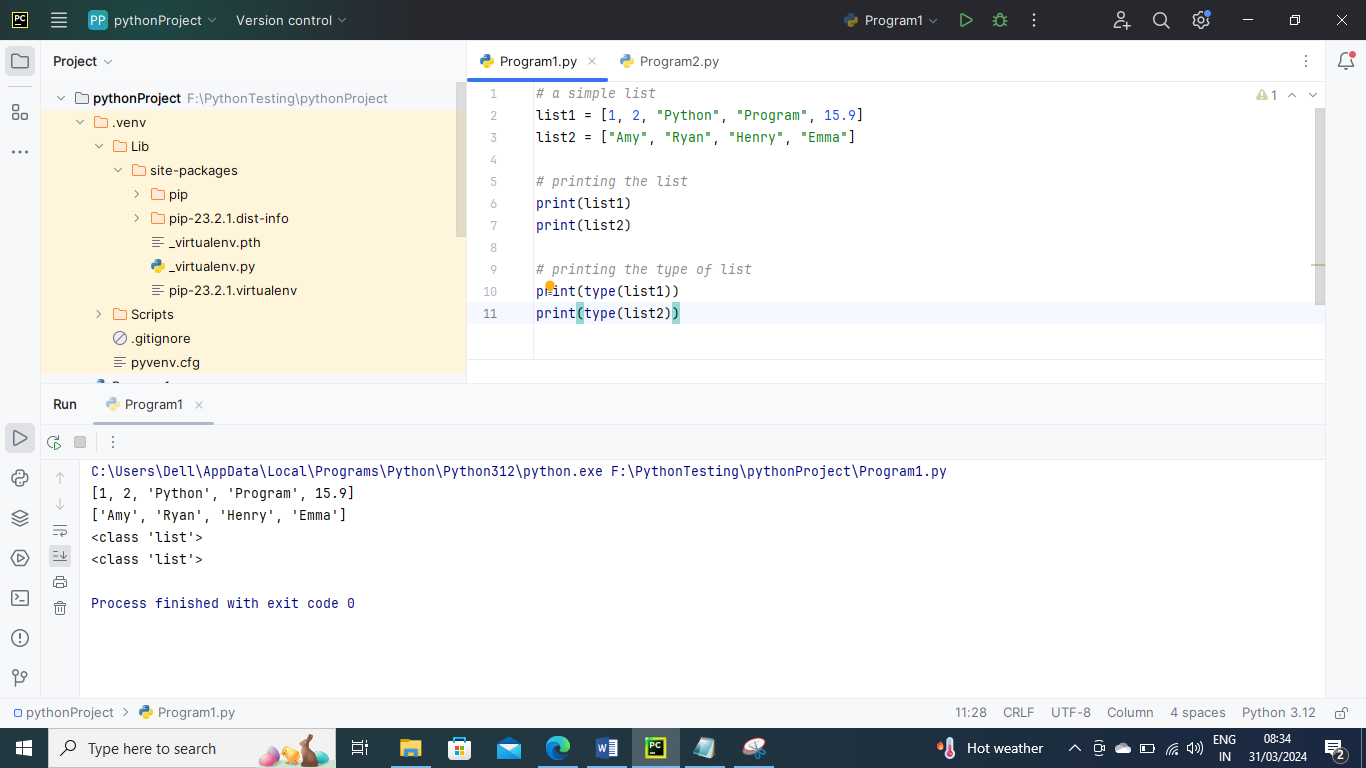
List -

In Python, the sequence of various data types is stored in a list. A list is a collection of different kinds of values or items. Since Python lists are mutable, we can change their elements after forming. The comma (,) and the square brackets [enclose the List's items] serve as separators.

* The lists are in order.
* The list element can be accessed via the index.
* The mutable type of List is
* The rundowns are changeable sorts.
* The number of various elements can be stored in a list.

