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.

Functions and other operations in Python

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."**

```
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.

Object-Oriented Programming

In your own words, what is object-oriented programming? What are the benefits of OOP?

Object-oriented programming is a software programming approach that uses methods and objects to structure a simple reusable codes that can be used through defined methods. OOP has an advantage of making complex codes of program simple through its code reusability approach with further enhances efficiency and scalability.

What are objects and classes in Python? Come up with a real-world example to illustrate how objects and classes work.

Classes are predefined structure of code made of methods (to manipulate data) and attributes that are used to act as arranged plans/templates for individual objects created. Objects are instances of Class. For example, if I created a class 'UsedCar' to contain all the properties of a used car, for a car dealership, like 'color', 'mileage', 'history'… Then I can make an object of a sample used car say 'Toyota corolla' to represent a car with its properties on the list of cars.

In your own words, write brief explanations of the following OOP concepts?

Inheritance: is a concept of OOP that enables user to use one method for two or more classes by inheriting a method from one class to another instead of having to write similar/the same codes twice.
Polymorphism: is a concept in which two or more methods from different classes have the same name but perform different actions when called depending on form which instance it's being called.
Operator-overloading: is a process in which a user defines a method to run when python built in operators are called form a class instance. This is because the built in python operators does not work with data type classes thus would throw error if user tries to use them.

Connecting to databases in Python

What are databases and what are the advantages of using them?

Databases are secure and standardized storage formats alternative to the norm local storages. Databases have far more advantages to regular local storage in a way that it's more secure where a user is required to pass through password access, have formats consistent with the international standards and can be accessed using applications.

List 3 data types that can be used in MySQL and describe them briefly:

Varchar(n): is string with maximum length restricted to n.

Int: Integers are standard zero to infinity and zero to negative infinity numbers without decimal points

Float: are numbers with decimal floating points ranging from negative infinity to positive infinity.

In what situations would SQLite be a better choice than MySQL?

SQLite is a smaller version that need no installation and stores data in .db files which can be modified directly with applications. It is useful for databases that are simple and don't require setting up an entire database.

Think back to what you learned in the Immersion course. What do you think about the differences between JavaScript and Python as programming languages?

I think the first thing that comes to mind focusing on difference be python and other languages is the simplicity and readability of python which makes its learning curve relatively flattened. Python also has many built in methods and functions that can be used without installing other packages. But the main difference could be that python is mainly used for backend/server side programming while JS on frontend programming and user interaction as it has full support for DOM manipulation. Both have fairly the same large community support.

What would you say are the limitations of Python as a programming language?

The fact that python runs and executes line by line makes it slow and on sector where efficiency is key, this is major limitation of python. Furthermore, access to databases through python is not user-friendly/ not straight forward and it's not used to make mobile and web application despite its simplicity and versatility.

Finalizing Python Program

What is an Object Relational Mapper and what are the advantages of using one?

ORM are used for database conversion in which it converts database structure and entries into classes and objects. Their advantage is it's simpler to use them, in that the database can be accessed and edited using python syntax rather than SQL syntax when converted using ORM.

By this point, you've finished creating your Recipe app. How did it go? What's something in the app that you did well with? If you were to start over, what's something about your app that you would change or improve?

The recipe app was easier I think to code once you've got the frame structured. I believe the error handlers are the most important part of the app and it's a good point to start learning about how to make user friendly apps using error handlers. I think the app is fine overall and there wouldn't be anything I would change.

Imagine you're at a job interview. You're asked what experience you have creating an app using Python. Taking your work for this Achievement as an example, draft how you would respond to this question.

I've made an app that runs in terminal, which uses python scripts to add and edit SQL database using sqlalchemy ORM. The app basically starts a table and lets user choose whether to add recipe, display recipes list from database, edit attributes (columns) of recipes form database and delete recipes from the database.

Take a moment to reflect on your learning in the course so far:

a. **What went well during this Achievement?**

The fact that I had some previous experience with python syntaxes helped much and this achievement was an added skills to my python knowledge. It was also helpful that each exercise does not cover too much stuff.

