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 mainly uses markup languages to create the user interface of a website and add programs for the page to interact with the user for dynamic websites. Backend on the other hand, deals with form validation, api's, databases and alike that deals with creating and delivering dynamic content to users from the server.

If I were to be hired as backend programmer I would create and/or maintain api's, dynamic databases and server-side functions.

1. **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")

Python is easier and versatile to work with as it comes with installed open source packages that save time, uses understandable key works, allows dynamic typing, easier to debug due to its comfortable readability and its strong community support.

1. **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 syntax of all necessary packages python backend developers use.
- I want to have enough python knowledge to get comfortable with senior python developers.
- In the future I want to be able to use python together with other programming languages to create interactive web apps.

Data types in Python

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 types the code inserted in highlighted colors for different features of python enhancing and readability and indents codes within nested programs. Moreover, it provides guidance and is faster to test or debag small codes as it prints right away expected responses without the need to execute on separate files.

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 (int): Are non-rational numbers from zero to infinity in both directions, negative and positive. This data type is *Scalar*.
Boolean (bool): Is a data type of value either 'True' or 'False' which is mainly used to store output of a condition checked. This data type is *Scalar.*
String (str): Is an array of characters and symbols enclosed in double or single quotes. This data type is *Non-Scalar.*
Tuples: Is a linear array of multiple values of any data type stored. This data type is *Non-Scalar*.

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.

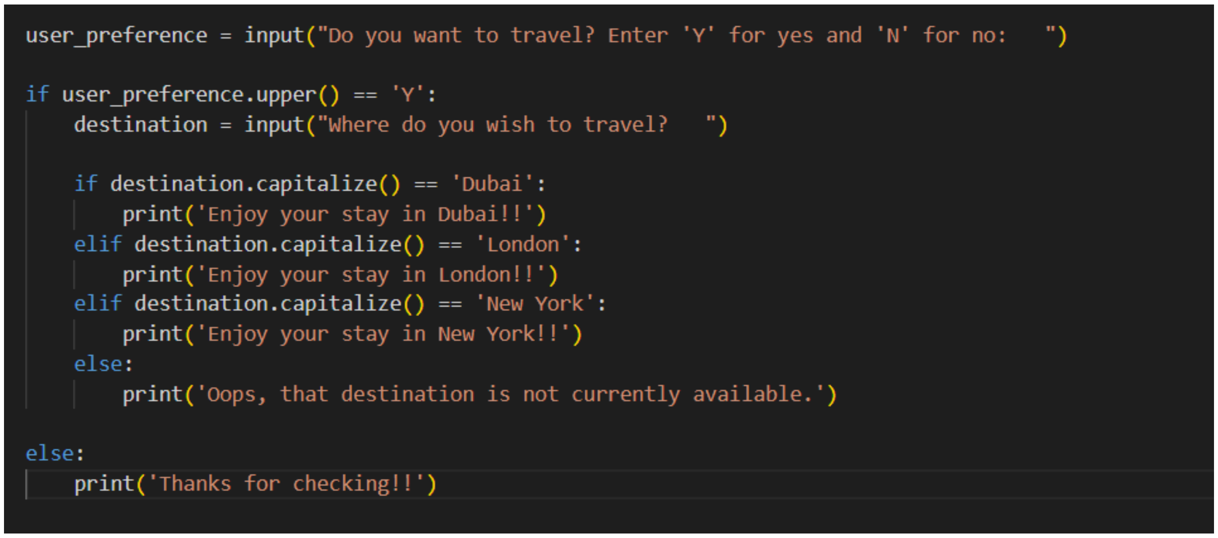
Lists and tuples are similar in a way that both are arrays stored data of any type. The difference is the internal elements of 'List' can be modified whereas 'Tuple' cannot be altered once it's defined (unless reassigned with the same variable name). i.e Lists are mutable and Tuples immutable.

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 this app I would use 'Dictionary' to store the information of each vocabulary in cards. Dictionaries have the advantage of storing date in key: value pairs which makes it easy to index and modify a particular attribute using its key name assigned. Also, dictionaries allow much more operations to be performed on the stored data enhancing flexibility that lists and tuples don't have.

In this Exercise, you learned how to use if-elif-else statements to run different tasks based on conditions that you define. Now practice that skill by writing a script for a simple travel app using an if-elif-else statement for the following situation:

- **The script should ask the user where they want to travel.**
- **The user's input should be checked for 3 different travel destinations that you define.**
- **If the user's input is one of those 3 destinations, the following statement should be printed: "Enjoy your stay in ______!"**
- **If the user's input is something other than the defined destinations, the following statement should be printed: "Oops, that destination is not currently available."**



```python
user_preference = input("Do you want to travel? Enter 'Y' for yes and 'N' for no:   ")

if user_preference.upper() == 'Y':
    destination = input("Where do you wish to travel?   ")

    if destination.capitalize() == 'Dubai':
        print('Enjoy your stay in Dubai!!')
    elif destination.capitalize() == 'London':
        print('Enjoy your stay in London!!')
    elif destination.capitalize() == 'New York':
        print('Enjoy your stay in New York!!')
    else:
        print('Oops, that destination is not currently available.')

else:
    print('Thanks for checking!!')
```

Imagine you're at a job interview for a Python developer role. The interviewer says "Explain logical operators in Python". Draft how you would respond.

Logical operators and keywords in python (mainly *in, and, not*) used to perform logical operations on two or more expressions based on their respective truth values. They return either True or False depending on the operator used. The 'not' operator checks if value of a variable is not the same to an expression/value. The 'and' operator checks if both expressions are of the same truth value (i.e both either True or False). And the 'or' operator checks whether at least one of the expressions is True or False.

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

Functions are block of codes defined by programmer that perform a specific task that runs only when it's called by its name (also predefined). Functions in Python helps to avoid excessive time consumptions made to write repetitive blocks of codes and facilitates code reusability.

File handling in Python

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

It's necessary to store data in file on local machine because python interpreter deletes all data when script files finishes running. If you didn't store files in local storage any data used on the script file will be erased and won't be available for later use.

In this Exercise you learned about the pickling process with the pickle.dump() method. What are pickles? In which situations would you choose to use pickles and why?

In complex data structures it's difficult to structure the details on text format file. Using pickle though, you can store complex data in binary format files which is not readable to humans, to machine only.

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

The function **'os.getcwd()'** - part of the **'os'** module - is used to locate the current directory you're working on. To change directory a function **'os.chdir()'** is used – again part of **'os'** module

Imagine 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 follow the **'try – except – else – finally'** approach to writing code. This should enable me to handle errors without terminating the whole script file rather continue to run without crashing.