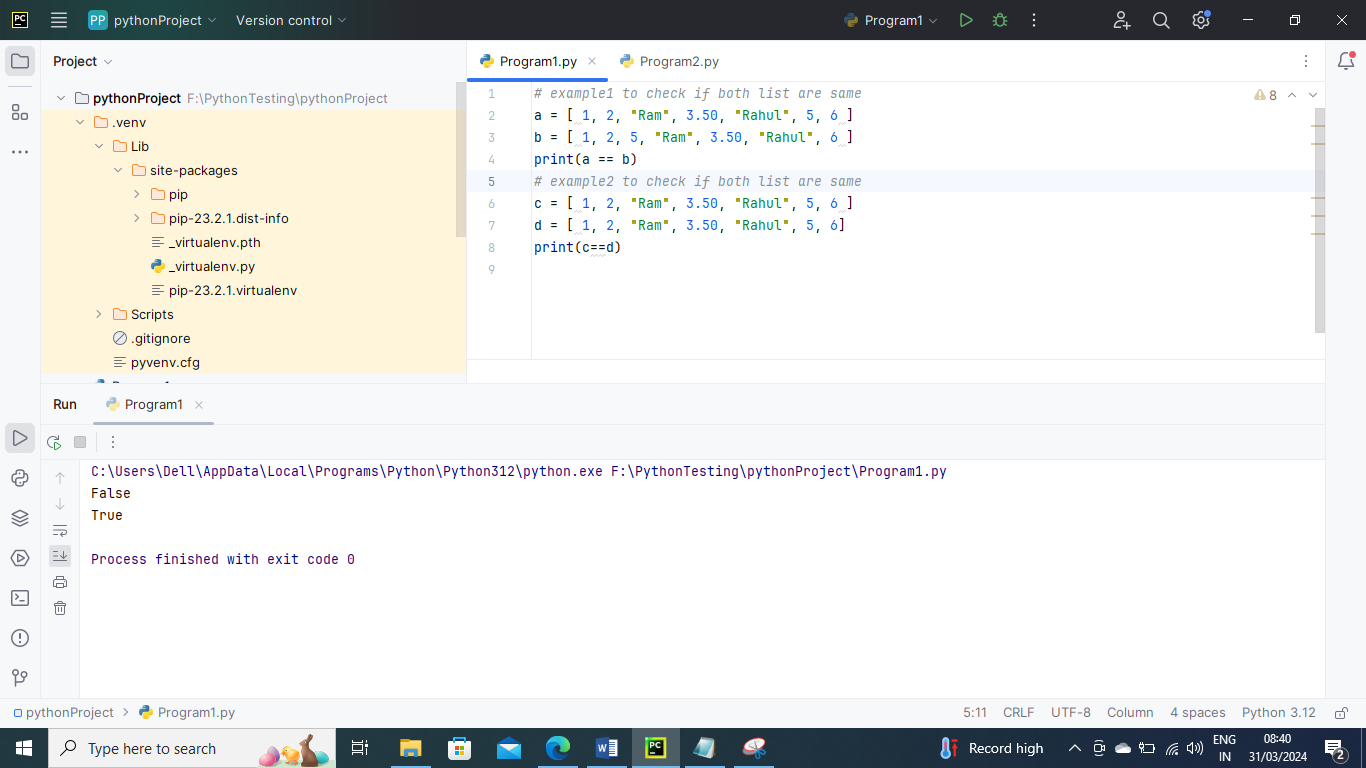
Below is an example

*# a simple list*list1 = [1, 2, "Python", "Program", 15.9]  
list2 = ["Amy", "Ryan", "Henry", "Emma"]  
  
*# printing the list*print(list1)   
print(list2)  
  
*# printing the type of list*print(type(list1))  
print(type(list2))



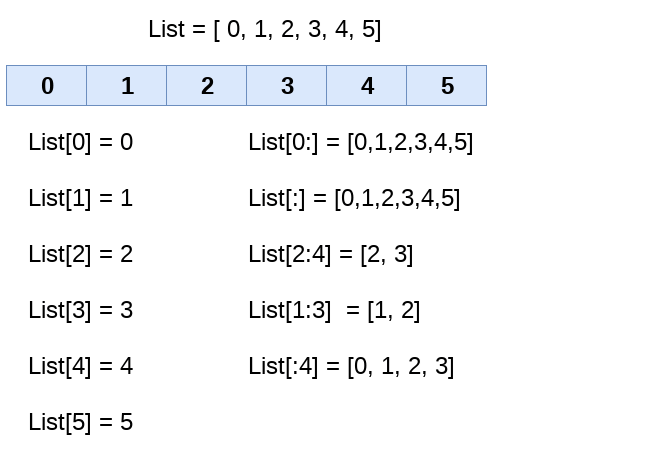
Below is an example to check if two lists are same

*# example1 to check if both list are same*a = [ 1, 2, "Ram", 3.50, "Rahul", 5, 6 ]  
b = [ 1, 2, 5, "Ram", 3.50, "Rahul", 6 ]  
print(a == b)  
*# example2 to check if both list are same*c = [ 1, 2, "Ram", 3.50, "Rahul", 5, 6 ]  
d = [ 1, 2, "Ram", 3.50, "Rahul", 5, 6]  
print(c==d)



List Indexing and Splitting –

The indexing procedure is carried out similarly to string processing. The slice operator [] can be used to get to the List's components.



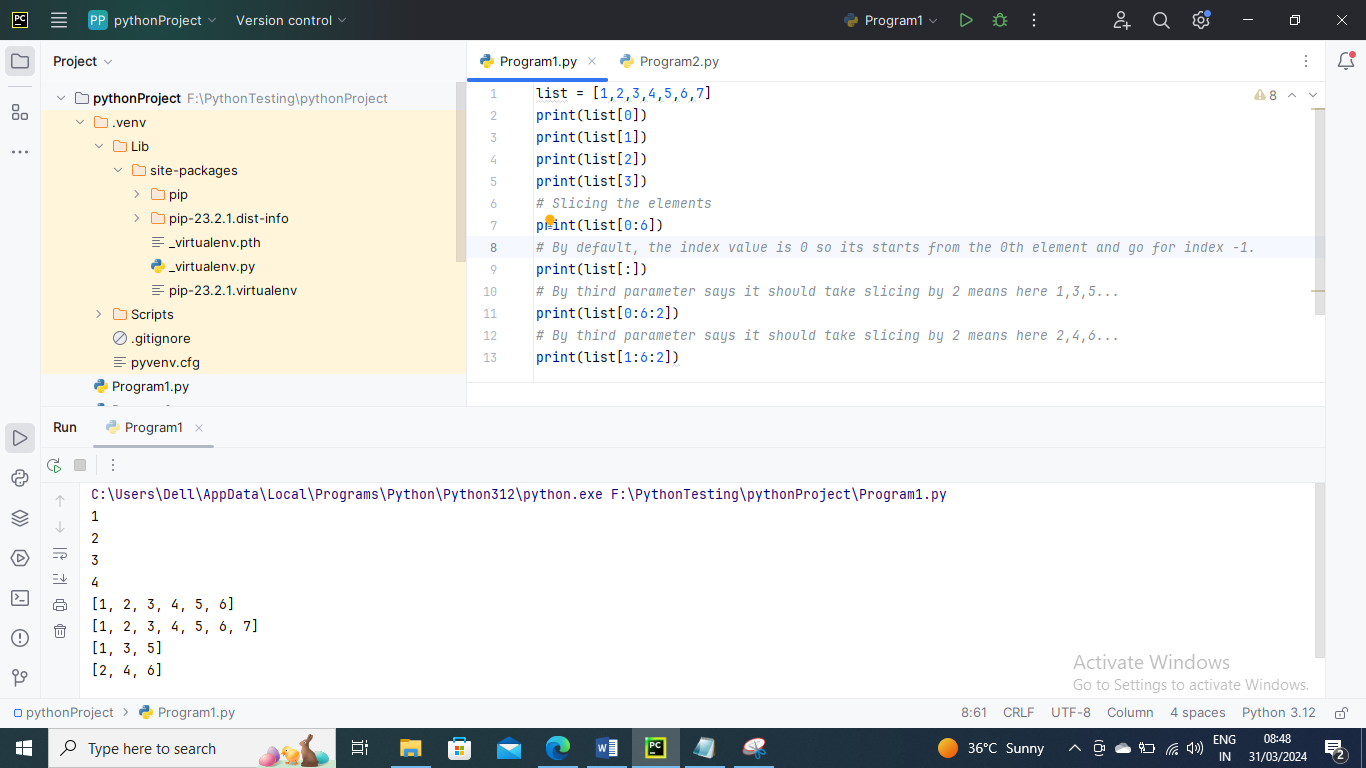
We can get the sub-list of the list using the following syntax.

