1. When I started building my first portfolio project, I knew I wanted to do something with cryptocurrency. Before coding a single line, I put together a small raspberryPi crypto-ticker and used someone else's code to run it, then I thought, "Hey I could write this too". 5 weeks into my coding journey I dared to accomplish this first task, completely unaware of what I was getting myself into. On day one I figured out how to make API calls and display the exact information I wanted, price, logo, and symbol. But that wasn't enough anymore. I had the taste of success lingering in my mouth and I wanted charts, real time prices, and to make trades. I soon learned that wasn't the taste of success, it was a foreboding waft of the challenges to come. I chose to use multiple APIs, Websockets, and Widgets to accomplish my lofty goals, and I did, but not a minute before learning how to use them properly. Three weeks of work and my persistence had finally paid off for the very first time. I developed what I call Cryptid, a cryptocurrency tracking app, that allows the user to search for thousands of popular and obscure coins. The primary takeaway from this process was the need for a solid plan, a definitive finish line. I dove in head first, without noticing which end of the pool I was plummeting towards. I've always been a dreamer, but now I'd like to consider myself a planner.

2. Diving head first into a project is always fun, right? But completing a project isn't always the funnest part of the process. It's great to have a final product, and that is what actually matters to the end user. For me however, I'd like to think what I get out of completing projects is all the problem solving along the way. All of those tedious bugs that throw red flags into the console; some of those bugs will not hinder user experience but they definitely impact the experience of the developer. Why is my app working but I'm logging literally thousands of errors per minute? It might seem like I'm speaking in hyperbole, but I'm not. When I first decided to use Binance Websockets in conjunction with a paginated list, this is the exact issue I faced. The data was coming in, and it was appending to my list, actually ,to every single page on my list. But my console was logging thousands of *'Undefined is not a node'* errors. I could tell why I was having these errors after some digging. My code was trying to append information to items on every single page, when only one was displayed at a time. I researched this problem for 5 days, I also made posts to see if any other developer had run into the same issue. I finally found the right documentation on stackoverflow. My issue was in my request, initially I was requesting information on every single item I needed, even those that were not currently displayed. What I needed to be doing was adjusting my request based on the start and end index of what I was looking at. After tinkering for another day my paginated list worked, and as a bonus my knowledge on websockets and the DOM had grown.