

CSS Animation

Animation Basics

Animations in CSS are handled in two major ways:

- Transitions - for simple animations
- KeyFrames - for more complex animations

Transitions

Transitions are simple animations that can be created for when the CSS properties change from one state to another.

Transition Properties

- `transition-property` - the name or names of the CSS properties to which transitions should be applied
- `transition-duration` - the amount of time a transition lasts
- `transition-delay` - the amount of time to wait before initiating a transition
- `transition-timing-function` - the animation function to use to initiate the transition (e.g `easeOutCubic`, `easeInSine`)

Transition Example

```
/*
  sets the width, height, background-color and transform properties
  to be transitioned over 1s
*/
.box {
  transition-property: width, height, background-color;
  transition-duration: 1s;
}

/* when hovering, the properties defined above will transition */
.box:hover {
  background-color: #FFCCCC;
  width: 200px;
  height: 200px;
}
```

Keyframes

Keyframes allow us to have more control over how animations take place. We can define the start state, end state as well as intermediate states of an animation; then the browser takes care of the rest.

Keyframe Properties

- `animation-name` - the name you chose for the animation
- `animation-duration` - the amount of time an animation lasts
- `animation-delay` - the amount of time to wait before initiating an animation

More Properties

Keyframes Example

```
/*
  sets the keyframe animation and the duration
*/
p {
  animation-name: slidein;
  animation-duration: 3s;
}

/* defines the keyframe animation */
@keyframes slidein {
  /* we start with the element at this state */
  from {
    margin-left: 100%;
    width: 300%;
  }

  /* we end with the element in this state */
  to {
    margin-left: 0%;
    width: 100%;
  }
}
```


Resources

[Mozilla Developer Network - CSS Transitions](#)

[Mozilla Developer Network - CSS Animations](#)