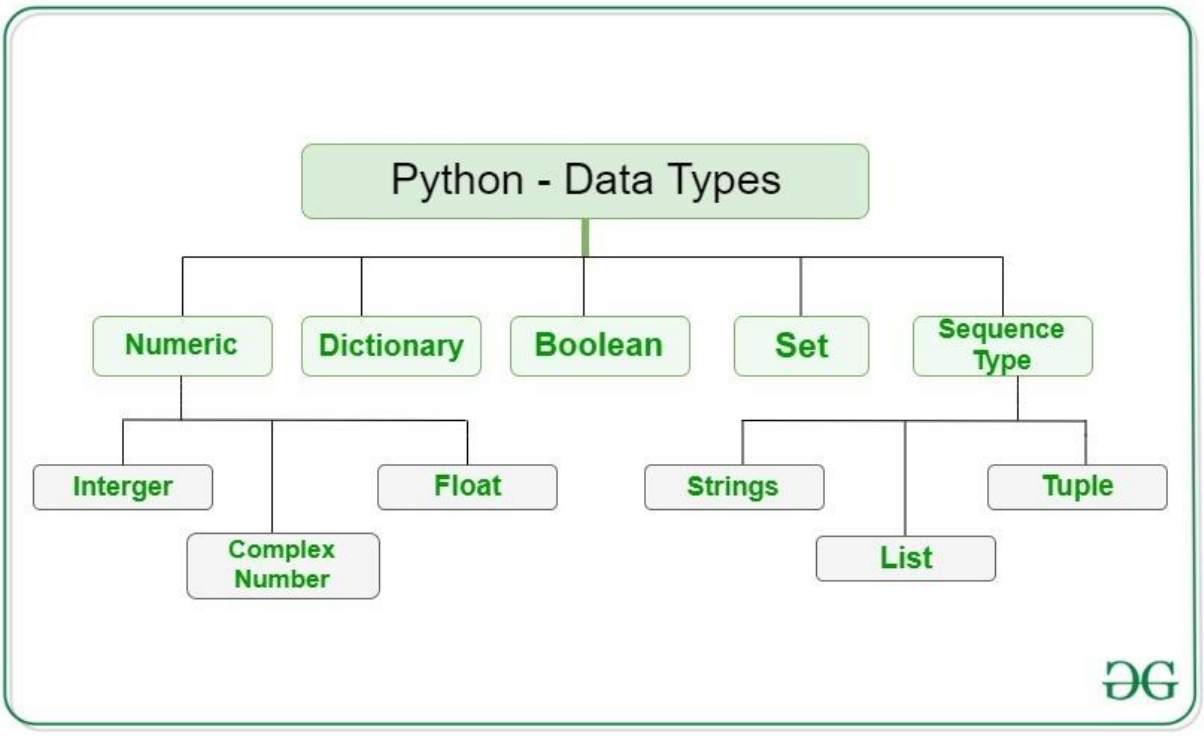
DATA TYPES

Python Data types are the classification or categorization of data items. It represents the kind of value that tells what operations can be performed on a particular data. Since everything is an object in Python programming, Python data types are classes and variables are instances (objects) of these classes. The following are the standard or built-in data types in Python. **Python Sets**