list\_varible(start:stop:step)

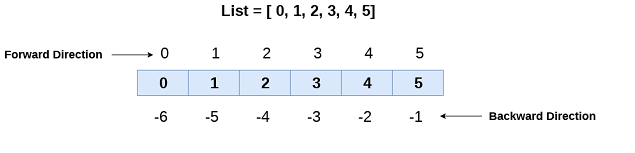
* The beginning indicates the beginning record position of the rundown.
* The stop signifies the last record position of the rundown.
* Within a start, the step is used to skip the nth element: stop.

Below is an example

list = [1,2,3,4,5,6,7]  
print(list[0])  
print(list[1])  
print(list[2])  
print(list[3])  
*# Slicing the elements*print(list[0:6])  
*# By default, the index value is 0 so its starts from the 0th element and go for index -1.*print(list[:])  
*# By third parameter says it should take slicing by 2 means here 1,3,5...*print(list[0:6:2])  
*# By third parameter says it should take slicing by 2 means here 2,4,6...*print(list[1:6:2])

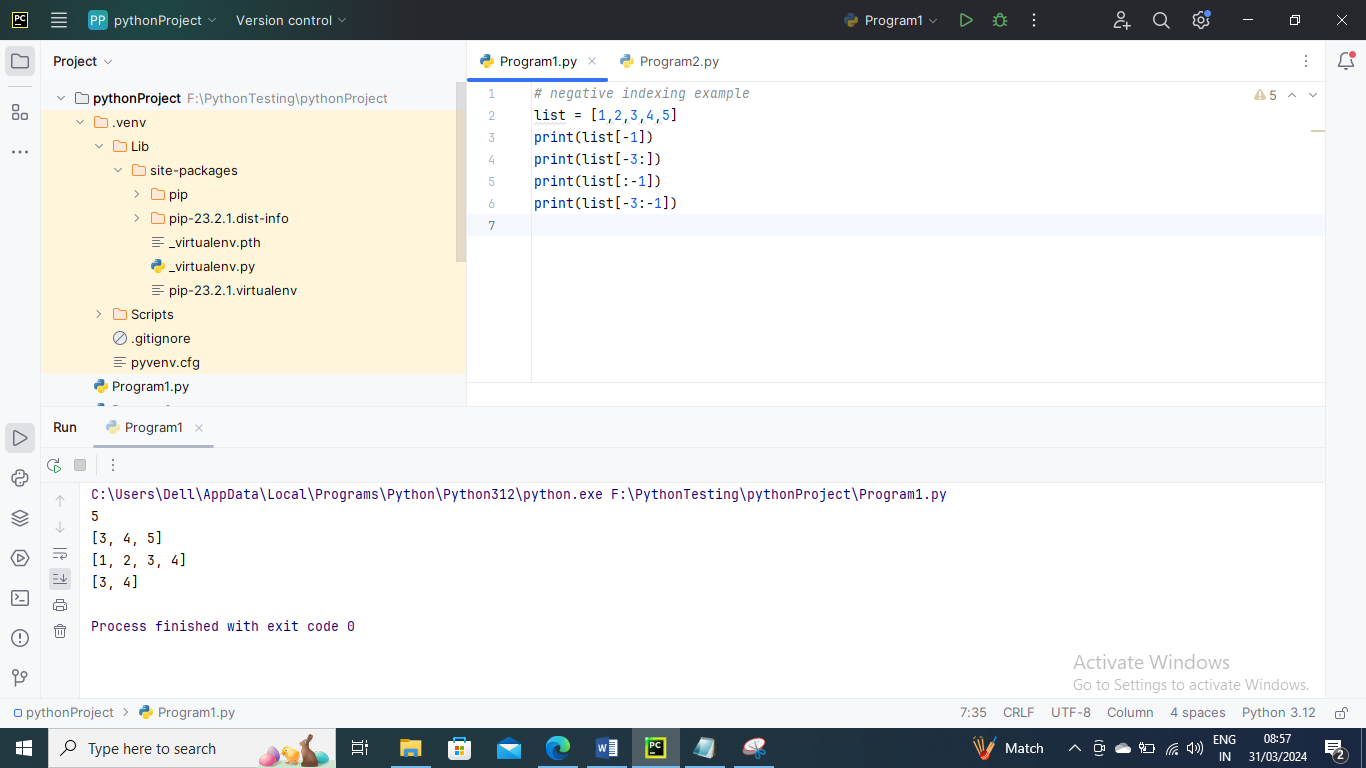


In contrast to other programming languages, Python lets you use negative indexing as well. The negative indices are counted from the right. The index -1 represents the final element on the List's right side, followed by the index -2 for the next member on the left, and so on, until the last element on the left is reached.



