**List Data Types : -**

**List** :

* List objects are used to store a group of objects.
* The objects which are used to store a group of objects are known as elements of the list.
* List objects are mutable objects but the elements of the list can be mutable or immutable .
* Insertion order is preserved in the list .
* Duplicate elements are allowed in the list.
* It supports both positive and negative indexing.
* List objects can be created by calling list functions or square brackats.

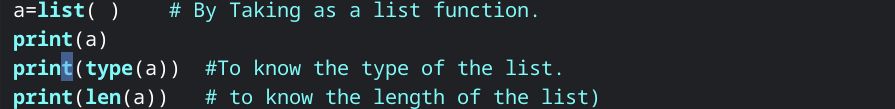
E1:

a=list( ) # By Taking as a list function.

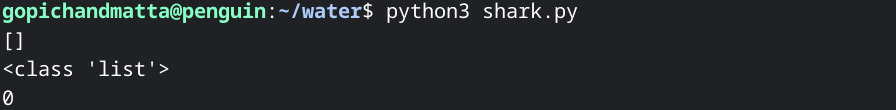
print(a)

Print(type(a)) #To know the type of the list.

print(len(a)) # to know the length of the list.



Output : -



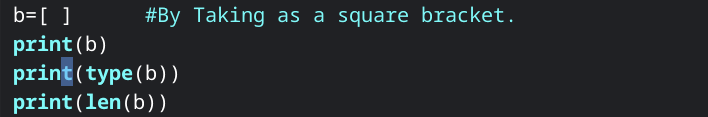
E2:

b=[ ] #By Taking as a square bracket.

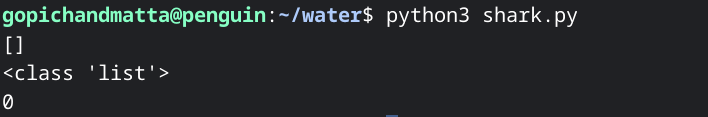
print(b)

Print(type(b))

print(len(b))



Output :



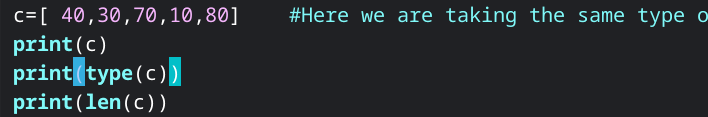
E3:

c=[ 40,30,70,10,80] #Here we are taking the same type of data

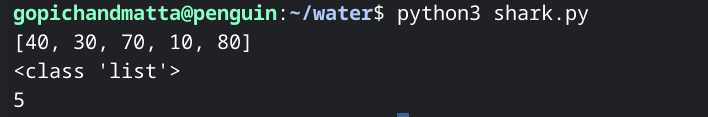
print(c)

Print(type(c))

print(len(c))



output :



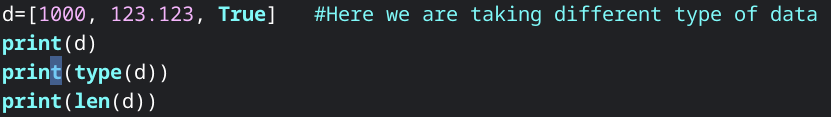
E4:

d=[1000, 123.123, True] #Here we are taking different type of data

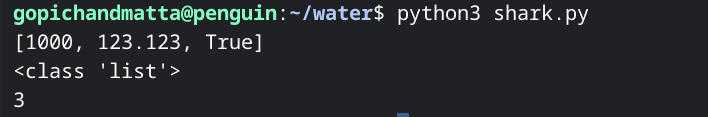
print(d)

Print(type(d))

print(len(d))



output :



**List\_Methods :**

x=[10, 20, 30, 40, 50 ]

print(x)

Print(type(x))

print(len(x))

x.append(80) #Appends only one value at the end of the list .

print(x)

print(len(x))

x.extend([ 5, 15, 25 ]) #extend is used if we have more than one data.

print(x)

print(len(x))

x.clear( ) # To clear all the data inside the variable.

print(x)

x.insert( 3, 20 ) #Initial one is index value and final one is value.

print(x)

x.count(20) #This counts the given value in the count.

print(x)

x.index(40) # it gives the index value of the given number.

print(x)

x.remove(40) #it removes the given value(it works on value).

print(x)

x.pop(2) #it deletes the particular index value(it works in index).

