February 1st, 2025

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?

Answer: I have completed the Full-Stack Immersion part of the Web Development program with Career Foundry. Prior to that, I did not have much experience at all. I had done a month of the FreeCodeCamp program which is what sparked my interest in programming and led me consequently to CareerFoundry.

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

Answer: Besides the fact that it is a programming language and a very popular one, I don't know anything about it.

3. 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.

Answer: I believe keeping myself motivated may be a challenge as the content of the course can sometimes present us with very complex ideas. I try to keep myself motivated by thinking that in a few years, I will be glad I started right now. I also try to think of the possibilities that completing this course will bring me and what possible doors might open when I can finally and confidently say that I am a web developer. But most of all, I know the importance of breaks and coming back to the course with a clear mind.

EXERCISE 1.1

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?

Answer: To put it shortly: Frontend is what users interact with on an application or web-page, while the backend is what happens backstage and makes the program work. In other words, all of the work that users don't need to see in order to interact with the platform.

2. 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?

Answer: The similarity between JavaScript and Python is that both are object oriented (which means they both work with sets of code and data). Also, both work with variables and functions and give the developer the ability to work with classes and properties. On the other hand, the main differences lie on:

- Speed: where JavaScript tends to perform faster,,
- Syntax: with Python offering a simpler syntax than JavaScript.

When it comes to convincing my team to use Python, I might argue that Python tends to be more suitable for large-scale projects and that working with Python for backend allows us to have versatility due to its very simple language in terms of syntax. I might also add that there are a lot of different frameworks available and resources on how to work with python; besides a strong and collaborative community online.

3. 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 reect 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?

Answer:

1. I want to learn enough about Python so that I can work with it and discuss it confidently with my future co-workers. I want to be able to understand and follow guidelines on possible Python documentations I may come across.
2. By working on this achievement I hope to get a good introduction about Python so that I can grow my confidence in web development and reach my goal in point number one.
3. After the completion of this achievement, I see myself applying for Technical Writing jobs that will allow me to grow my knowledge even more. I also hope to work on smaller projects for myself and my portfolio using the skills that I have acquired throughout the course.

February 4th, 2025

EXERCISE 1.2

1. 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?

Answer: To put it shortly, the iPython Shell is more practical and user-friendly than the default one. IPython's syntax highlighting makes it easier to read the code and it indents the code automatically. IPython Shell is also faster when it comes to testing small pieces of code.

2. 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?
int	Represents integers, includes both negative and non-negative numbers.	Scalar
float	Holds decimal numbers and includes both negative and non-negative decimal numbers.	Scalar
bool	Represents boolean where data can only be either True or False.	Scalar
dictionary	Unordered set of items that requires a key-value pair.	Non-Scalar

3. 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.

Answer: The difference between Lists and Tuples lies in the fact that lists are mutable and tuples are not. Elements within a list can be modified, deleted or rearranged.

4. 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.

Answer: I would choose to go with dictionaries as they don't require data to be sequential. This particular feature of dictionaries may come in handy as we continue to develop and grow the app's data.