Lesson One Questions

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

Frontend development is about what those who visit a site see and use directly. This could include the programming of visual features like buttons, forms and images and the logic connected directly to them. In final form it involves HTML, CSS and JS. The back end is not affected directly by users and users cannot access the programming of it. It can involve different languages and involve work with the servers and APIs (eg requests). It can also involve connecting and interacting with databases.

1 Imagine that you are working like a full stack developer soon. Your team is asking for your advice on whether to use JavaScript or Python for a project and you think that Python would be the better choice. How would you explain the similarities and differences between the two languages to your team? Based on what you saw in this lesson what reasons would you give to convince your team that Python is the better option?

One benefit of Python is that it is simple in format and can be quick to learn and understand. The readable design of the language allows those who see it to basically be able to understand what is happening without too much explanation needed. It is dynamically typed like JavaScript but in a simpler way of writing it. Packages are also convenient because they add many features and come with Python. Management is also simple. Where being efficient, readable and simple are priorities Python might be the best choice.

2 Now that you got an introduction to Python write three goals which you made for yourself and learning during this Achievement. You can use these questions to reflect if you want. 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?

One goal is that I want to learn how to work with if/else statements and custom reuseable functions because these are helpful tools in so many languages. I also want to be able to take inputs and be prepared to process them effectively by the end of the achievement. I also hope to use Python for the back end a personal app which enables anonymous sharing of information.

Lesson Two Questions

0 Pretend like you are 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?

One benefit of iPython is how it uses different colours for text to indicate what the purpose of a specific part of the programming is. For example it might highlight errors, in/out and things with different types. The in and out are helpful for tracking because of matching numbers. Indentations are also another helpful feature. Tab completing can give helpful reminders, increase speed and help prevent spelling errors of commands.

1 Python has got many different data types which allow you to store and organise information. Say four examples of data types which Python recognises, define them and say whether they are scalar or non-scalar.

One data type is str or string and it means strings of characters in a readable format; they are in a structure which is non-scalar and includes different values in an arrangement. Another is int which means an integer (numbers with no point in them) and can be positive or negative. It is scalar and just contains a single value. Similarly the float type contains a decimal point and could be for a positive or negative number; it is also scalar. Tuples are arrays of elements which are not possible to modify and delete (they use round brackets) and lists are arrays of elements which I could modify or delete and even re-order the list (they use square brackets). Both are non-scalar. (Dictionaries are key value pair arrays which are also non-scalar.) Also the scalar bool (boolean) type can be either True or False. The NoneType is for None (basically Python's equivalent of null).

2 A frequent question at job interviews for Python developers is what is the difference of lists to tuples in Python? Write how you would respond.

Both lists and tuples are a form of array of elements in Python indicated by their use of square or round brackets in order mentioned. A list can be modified where elements can be changed and deleted (eg by popping them) and even re-ordered. A tuple is unmodifiable once it gets defined. It cannot get edited, sorted or other similar actions.

3 In the task for this Exercise you decided what you thought was the best data structure for storing all the information for a recipe. Now imagine that you are creating a language-learning app that helps users memorise 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 one would you choose? Think about their respective advantages and limitations and where flexibility might be useful if you continued developing the language-learning app beyond vocabulary memorisation.

I would probably use a list of dictionaries. The reason why I would use a list is because it is modifiable and flexible which would be good for an app where users could add and remove words, definitions and more. The dictionaries could be added to and removed from the original list. Dictionaries are arrays of key value pairs which is helpful because the keys could be used to label the purpose of each value (eg 'Word', 'Definition', 'Category' being keys for values: 'Edgy', 'Simplistically modern in design with an innovative quality', 'Adjective'). It could also be expanded to include other labels if the app went past vocabulary memorisation…like an app about computers (with keys 'Memory', 'Size', 'Type') or lightbulbs (with keys 'Brightness', 'Light Source', 'Colour Temperature', 'Attachment').