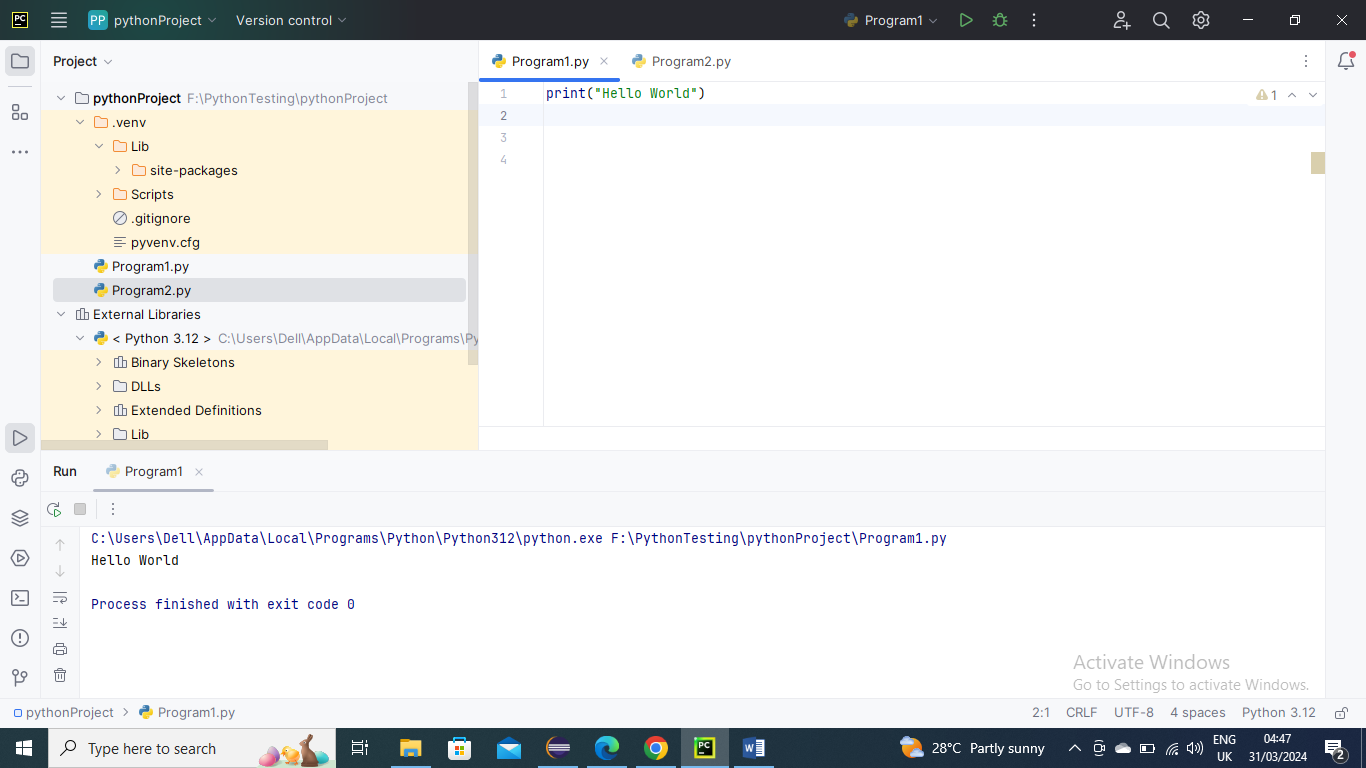
Python is a general-purpose, dynamically typed, high-level, compiled and interpreted, garbage-collected, and purely object-oriented programming language that supports procedural, object-oriented, and functional programming.

