Table of Contents

Overview of Scrollytelling Project

Steps to take

1. Animate in Adobe After Effects.

2. Install the bodymovin add on.

3. Convert animation to json file.

4. Drag and drop it in lottie website to view the animation.

5. Use code to animate it on scroll.

Lottie Animations

Desktop

01_Cover_d - https://app.lottiefiles.com/share/ed117a21-f5aa-46bb-99af-a7cec72f536c

02_Gov_d - https://lottiefiles.com/share/jgitxsov

03_Gov_d - https://lottiefiles.com/share/pzzy0w7w

04_Cover_d - https://app.lottiefiles.com/share/3f734b3b-a8d7-4f6b-9486-fb11ec80f21d

05_LovedOnes_d - https://lottiefiles.com/share/dftwvmpm

06_Key Loved Ones_d - https://app.lottiefiles.com/share/dbe5fa3f-de12-458a-8e82-bad995e2d3e3

07_Cover_d - https://app.lottiefiles.com/share/40d05ab7-456e-494d-a361-10c9989dc36e

08_GovLovedOnes_d- https://lottiefiles.com/share/tht2czrc

09_Cover_d - https://app.lottiefiles.com/share/83069614-a4f8-4ed9-bc22-ae422e235707

Mobile

01_Cover_m - https://app.lottiefiles.com/share/df5d8885-3de0-4352-bf70-51dffd02c277

02_Gov_m - https://app.lottiefiles.com/share/14f92251-7abb-4d54-be49-cc2950b21acf

03_Gov_m - https://app.lottiefiles.com/share/c9822f80-7cd3-46a3-b406-ff1a275545e8

04_Cover_m - https://app.lottiefiles.com/share/00ec2a22-d0ef-4869-a1d4-5d2a7a0cc1ec

05_Loved Ones_m - https://app.lottiefiles.com/share/6b891943-b1cf-4d1e-92f8-d013b2afa759

06_Loved Ones_m - https://app.lottiefiles.com/share/283d3926-a780-44da-a86f-942a048938dd

07_Cover_m - https://app.lottiefiles.com/share/f5acec54-1d49-4015-8bc7-c4be30d5c6a1

08_GovLovedOnes_m - https://app.lottiefiles.com/share/6acb9ebe-58e2-4ff9-8943-bf813835bd10

09_Cover_m - https://app.lottiefiles.com/share/a931a8fa-88a9-43a7-9357-244845f5ffe7

Bodymovin

1. Bodymovin is an Adobe After Effects extension. It is used to export Adobe After Effects animations as .json files. The download link is in here https://aescripts.com/bodymovin/.
2. Not all Adobe After Effects features are supported. The supported features are as listed in the link above.
3. While we were testing the webpage on different devices, we realised that a mask effect in adobe after effect when exported as json files, caused a huge lag in ios devices. Do take note of this, and make sure the exported json files are not too big to reduce lag. (There are steps that can be taken in after effects to reduce a file size, for example using a smaller fps like 25fps etc…).

JavaScript Libraries

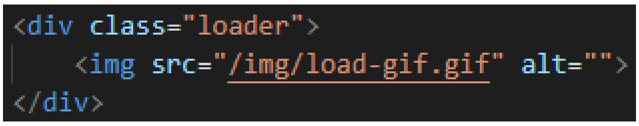
1. Lottie-web is used to load the .json files onto the webpage.
2. ScrollTrigger by GSAP is used to sync the .json files (the animations) to the user's scroll, as well as to set different parameters to tweak the scrollytelling experience.
3. The links to the respective libraries are as follows.
   a. Lottie-Web: https://github.com/airbnb/lottie-web/wiki/loadAnimation-options
   b. ScrollTrigger: https://greensock.com/docs/v3/Plugins/ScrollTrigger
4. The code that I used for scrollytelling is a combination of both lottie-web and scrolltrigger. Most of the code that I have used came from here. https://github.com/chrisgannon/ScrollLottie. I will explain the codes more in depth later in this guide.

Useful Links for Beginners to Web Development

1. Knowing basic HTML, CSS and JavaScript will help greatly.
2. These are the two crash courses I feel is good for beginners to web development.
    a. HMTL&CSS: https://www.youtube.com/playlist?list=PL4cUxeGkcC9ivBf_eKCPIAYXWzLlPAm6G
    b. JavaScript: https://www.youtube.com/playlist?list=PL4cUxeGkcC9haFPT7J25Q9GRB_ZkFrQAc
    c. You do not need to watch all the videos, just watch to the point you feel that you understand enough to continue on with this project.