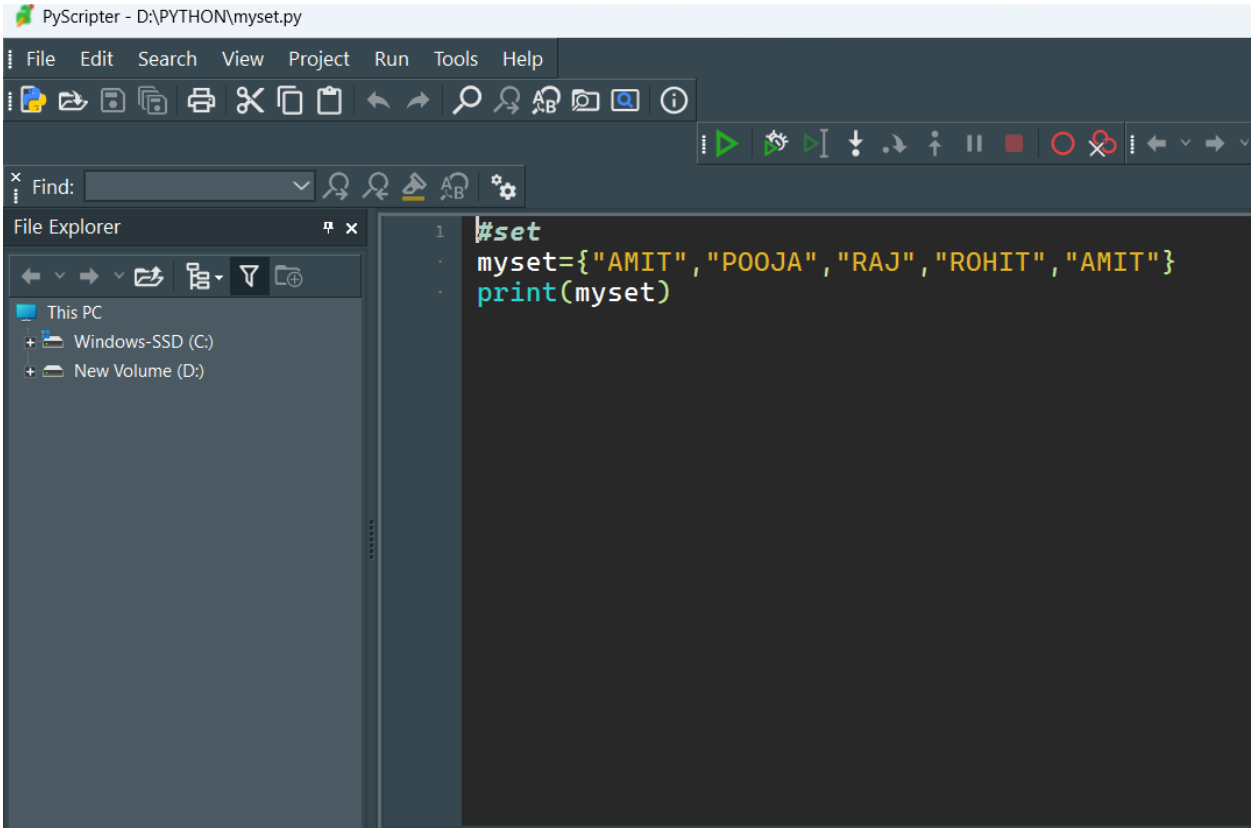
Python Sets

Python Set is an unordered collection of data types that is iterable, mutable, and has no duplicate elements. The order of elements in a set is undefined though it may consist of various elements. The major advantage of using a set, as opposed to a list, is that it has a highly optimized method for checking whether a specific element is contained in the set. Here, we will see what is a set in Python and also see different examples of set Python.

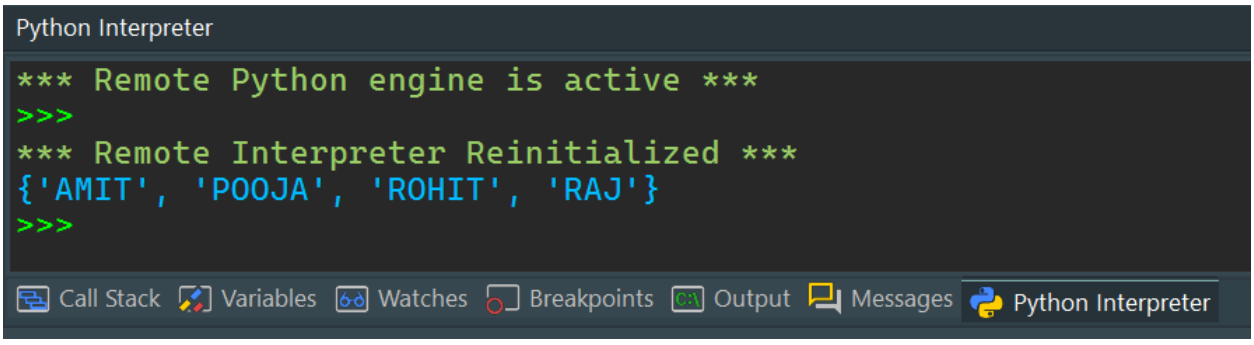
Creating a Set in Python

Python 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.

