Dashboard Circle Rotate

Development

November 26, 2019

Overview

This document is intended to explain how I developed the rotating the SVG Icons.

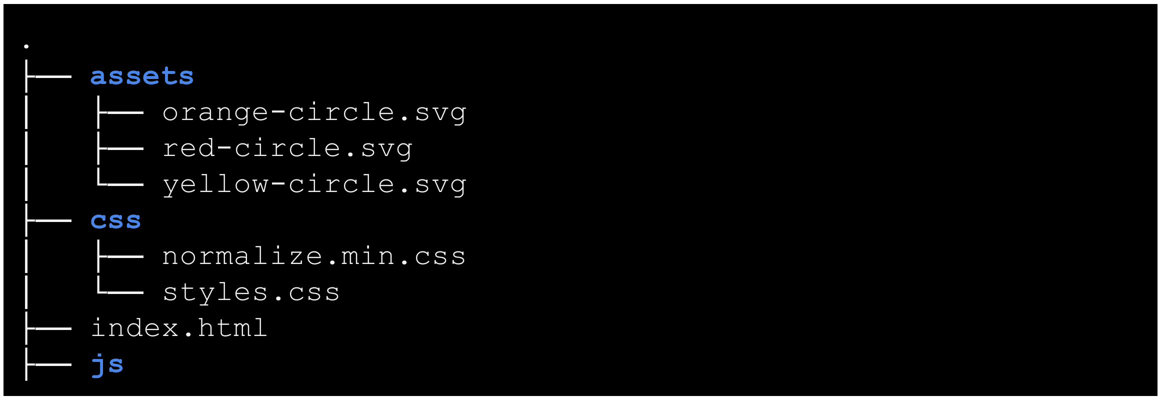
Steps (Pseudo Code)

1. **Download the SVG icons and group them:** Upon downloading the SVG icons, I grouped the multiple svg icons into one svg icon by color.
   a. yellow-circle.svg (consists of all svgs related to yellow circle)
   b. red-circle.svg (consists of all svgs related to red circle)
   c. orange-circle.svg (consists of all svgs related to orange circle)

Upon grouping them, I assign a specific ID for every specific circle (outer-circle, middle graph etc).

I used this tool to group them as export as SVG.

2. **Setup VS code for the actual coding:** This is the directory structure I initially created.



```
.
├── assets
│   ├── orange-circle.svg
│   ├── red-circle.svg
│   └── yellow-circle.svg
├── css
│   ├── normalize.min.css
│   └── styles.css
├── index.html
├── js
```

```
│   └── main.js
└── README.md

3 directories, 8 files
```

3. **Create index.html:** I used this boilerplate to structure my index.html and its DOM.

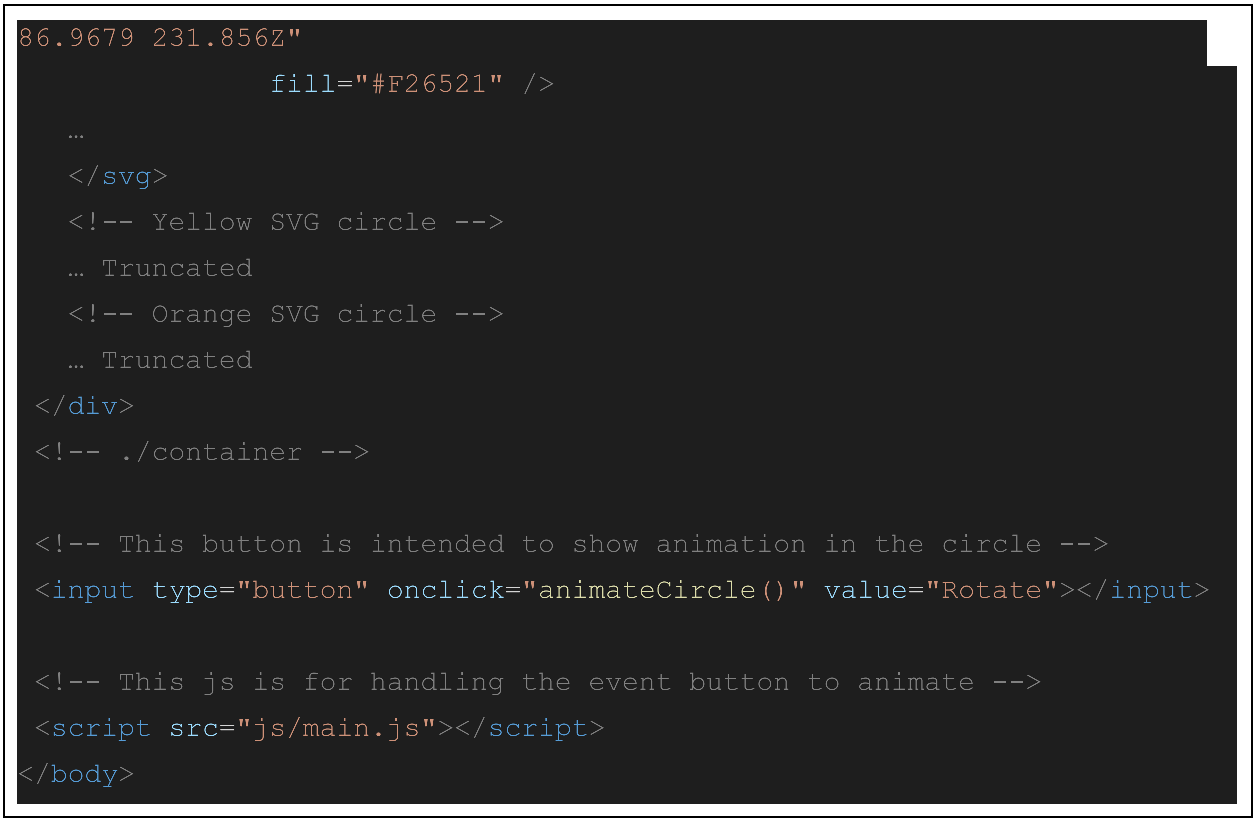
4. **Create css:** the css files consists of styles.css and normalize.min.css. **styles.css** is for my custom page layout and animation itself, whereas the **normalize.min.css**, which I downloaded via CDN is for cross browser consistency for default html styling.