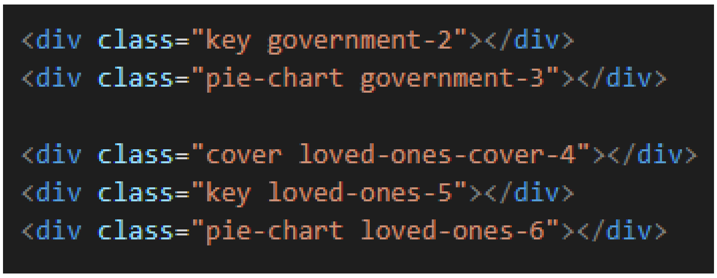
HTML&CSS

I used divisions (div) to hold different animations.



```
<div class="loader">
    <img src="/img/load-gif.gif" alt="">
</div>
```

The loader class is used to house the loading placeholder. This gif I used will need to be replaced by cpf's current placeholder.

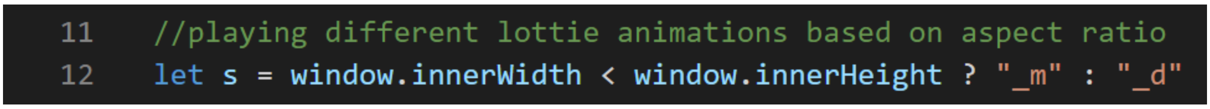


```
<div class="key government-2"></div>
<div class="pie-chart government-3"></div>

<div class="cover loved-ones-cover-4"></div>
<div class="key loved-ones-5"></div>
<div class="pie-chart loved-ones-6"></div>
```

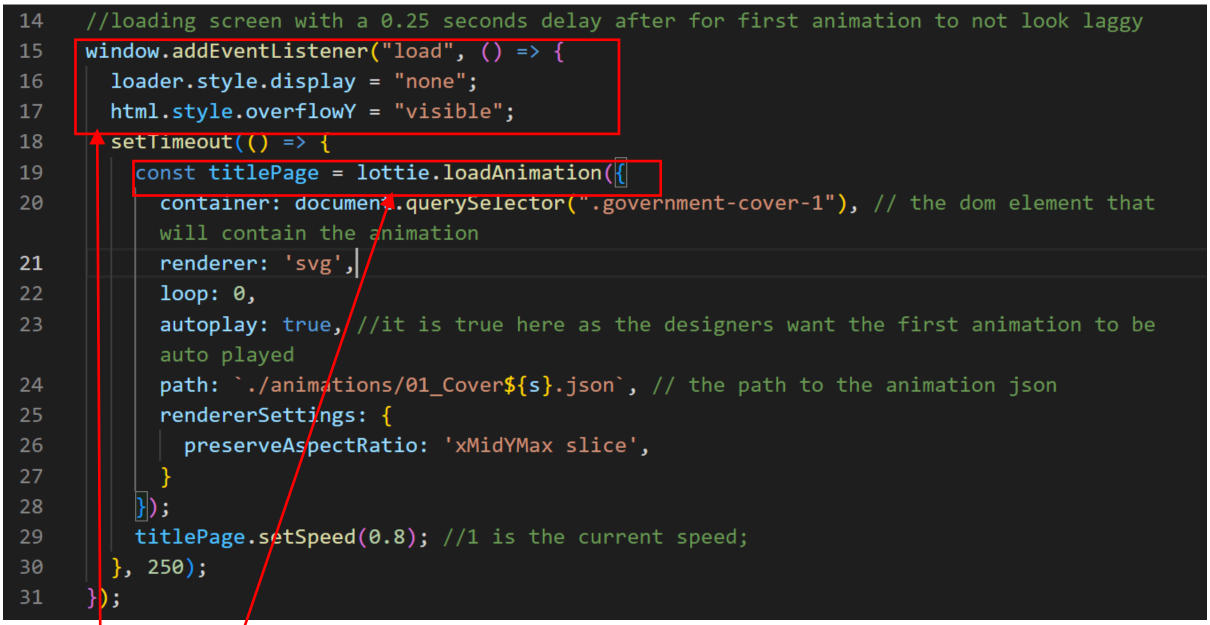
The rest of the divisions holds the different animations. Classes like government-2 and loved-ones-5 need to be created to be referred to later. Classes like key and pie-chart are being reused.