Python is case sensitive

Below is an example of Hello World

print("Hello World")



Python Variables:

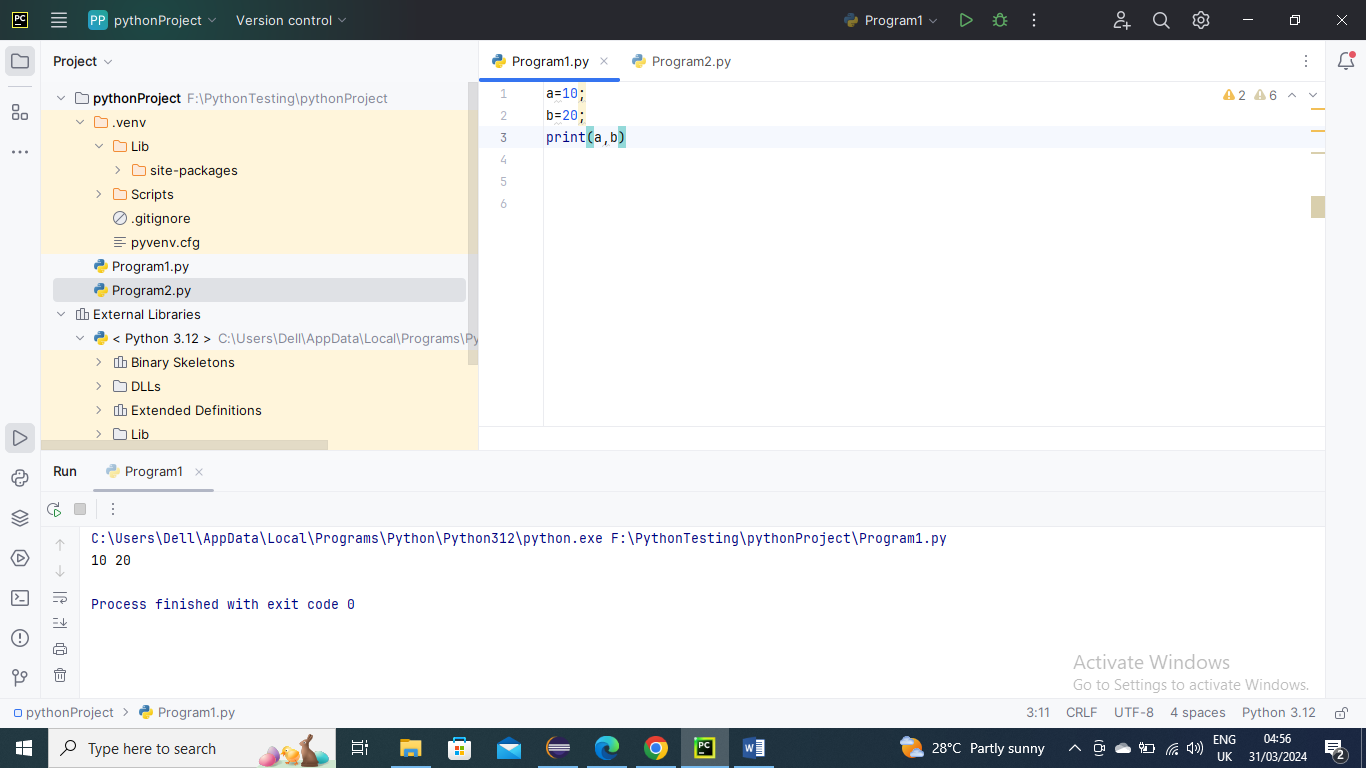
A variable is the name given to a memory location. A value-holding Python variable is also known as an identifier.

Variable names must begin with a letter or an underscore, but they can be a group of both letters and digits.

The name of the variable should be written in lowercase. Both Rahul and rahul are distinct variables.

Below is an example here “a” and “b” are varaiables

a=10;  
b=20;  
print(a,b)