Below is an example

*# negative indexing example*list = [1,2,3,4,5]  
print(list[-1])  
print(list[-3:])  
print(list[:-1])  
print(list[-3:-1])

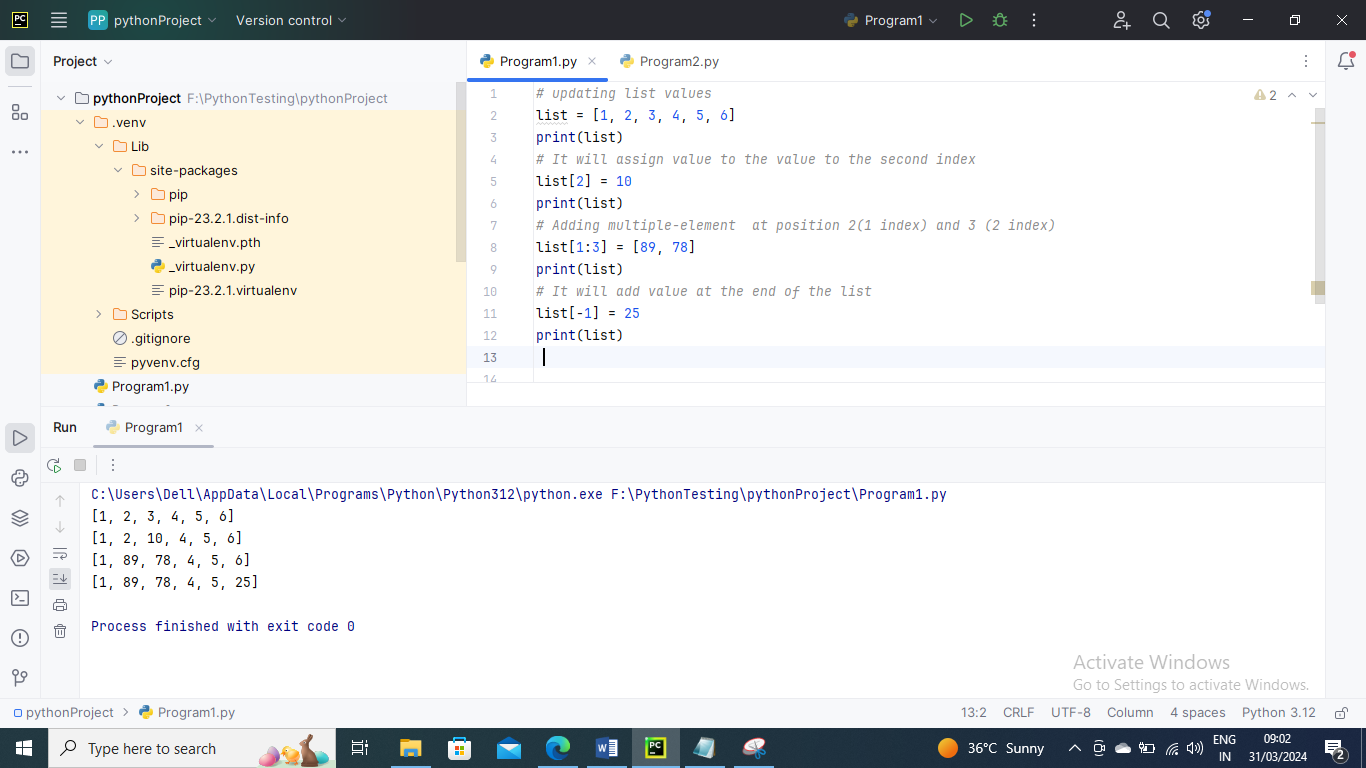


Updating List Values –

Due to their mutability and the slice and assignment operator's ability to update their values, lists are Python's most adaptable data structure. Python's append() and insert() methods can also add values to a list.

Below is an example

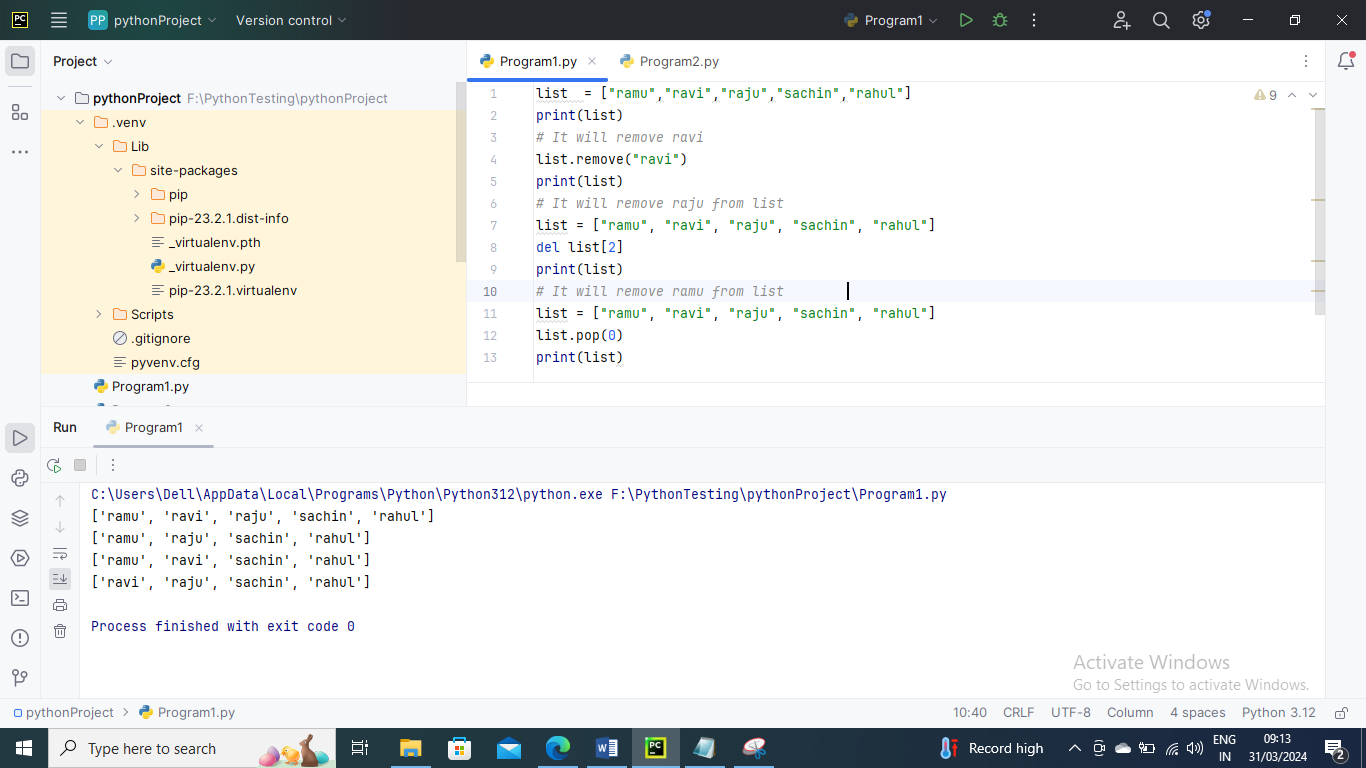
*# updating list values*list = [1, 2, 3, 4, 5, 6]  
print(list)  
*# It will assign value to the value to the second index*list[2] = 10  
print(list)  
*# Adding multiple-element at position 2(1 index) and 3 (2 index)*list[1:3] = [89, 78]  
print(list)  
*# It will add value at the end of the list*list[-1] = 25  
print(list)