JavaScript In-Depth



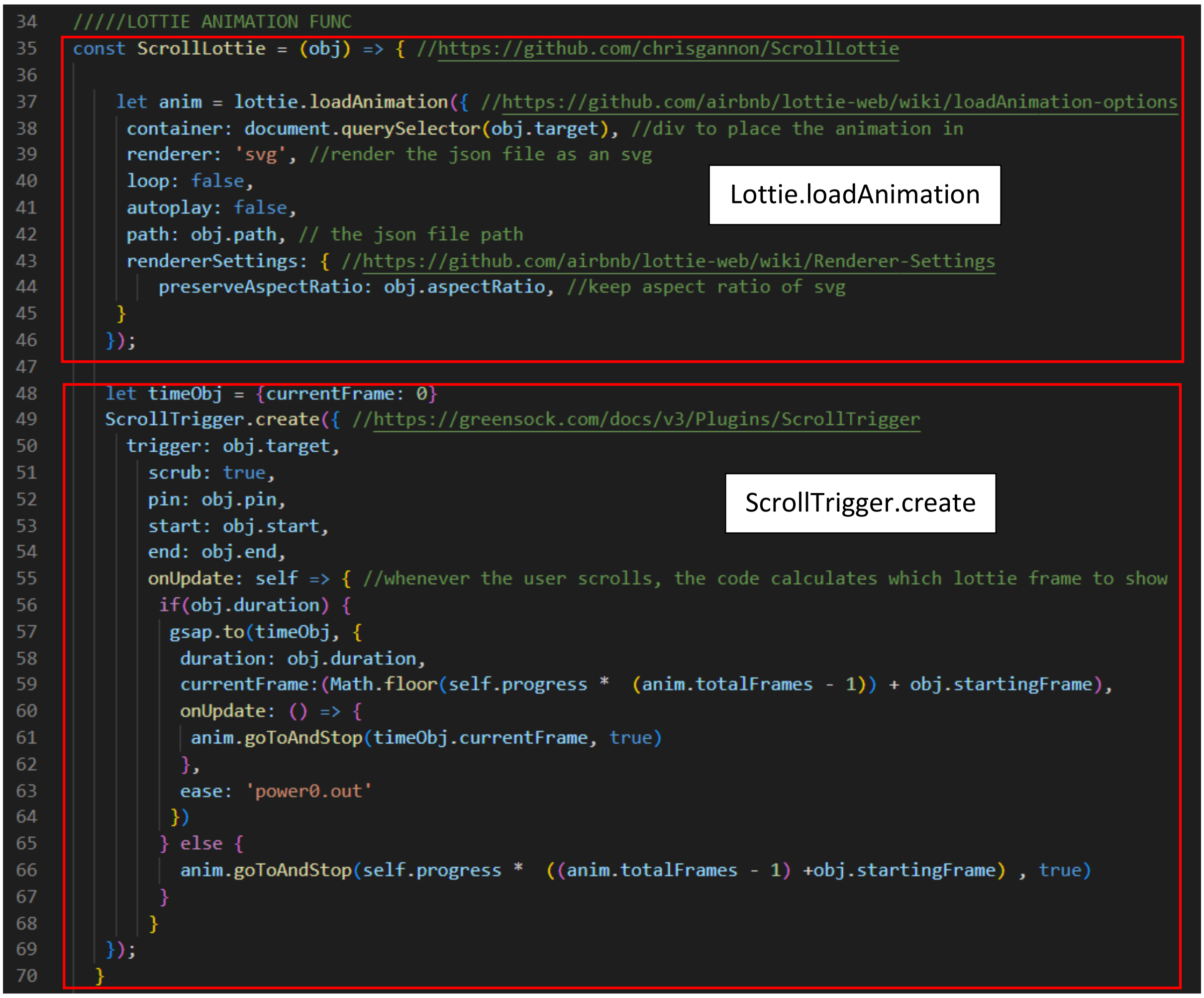
```javascript
//playing different lottie animations based on aspect ratio
let s = window.innerWidth < window.innerHeight ? "_m" : "_d"
```

