Exercise 1.1

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 had no knowledge of coding before registering for the Full-stack developer course with careerFoundry. I worked as an Account Executive for FMCG companies which is not related to programming.

What do you know about Python already? What do you want to know?

I have heard that Python is a popular programming language that is beginner-friendly for non-programmers to learn.
I would like to know how to use Python to build a game because I would like to create one.

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 the week that might be favorable to your facing challenges and growing. Plan for how to solve challenges that arise.

The biggest challenge for me is that I have only 3 weeks to complete this course. (If not then I need to pay extra money and void the job guarantee.)
Moreover, I am not so good at learning anything new with reading only texts and this course is full of heavy texts. My plan is to submit at least 5 tasks per week. There is no space for failing.

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 focuses on creating the user interface and experience of a website or application, involving technologies like HTML, CSS, and JavaScript to design and implement the visual elements and interactivity that users interact with directly.
Backend web development, on the other hand, deals with server-side logic and database management, using languages like Python, Ruby, or Node.js to handle data, user authentication, and the overall functionality that supports the front end.

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?
Python and JavaScript are both high-level programming languages with versatile applications, yet they serve distinct purposes. They share similarities in being interpreted, dynamically typed, and having active communities.

Python stands out as the superior choice due to its unparalleled readability, versatility, and robust ecosystem. Its clean and intuitive syntax fosters efficient collaboration and reduces development time, making it an ideal language for teams.

Write down 3 goals you have for yourself and your learning during this Achievement.

1. Be able to code with Python effectively
2. Have Python projects on my portfolio
3. Convince the HR and IT team lead that I am well-versed in Python

Exercise 1.2

Imagine you're having a conversation with a future colleague about whether to use the iPython Shell instead of Python's default shell. What reasons would you give to explain the benefits of using the iPython Shell over the default one?

The iPython Shell offers numerous benefits over the default Python shell due to its enhanced features and capabilities. It provides an interactive and user-friendly environment with features like tab completion, syntax highlighting, and history navigation, making code exploration and debugging more efficient.

Python has a host of different data types that allow you to store and organize information. List 4 examples of data types that Python recognizes, briefly define them, and indicate whether they are scalar or non-scalar.

Data type	Definition	Scalar or Non-Scalar?
Integer (int)	whole numbers, both positive and negative, without any decimal points	scalar
Float (float)	numbers with decimal points, allowing for representation of real numbers	scalar
String (str)	sequences of characters, such as text	non-scalar
List (list)	ordered collections of items that can be of any data type, including other lists	non-scalar

A frequent question at job interviews for Python developers is: what is the difference between lists and tuples in Python? Write down how you would respond.
The key difference between lists and tuples in Python lies in their mutability. Lists are mutable, meaning their elements can be modified, added, or removed after creation, making them suitable for situations where data needs to change. Tuples, on the other hand, are immutable, meaning their elements cannot be modified after creation, providing stability to the data. This makes tuples suitable for cases where data should remain constant throughout the program. Lists are defined using square brackets [], while tuples are defined using parentheses ().

In the task for this Exercise, you decided what you thought was the most suitable data structure for storing all the information for a recipe. Now, imagine you're creating a language-learning app that helps users memorize vocabulary through flashcards. Users can input vocabulary words, definitions, and their category (noun, verb, etc.) into the flashcards. They can then quiz themselves by flipping through the flashcards. Think about the necessary data types and what would be the most suitable data structure for this language-learning app. Between tuples, lists, and dictionaries, which would you choose? Think about their respective advantages and limitations, and where flexibility might be useful if you were to continue developing the language-learning app beyond vocabulary memorization.

The necessary data types would be String('str') and Dictionary('dict') to represent the information
String: To store vocabulary words, definitions, and categories. These textual data can be stored using the string data type.
Dictionary: Dictionaries would be a suitable data structure to store flashcards. Each flashcard can be represented as a dictionary with keys like "word," "definition," and "category," where the corresponding values hold the actual word, definition, and category information. Dictionaries allow you to organize and access flashcard data efficiently using meaningful keys.

