PYTHON ASSIGNMENT : AYISHA NAJEEHA

1. What is Python?

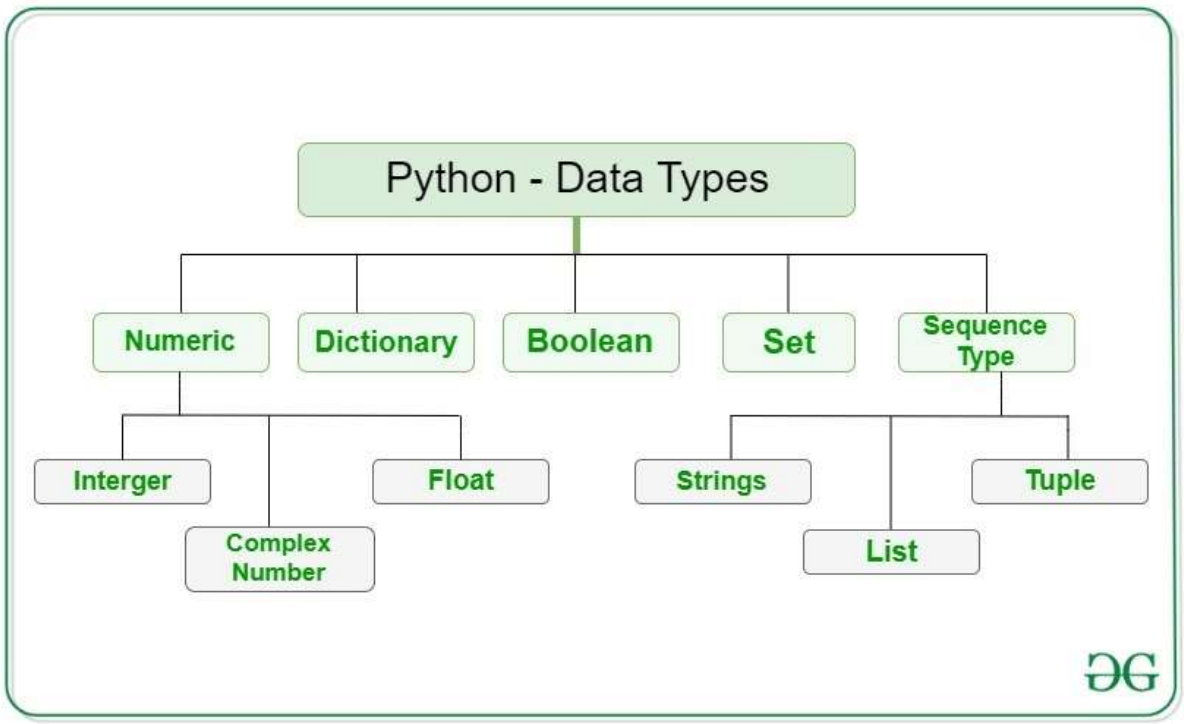
    a. Python is a scripting language which uses interpreter to convert to machine language

2. What is the purpose of

- PYTHONPATH --- It is used to set the path for the user-defined modules so that it can be directly imported into a Python program. It is also responsible for handling the default search path for Python Modules.

- PYTHONSTARTUP----PYTHONSTARTUP:
    1. This variable contains path of an initialization file containing Python source code.
    2. When we start the Python interpreter, initialization is executed each time.
    3. In the Unix system, it is named as pythonrc.py
    4. The commands are included to load utilities or change PYTHONPATH
- PYTHONCASEOK:
    1. It is used in the Windows operating system.
    2. It is used to find the first case-insensitive match in the import statement.
    3. To activate this variable, we can set it to any value.
- PYTHONHOME:
    1. It is normally used within PYTHONSTARTUP or PYTHONPATH directories.
    2. It is used to make easiness in switching module libraries.

3. What are the supported data types in Python?



4. What is the difference between list and tuples?
   The key difference between the tuples and lists is that while the **tuples are immutable objects the lists are mutable**. This means that tuples cannot be changed while the lists can be modified.
5. How is memory managed in Python?
   Memory allocation can be defined as allocating a block of space in the computer memory to a program. In Python memory allocation and deallocation method is automatic as the Python developers created a garbage collector for Python so that the user does not have to do manual garbage collection.
6. Explain Inheritance in Python with an example.

   Inheritance is a powerful feature in object oriented programming.It refers to defining a new class with little or no modification to an existing class. The new class is called **derived (or child) class** and the one from which it inherits is called the **base (or parent) class**.

   class baseclass:

   body of base class

   class derivedclass(baseclass):

body of derived class

example

```
class vehicle:
        def feature(self):
                print("I can ride")
class  motorbike(vehicle):
        def twowheeler(self):
                print("I have two wheels")
class car(vehicle):
        def fourwheeler(self):
                print("I have four wheels")
m=motorbike()
m. twowheeler()
m. feature()
c=car()
c. fourwheeler()
c. feature()
```

7. Whenever Python exits, why isn't all the memory de-allocated?

   Whenever Python exits, especially those Python modules which are having circular references to other objects or the objects that are referenced from the global namespaces are not always de-allocated or freed.