This code is placed at the start, and it will choose which set of animations to play based on the aspect ratio of the viewport.



```javascript
//loading screen with a 0.25 seconds delay after for first animation to not look laggy
window.addEventListener("load", () => {
  loader.style.display = "none";
  html.style.overflowY = "visible";
  setTimeout(() => {
    const titlePage = lottie.loadAnimation({
      container: document.querySelector(".government-cover-1"), // the dom element that will contain the animation
      renderer: 'svg',
      loop: 0,
      autoplay: true, //it is true here as the designers want the first animation to be auto played
      path: `./animations/01_Cover${s}.json`, // the path to the animation json
      rendererSettings: {
        preserveAspectRatio: 'xMidYMax slice',
      }
    });
    titlePage.setSpeed(0.8); //1 is the current speed;
  }, 250);
});
```

This portion houses the code for the loading placeholder, as well as for the first animation to be played. What the code does is once the JavaScript is loaded, I will set the placeholder class display to be none and start the first animation.



```
/////LOTTIE ANIMATION FUNC
const ScrollLottie = (obj) => { //https://github.com/chrisgannon/ScrollLottie

    let anim = lottie.loadAnimation({ //https://github.com/airbnb/lottie-web/wiki/loadAnimation-options
     container: document.querySelector(obj.target), //div to place the animation in
     renderer: 'svg', //render the json file as an svg
     loop: false,
     autoplay: false,
     path: obj.path, // the json file path
     rendererSettings: { //https://github.com/airbnb/lottie-web/wiki/Renderer-Settings
        preserveAspectRatio: obj.aspectRatio, //keep aspect ratio of svg
    }
   });

   let timeObj = {currentFrame: 0}
   ScrollTrigger.create({ //https://greensock.com/docs/v3/Plugins/ScrollTrigger
     trigger: obj.target,
       scrub: true,
       pin: obj.pin,
       start: obj.start,
       end: obj.end,
       onUpdate: self => { //whenever the user scrolls, the code calculates which lottie frame to show
        if(obj.duration) {
         gsap.to(timeObj, {
          duration: obj.duration,
          currentFrame:(Math.floor(self.progress *  (anim.totalFrames - 1)) + obj.startingFrame),
          onUpdate: () => {
           anim.goToAndStop(timeObj.currentFrame, true)
          },
          ease: 'power0.out'
         })
        } else {
          anim.goToAndStop(self.progress *  ((anim.totalFrames - 1) +obj.startingFrame) , true)
        }
       }
   });
  }
```

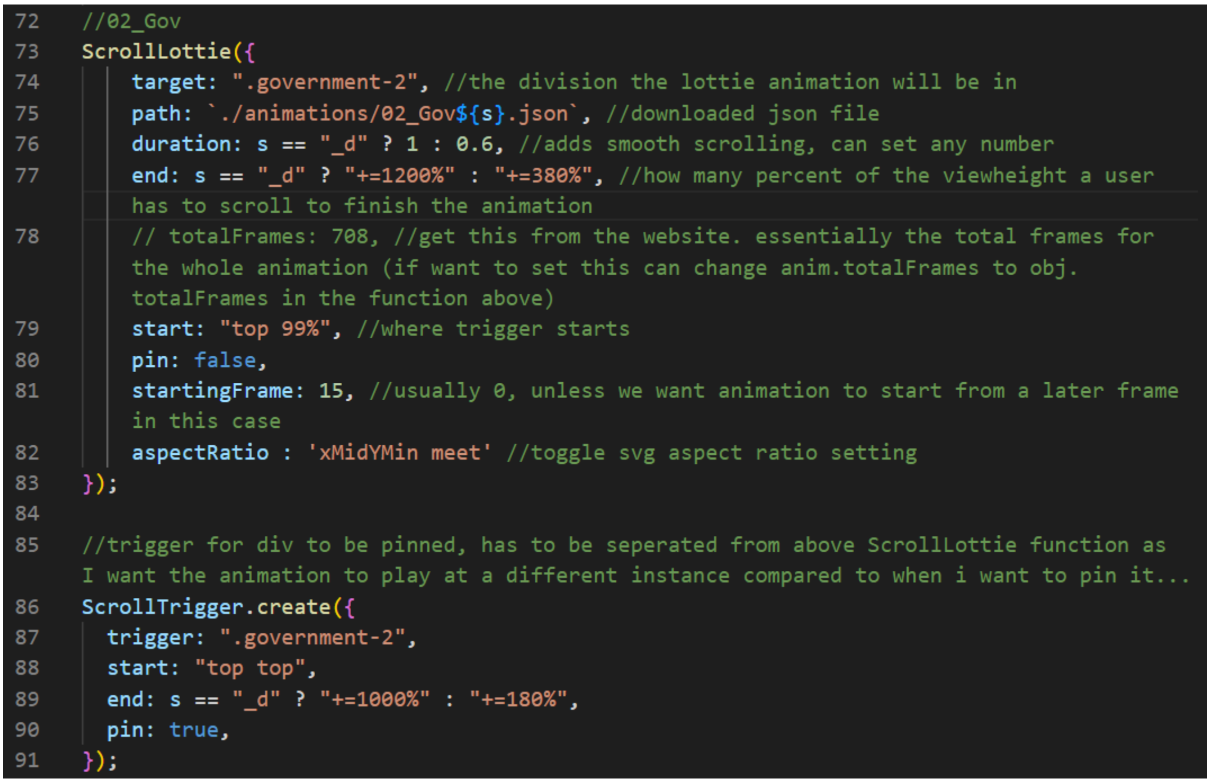
This is the code for the loading of lottie animations onto the webpage. It contains two functions from lottie-web and ScrollTrigger. lottie.loadAnimation (from lottie-web) is used to load lottie animations (json files) onto the webpage. and ScrollTrigger.create (from ScrollTrigger) allows us to set up different parameters for the scrollytelling experience.

