

CSS Animation

Web Design | HW Summer

2021-06-30 | Henry Woody

Intro

- Animations can make our sites feel more alive and make transitions feel more natural
- Can also use animation to make an element stand out relative to others
- Useful when content is loading

Transition

- One way to animate an element is to set styling on that element when in a different state (different class or pseudo-class) and set the transition property
- When the element gets a new style applied to it (via a state change), CSS will animate the transition between the two styles
- Typically only used in response to user-events (hover, click, etc)

```
.element {  
  color:  black;  
  transition: color .25s ease 0s;  
}  
  
.element.active {  
  color:  red;  
}  
  
.element:hover {  
  color:  blue;  
}
```

Keyframes

- To animate an element continuously or when it first is loaded/created, we can define a keyframes animation
- Keyframes take a set of style rules and transition between them

```
.spinning {  
  animation: spin 2s linear 0s infinite;  
}  
  
@keyframes spin {  
  from {  
    transform: rotate(0deg);  
  }  
  
  to {  
    transform: rotate(360deg);  
  }  
}
```

```
.moving {  
  animation: move-around 10s linear 0s 2 alternate both;  
}  
  
@keyframes move-around {  
  0% {  
    transform: translate(0, 10rem);  
  }  
  
  50% {  
    transform: translate(10rem, 10rem);  
  }  
  
  100% {  
    transform: translate(0, 0);  
  }  
}
```

Restrictions

- Not all properties are animatable
 - For example, there is no way to animate between `display: inline;` and `display: block;`
 - Properties with numeric units are animatable
- To check if a property is animatable, check the documentation
- However, non-animatable properties can still be included in animation styles, they will just transition instantaneously rather than gradually

Animation Performance

- We want animations to be as close to 60fps on most computers as possible
- For best performance when animating, it's best to only animate the following properties:
 - transform
 - opacity
- Changing layout properties causes CSS to have to recalculate the layout of all other elements on the page, which slows our animations down
 - examples: height, width, padding, margin, position, display