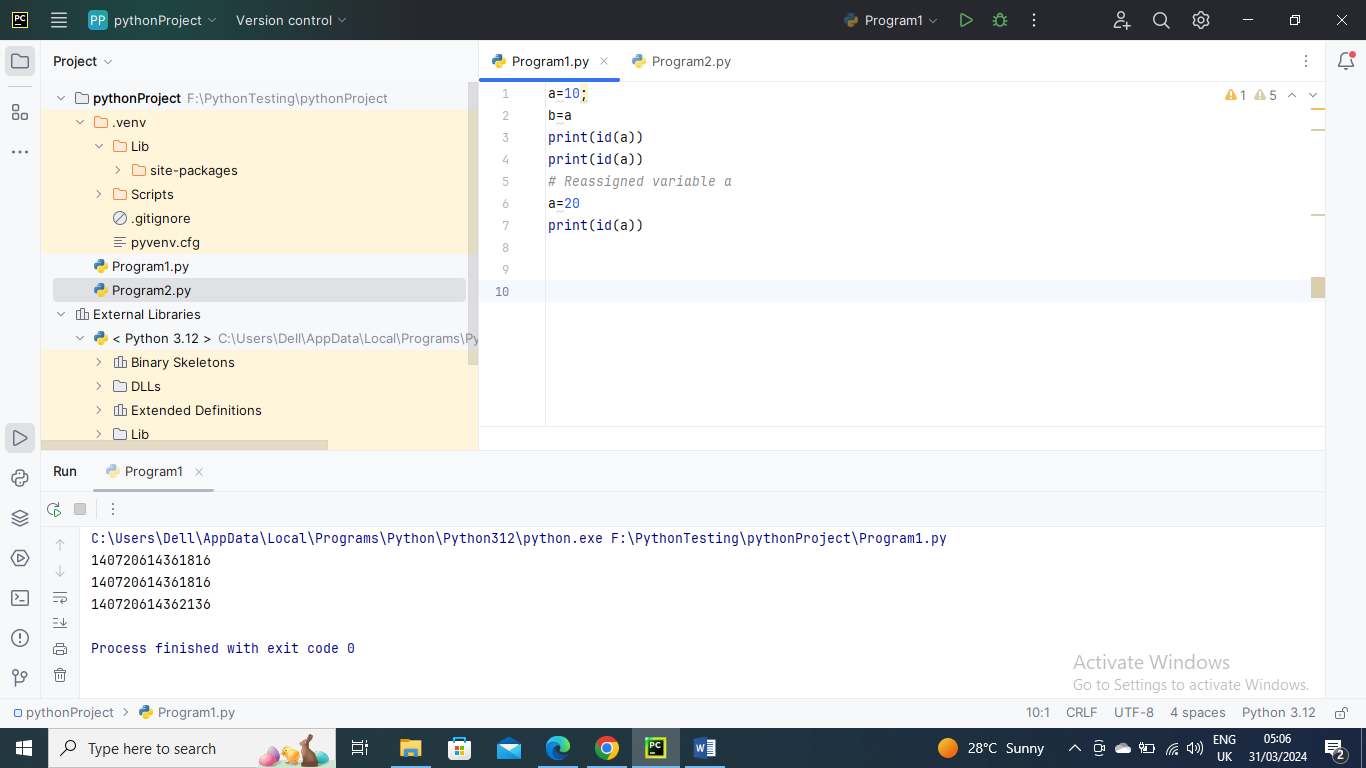


Below is the example where I have assigned a to 10 and B to a, here variable b refers to the same object that a points to because Python does not create another object.

a=10;  
b=a  
print(a,b)

Below is the picture representation for above where variables a and b use same object when I re-assigned a to different value to creates a new object…..As you can see in screenshot the id values for a and b are same and then when value a is assigned to different value it made a new id

a=10;  
b=a  
print(id(a))  
print(id(a))  
*# Reassigned variable a*a=20  
print(id(a))