EXAMPLE:



OUTPUT:



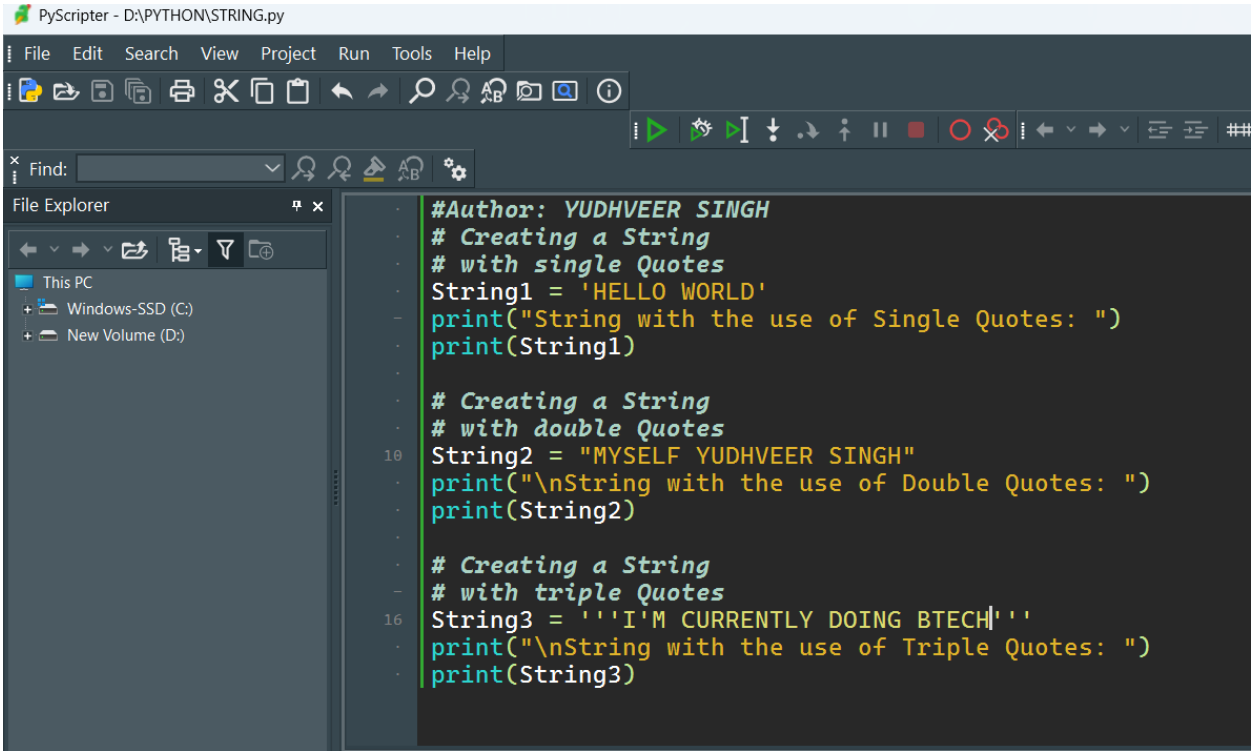
Python String

A String is a data structure in Python Programming that represents a sequence of characters. It is an immutable data type, meaning that once you have created a string,

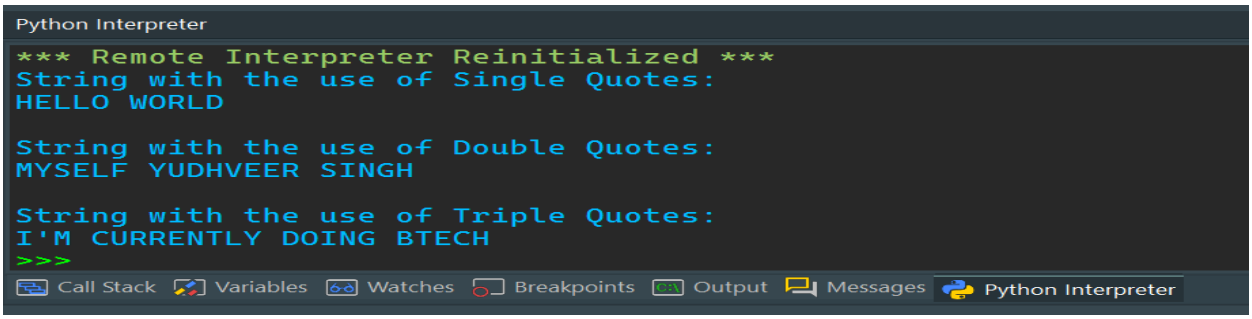
you cannot change it. Python String are used widely in many different applications, such as storing and manipulating text data, representing names, addresses, and other types of data that can be represented as text