The in-depth explanation of these codes can be found here.

a. Lottie-Web: https://github.com/airbnb/lottie-web/wiki/loadAnimation-options
b. ScrollTrigger: https://greensock.com/docs/v3/Plugins/ScrollTrigger

Reusable JavaScript Parameters

ScrollLottie(), ScrollTrigger.create()

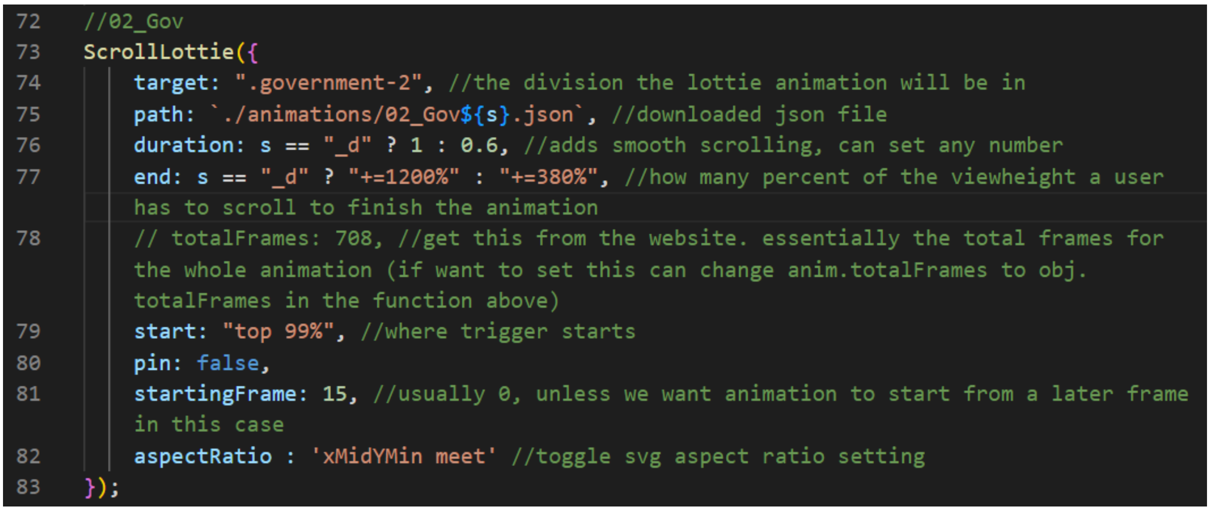


```
//02_Gov
ScrollLottie({
    target: ".government-2", //the division the lottie animation will be in
    path: `./animations/02_Gov${s}.json`, //downloaded json file
    duration: s == "_d" ? 1 : 0.6, //adds smooth scrolling, can set any number
    end: s == "_d" ? "+=1200%" : "+=380%", //how many percent of the viewheight a user has to scroll to finish the animation
    // totalFrames: 708, //get this from the website. essentially the total frames for the whole animation (if want to set this can change anim.totalFrames to obj.totalFrames in the function above)
    start: "top 99%", //where trigger starts
    pin: false,
    startingFrame: 15, //usually 0, unless we want animation to start from a later frame in this case
    aspectRatio : 'xMidYMin meet' //toggle svg aspect ratio setting
});

//trigger for div to be pinned, has to be seperated from above ScrollLottie function as I want the animation to play at a different instance compared to when i want to pin it...
ScrollTrigger.create({
  trigger: ".government-2",
  start: "top top",
  end: s == "_d" ? "+=1000%" : "+=180%",
  pin: true,
});
```

This is the reusable portion of the code. Whenever I add in a new animation, I will have to do so in this format. I will call the ScrollLottie function, and then call the ScrollTrigger.create function.

