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').

Lesson Three Questions

0 Write script for a simple travel app using an if elif else statement for this situation.

The script should ask the user where the user wants to travel.
The user's input should be checked for three different travel destinations which you define.
If the user's input is one of those three destinations it should print 'You shall stay in _'
If the user's input is something other than the defined destinations it should print that: 'That destination is not available.'

```
print('You can choose to go to a city called Py, Trainbus or Fashion.')
choice = input('Type the destination which you want to go to with your ticket --> ')
destinations = ['py', 'trainbus', 'fashion']
if choice.lower() in destinations:
    print('You shall stay in', choice.lower().capitalize() + '.')
else:
    print('That destination is not available.')
```

1 Imagine that you are at an interview for a Python developer role. The interviewer says to 'Explain logical operators in Python'. Write how you would respond.

One logical operator in Python is 'and'. It is used to check if two conditions are true. It requires both to be true to return 'True'. Another is 'or' which checks if either one, the other or both conditions are true to return 'True'. The third is 'not' which is used to reverse 'True' to 'False' and 'False' to 'True'

2 What are functions in Python? When and why are they useful?

Functions in Python are basically reuseable units of programming which can be called to run by their name. They can accept and respond to (optional) params, perform tasks and return values. They are useful for nearly all programmes (other than small and simple ones) to avoid repetition. If there is a task which might need to be repeated or a situation where you need to respond to an input those might be hints to use a function. It also helps with problem identification: if you wrote a programme without functions it could get very difficult to find the source but functions help simplify the process of identifying issues.

3 In the section for the first lesson in the journal you had 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.

For my goal of learning about if/else statements I was able to learn about them and applied them to problem solving. I also made custom reuseable functions to make programming more organised and compact. I additionally met my goal of processing and responding to inputs with functions. I currently do not know how to use it for the back end but am excited to learn how to use it.

Lesson Four Questions

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

File storage is important when using Python because it ensures that the information does not disappear when a programme starts running. It is stored on the computer's memory and would only disappear if the user chose to do it. If I did not store local files the information generated by a programme would disappear if I closed the window running it. I would need to copy and paste it if I wanted to save it and that would be inefficient and lead to a risk of making mistakes.

1 In this Exercise you learnt 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 sets of bytes used to store complex sets of information in Python. I would use them for situations where the information is stored in a relatively complex format which would not translate well to more 'readable' file types like csv or txt. They are not readable when stored—making them impractical for very simple applications where it may be important for the user to understand the text in the file. They can be used for situations like dictionaries and—with the help of Python programming—the contents can be displayed correctly by an effective programmer.

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

The os.getcwd() function is used to determine the current directory. I could change it with os.chdir() with the content being the path to the directory wanted.

3 Imagine that 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 include it in a try block and place an except block after it. This means that it would attempt to run the block and—if an issue with it occurred—it would respond with the programming in the except block instead of the programme termination. It could inform the user than an issue happened instead of potentially very unreadable and confusing error notifications.

4 You're now more than halfway through achievement one! 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 practise with? Feel free to use these notes to guide your next mentor call.

My learning in the course is productive. I am taking my time to ensure that I re-learn Python well and not poorly. I am proud of the extra steps which I took to improve the user experience like formatting. I am currently not struggling but always think that there is room to learn more. I need to practise more problem solving because miniature challenges are a good way to learn.

Lesson Five Questions

0 What is object orientated programming? What are benefits of OOP?

Object orientated programming is programming where information and methods are made into structures called classes. The classes can store both data attributes and procedural attributes. A data attribute is basically a value or piece of information in the class. A procedural attribute is basically something which can achieve tasks like updating a data attribute. A benefit of it is that a class is like a template which can be used to create many objects each which their own characteristics. The objects are very much like real world objects (which possess dimensions, material, state of matter etc) because they can be made, possess characteristics and be modified. Another benefit is that it can easily be used for modelling real world objects because of this ability.

1 What are objects and classes in Python? Make a real-world example to demonstrate how objects and classes work.

An object in Python is something which is made from a class and possesses data/procedural attributes. There are custom ones from custom classes in addition to floats, integers, strings, lists etc and these are predefined. A class is a template which defines how to create an object in addition to the characteristics which an object can possess (data attributes) and methods which can be used for performing tasks (many times to get or set a data attribute). These are the procedural attributes. One real example is that someone decided to make a custom class for a lightbulb. It could possess the data attributes (model, category, weight, colour temperature, lumens) and methods for getting them and setting them. A person could then create a new lightbulb object and then use the setting methods to give that object its data attributes. That person could also create other lightbulb objects with different characteristics but based on the same class.

2 Write explanations (100-200 words) of these OOP concepts.

Inheritance is how attributes can be inherited by subclasses of a class. The default example is how a custom class inherits from an object. Subclasses of that custom class can then inherit the attributes of the parent class. For example if there was a parent class for vehicles which contained common characteristics and methods for getting/setting them (eg number of wheels, power, top speed, fuel) I could make a subclass to inherit these characteristics—like the ability to set number of wheels or increase fuel level—for different sorts of vehicles (eg ships, trains, buses) and then be able to use the methods from the vehicle class in addition to those which are specific to a category of vehicle (like 'throttle position' for a train or 'seating capacity' for a bus). Polymorphism is simply how a single name can be used in Python or other programming languages with independent definitions for a data attribute or method but in different classes. For example there can be different methods in the classes 'A', 'B' and 'C' all called the same exact name 'x' which does something different in each one of those three distinct classes. The definition within one class does not affect the function of the methods with identical names in other classes. A single name could essentially get used to do an infinite amount of things as long as it is in its own class. Operator overloading is how the special operators of Python like the addition or '+' symbol and others like the multiply '*' symbol can all get defined to function differently with different classes. The most memorable example is probably how the addition '+' symbol can get used to both join strings and add the floats and integers. The task of doing operator overloading involves using pre-defined method names from Python like __add__ for the addition symbol or __sub__ for the subtraction symbol surrounded by two underlines on both sides. The definition of these methods determines the behaviour when the operator is used with a specific class.