What is a String in Python?
Python Programming does not have a character data type, a single character is simply a string with a length of 1.
Syntax of String Data Type in Python
string_variable = 'Hello, world!'
Create a String in Python
Strings in Python can be created using single quotes or double quotes or even triple quotes. Let us see how we can define a string in Python or how to write string in Python.
Example:
In this example, we will demonstrate different ways to create a Python String. We will create a string using single quotes (‘ ‘), double quotes (” “), and triple double quotes (“”” “””). The triple quotes can be used to declare multiline strings in Python.

EXAMPLE:



OUTPUT:



Python Lists

Python Lists are just like dynamically sized arrays, declared in other languages (vector in C++ and Array List in Java). In simple language, a list is a collection of things, enclosed in [ ] and separated by commas.

The list is a sequence data type which is used to store the collection of data.

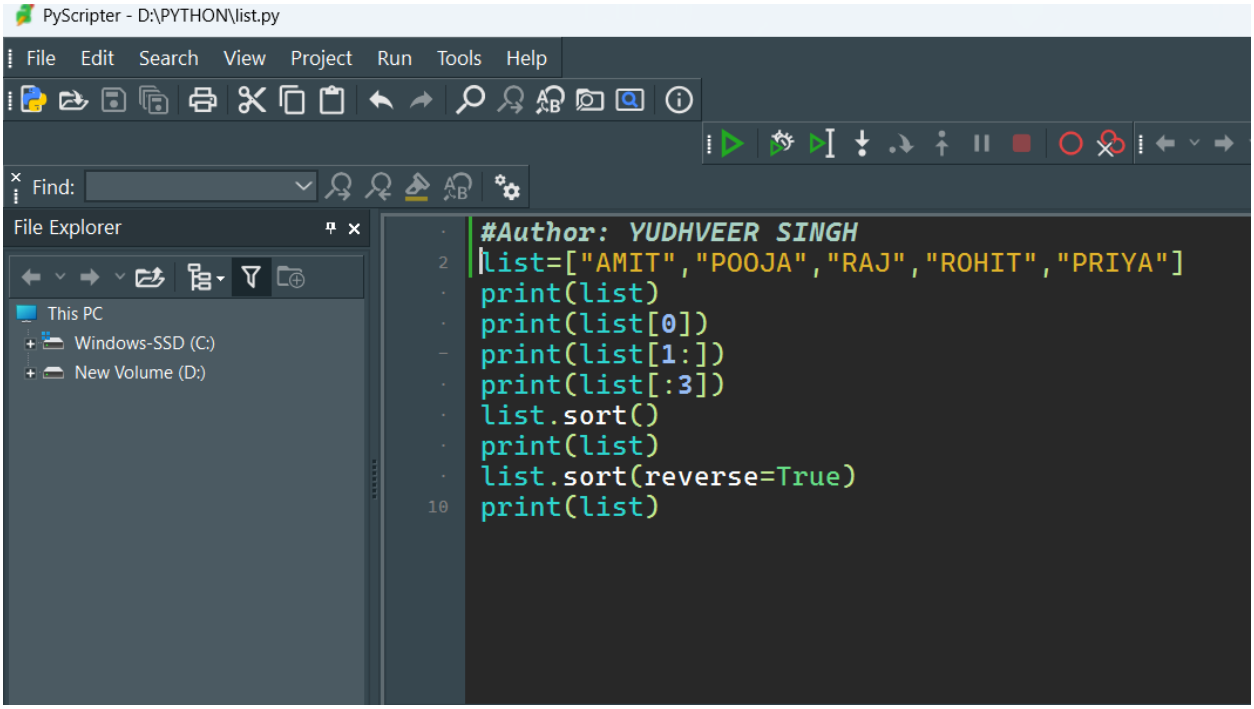
Creating a List in Python

Lists in Python can be created by just placing the sequence inside the square brackets[]. Unlike Sets, a list doesn't need a built-in function for its creation of a list

