## Pre-Work: Before You Start the Course

Reflection questions (to complete before your first mentor call)

1. **What experiences have you had with coding and/or programming so far? What other experiences (programming-related or not) have you had that may help you as you progress through this course?**

I’ve completed a bachelor’s degree in Computer Science where I studied some Python and Java, completed a small JavaScript course from FreeCodeCamp, and have worked on several personal mini projects in various languages.

1. **What do you know about Python already? What do you want to know?**

I am aware of the fundamentals of Python and that it’s one of the most popular and versatile programming languages in the world, with its easy to digest syntax and extensive standard library and an active community contributing to it.

1. **What challenges do you think may come up while you take this course? What will help you face them? Think of specific spaces, people, and times of day of week that might be favorable to your facing challenges and growing. Plan for how to solve challenges that arise.**

I am confident that I can tackle any challenges that arise throughout the course with my solid foundation in Python from university and the help of my mentor.

### Exercise 1.1: Getting Started with Python

Reflection questions

1. **In your own words, what is the difference between frontend and backend web development? If you were hired to work on backend programming for a web application, what kinds of operations would you be working on?**

Frontend web development involves creating the visual elements and user interface of a website that users interact with directly, typically using HTML, CSS, and JavaScript. Backend web development, on the other hand, focuses on the server-side operations that power the frontend, such as database management, user authentication, and handling requests. If hired to work on backend programming for a web application, I would be responsible for tasks like designing and maintaining the database structure, implementing server-side logic to process user input, ensuring data security, and integrating with external APIs or services.

1. **Imagine you’re working as a full-stack developer in the near future. Your team is asking for your advice on whether to use JavaScript or Python for a project, and you think Python would be the better choice. How would you explain the similarities and differences between the two languages to your team? Drawing from what you learned in this Exercise, what reasons would you give to convince your team that Python is the better option?**

I would explain that both JavaScript and Python are versatile programming languages commonly used in web development, but they have distinct strengths and use cases. While JavaScript excels in frontend development with its browser compatibility and asynchronous capabilities, Python offers a more cohesive ecosystem for backend development, with powerful frameworks like Django, as well as broader applicability in other non-web related fields. Additionally, Python's clean syntax and emphasis on readability can lead to faster development, easier maintenance, and smoother collaboration among team members, making it the better choice for our project's backend requirements.

1. **Now that you’ve had an introduction to Python, write down 3 goals you have for yourself and your learning during this Achievement. You can reflect on the following questions if it helps you. What do you want to learn about Python? What do you want to get out of this Achievement? Where or what do you see yourself working on after you complete this Achievement?**
2. Master Data Structures and Algorithms: I aim to deepen my understanding of fundamental data structures such as lists, dictionaries, sets, and tuples, as well as algorithms like sorting and searching. This knowledge will enhance my problem-solving skills and enable me to write more efficient and scalable code.
3. Solidify Object-Oriented Programming Concepts: I intend to strengthen my grasp of object-oriented programming principles, including inheritance, encapsulation, and polymorphism. By practicing these concepts extensively, I aim to write cleaner, modular, and more maintainable code for the Recipe app project.
4. Learn Database Interaction: I aspire to become proficient in interacting with databases using Python, including querying, updating, and managing data. Understanding how to integrate databases into my applications will be crucial for the Recipe app project and will prepare me for working with frameworks like Django in the future.