Python Learning Journal

Exercise 1.1

Learning Goals;

- Summarize the uses and benefits of Python for web development
- Prepare your developer environment for programming 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?**

   Back-end development focuses more on server-side programming and infrastructure, while a front end developer is more involved in the UI and user facing elements of the website/app. Operations that a back end developer might be working on include; database management, user information handling, building and maintaining servers, working to improve website security and monitoring and improving performance and functionality.

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? (Hint: refer to the Exercise section "The Benefits of Developing with Python")**

   If I was recommending Python for a project, I would look to highlight the ease of use, built in package manager, simplicity and support. Python was designed to be easy to learn and the standardized syntax of the language enables anyone with some coding experience to understand the code at a glance. This simplicity extends to the built in package manager - PIP. The base install of Python comes with many useful packages and it is very easy to install further packages through the console. Python is also well supported by an active online community that continually improve the language.

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 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?**

a) My first goal is to make notes throughout the project on how I think the things learnt would be useful in other situations or projects.
b) I want to understand the language to the point where I am seeing solutions to tasks and issues without needing to rely solely on the learning materia**l.**
c) Ultimately, I want to be confident about putting Python on my CV/portfolio and comfortable using the language and answering questions about it.

Exercise 1.2

Learning Goals;

● Explain variables and data types in Python
● Summarize the use of objects in Python
● Create a data structure for your Recipe app

Reflection Questions
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?

I would make note that the ipython shell is considerably more user friendly and practical. The colour coding within the ipython shell makes it easier to read, and the ipython shell contains automatic indenting and syntax highlighting**.**

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.

Integer - whole numbers - Scalar
Bool - Boolean, returns either True or False - Scalar
Tuples - linear arrays to store multiple types of data - Non-scalar
Lists - mutable arrays of data - 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.

Both Lists and Tuples are used to store collections of data/items but the key difference is that Lists are mutable, meaning that they can be modified once the list is created. Tuples are immutable, which means that once created, their elements cannot be changed. Visually in code, lists are defined using square brackets, while tuples use parentheses.

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

For the language-learning app, I would choose dictionaries as the most suitable data structure. Dictionaries can store the vocabulary words as keys and the corresponding definitions and categories as values. This allows for efficient lookup and easy organization of data. Additionally, dictionaries' flexibility allows users to update or add new words and definitions, enhancing the learning experience.