The list elements can also be deleted by using the **del** keyword. Python also provides us the **remove()** method if we do not know which element is to be deleted from the list. You can also use **pop()** method to remove based on index

Below is an example to delete list elements

list = ["ramu","ravi","raju","sachin","rahul"]  
print(list)  
*# It will remove ravi*list.remove("ravi")  
print(list)  
*# It will remove raju from list*list = ["ramu", "ravi", "raju", "sachin", "rahul"]  
del list[2]  
print(list)  
*# It will remove ramu from list*list = ["ramu", "ravi", "raju", "sachin", "rahul"]  
list.pop(0)  
print(list)



List Operators –

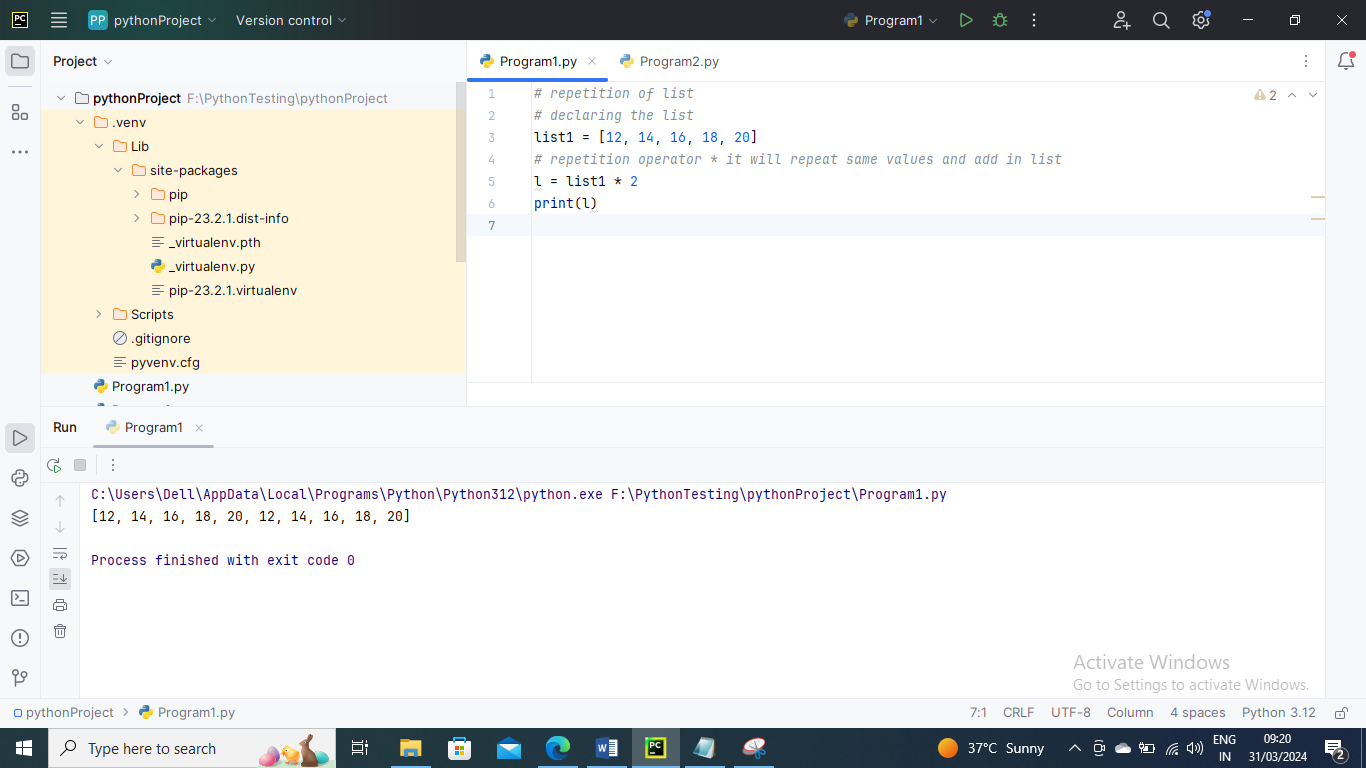
The concatenation (+) and repetition (\*) operators work in the same way as they were working with the strings. The different operations of list are

1. Repetition
2. Concatenation
3. Length
4. Iteration
5. Membership
6. Repetition

The redundancy administrator empowers the rundown components to be rehashed on different occasions.

Below is the example

*# repetition of list  
# declaring the list*list1 = [12, 14, 16, 18, 20]  
*# repetition operator \* it will repeat same values and add in list*l = list1 \* 2  
print(l)