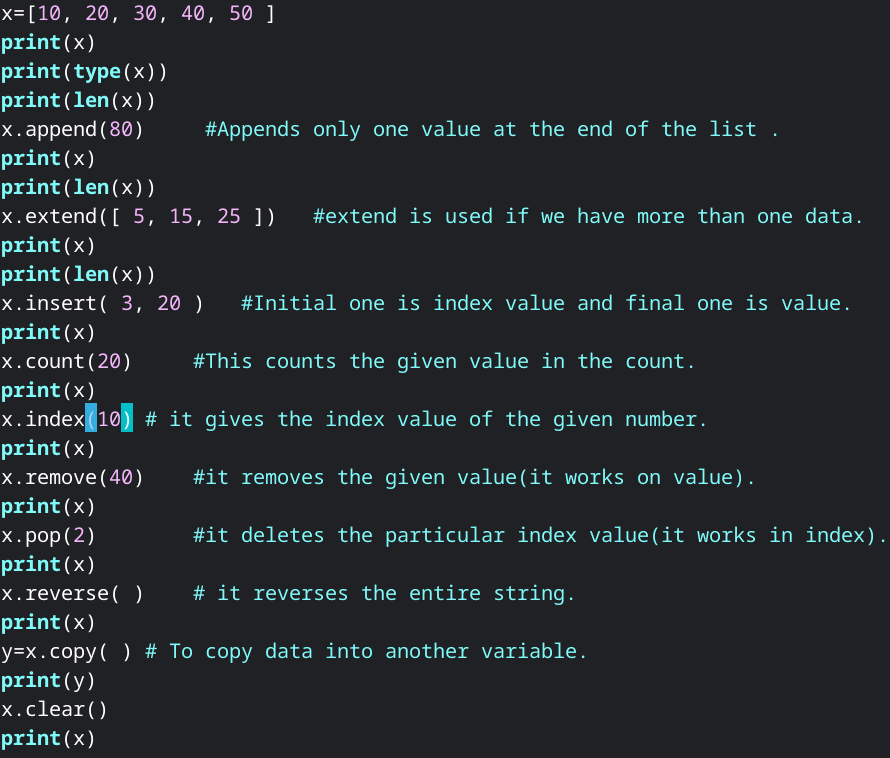
print(x)

x.reverse( ) # it reverses the entire string.

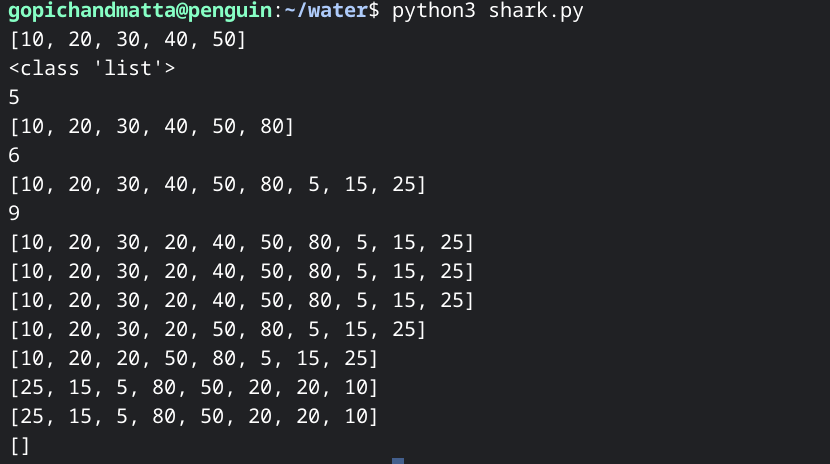
print(x)

y=x.copy( ) # To copy data into another variable.

print(y)



output :



**Tuple :**

**Tuple is used to store multiple data into a single variable. Once the value is initialized to the tuple we can not modify it.Tuple is immutable.Tuple is used as “keys” in the Dictionary. Insertion order is preserved in the tuple. We can create a Tuple by calling the Tuple function or by calling parenthesis.It supports both positive and negative indexing.**

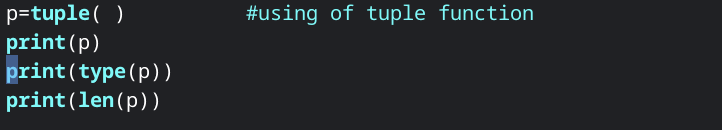
E1:

p=tuple( ) #using of tuple function

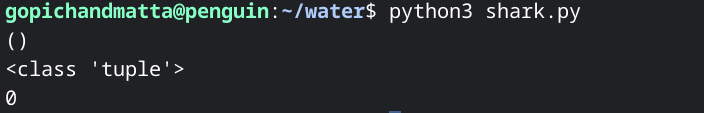
print(p)

Print(type(p))

print(len(p))



output :



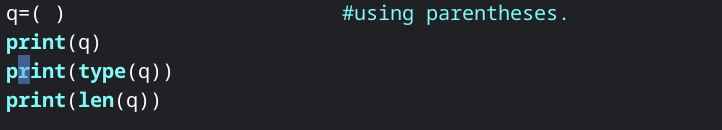
E2:

q=( ) #using parentheses.

