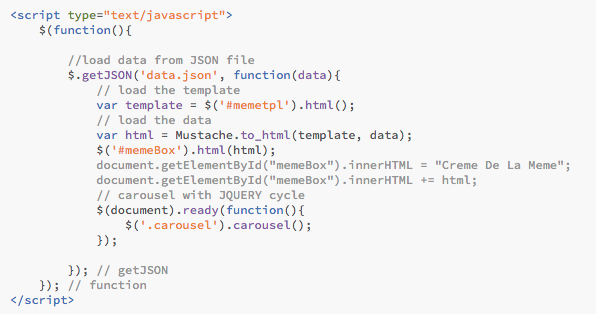
Honestly, initially the first time I came to tackle this, I struggled a lot. I did a lot of debugging and a lot of trial and error. I ended up deleting the entire project and tried it again and got it to work. Mustache is fairly people to comprehend and very easy to use. It helped filter out pictures I was using and not using with the logicless templating.



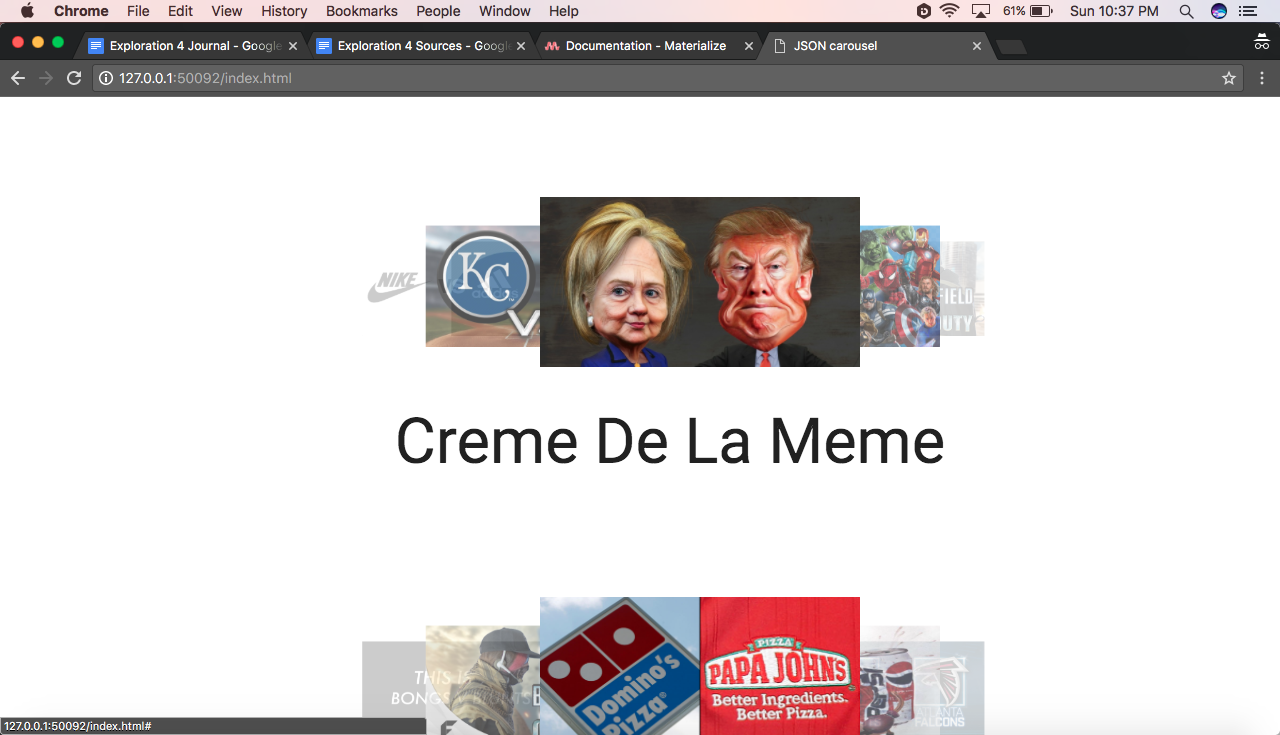
This is the function that I used to pull in the data and mustache template and display that to the DOM in html. It’s a pretty standard templating schema so I won’t go into it in depth.



The div #memebox is where the javascript loads the html data when the page loads. The #memetpl is the template I used with the help of mustache. There’s an object called carousels and for every pictures object, it will create a new carousel. I have variables called href, shortname, and uname. I will put a screenshot below. Basically, it feeds the name of the pic dynamically and will pull that image into the carousel dynamically using the shortname.



This is the json file that I used and I will include the images in the zip file. I got this idea from mustache.js website



This is the finished product and I am actually really proud of it. Each image is an html link, and will take you to that page, which I did not include in this example.

Recap:

I think the hardest thing was displaying more than one carousel. What I ended up doing was creating one json object called carousels with objects in that called pictures. For every picture object, it would create a carousel for that object. Before, all I had were the carousel objects, and it would only create one and that was it.