Library research

Robin Friesen – Personal project

The library that I needed was an animation library. I'd like to add some on-scroll animations, because I think they make a site feel more alive. I've generally developed static websites, so I'm in for something new.

To find some good animation libraries, I've done a google search for 'HTML animation library' (maybe 'CSS animation library' would have been better, but I didn't think of that at the time). This resulted in the following libraries:

Animate on Scroll library

The Animate on Scroll Javascript library – as the name suggests – focusses on animations when a user is scrolling through a page. The library works on desktop and mobile and doesn't have any other dependencies. You have several effects available, like fade (up, down, left or right), zoom in up/down and flip left or right. While this would be a useful library, it is a bit restrictive. It does what it says, and nothing more (like other types of animations that don't have to do with scrolling). The library seems very easy to work with though, as you can simply add effects with an additional bit of code.

Animate on Scroll can be installed via npm, yarn and bower. It can also be added to a web page by linking the stylesheet.

The library works by putting text inside of a div with the name of the animation, like this:

```
<div data-aos="fade-up"></div>
```

Settings like the duration of the animation can be changed by adding the setting to the div:

```
<div data-aos="fade-up"
     data-aos-duration="3000">
</div>
```

(Sajnóg, n.d.)

Anime.js

Anime.js is a more extensive Javascript library than the Animate on Scroll library. With Anime.js a developer can make a lot of different animations (for example with text, symbols or pictures). Anime.js is especially useful for making very complex animations. While this would be very interesting to try out, I personally don't like websites where a lot is going on (as it takes away attention from the content). While I would definitely like to try to make animations, I think I'll keep that for another project. It also seems a bit harder to work with (based on the documentation), and it may be too complex to use just to add some small animations.

Anime.js doesn't have any standard animations. An example of a programmed animation:

```
var animation = anime({
  targets: '.play-pause-demo .el',
  translateX: 270,
  delay: function(el, i) { return i * 100; },
  direction: 'alternate',
  loop: true,
  autoplay: false,
  easing: 'easeInOutSine'
});

document.querySelector('.play-pause-demo .play').onclick = animation.play;
document.querySelector('.play-pause-demo .pause').onclick = animation.pause;
```

This animation shows three blocks that move from left to right. The animation is much more complex than what I need, but the website doesn't have any examples of simple animations available.

(Garnier, n.d.)

Animate.css

Animate.css is a bit of an in-between of the two other libraries. It's easy to work with like Animate on Scroll, but has more features. The library can be installed via npm or yarn. Alternatively it can be added to a webpage by linking the external stylesheet. The library works via css. The animations can be added by adding the class name of the animation to text, like this:

```
<h1 class="animate__animated animate__bounce">An animated element</h1>
```

Settings of the animations can also be changed easily by changing the properties via the css, like this:

```
/* This only changes this particular animation duration */
.animate__animated.animate__bounce {
  --animate-duration: 2s;
}

/* This changes all the animations globally */
:root {
  --animate-duration: 800ms;
  --animate-delay: 0.9s;
}
```

The documentation is very detailed, which makes the library even easier to add to and use on my website.

(Eden, n.d.)

Conclusion

Between these three libraries, I prefer Animate.css.

Animate on Scroll would work just fine, as it's easy to use and has all the features that I need. However, Animate.css is about as easy to use and it has built-in features to add your own animations as well. I prefer to work with something that has features that I can add with ease later while doing what I want it to do now, rather than having to change or add libraries later on.

Anime.js is too complicated for what I need on my website. It would be cool to use in a side project where animation is the main point, but for my website it's too complex.

Bibliography

Eden, D. (n.d.). *Documentation*. Animate.css. Retrieved October 20, 2023, from https://animate.style/

Garnier, J. (n.d.). *Anime.js*. anime.js • JavaScript animation engine. Retrieved September 26, 2023, from https://animejs.com/documentation/

Sajnóg, M. (n.d.). AOS. Retrieved September 26, 2023, from https://michalsnik.github.io/aos/