CheatSheet

Binaryhood - [Faseeh Lodhi]

March 31, 2023

1 iOS development

- Here are some professional tips for learning iOS development using Xcode and Swift:

1. Start with the basics: Learn the basics of Swift programming language such as variables, data types, loops, functions, etc. Once you have a good understanding of the language, move on to iOS app development.
2. Get familiar with Xcode: Xcode is the official IDE for developing iOS apps. Learn the basics of Xcode such as creating projects, building user interfaces, running and debugging apps, etc.
3. Learn iOS app architecture: iOS app architecture is based on the Model-View-Controller (MVC) pattern. Learn how to use the MVC pattern to structure your app's code.
4. Work with storyboards: Storyboards are visual tools that allow you to design your app's user interface. Learn how to create and use storyboards to design your app's screens.
5. Use Auto Layout: Auto Layout is a tool that helps you create adaptive interfaces that work across different devices and screen sizes. Learn how to use Auto Layout to design your app's user interface.
6. Understand view controllers: View controllers are the building blocks of your app's user interface. Learn how to use view controllers to manage your app's screens and handle user interactions.
7. Work with data: Learn how to work with data in your app using Core Data or other data persistence frameworks.
8. Use networking: Learn how to use URLSession to make network requests and fetch data from APIs.
9. Practice coding: The best way to learn iOS development is by writing code. Practice building small apps, experimenting with different frameworks, and trying out new features of the language and the platform.
10. Stay up-to-date: iOS development is a constantly evolving field, with new features and frameworks being released regularly. Stay up-to-date with the latest trends and best practices by reading blogs, attending conferences, and joining online communities.

2 DevOps Engineer

2.0.1 Becoming a good DevOps engineer requires a combination of technical skills, soft skills, and experience. Here are some steps you can take to become a skilled DevOps engineer

1. Learn the basics: Start by learning the fundamental concepts of DevOps, including continuous integration, continuous delivery, automation, and monitoring. You should also familiarize yourself with popular DevOps tools such as Jenkins, Ansible, Docker, and Kubernetes.

2. Develop programming skills: A good DevOps engineer should be proficient in at least one programming language. Python, Ruby, and Java are all popular choices in the DevOps community.

3. Build your automation skills: DevOps is all about automation, so it's essential to have strong scripting skills. Bash, Python, and PowerShell are all widely used scripting languages in DevOps.

4. Learn cloud computing: Cloud computing is a fundamental component of modern DevOps, so you should become familiar with public cloud platforms such as AWS, Azure, and Google Cloud.

5. Practice continuous integration and delivery: Continuous integration and delivery are at the heart of DevOps, so it's important to have hands-on experience setting up and managing CI/CD pipelines.

6. Hone your collaboration skills: DevOps is a team sport, so collaboration skills are critical. You should be comfortable working with developers, operations teams, and other stakeholders.

7. Focus on security: Security is a critical aspect of DevOps, so it's essential to have a solid understanding of security best practices and how to integrate security into your DevOps processes.

8. Stay current: DevOps is a rapidly evolving field, so it's important to stay current with new tools, technologies, and trends. Attend conferences, meetups, and other industry events to stay up to date.

9. Build a portfolio: Build a portfolio of your work to showcase your skills and experience to potential employers. Include examples of automation scripts, CI/CD pipelines, and other DevOps-related projects.

10. Get certified: Consider getting certified in DevOps tools such as AWS, Ansible, or Kubernetes. Certifications can help demonstrate your skills and expertise to potential employers.