Exercise 1.3

In this exercise, you learned how to use if-elif-else statements to run different tasks based on the conditions that you define. Now practice that skill by writing a script for a simple travel app using an if-elif-else statement for the following situation:

●The script should ask the user where they want to travel.
●The user's input should be checked for 3 different travel destinations that you define.
●If the user's input is one of those 3 destinations, the following statement should be printed: "Enjoy your stay in _____!"
●If the user's input is something other than the defined destinations, the following statement should be printed: "Oops, that destination is not currently available."

```
available_destinations = ["Bangkok", "Tokyo", "London"]

user_input = input("Where do you want to travel? ")

if user_input in available_destinations:
    print(f"Enjoy your stay in {user_input}!")
else:
    print("Oops, that destination is not currently available.")
```

Imagine you're at a job interview for a Python developer role. The interviewer says "Explain logical operators in Python". Draft how you would respond.

Logical operators in Python are used to combine and manipulate boolean values (True or False). There are three main logical operators: "and", "or", and "not".

The "and" operator returns True only if both operands are True.

The "or" operator returns True if at least one of the operands is True.

The "not" operator is a operator that negates the boolean value, turning True into False and False into True.

In the section for Exercise 1 in this Learning Journal, you were asked in question 3 to set some goals for yourself while you complete this course. In preparation for your next mentor call, make some notes on how you've progressed towards your goals so far.

Today I gained a deeper understanding of conditional statements in Python, mastering the art of making decisions within my code. Explored the realm of loops in Python programming. I feel more confident in my grasp of these fundamental concepts, ready to apply them creatively to a variety of coding challenges.

Exercise 1.4

Why is file storage important when you're using Python? What would happen if you didn't store local files?

File storage is really important when I use Python because it lets me keep information even after my program finishes running. If I don't store files locally, any data I use or create during our program's time won't stay once the program stops. This means I won't be able to save data for later use, share it with others, or use it in different sessions. Storing files helps Python programs to work with information over time and do more useful things.

In this exercise, you learned about the pickling process with the pickle.dump() method. What are pickles? In which situations would you choose to use pickles and why?

Pickles are a type of file format in Python that's used for storing data in a way that preserves its structure and type. It's like putting data in a container I can open later and get the same data back. I would choose to use pickles when I want to save complex data, like lists or dictionaries, to a file so that I can use it again later, even in a different session.

In Python, what function do you use to find out which directory you're currently in? What if you wanted to change your current working directory?

I can use the function getcwd() to discover the current. If I wanted to change my current working directory I would use chdir(new_directory_path).

Imagine you're working on a Python script and are worried there may be an error in a block of code. How would you approach the situation to prevent the entire script from terminating due to an error?

I would enclose the potentially problematic code with a "try" section, and if an error occurs, the script won´t terminate but will instead move to the corresponding "except" section where I can handle the error.

You're now more than halfway through Achievement 1! Take a moment to reflect on your learning in the course so far. How is it going? What's something you're proud of so far? Is there something you're struggling with? What do you need more practice with? Feel free to use these notes to guide your next mentor call.

I find Python readable and enjoy learning. I am proud of my cocktail recipe and how I could fix the bugs on my own so far. To use Python professionally, I would need to practice doing more projects.

Exercise 1.5

In your own words, what is object-oriented programming? What are the benefits of OOP?
Object-oriented programming (OOP) is a way of organizing and designing computer programs by using objects. In OOP, these objects can interact with each other to perform tasks, making it easier to manage complex systems and promote code reusability.

What are objects and classes in Python? Come up with a real-world example to illustrate how objects and classes work.
In Python, objects are individual instances of a class, which is a blueprint or template defining the structure and behavior of objects. For instance, imagine a class named "Car" that has attributes like "color" and "brand." An object created from this class could represent a specific car, such as a red Toyota, which embodies the characteristics and behaviors defined by the class.

In your own words, write brief explanations of the following OOP concepts.

Method	Description
Inheritance	A new class can inherit the properties and methods of an existing class, allowing for the reuse of code.
Polymorphism	The ability of different objects or classes to be used interchangeably through a common interface.
Operator Overloading	Giving special meanings to operators when used with custom objects, allowing them to perform specific actions beyond their usual functionality.