8. What is dictionary in Python?
   Dictionaries are used to store data values in key:value pairs.
   A dictionary is a collection which is ordered*, changeable and do not allow duplicates.

```
dictionary = {"brand": "Ford","model": "Mustang","year": 1964}
print(dictionary)
```

9. Write a one-liner that will count the number of capital letters in a file. Your code should work even if the file is too big to fit in memory.

```
With open(some_large_file) as fh:
        Count=0
Text=fh.read()
for character in text:
        if character.isupper():
                count+=1
```

10. Write a sorting algorithm for a numerical dataset in Python.

    List name.sort() --- will sort the list

```
List1=[23,45,56,85,64,75]
List1.sort()
Print(list1)
```

11. How will you reverse a list?
    Listname.reverse()---- will reverse the list

```
List1=[23,45,56,85,64,75]
List1.reverse()
Print(list1)
```

12. How will you remove last object from a list?
    Listname.pop()--- will remove last element

```
List1=[23,45,56,85,64,75]
List1.pop()
Print(list1)
```

13. What are negative indexes and why are they used?

Python programming language supports negative indexing of arrays, something which is not available in arrays in most other programming languages. This means that the index value of -1 gives the last element, and -2 gives the second last element of an array. The negative indexing **starts from where the array ends**.

14. Explain split(), sub(), subn() methods of "re" module in Python.
    Split(): to split a given string into a list
    Sub():**finds all substrings where the regex pattern matches** and then replace them with a different string.
    Subn():being similar to sub() it also returns the new string along with the number of replacements

15. What is the difference between range & xrange?

range()	**xrange()**
Returns a list of integers.	Returns a generator object.
Execution speed is slower	Execution speed is faster.

16. What is a Python module?
    A **Python module** is a file containing Python definitions and statements. A module can define functions, classes, and variables.
17. Name the File-related modules in Python?
    Python has several built-in modules and functions for handling files. These functions are spread out over several modules such as **os , os. path , shutil , and pathlib**
18. Explain the use of with statement?
    with statement **helps avoiding bugs and leaks** by ensuring that a resource is properly released when the code using the resource is completely executed. The with statement is popularly used with file streams
19. Explain all the file processing modes supported by Python?
    In Python, there are 3 modes that allow you to process files: **read-only mode, write-only mode, read-write mode, and append mode** by specifying the flags "r", "w", "rw", "a" respectively
20. How many kinds of sequences are supported by Python? What are they?
    Python supports **six different types** of sequences. These are strings, lists, tuples, byte sequences, byte arrays, and range objects.
21. How do you perform pattern matching in Python? Explain
    Pattern matching is **the process of checking whether a specific sequence of characters/tokens/data exists among the given data**.

Regular programming languages make use of regular expressions (regex) for pattern matching.

22. How to display the contents of text file in reverse order?
    Use reversed() to read a file line by line backwards

23. Which of the following is an invalid statement?

a) abc = 1,000,000 ---wrong

b) a b c = 1000 2000 3000---wrong

c) a,b,c = 1000, 2000, 3000 ----true

d) a_b_c = 1,000,000 ---wrong

24. What is the output of the following? try: if '1' != 1: raise a) some Error has occured b) some Error has not occured c) invalid code d) none of the above

c-invalid code

25. Suppose list1 is [2, 33, 222, 14, 25], What is list1[-1] ?

List[-1]: 25

26. To open a file c:\scores.txt for writing?

open("c:\\scores.txt", "w")

27. Is Python object oriented? what is object oriented programming?

Yes ,

Object-oriented programming (OOP) is **a programming paradigm based on the concept of "objects"**, which can contain data and code: data in the form of fields (often known as attributes or properties), and code, in the form of procedures

28. What is multithreading? Give an example.

Multithreading **enables us to run multiple threads concurrently**. For example in a web browser, we can have one thread which handles the user interface, and in parallel we can have another thread which fetches the data to be displayed. So multithreading improves the responsiveness of a system

29. Does Python supports interfaces like in Java? Discuss.

No, python does not have any equivalent of **interfaces** . Since Python does support multiple inheritance, you can easily emulate the equivalence of interfaces. interfaces are **implicit in Python** : if an object conforms to an interface, you can

use it, no need to declare it like you would in statically typed languages such as **Java** or **C#** .

30. Which of the following is not the correct syntax for creating a set