ScrollLottie()



```
//02_Gov
ScrollLottie({
    target: ".government-2", //the division the lottie animation will be in
    path: `./animations/02_Gov${s}.json`, //downloaded json file
    duration: s == "_d" ? 1 : 0.6, //adds smooth scrolling, can set any number
    end: s == "_d" ? "+=1200%" : "+=380%", //how many percent of the viewheight a user has to scroll to finish the animation
    // totalFrames: 708, //get this from the website. essentially the total frames for the whole animation (if want to set this can change anim.totalFrames to obj.totalFrames in the function above)
    start: "top 99%", //where trigger starts
    pin: false,
    startingFrame: 15, //usually 0, unless we want animation to start from a later frame in this case
    aspectRatio : 'xMidYMin meet' //toggle svg aspect ratio setting
});
```

This is the ScrollLottie function. For each new animations, I just need to call ScrollLottie, which is a function already created above.

1. target: the division the lottie animation will be in
2. path: the location of the json file

3. duration: adds smooth scrolling, can be set to any number
4. end: how many percent of the viewheight the user must scroll to finish an animation
5. start: where the scrolltrigger starts (https://greensock.com/docs/v3/Plugins/ScrollTrigger/start)
6. pin: determine if we want to pin the div when scrolltrigger starts
7. aspectRatio: toggle the svg aspect ratio (https://developer.mozilla.org/en-US/docs/Web/SVG/Attribute/preserveAspectRatio)

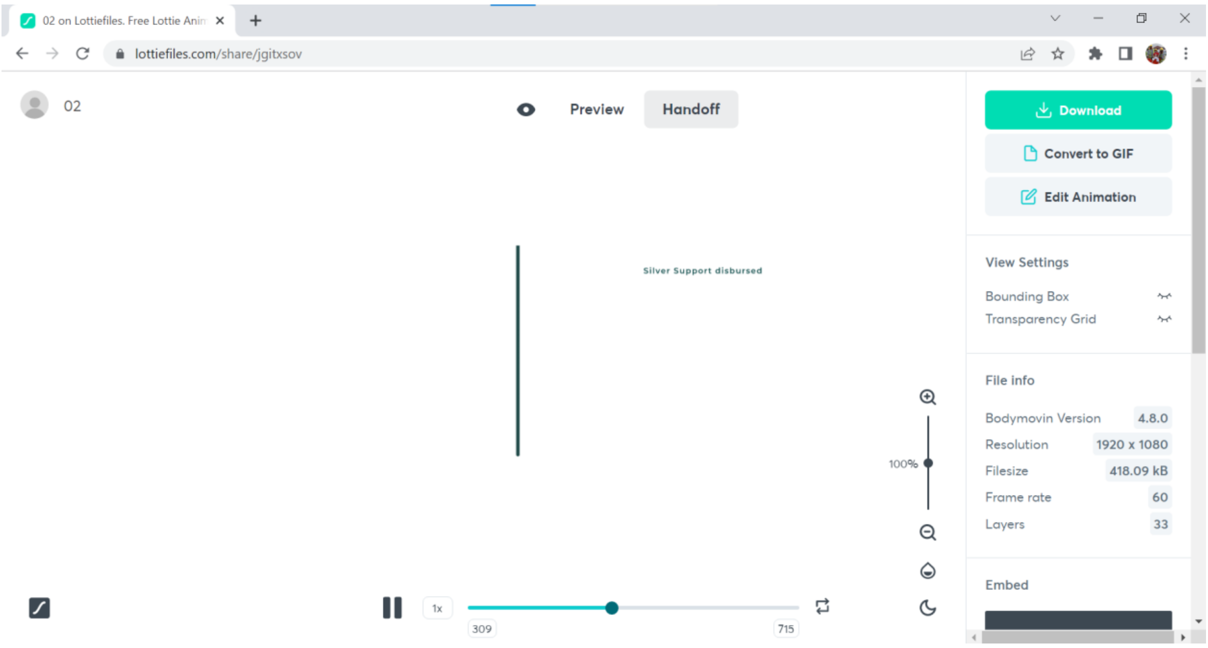
ScrollTrigger.create()

```
//trigger for div to be pinned, has to be seperated from above ScrollLottie function as I
want the animation to play at a different instance compared to when i want to pin it...
ScrollTrigger.create({
  trigger: ".government-2",
  start: "top top",
  end: "1000%",
  pin: true,
});
```

This is the trigger for pinning the division. I have this as I want the animation to be played early, but I only want the division to be pinned only when it resides in 100% of my viewport.

