Python Learning Journal

Exercise 1.1: Getting Started with Python

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 is what the user sees and interacts with. Backend is where operations that occur for handling requests, interacting with databases and files the server. If I was hired as a backend programmer, I would anticipate working on CRUD operations, working with databases, security concerns, and working with servers.

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?**

   JavaScript and Python are both scripting languages that use dynamic typing. JavaScript is not optimized for backend development, however Python's straightforward syntax and high readability and range of libraries and frameworks may make more sense in this situation.

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?**

   - I want to learn basic Python syntax.
   - In this achievement I want to be able to setup an environment to get started using Python.
   - After this achievement I want to continue to build my knowledge and skillset to become marketable as a backend developer.

Exercise 1.2: Data Types in Python

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?**

iPython is more user-friendly. It utilizes different colors making reading the code easier. The lines are numbered. There is syntax highlighting and indenting is done automatically.

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?
String str	immutable, can be composed of alphanumeric characters and symbols contained in single or double quotes	Non-Scalar
Numeric int	integer (whole number) negative or positive (to infinity)	Scalar
Boolean	True or False statement	Scalar
Sequence list	mutable, ordered sequence of data using square brackets and commas	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.**

While both are a sequence of data, a tuple is immutable while a list is mutable. Data in a tuple is write-protected. A list can be more easily manipulated. In the code, a tuple is written using parenthesis and commas while a list utilizes brackets and commas.

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.**

For each input they would be stored as strings. Having words, definitions, and category may get confusing if they were just stored as a list. If you used a dictionary you can use key-value pairs to distinguish between categories. Also

with a dictionary you can always extract the keys or values and place those into a list if needed in the future.