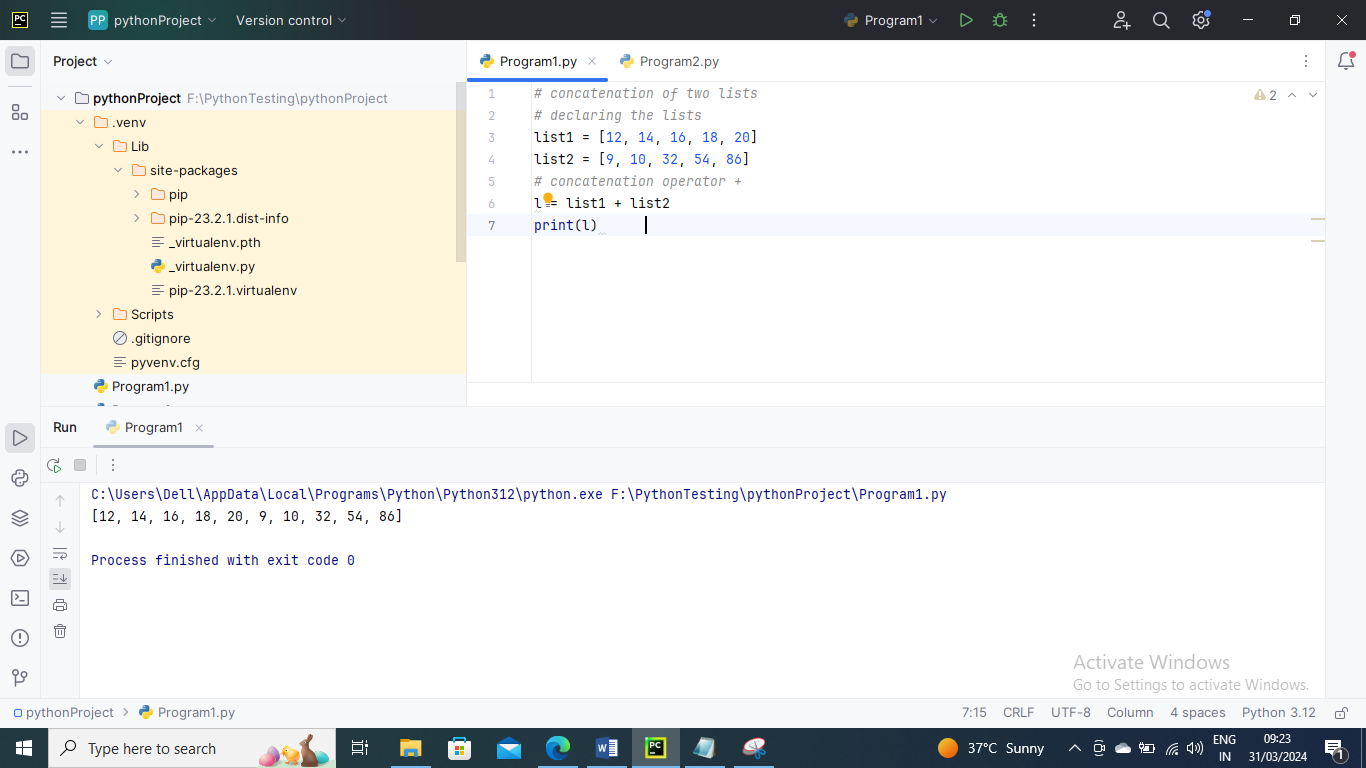


1. Concatenation

It concatenates the list mentioned on either side of the operator.

Below is the example

*# concatenation of two lists  
# declaring the lists*list1 = [12, 14, 16, 18, 20]  
list2 = [9, 10, 32, 54, 86]  
*# concatenation operator +*l = list1 + list2  
print(l)

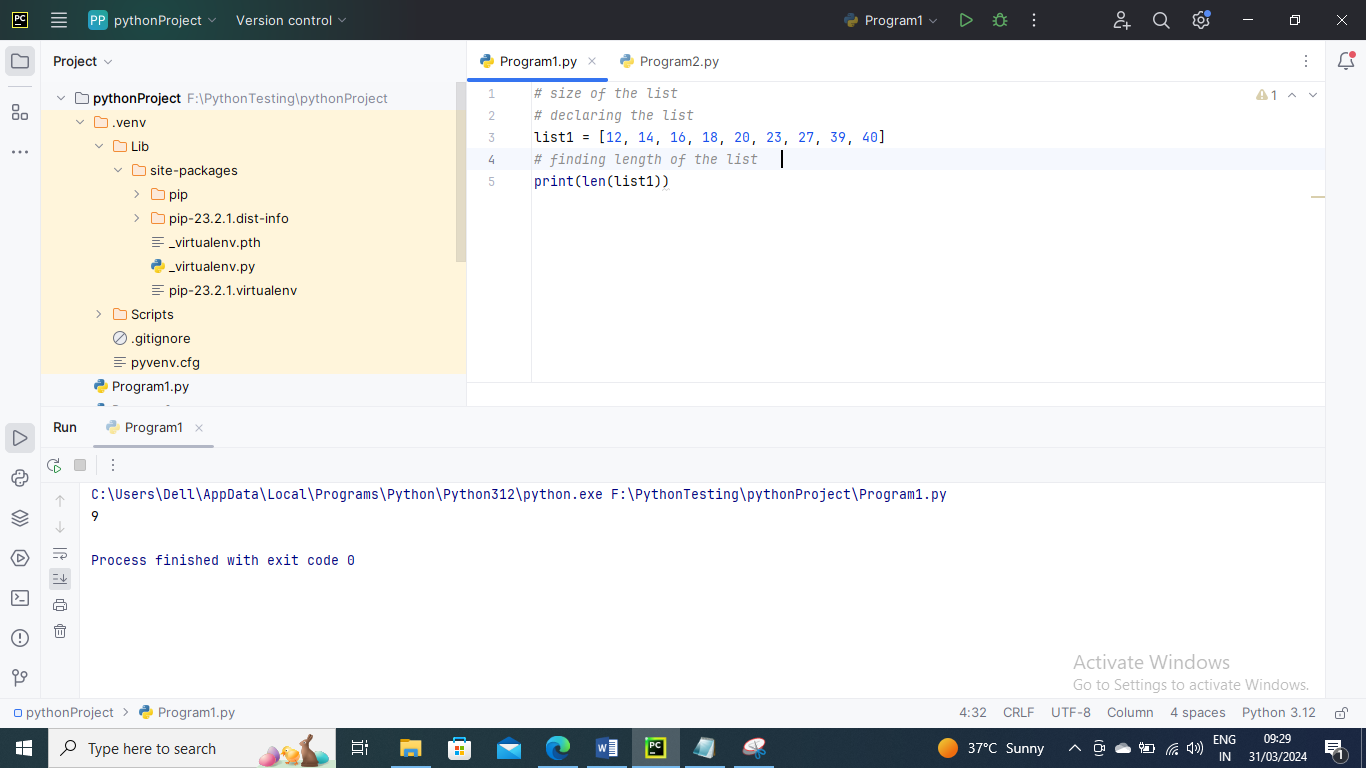


1. Length

It is used to get length of the list

Below is an example

*# size of the list  
# declaring the list*list1 = [12, 14, 16, 18, 20, 23, 27, 39, 40]  
*# finding length of the list*print(len(list1))

