Alexander Pearsall

Gavin Danner-Rivers

SWE2511-121

**Lab 5: Styling Libraries**

**Content Delivery Networks**

**Research CDNs from a general sense. What are some (at least 2) security concerns that you have in regard to using CDNs?**

A content delivery network (CDN) is a group of servers that are able to cache content to end users. Content such as HTML pages, JavaScript files, stylesheets, etc., are all able to be transferred from server to user over the internet. While CDNs are a quick and easy way to transfer data, there are some security concerns around them. One of these being the need to trust third-party companies. After handing over your content to a third-party company, you now have to put your trust in them to not do anything malicious when sending out your data to other users. You’ll also have to trust the third party's own security to prevent any major security breaches from happening. Some other security concerns that may affect you, the company you're working with, and the users that are using your page, are data theft, data loss, blockage from data, reduced bandwidth, and many others. When it comes to security, you want to make sure you are making the correct decisions in order to prevent any of these from happening to keep everyone involved in your CDN safe.

**Research the *link* and *script* HTML tags**

The link HTML tag enables the ability to insert a link into the webpage.

The script HTML tag is used to embed a client-side script (JavaScript).

* **Describe the *integrity* attribute and the *crossorigin* attribute used to include the Bootstrap library via CDN.**

The integrity attribute allows the browser to check the files source in order to ensure that the code is never loaded if the source has been manipulated.

The crossorigin attribute sets the mode of the request to an HTTP CORS Request.

* **What security concern(s) to these attributes attempt to fix? Explain how this works.**

Both of these attributes attempt to block any malicious scripts that may try to run when loading the webpage. They can identify if anything malicious has been sent by checking if anything has been tampered with when delivering the content from the server to your browser.

**Other Styling Libraries**

**Describe how to install the library (CDN or direct download)**

The CSS styling library Animate.css can be installed in four different ways:

1. With npm



1. With Yarn



1. With your own file



1. Directly using a CDN

A computer screen with white text

Description automatically generated

**Research and learn about the components and/or other styling the library provides**

* **Find 5 separate components that you find interesting and describe how to use them the library you chose**

1. animate\_\_animated class
   1. Adding this class to an element’s tag allows the ability to begin animating with Animate. This class will be used off of Animate’s custom CSS sheet.

Ex:

<div><h1 class="animate\_\_animated animate\_\_rubberBand">An animated element</h1></div>

* 1. This allows a level 1 heading to “rubber band” once the page is loaded.

1. Properties in Tags
   1. If we don’t want to specify a property in the CSS sheet, we can put it directly in the elements tag.

Ex:

<div><h1 class="animate\_\_animated animate\_\_rubberBand animate\_\_delay-2s">An animated element</h1></div>z

1. Animation CSS Customization
   1. If the default values in the Animate sheet aren’t what we are looking for, then we can customize and create our own with keyframes.

Ex:

h2 {  
 animation-name: moving;  
 animation-duration: 4s;  
}  
  
@keyframes moving {  
 0% {transform: translateX(200px)}  
 50% {transform: translateX(-200px)}  
 100% {transform: translateX(0px)}  
}

* 1. With this example, we first define an animation element with the @keyframe tag along with its name. In this example, we are translating an element along the x axis 400 pixels in total. We can then assign this animation to the header element. Since the animation is then too fast, we can then slow it down by setting the animation duration to 4 seconds.

1. Hover Animation
   1. We can create a hover animation that will then allow use to modify an elements state when someone hovers their mouse over the element.

Ex:

.box:hover {  
 animation-name: rotate;  
 animation-duration: 1s;  
}  
  
@keyframes rotate {  
 0% {  
 transform: rotate(0);  
 }  
 100% {  
 transform: rotate(360deg);  
 }

* 1. With this example, we create a rotation animation that will rotate the element. Instead of having the element only once, we can assign it to play whenever the mouse hovers over it.

1. Infinite and Glow
   1. We can change the properties of an animation by adding extra states to them. One of these states are the infinite alternate. This will loop through the animation infinitely. We can pair this up with a glowing boarder to make it look like it is pulsing.

Ex:

div {  
 animation: glow 1s infinite alternate;  
 position: absolute;  
 left: 30%;  
}  
  
@keyframes glow {  
 from {  
 box-shadow: 0 0 10px -10px #620000;  
 }  
 to {  
 box-shadow: 0 0 10px 10px #ff0000;  
 }  
}

* 1. Here, we create an animation called glow that changes color and scales up and down. We then apply this to all the div elements with a 1 second and infinite alternate duration.

**Bootstrap Exercises**

A screenshot of a phone

Description automatically generated

**Suggestions**

Overall, we enjoyed this lab, as it gave us a great introduction into bootstrap, along with introducing us into applying other variants of CSS styling libraries.

**Sources**

<https://www.cloudflare.com/learning/cdn/what-is-a-cdn/>

[What is CDN Security? - VERIMATRIX](https://www.verimatrix.com/knowledge-base/streaming/what-is-cdn-security/)

<https://animate.style/>