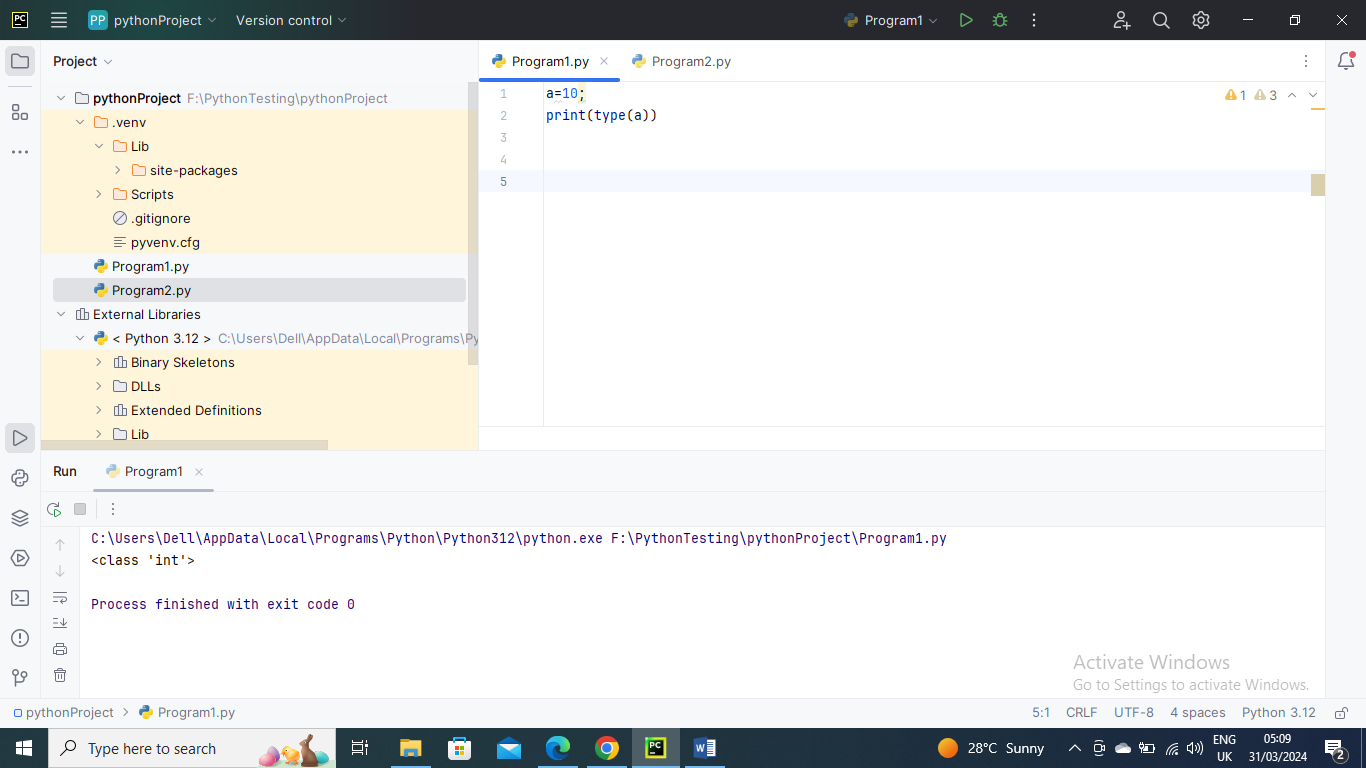


Python Data Types:

Every value has a datatype, and variables can hold values. Python is a powerfully composed language; consequently, we don't have to characterize the sort of variable while announcing it. The interpreter binds the value implicitly to its type.