Example of how to add in a new Animation

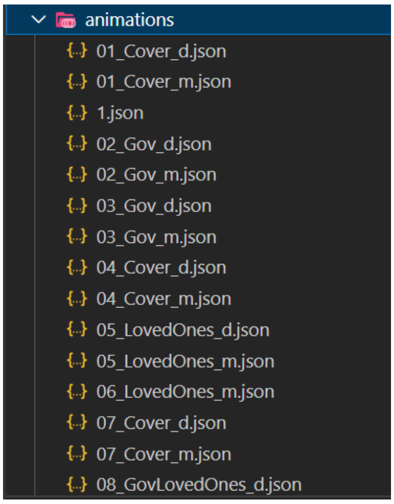
1. DXC wants a new animation added, the lottie link is as shown. https://lottiefiles.com/share/jgitxsov



File info

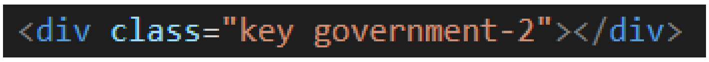
Bodymovin Version 4.8.0
Resolution 1920 x 1080
Filesize 418.09 kB
Frame rate 60
Layers 33

2. Upon clicking the link, you will get sent to this webpage. Click Handoff, and download the lottie animation as a json file.



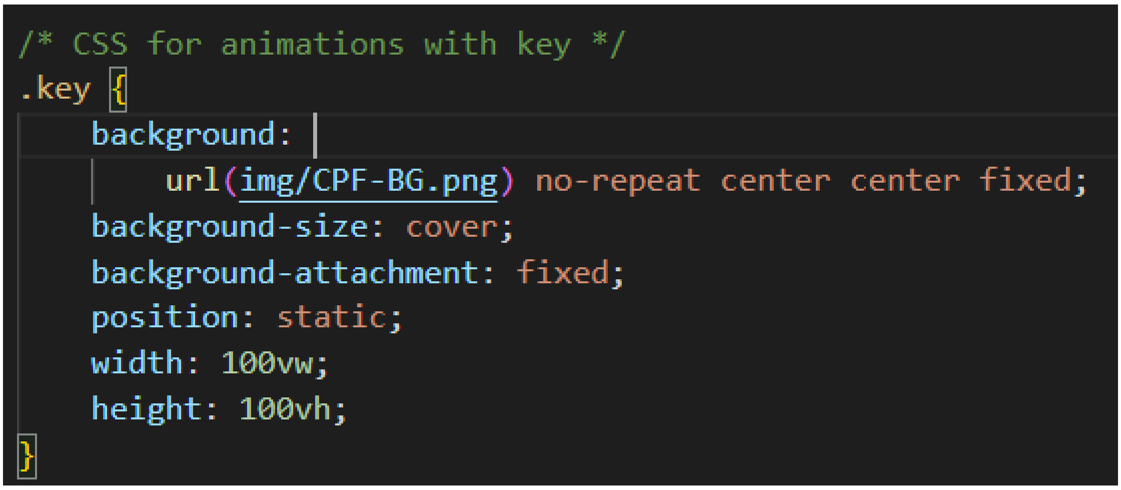
```
animations
    01_Cover_d.json
    01_Cover_m.json
    1.json
    02_Gov_d.json
    02_Gov_m.json
    03_Gov_d.json
    03_Gov_m.json
    04_Cover_d.json
    04_Cover_m.json
    05_LovedOnes_d.json
    05_LovedOnes_m.json
    06_LovedOnes_m.json
    07_Cover_d.json
    07_Cover_m.json
    08_GovLovedOnes_d.json
```

3. Create an animation folder and save the downloaded json file in that folder. In this case, you can see that 02_Gov.json is saved in the animations folder.



