

Animations Keyframes

To create a CSS3 animation, you will need a specialized CSS @ rule that starts with a @, followed by a name of your choice.

```
@keyframes sunrise {  
    //Your code goes here  
}
```

Animations Keyframes

To set multiple points for our animation, you should include animation breakpoints and the properties intended to be animated. Here's an example:

```
@keyframes sunrise {  
  0% {  
    bottom: 0;  
    left: 340px;  
    background: #f00;  
  }
```

Animations Keyframes

Let's code

Create a circle with a div element in the same files you were using for your button exercise

```
.circle
{
    background-color: #09B8C1;
    height: 50px;
    width: 50px;
    border-radius: 100%;
    index-left: 250px;
}
```

Animations Keyframes

Let's code

Create an animation called fly to be used in our circle element

*0% defines the
begin of the
animation and
100% defines
where it ends*

```
@keyframes fly {  
  0%, 100% {  
    transform: rotate(0deg) translateX(0px);  
  }  
  50% {  
    transform: rotate(0deg) translateX(150px);  
  }  
}
```

*If a set of rules
applies for 0% and
100%, the
animation starts
and ends in the
same style*

Animations Keyframes

Let's code

Now, add the animation to your circle element. You can set up all these features to your animation:

```
animation-name: fly;  
animation-duration: 4s;  
animation-iteration-count: 3;  
animation-direction: alternate;  
animation-timing-function: ease-out;  
animation-fill-mode: forwards;  
animation-delay: 2s;
```

Animations Keyframes

Exercise

Change your animation to make the circle move in circles instead of a straight line



Want some more?

- A very colorful donut [here](#)
- [BB-8](#) pure CSS animation
- Floating in CSS [jellyfishes](#)