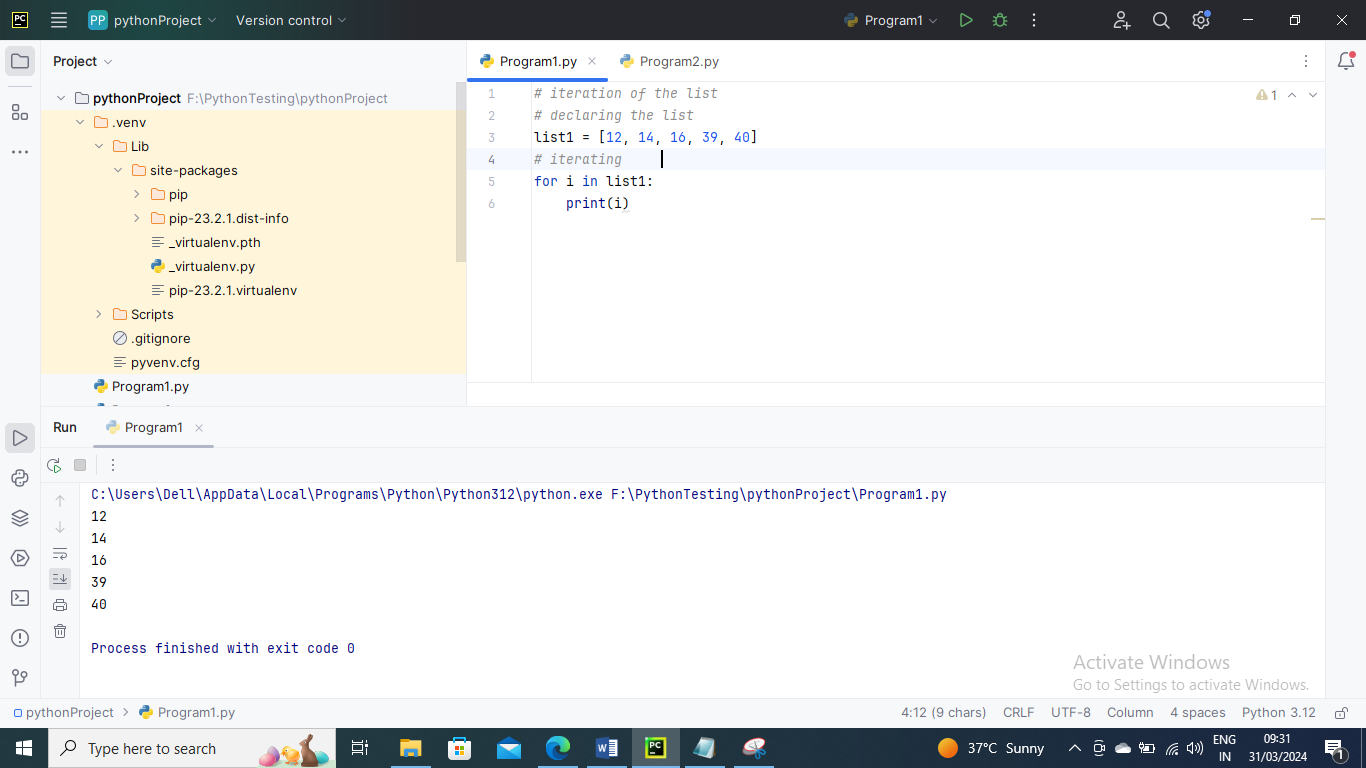


1. Iteration

The for loop is used to iterate over the list elements.

Below is an example

*# iteration of the list  
# declaring the list*list1 = [12, 14, 16, 39, 40]  
*# iterating*for i in list1:   
 print(i)

