PYTHON DATATYPES

-->>In python data type,specify type of data that can be stored in a variable

-->>Python Data Types

Data Types	Classes	Description
Numeric	int, float, complex	holds numeric values
Sequence	list, tuple, string	holds collection of items
Mapping	dict	holds data in key-value pair form
Boolean	bool	holds either True or False
Set	set	hold collection of unique items

1-->>Python Numeric Data type-

In Python, numeric data type is used to hold numeric values

--> int - holds signed integers of non-limited length.
-->float - holds floating decimal points and it's accurate up to 15 decimal places.
--> complex - holds complex numbers.

NOTE-->type() function is used to find the type of numeric data type
example num1 = 5
print(num1, 'is of type', type(num1))
output is 5 is of type <class 'int'>

2-->>Python Dictionary Data Type

--> Python dictionary is an ordered collection of items. It stores elements in key/value pair

--> Here, keys are unique identifiers that are associated with each value.
for example -# create a dictionary named capital_city
capital_city = {'Nepal': 'Kathmandu', 'Italy': 'Rome', 'England':'London'}
print(capital_city)
output {'Nepal': 'Kathmandu', 'Italy': 'Rome', 'England': 'London'}

3-->>Boolean Data Type in Python-

--> Data type with one of the two built-in values, True or False.
--> Boolean objects that are equal to True are truthy (true), and those equal to False are falsy (false).
--> But non-Boolean objects can be evaluated in a Boolean context as well and determined to be true or false. It is denoted by the class bool.
for example print(type(True))

print(type(true))
output- <class 'bool'>
error true not defined

4-->>Set Data Type in Python-

-->>In Python, a Set is an unordered collection of data types that is iterable, mutable and has no duplicate elements
-->>Sets can be created by using the built-in set() function with an iterable object or a sequence by placing the sequence
inside curly braces, separated by a 'comma'
for example set1 = set("GeeksForGeeks")
print("\nSet with the use of String: ")

```
                    print(set1)
           output   Set with the use of String:
                {'F', 'o', 'G', 's', 'r', 'k', 'e'}
```

5-->>Sequence Data Type in Python
                -->>The sequence Data Type in Python is the ordered collection of similar or differe
nt data types.
                Sequences allow storing of multiple values in an organized and efficient fashion

        5.1--> String Data Type
                --> Strings in Python are arrays of bytes representing Unicode characters.
                -->A string is a collection of one or more characters put in a single quote, d
ouble-quote, or triple-quote
                -->In python there is no character data type, a character is a string of length
 one.

                for example String1 = 'Welcome to the Geeks World'
                        print("String with the use of Single Quotes: ")
                        print(String1)
                    output   String with the use of Single Quotes:
                            Welcome to the Geeks World

        5.2--> List Data Type
                --> Lists are just like arrays,which is an ordered collection of data
                --> List are mutable i.e. can be modified after it is created
                --> Lists in Python can be created by just placing the sequence inside the sq
uare brackets[].
                for example
                        List = [['Geeks', 'For'], ['Geeks']]
                        print("\nMulti-Dimensional List: ")
                        print(List)
                    output   Multi-Dimensional List:
                            [['Geeks', 'For'], ['Geeks']]

        5.3--> Tuple Data Type
                --> Just like a list, a tuple is also an ordered collection of Python objects.
                --> The only difference between a tuple and a list is that tuples are immutabl
e i.e. tuples cannot be modified after it is created
                --> In Python, tuples are created by placing a sequence of values separated
 by a 'comma'
                    with or without the use of parentheses for grouping the data sequence.
                for example list1 = [1, 2, 4, 5, 6]
                        print("\nTuple using List: ")
                        print(tuple(list1))
                    output   Tuple using List:
                            (1, 2, 4, 5, 6)