Exercise 1.6

What are databases and what are the advantages of using them?
A database is a structured collection of organized information that can be stored, managed, and retrieved electronically. In Python, databases offer benefits like efficiently storing large amounts of data in a structured manner and providing quick data access and manipulation through queries.

List 3 data types that can be used in MySQL and describe them briefly:

Data type	Definition
INT (Integer)	Used for holding whole numbers.
VARCHAR(n)	String of variable length, with n representing the maximum number of characters.
DATE	Stores calendar dates like 'YYYY-MM-DD' for recording specific days.

In what situations would SQLite be a better choice than MySQL?
SQLite would be a better choice than MySQL in simpler projects or when simplicity is important. For instance, if you need a database for a small mobile app or a personal project, SQLite is handy as it's embedded directly into the application, requiring no separate server setup. It's also easier to manage because it's a single file, while MySQL needs a separate server to be set up. However, for larger projects with many users and complex needs, MySQL is more suitable due to its client-server architecture and advanced features.

Think back to what you learned in the Immersion course. What do you think about the differences between JavaScript and Python as programming languages?
I find Python often chosen for its simplicity and readability, making it suitable for beginners. On the other hand, JavaScript is mainly used for web development and focused on web-related tasks and can be more complex due to its asynchronous nature.

Now that you're nearly at the end of Achievement 1, consider what you know about Python so far. What would you say are the limitations of Python as a programming language?
I've learned the fundamentals of Python to construct a database. I used my knowledge of Python to create a structured storage system for organizing data. This involved understanding how to define, add, retrieve, and modify information within the database. By applying my Python skills, I gained the ability to design a simple but functional database that can store and manage data effectively.

My point of view is that the limitation of Python is the execution environment. JavaScript runs directly in browsers, making it easily accessible for web development, while Python often requires additional tools and frameworks to run in a browser context.

Exercise 1.7

What is an Object Relational Mapper and what are the advantages of using one?
An Object Relational Mapper (ORM) in Python is a helpful tool that links software code with databases. It makes it simpler for developers to manage and retrieve data from databases (like MySQL or PostgreSQL) using familiar programming objects instead of writing complex database queries. This makes coding faster and less error-prone. In Python, using an ORM like SQLAlchemy or Django's ORM provides advantages such as easier code maintenance, increased productivity, and improved data security through built-in features for data validation and protection.

By this point, you've finished creating your Recipe app. How did it go? What's something in the app that you did well with? If you were to start over, what's something about your app that you would change or improve?
At this stage, I have successfully completed building the Recipe app using Python. I am pleased with how smoothly I managed to store and retrieve recipe information in the app's database. However, in hindsight, I would prioritize incorporating more comprehensive unit and integration testing during development to ensure the app's reliability and minimize potential bugs.

Imagine you're at a job interview. You're asked what experience you have creating an app using Python. Taking your work for this Achievement as an example, draft how you would respond to this question.
During my previous projects, I gained valuable experience in creating a database for a recipe app using Python. I successfully designed and implemented a database schema to efficiently store various aspects of recipes, such as ingredients, difficulties, and user preferences.

I utilized Python's database libraries, like SQLAlchemy, to establish connections between the app and the database, allowing seamless data retrieval and manipulation. I ensured data integrity and security by implementing proper validation and encryption techniques. Through this project, I honed my skills in database design, Python programming, and data management within the context of recipe applications.

You've finished Achievement 1! Before moving on to Achievement 2, take a moment to reflect on your learning in the course so far:

a. **What went well during this Achievement?**
   It went well in understanding the concept of data type in Python and applied it to the project including methods because of how readable Python language is.

b. **What's something you're proud of?**
   I often struggle to learn new programming languages but I am proud every time when I find the solution even if I know it from StackOverflow or Google.

c. What was the most challenging aspect of this Achievement?

To complete this achievement within 10 days and I did it.

d. Did this Achievement meet your expectations? Did it give you the confidence to start working with your new Python skills?

I would be more confident starting from the front end because it is also an essential part of knowing Python.

e. What's something you want to keep in mind to help you do your best in Achievement 2?

The deadline will keep me running. I must finish the course before next week.