

CSS IN JAVASCRIPT

TOPICS 7 AND 8 BY MELISSA HOLLAND

2 WAYS TO ALTER CSS PROPERTIES IN JAVA

- Setting Styles Directly
- Adding and removing class values

SETTING STYLES DIRECTLY

- Every DOM element has a style object that allows you to set CSS properties on them directly.
- Single Element Example: (this is what setting the background color of an HTML element whose id value is **superman** looks like)

```
varmyElement = document.querySelector("#superman");
myElement.style.backgroundColor ="#D93600";
```

- Multiple Elements Example:

```
varmyElements = document.querySelectorAll(".bar");
for(vari = 0; i<myElements.length; i++) {
    myElements[i].style.opacity = 0;
}
```

SPECIAL CASING SOME NAMES IN JAVA

- Anything that contains a dash in CSS like background-color needs to be changed to camel case and written without a dash like so: backgroundColor
- Certain CSS properties like the float property need to be written as cssFloat in Java because float is a reserved word in Java that already has a different meaning

ADDING & REMOVING CLASS VALUES

- Let's say you have a style rule that looks as follows:

```
.disableMenu {  
    display:none;  
}
```

In html you have a menu whose ID is dropDown:

```
<ul id="dropDown">  
    <li>One</li>  
    <li>Two</li>  
    <li>Three</li>  
    <li>Four</li>  
    <li>Five</li>  
    <li>Six</li>  
</ul>
```

CONTINUED

Now, if we wanted to apply our `.disableMenu` style rule to this element, all you would need to do is add **disableMenu** as a class value to the **dropDown** element:

```
<ul class="disableMenu" id="dropDown">  
    <li>One</li>  
    <li>Two</li>  
    <li>Three</li>  
    <li>Four</li>  
    <li>Five</li>  
    <li>Six</li>  
</ul>
```

DO THIS SAME THING IN JAVA WITH CLASSLIST API

To add the **disableMenu** class name to our **dropDown** element, use the `add` method on the HTML element's `classList` property:

```
var theDropDown = document.querySelector("#dropDown");
theDropDown.classList.add("disableMenu");
```

To remove the the **disableMenu** class name, we can call the `classList` API's `remove` method:

```
var theDropDown = document.querySelector("#dropDown");
theDropDown.classList.remove("disableMenu");
```

CREATING ANIMATIONS IN CSS

Adding a CSS animation is made up of two steps. The first step is to set the animation property, and the second step is to define the keyframes that specify exactly what gets animated.

From the markup you added, find the `#bigCloud` style rule add the following highlighted line:

```
animation: bobble 2s infinite;
```

The animation property is made up of 3 things:

- The name of your animation
- The duration
- The number of times your animation will loop

Next, add the following `@keyframes` style rule:

(See pic to the right)

```
@keyframes bobble {  
    0% {  
        transform: translate3d(50px, 40px, 0px);  
        animation-timing-function: ease-in;  
    }  
    50% {  
        transform: translate3d(50px, 50px, 0px);  
        animation-timing-function: ease-out;  
    }  
    100% {  
        transform: translate3d(50px, 40px, 0px);  
    }  
}
```

ANIMATIONS CONTINUED

- In our example, the percentage values for our keyframe selectors are 0%, 50%, and 100%. What they represent is the percentage of the animation that has completed. When your animation is just starting, you have completed 0% of your animation. The 0% keyframe will become active. When your animation is half-way done, the 50% keyframe becomes active. At the end of your animation, the 100% keyframe becomes active instead.
- Show example working

CREATING TRANSITIONS IN CSS

- Transitions slow down the sudden change in properties. They allow you to specify how long a particular property change will take place. They allow you to specify what kind of easing will be applied in going from the current property value to another. **Transitions basically allow you to animate the property value changes.**
- Add this line of code to create a transition:
 - `transition: transform .5s ease-in;`
 - Show working example

CONTINUED

- A typical CSS transition defines the following three properties:
 - The property to apply the transition to
 - How long the transition will last
 - What kind of a timing function (aka easing function) to use

The diagram illustrates the structure of a CSS transition declaration. It shows the word "transition:" followed by the value "transform .5s ease-in;". Three horizontal brackets with labels point to specific parts of the value:

- A bracket above "transform" is labeled "transition property".
- A bracket above ".5s" is labeled "transition duration".
- A bracket above "ease-in;" is labeled "transition timing function".

```
transition: transform .5s ease-in;
```

transition property transition timing function
transition duration

BENEFITS TO ADDING JAVASCRIPT

- Having our animations play at a time other than during page load.
- Triggering our transitions outside of simple gestures like hover.
- Defining the properties we animate, their values, or both on the fly as opposed to having them always be predefined in our CSS.
- Having the element whose properties we animate be different than the element we trigger the animation on.

TRIGGERING ANIMATIONS/TRANSITIONS IN JAVA

- Essentially you will be following this pattern:
 - Define the initial state using CSS.
 - Make changes using JavaScript.
- Show working example and the Java code added to the doc.