

## **Describe a time you experienced a challenge while building a product/project and how you overcame it.**

I started coding because I was inspired to make a very specific "Name That Truck" game for my son. Since I had zero coding experience, I started with HTML5/CSS3 and JavaScript with the goal of making a website version of the game. However, I simply couldn't make "Name That Truck"'s JavaScript work. At the time, I maybe had 3 months of experience coding, and writing in-browser games was a newish concept, so there weren't very many resources on what I might be doing wrong, or how I could improve my code. I was stuck. But I didn't want my dream to die...

So I learned Python. I know what you're thinking -- that seems extreme! But learning Python gave me two key advantages: Python could do everything I wanted JavaScript to do and more, and also, there were a seemingly endless number of Python resources available to me. If I didn't understand what was going wrong with my Python code, I could visit the PyLadies Wednesday night session at my local library, or I could check out a Python book designed to answer the question... at a level even children could understand.

Learning Python because I was having trouble writing JavaScript was the best decision I ever made. I worked my way through every single Python book at the Pasadena library. I worked through all the courses I could find online. Python taught me computer science. It also taught me that coding is an art equivalent to sketching or writing poetry.

When I revisited my JavaScript version "Name That Truck" after adding a year of Python to my tool belt -- I could code "Name That Truck". It was easy. So easy, in fact, that I decided to take on Swift and make "Name That Truck" into an iOS app instead!

**What are the qualities you most desire in a work environment and/or manager?**

The single most exciting thing about iOS Development to me is how the iOS world embraces sharing their knowledge. I love how many Swift and iOS communities exist online and in real life that I can turn to in order to learn how to solve a specific problem or be exposed to new and exciting techniques and technologies.

I would absolutely love to see that same passion for new Swift and iOS knowledge in a work environment. It would thrill me to have a manager who not only critiques my code but encourages my team to work together, teach each other new things, improve each other's code and bring a fresh perspective to maintaining legacy code and creating new apps and features.

I'm excited about working with a team of coders and seeing how everyone in the work environment can push each other to write better, newer, fresher, safer code. I don't just want a job where I can practice and show off my existing skills, I want to keep learning and pushing new boundaries.

**Describe your favorite mobile app: how you use it, why you like it, and how it inspires your own development.**

My favorite mobile app is Feedly. Feedly is my preferred method of interacting with media. It's like having an entire library or cable channel at my fingertips at all times. It collects news articles or blog posts on topics that I find interesting, from major sources or just smart, cool people, and shows them to me when I'm ready, automatically making sure that I don't miss a single thing. Feedly is how I keep up with news stories, iOS trends, new technology, new creative ideas or my friends' blogs. I can even save the stories I like most in order to share them or follow up on them later. For me, Feedly is the key to keeping up with all the important things I need to know in life.

## **What data persistence method do you prefer, and why?**

In developing my own apps, I have primarily used a web-hosted database like Parse and/or Firebase to persist my data. The Parse and Firebase databases have all kinds of options, an easy interface, and fairly fast and reliable storage. This method does need to rely on a network connection, which has strong advantages for constantly-updating data among a large number of users as is the case in a social-media app.

For projects that don't rely on a database that users are updating at all times, I like Core Data. Apple has made sure Core Data is relatively simple and safe. It presents data quickly and efficiently through simple but powerful fetch requests. The `NSFetchedResultsController` makes displaying data in table views incredibly easy. Plus, Apple's support is making the framework better and better every single day.

## **Explain a MV\* architecture and why it is useful.**

MV\* architecture - MVC: Model-View-Controller or MVVM: Model-View-ViewModel - is an incredibly useful structure for app development because it takes the three main sections of your code and separates them into interconnected modules.

The model is used to structure the data, the view is used to present the data, and the controller, or view model, is used to communicate between them - to send the view the data to display, or to send the model data to process. Each unit needs to know only the information regarding its own function, which leads to writing clean and reusable code.