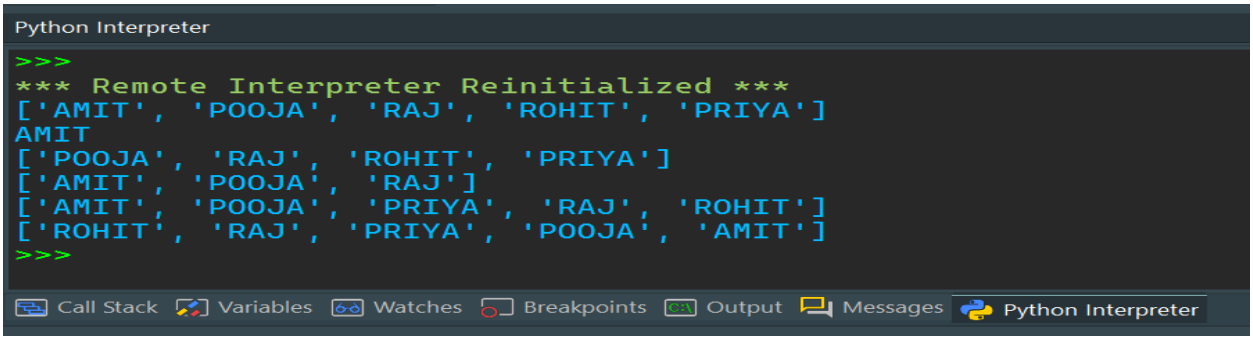
Accessing elements from the List

In order to access the list items refer to the index number. Use the index operator [ ] to access an item in a list. The index must be an integer. Nested lists are accessed using nested indexing.

EXAMPLE:



OUTPUT:



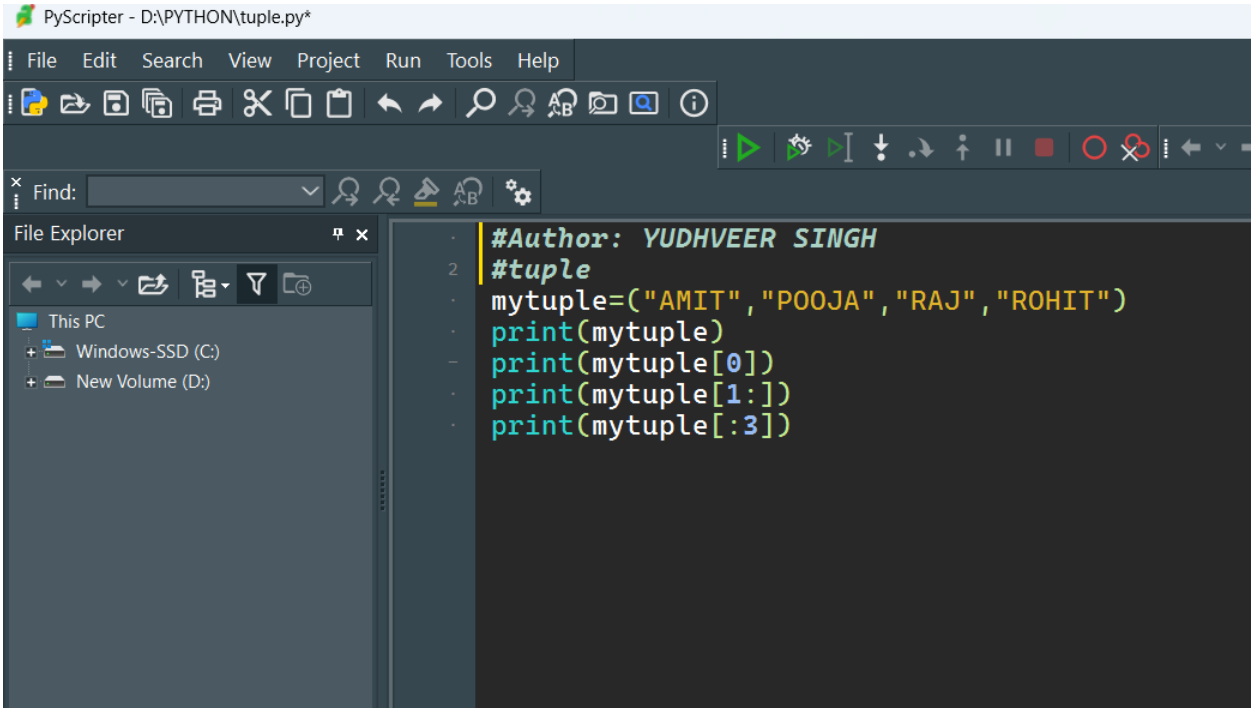
Python Tuples

Python Tuple is a collection of Python Programming objects much like a list. The sequence of values stored in a tuple can be of any type, and they are indexed by integers. Values of a tuple are syntactically separated by '**commas**'. Although it is not necessary, it is more common to define a tuple by closing the sequence of values in parentheses. This helps in understanding the Python tuples more easily.

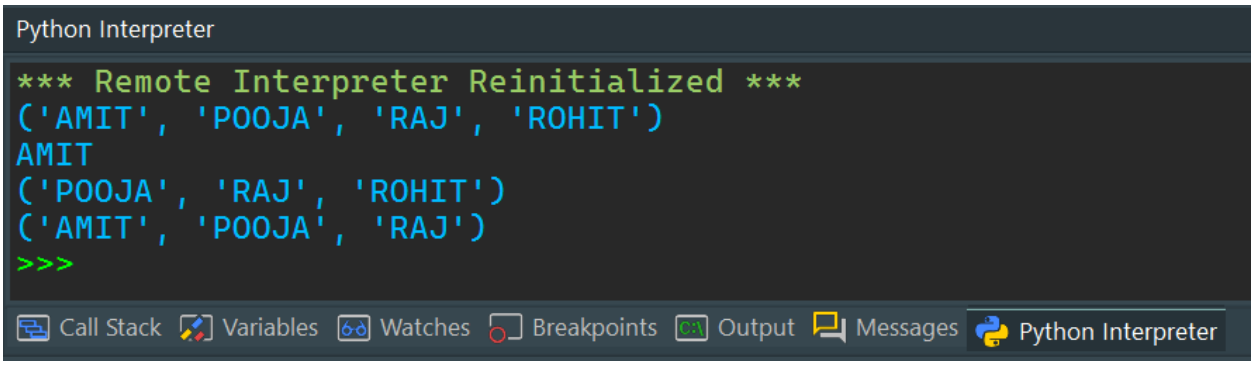
Creating a Tuple

In Python Programming, tuples are created by placing a sequence of values separated by 'comma' with or without the use of parentheses for grouping the data sequence.

EXAMPLE:



OUTPUT:



Dictionaries in Python

Dictionaries in Python is a data structure, used to store values in key :value format. This makes it different from lists, tuples, and arrays as in a dictionary each key has an associated value.