b. **What's something you're proud of?**

I think the fact that I was able to finish this achievement in less than 2 weeks is worth being proud of. Fair play I had some experience with python but still I had some moments where I was stuck for a day for silly issues and that taught me patience and taking some time off.

c. **Did this Achievement meet your expectations? Did it give you the confidence to start working with your new Python skills?**

To be honest I don't think I'm comfortable enough to work databases using ORM's, other than that I'm very confident I'll be able to work with python.

d. **What's something you want to keep in mind to help you do your best in Achievement 2?**

Self-discipline helped me finish this achievement well before scheduled time and I think I should continue practicing it,

Getting Started with Django

Suppose you're a web developer in a company and need to decide if you'll use vanilla (plain) Python for a project, or a framework like Django instead. What are the advantages and drawbacks of each?

Django is made to help developers create their apps fast with robust security and ease of scalability. It also comes loaded with a large library of tools and features you can use. Moreover, it provides developers its 'Admin interface' which an authorized person can easily manage contents of the page. This makes django cost effective and efficient if used in large projects with short deadlines. Its disadvantages on the other hand are: django has its own ways of performing tasks – Knows as "The Django way" – which further steepens its learning curve and it takes large portion of server processing and time for small projects & its strength is seen as the code grows.

In your own words, what is the most significant advantage of Model View Template (MVT) architecture over Model View Controller (MVC) architecture?

In MVC structure you have to write the control that stands in the middle of model (which interacts with the data base and fetches data per request) and view (which prepares the presentation of the page), coordinates both and send back HTTPS for every request received. On the other hand, in MVT structure the view stands in the middle of model (interacts with database and fetches data) and template (prepares presentation), coordinate them appropriately and send back a response URL. The advantage of MVT lies in that when request comes the view matches the URL with specific request and returns response, so here you don't need to write code for the view part like you do for the controller part in MVC.

Now that you've had an introduction to the Django framework, write down three goals you have for yourself and your learning process during this Achievement.

By the end of this achievement I should be confident enough to work with different features and tools of django. I think if I can make applications with heavy traffic requests or large amount of data to manipulate and scale up or down apps made using django, this achievement is a win for me.

Django Project Setup

Suppose you're in an interview. The interviewer gives you their company's website as an example, asking you to convert the website and its different parts into Django terms. How would you proceed? For this question, you can think about your dream company and look at their website for reference.

If I can take the citi bank website for example, the whole web application is the 'project'. Inside this project there are multiple individual 'apps' that perform different tasks like the 'login' app let you get authenticated and login to your profile page and the 'portfolio' app let you see your investments holdings page

In your own words, describe the steps you would take to deploy a basic Django application locally on your system.

I first need to install django then start a new project using commands from django's built in tools. This will create the general file structure for my app. I would then run migrations to create database for the app of the type I configured on the settings.py file of my project. Once I've created database I would create super-user for the admin and update my database. Then I would add apps to my project that perform specific functions within the project. Finally, I would run the server in local host.

Do some research about the Django admin site and write down how you'd use it during your web application development.

The django admin panel is an interface where users can create, read, update and delete on the model directly. It is an instant interface where user can manage contents of the site using CRUD operations, which saves a lot of development time.

Django Models

Do some research on Django models. In your own words, write down how Django models work and what their benefits are.

Django models is the similar for SQL that developers use with Django to create tables and their attributes. Django models simplify the task of sending SQL queries to perform CRUD and similar operations on databases on Django's admin panel.
To use models you should have project and an app started within and you use two commands: 'makemigrations' and 'migrate' to update changes to the database. You can then add models to the models.py file of the app and register your model to the admin.py file to easily modify the database using the admin UI.
In creating models, the whole model is defined as a class with database fields specified as class attributes (ex. 'CharField', 'DateField',..) and each field have field options which are arguments used to add constraints or characteristics to the corresponding field.

In your own words, explain why it is crucial to write test cases from the beginning of a project. You can take an example project to explain your answer.

It is recommended to write automated tests starting early in a project and then write as much code needed to pass the test to avoid sudden issue that lead the app to break or simply prevent bugs form the get go.
It's important as it speeds up development time as programmers avoid patterns that produce errors and adopt best practices that prevent errors. Furthermore, it ensures fully functional, tested and robust program, starting from the basics to the complex features of an app.