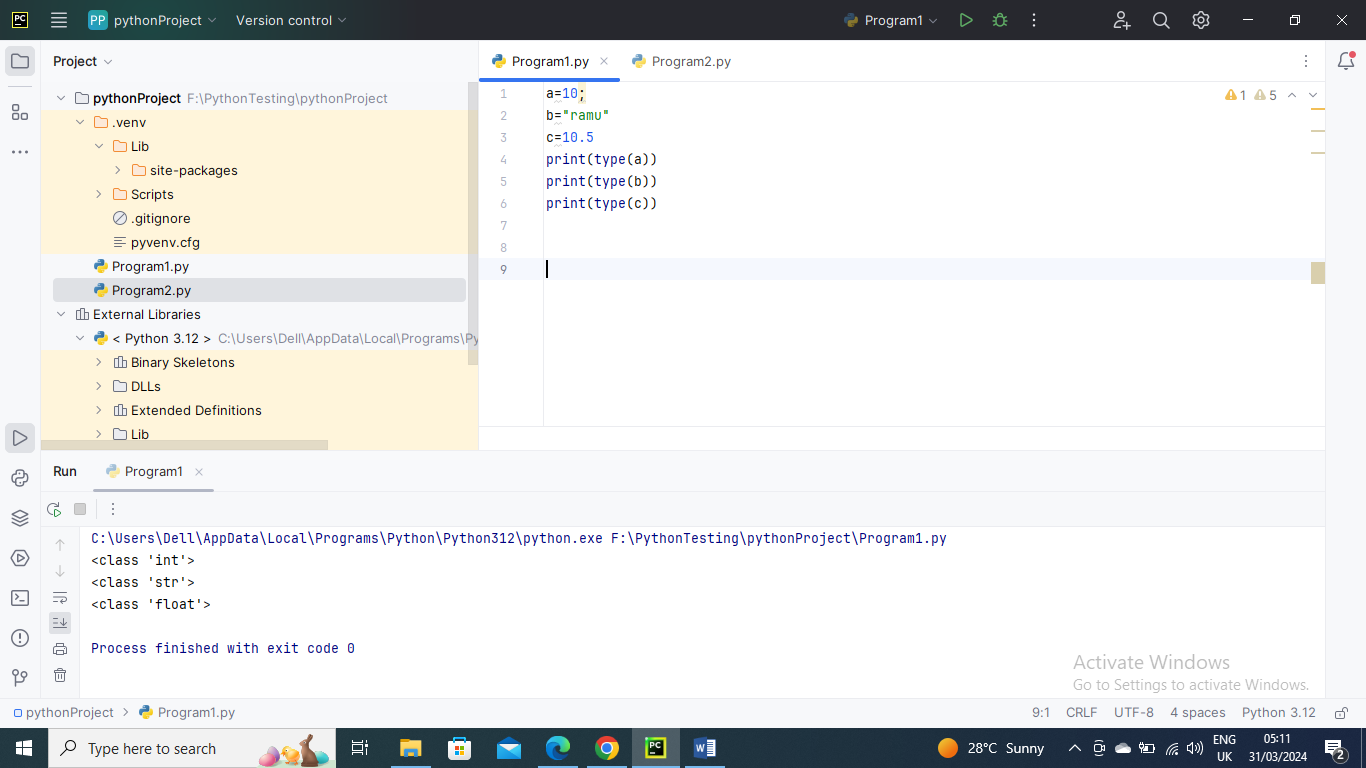
Below is the example where we did not specify the type of the variable a, which has the value five from an integer. The Python interpreter will automatically interpret the variable as an integer.

a=10;  
print(type(a))



Below is another example

a=10;  
b="ramu"  
c=10.5  
print(type(a))  
print(type(b))  
print(type(c))



The following is a list of Python defined data types

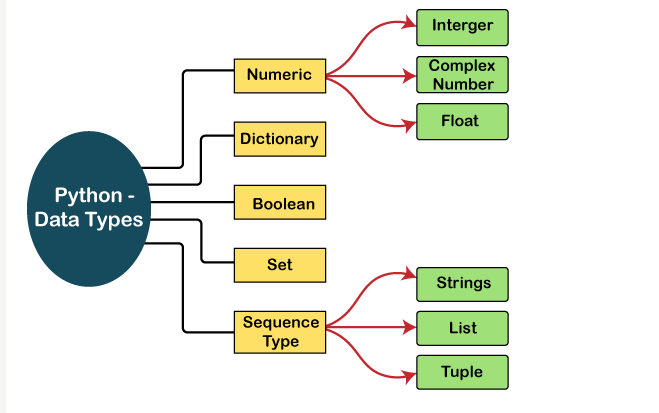
Numbers

Sequence Type

Boolean

Set

Dictionary



Numbers-

Numeric values are stored in numbers. The whole number, float, and complex qualities have a place with a Python Numbers datatype. Python offers the type() function to determine a variable's data type. The instance () capability is utilized to check whether an item has a place with a specific class.