5. **Create javascript:** The javascript file main.js is just for event trigger. Once you clicked the button, the circle will animate by assigning the css class for animation.

6. **Add SVG circles to html:** The svg circle files are added in index.html with its specific id attributes.

HTML Code Snippet

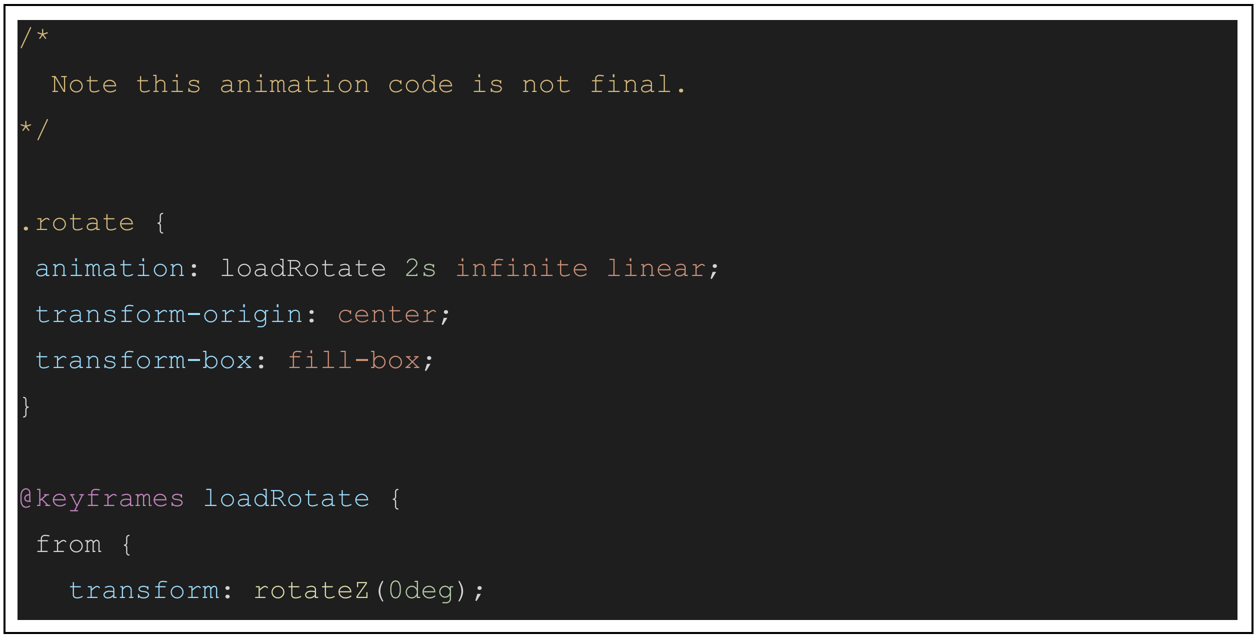
```
<body>
 <div class="container">
   <!-- Red SVG circle -->
   <svg width="300" height="242" viewBox="0 0 300 242" fill="none"
xmlns="http://www.w3.org/2000/svg">
     <g id="red-whole-circle">
       <g id="red-inner-circle">
         <g id="red-inner-circle-icons">
           <g id="-red inner-circle-graph">
             <path id="Line" d="M114.5 188.5L88.0969 232.963"
stroke="white" stroke-linecap="square" />
             <path id="Rectangle" d="M91.528 240.756L96.7885
229.97L86.0029 224.71L80.7425 235.496L91.528 240.756Z"
               stroke="#F26521" />
             <path id="Oval"
               d="M86.9679 231.856C86.4837 232.849 86.896 234.046
87.8887 234.531C88.8815 235.015 90.0789 234.603 90.5631 233.61C91.0473
232.617 90.635 231.42 89.6422 230.935C88.6494 230.451 87.4521 230.863
```