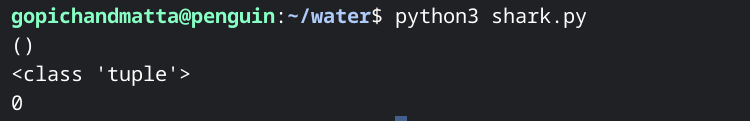
print(q)

Print(type(q))

print(len(q))



output :



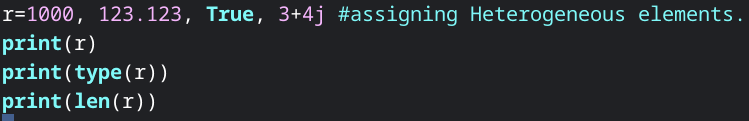
E3:

r=1000, 123.123, True, 3+4j #assigning Heterogeneous elements.

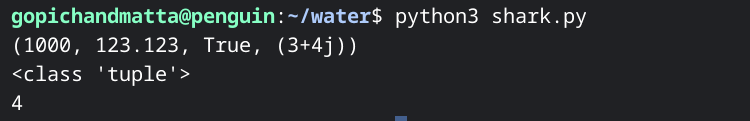
print(r)

print(type(r))

print(len(r))



Output:



E4:

s=(10, 20, 30, 40, 40,50) #by using parentheses.

print(s)

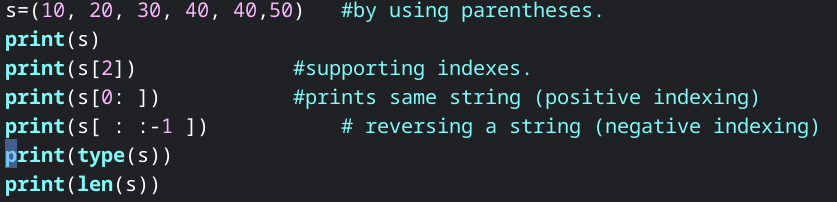
print(s[2]) #supporting indexes.

print(s[0: ]) #prints same string (positive indexing)

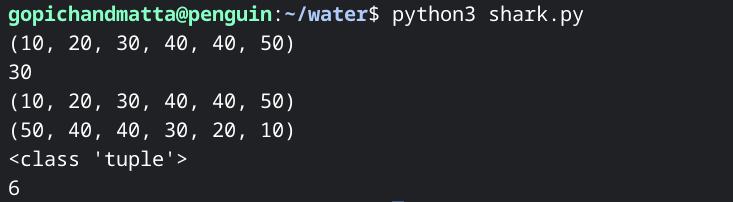
print(s[ : :-1 ]) # reversing a string (negative indexing)

Print(type(s))

print(len(s))



output :



E5:

x=(10, 20, 30, 40, 50)

print(x)

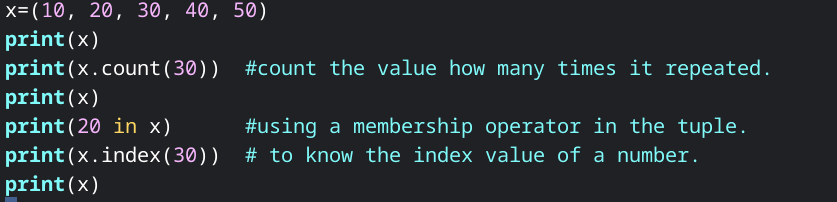
print(x.count(30)) #count the value how many times it repeated.

print(x)

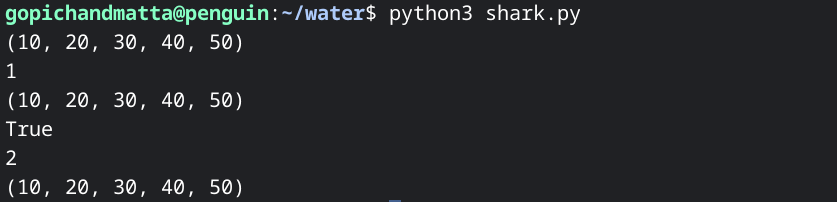
print(20 in x) #using a membership operator in the tuple.

print(x.index(30)) # to know the index value of a number.

print(x)



output:



E6:

y=( 1000, 123.123, True ) # This is packing of elements .

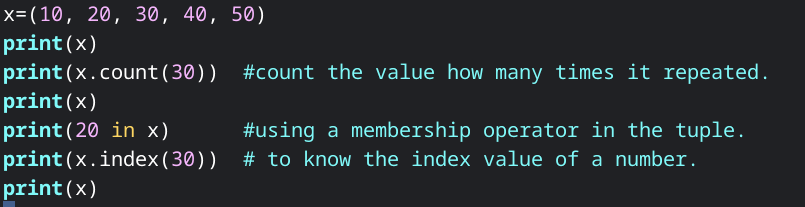
print(y)

i, j, k = y #unpacking of elements.

print(i,type(i))

print(j,type(j))

print(k,type(k))



output :