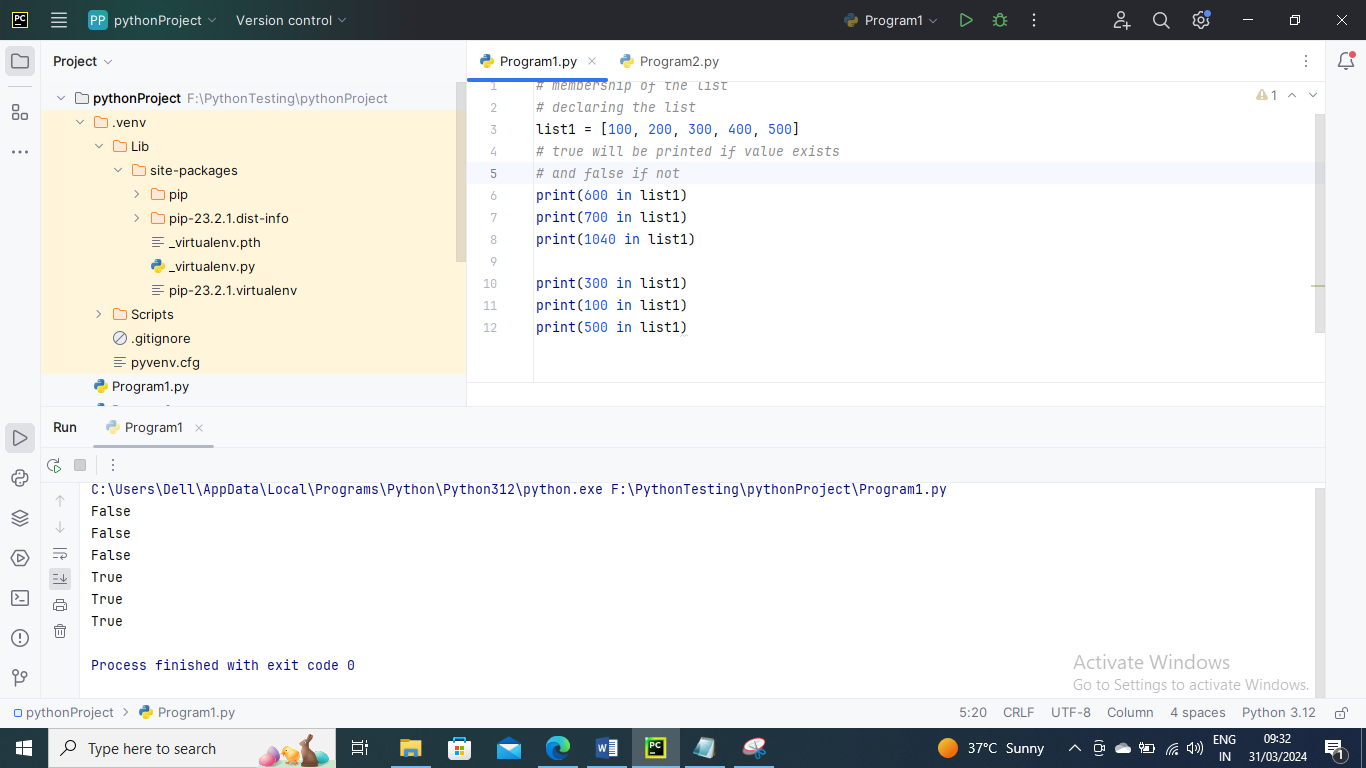


1. Membership

It returns true if a particular item exists in a particular list otherwise false

Below is an example

*# membership of the list  
# declaring the list*list1 = [100, 200, 300, 400, 500]  
*# true will be printed if value exists  
# and false if not*print(600 in list1)  
print(700 in list1)  
print(1040 in list1)  
  
print(300 in list1)  
print(100 in list1)  
print(500 in list1)



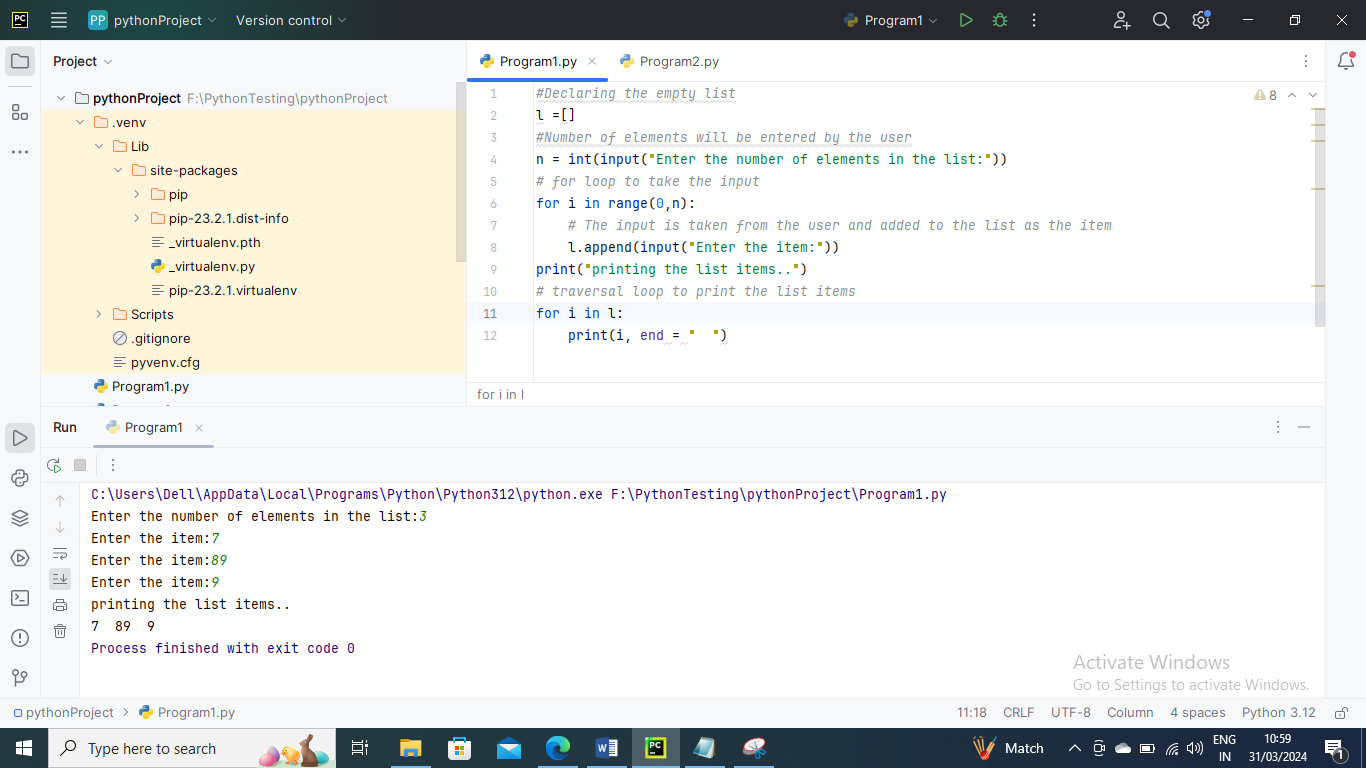
Adding elements to the list

The append() function in Python can add a new item to the List. In any case, the annex() capability can enhance the finish of the rundown

Below is an example

*#Declaring the empty list*l =[]  
*#Number of elements will be entered by the user*n = int(input("Enter the number of elements in the list:"))  
*# for loop to take the input*for i in range(0,n):  
 *# The input is taken from the user and added to the list as the item* l.append(input("Enter the item:"))  
print("printing the list items..")  
*# traversal loop to print the list items*for i in l:  
 print(i, end = " ")

, end = " ")



List Built in Functions

len() – It is used to calculate length of list

max() – It is used to get maximum value in the list

min() – it is used to get minimum value in list

Below is an example

list1 =[13,16,27,21,43,9,33]  
print(list1)  
*#length of list*print(len(list1))  
*#maximum value in list*print(max(list1))  
*#minimum value in list*print(min(list1))