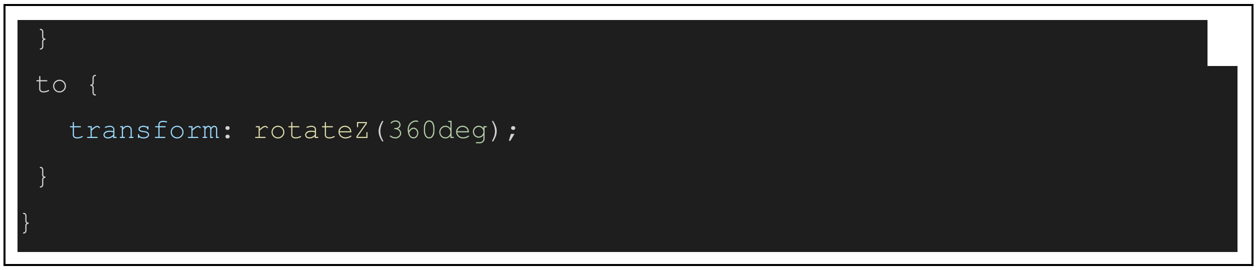
```
86.9679 231.856Z"
               fill="#F26521" />
   …
   </svg>
   <!-- Yellow SVG circle -->
   … Truncated
   <!-- Orange SVG circle -->
   … Truncated
 </div>
 <!-- ./container -->

 <!-- This button is intended to show animation in the circle -->
 <input type="button" onclick="animateCircle()" value="Rotate"></input>

 <!-- This js is for handling the event button to animate -->
 <script src="js/main.js"></script>
</body>
```

7. **Animate using css:** The method I used for rotating the circles is CSS animations. Using @keyframes and the animation properties.

CSS Code snippet



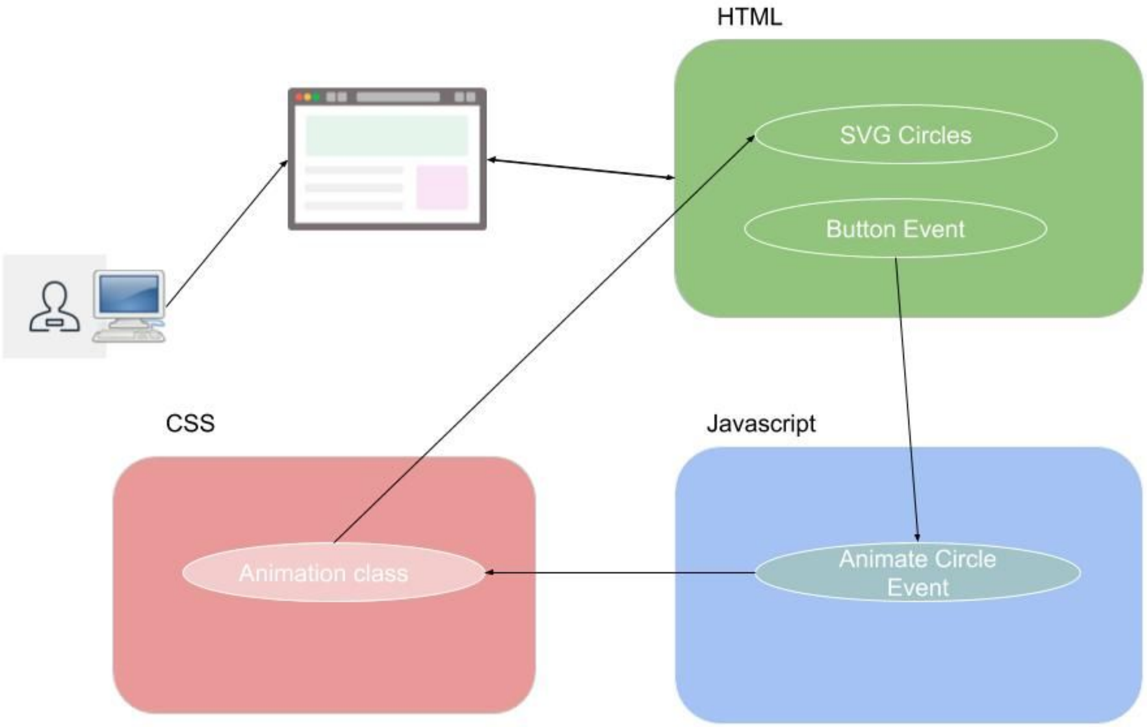
```
/*
  Note this animation code is not final.
*/

.rotate {
 animation: loadRotate 2s infinite linear;
 transform-origin: center;
 transform-box: fill-box;
}

@keyframes loadRotate {
 from {
   transform: rotateZ(0deg);
```



```
  }
  to {
    transform: rotateZ(360deg);
  }
}
```

8. **Rotate the circle on button click using javascript.** Created a button with onclick function. Upon click, I assign the **.rotate css class** to svg circle id to rotate.

9. **Render the html to browser.** I used to python's simple http server to create a localhost server and serve the html.


HTML
SVG Circles
Button Event
CSS
Javascript
Animation class
Animate Circle Event