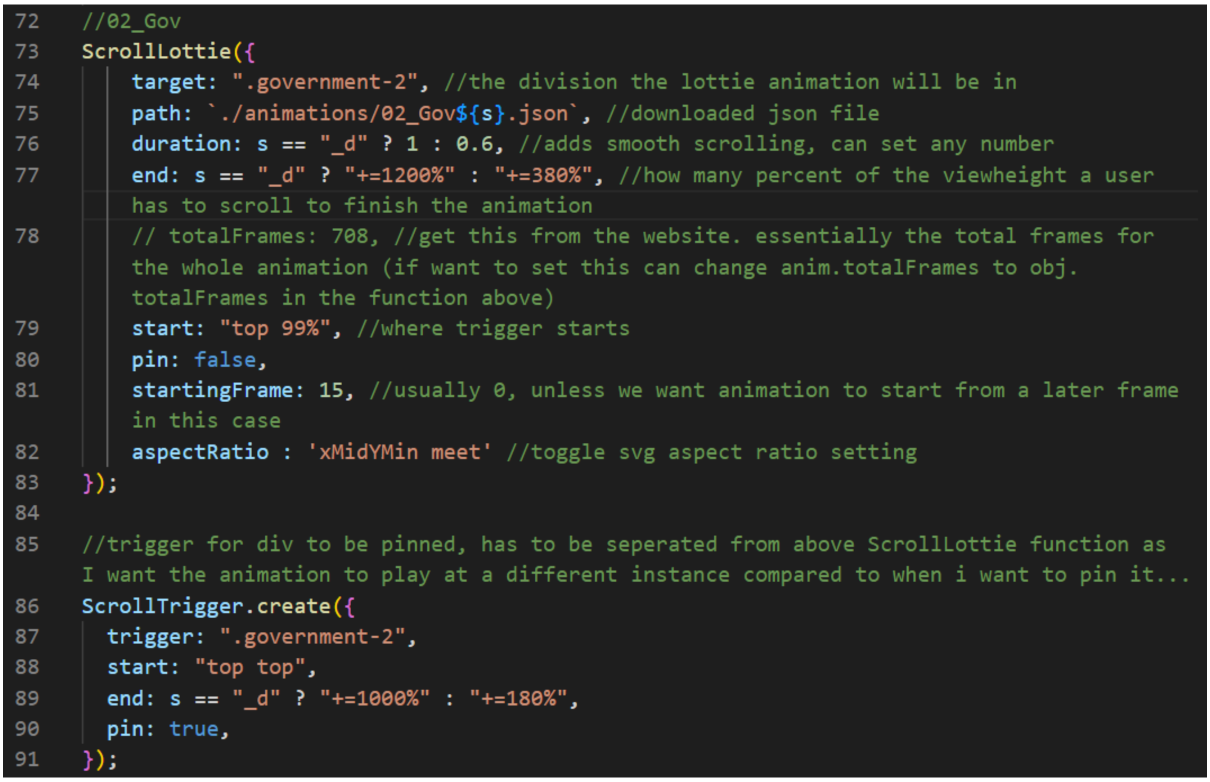
```
<div class="key government-2"></div>
```

4. A div is created to house the animation, and give it a class of government-2. Key is also a class given to it, to be able to set the background image and vice versa in css.



```
/* CSS for animations with key */
.key {
    background: 
        url(img/CPF-BG.png) no-repeat center center fixed;
    background-size: cover;
    background-attachment: fixed;
    position: static;
    width: 100vw;
    height: 100vh;
}
```

CSS for key class



```javascript
//02_Gov
ScrollLottie({
    target: ".government-2", //the division the lottie animation will be in
    path: `./animations/02_Gov${s}.json`, //downloaded json file
    duration: s == "_d" ? 1 : 0.6, //adds smooth scrolling, can set any number
    end: s == "_d" ? "+=1200%" : "+=380%", //how many percent of the viewheight a user has to scroll to finish the animation
    // totalFrames: 708, //get this from the website. essentially the total frames for the whole animation (if want to set this can change anim.totalFrames to obj.totalFrames in the function above)
    start: "top 99%", //where trigger starts
    pin: false,
    startingFrame: 15, //usually 0, unless we want animation to start from a later frame in this case
    aspectRatio : 'xMidYMin meet' //toggle svg aspect ratio setting
});

//trigger for div to be pinned, has to be seperated from above ScrollLottie function as I want the animation to play at a different instance compared to when i want to pin it...
ScrollTrigger.create({
  trigger: ".government-2",
  start: "top top",
  end: s == "_d" ? "+=1000%" : "+=180%",
  pin: true,
});
```

5. The final thing left to do is to call the 2 functions, ScrollLottie() and ScrollTrigger.create().