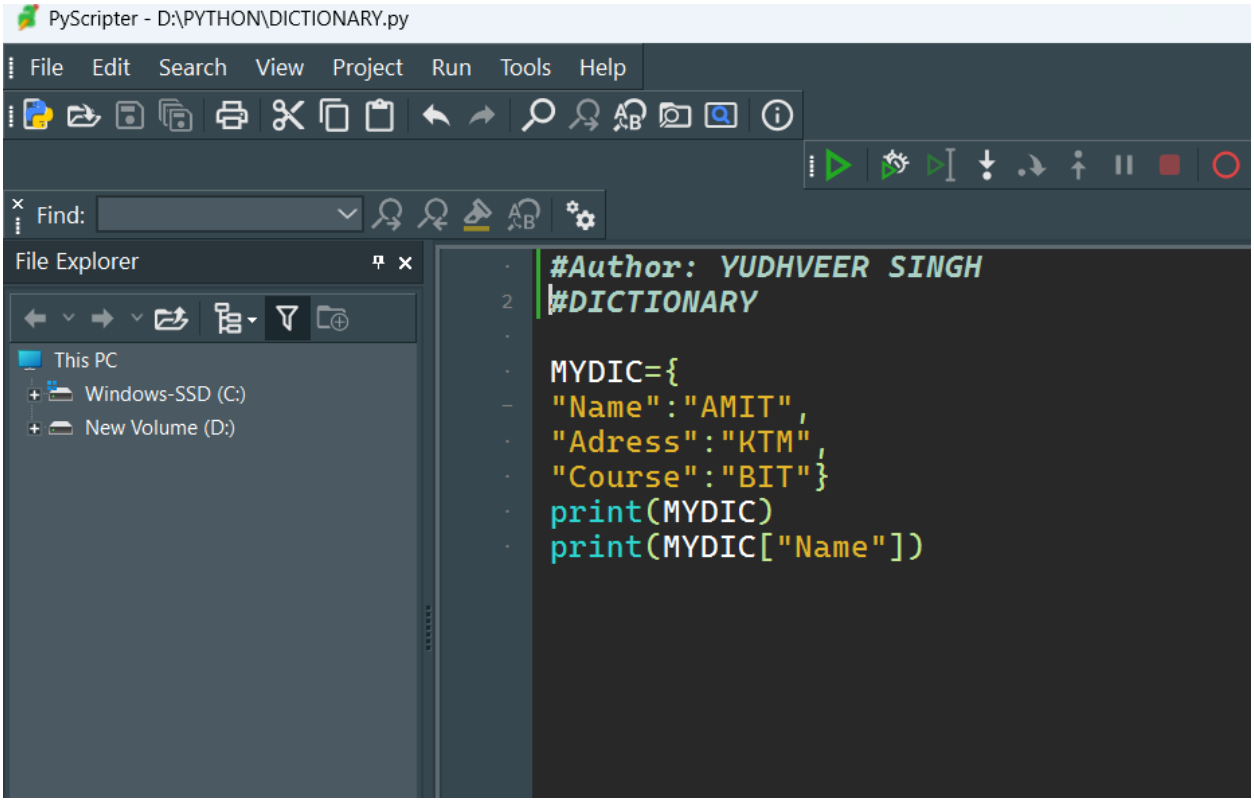
How to Create a Dictionary

In Python, a dictionary can be created by placing a sequence of elements within curly **{}** braces, separated by a 'comma'.

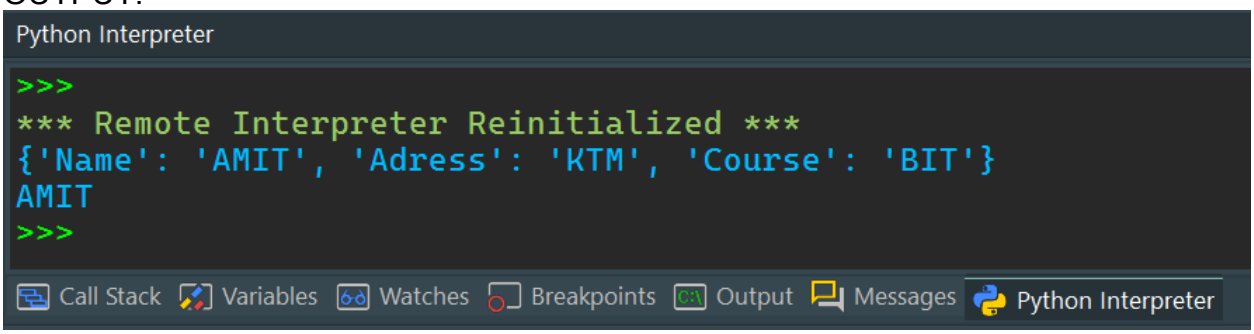
The dictionary holds pairs of values, one being the Key and the other corresponding pair element being its **Key:value**.

Values in a dictionary can be of any data type and can be duplicated, whereas keys can't be repeated and must be *immutable*.

EXAMPLE:


PyScripter - D:\PYTHON\DICTIONARY.py
File Edit Search View Project Run Tools Help
Find:
File Explorer
This PC
Windows-SSD (C:)
New Volume (D:)
#Author: YUDHVEER SINGH
#DICTIONARY
MYDIC={
"Name":"AMIT",
"Adress":"KTM",
"Course":"BIT"}
print(MYDIC)
print(MYDIC["Name"])

OUTPUT:


Python Interpreter
>>>
*** Remote Interpreter Reinitialized ***
{'Name': 'AMIT', 'Adress': 'KTM', 'Course': 'BIT'}
AMIT
>>>
Call Stack
Variables
Watches
Breakpoints
Output
Messages
Python Interpreter