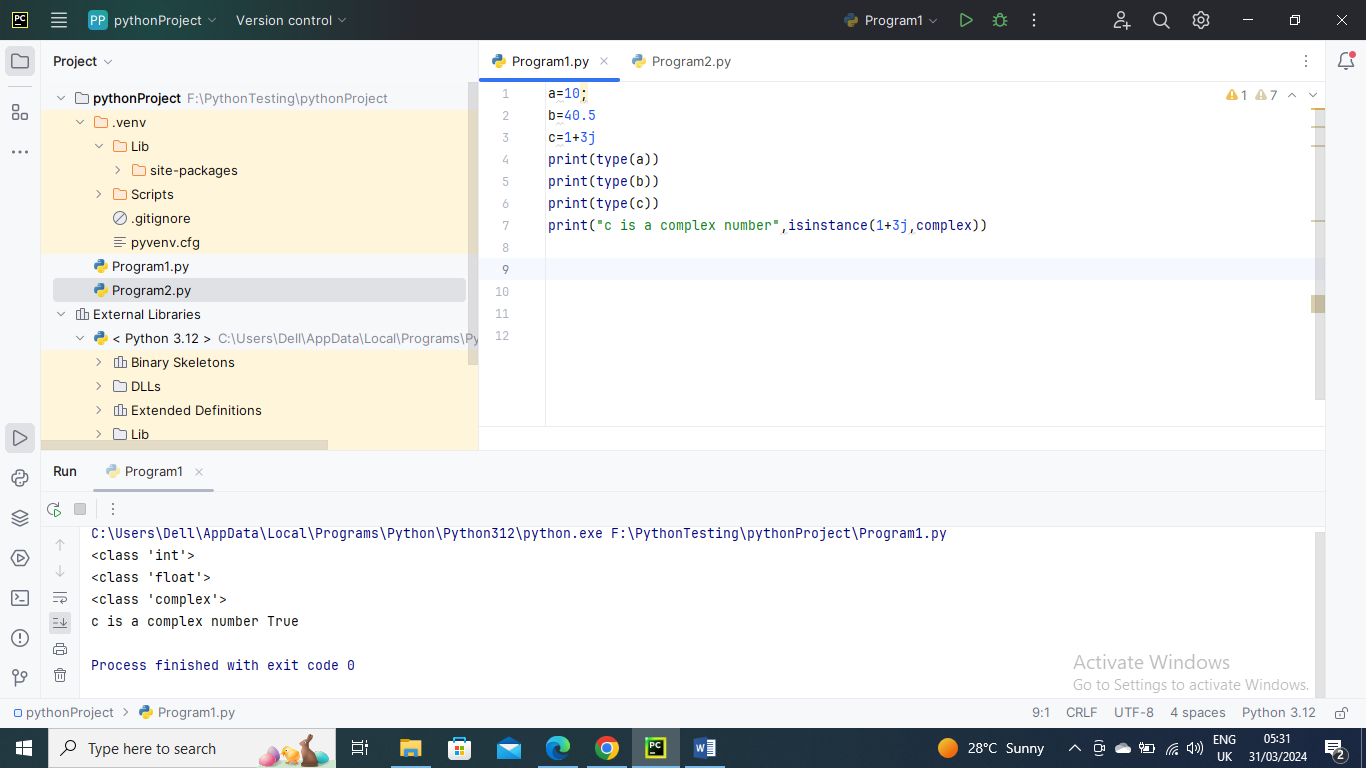
Python supports three kinds of numerical data.

**Int:** Whole number worth can be any length, like numbers 10, 2, 29, - 20, - 150, and so on. An integer can be any length you want in Python. Its worth has a place with int.

**Float:** Float stores drifting point numbers like 1.9, 9.902, 15.2, etc. It can be accurate to within 15 decimal places.

**Complex:** An intricate number contains an arranged pair, i.e., x + iy, where x and y signify the genuine and non-existent parts separately. The complex numbers like 2.14j, 2.0 + 2.3j, etc.

Below is an example for all three data types

a=10;  
b=40.5  
c=1+3j  
print(type(a))  
print(type(b))  
print(type(c))  
print("c is a complex number",isinstance(1+3j,complex))  
  
  


Sequence Type-

Sequence file consist of String , List, Tuple

String – You can see entire String in another Python String document

List – You can see entire List in another Python List document

Tuple – You can see entire Tuples in another Python Tuples document

Set – You can see entire Sets in another Python Sets document

Boolean –

True or false are the two default values for the Boolean Type. False an be represented by “0” or the letter “F” while true can be represented by any value that is not zero

