Teaching Presentation for CIT 261-week 8

CSS3 Transitions and Animations

triggering them with JavaScript

Cristina Irwin

CSS Properties

The CSS properties can be accessed using the **DOM** by using the *.style* element in JavaScript. By adding ***.transition,*** ***.transform*** or ***.animation*** to the *.style,* **javascript** can trigger the transitions, transformations and/or animations. Using Javascript allows for targeted interaction because the javascript calls the transitions, transformations and animations.

Example of a JavaScript triggering CSS properites.

```
const box = document.getElementById("box");
const btngo = document.getElementById("btngo");
const btnback = document.getElementById("btnback");

btngo.addEventListener("click", () => box.style.transform ="rotate(90deg)");
btngo.addEventListener("click", () => box.style.backgroundColor ="rgba(255,0,0,0)");

btnback.addEventListener("click", () => box.style.transform ="rotate(-90deg)");
btnback.addEventListener("click", () => box.style.backgroundColor ="rgba(113,23,114,1)");
```

Animations

Animations contain a style and a set of @keyframes.

Style

The animation property is shorthand for a **collection of properties.** Those properties are **duration, timing function, delay, iteration-count, direction, fill-mode, play-state** and **name.**

Duration is how long the animation lasts. The **timing function** determines how fast the animation moves depending on the progress of the animation. Animation **delay** is how long to wait for animation to start (in seconds or milliseconds). The **iteration count** is how many cycles the animation goes through. Animation **direction** refers to the behavior between cycles (normal, reverse, alternate and alternate-reverse). **Fill-mode** refers to how the element will be styled after the element completes the animation (styles last frame). **Play-state** controls whether the animation is playing or is paused. **Name** refers to the behavior of an animation.

```
.object {
    height: 300px;
    width: 300px;
    margin: 0 auto;
    background-color: purple;
    animation-name: change;
    animation-duration: 2.5s;
    animation-timing-function: linear;
    animation-delay: 0s;
    animation-direction: alternate;
    animation-iteration-count: infinite;
    animation-fill-mode: none;
    animation-play-state: running;
}
```

Animations

Animations contain a style and a set of @keyframes.

@keyframes

Each @keyframe defines what happens at a specific moment (0%, 25%, 75%, 100% and anywhere in between).

```
@keyframes change {
    0% {
        transform: scale(.1);
        background-color: purple;
        border-radius: 100%;
    }

    20% {
        background-color: blue;
    }


    50% {
        background-color: green;
    }

    100% {
        transform: scale(1.5);
        background-color: yellow;
    }
}
```

Transitions

Transitions are changes that allow property values to change over time.

Transitions properties are **name** (name of CSS property that is being affected), **duration** (how many seconds or milliseconds the transition takes), **timing-function** (speed) and **delay** (transition will start).

Transformations

Transform property applies to 2D or 3D elements. A transformation is an effect that lets an element change shape, size and position.

Transform properties are **rotate** (transform rotates an element clockwise or counterclockwise by a degree number)**, skew** (tilts an object based on the x and y axes values)**, translate** (moves the element right, left, down or up)**, scale** (transform increases or decreases the size of an element.)**.**

rotate

```
.selected {
    background-color: lightgreen;
    transform: rotate(360deg);
    }
```

scale

```
#scale {
    transform: scaleX(1.5);
    transform: scaleY(0.4);
}
```

Triggering CSS properties with

JavaScript

JavaScript can be used to trigger the animation, transition or transformation by using the DOM and the .style tag and one of the sub properties.

HTML

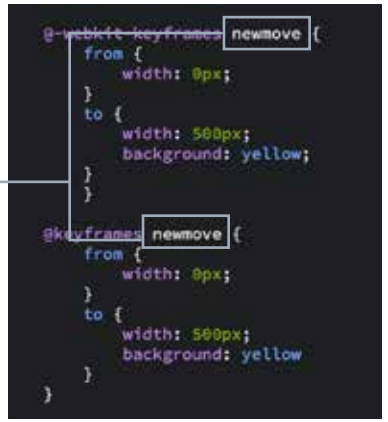
```
<button onclick="stretch()">Try it</button>
```

JavaScript

```
function stretch() {
    document.getElementById("box").style.WebkitAnimationName = "newmove";
    document.getElementById("box").style.animationName = "newmove";
    document.getElementById("text").innerHTML = "Caution";
```

animation name:
stretch() changes the animation name to **newmove**

CSS



Triggering CSS properties with

JavaScript

JavaScript can be used to trigger the transformation. This example uses an event listener.

HTML

```
<h1>Event Transition</h1>
<div id= "box"></div>
<button id="btngo">Watch me Disappear</button>
<button id="btnback">Watch me Re-appear.</button>
```

button is clicked

CSS

```
#box {
  margin: 20px;
  width: 200px;
  height: 200px;
  background: rgba(113,23,114,1);
  transition-property: transform background;
  transition-duration: 3s;
  transition-delay: 1s;
}
```

JavaScript

```
const box = document.getElementById("box");
const btngo = document.getElementById("btngo");
const btnback = document.getElementById("btnback");

btngo.addEventListener("click", () => box.style.transform ="rotate(90deg)");
btngo.addEventListener("click", () => box.style.backgroundColor ="rgba(255,0,0,0)");

btnback.addEventListener("click", () => box.style.transform ="rotate(-90deg)");
btnback.addEventListener("click", () => box.style.backgroundColor ="rgba(113,23,114,1)");
```

transform: when button is clicked, change rotate element.

Triggering CSS properties with

JavaScript

JavaScript can be used to trigger the transition.

element

CSS

```
.buttoncontainer {
    margin: 10px;
}
```

HTML

```
<button class="buttoncontainer" onclick="fadeOutColor('box1')">Fade Out</button>
```

Function fadeOutColor

JavaScript

```
function fadeOutColor(el){
    var elem = document.getElementById(el);
    elem.style.transition = "opacity 1.0s linear 0s";
    elem.style.opacity = 0;
}
```

changes the transition