1. Write a Python program to sum all the items in a list.

def sum\_list(items):

sum\_numbers = 0

for x in items:

sum\_numbers += x

return sum\_numbers

print(sum\_list([1,2,-8]))

1. Write a Python program to multiplies all the items in a list

def multiply\_list(items):

tot = 1

for x in items:

tot \*= x

return tot

print(multiply\_list([1,2,-8]))

1. Write a Python program to get the largest number from a list.

def max\_num\_in\_list( list ):

max = list[ 0 ]

for a in list:

if a > max:

max = a

return max

print(max\_num\_in\_list([1, 2, -8, 0]))

**4.** Write a Python program to get the smallest number from a list.

def smallest\_num\_in\_list( list ):

min = list[ 0 ]

for a in list:

if a < min:

min = a

return min

print(smallest\_num\_in\_list([1, 2, -8, 0]))

1. Write a Python program to count the number of strings where the string length is 2 or more and the first and last character are same from a given list of strings.    
   Sample List : ['abc', 'xyz', 'aba', '1221']  
   Expected Result : 2

def match\_words(words):

ctr = 0

for word in words:

if len(word) > 1 and word[0] == word[-1]:

ctr += 1

return ctr

print(match\_words(['abc', 'xyz', 'aba', '1221']))

1. Write a Python program to get a list, sorted in increasing order by the last element in each tuple from a given list of non-empty tuples.    
   Sample List : [(2, 5), (1, 2), (4, 4), (2, 3), (2, 1)]  
   Expected Result : [(2, 1), (1, 2), (2, 3), (4, 4), (2, 5)]

## Parameters for the sorted() function

sorted() can take a maximum of three parameters:

* **iterable** - A sequence ([string](https://www.programiz.com/python-programming/string), [tuple](https://www.programiz.com/python-programming/tuple), [list](https://www.programiz.com/python-programming/list)) or collection ([set](https://www.programiz.com/python-programming/set), [dictionary](https://www.programiz.com/python-programming/dictionary), [frozen set](https://www.programiz.com/python-programming/methods/built-in/frozenset)) or any other iterator.
* **reverse (Optional)** - If True, the sorted list is reversed (or sorted in descending order). Defaults to False if not provided.
* **key (Optional)** - A function that serves as a key for the sort comparison. Defaults to None.

**Examples -1**

x = [2, 8, 1, 4, 6, 3, 7]

print ("Sorted List returned :"),

print (sorted(x))

print ("\nReverse sort :"),

print (sorted(x, reverse = True))

# Dictionary

x = {'q':1, 'w':2, 'e':3, 'r':4, 't':5, 'y':6}

print (sorted(x))

# Set

x = {'q', 'w', 'e', 'r', 't', 'y'}

print (sorted(x))

# Frozen Set

x = frozenset(('q', 'w', 'e', 'r', 't', 'y'))

print (sorted(x))

**Examples -2 – sort based on length of string**

L = ["cccc", "b", "dd", "aaa"]

print ("Normal sort :", sorted(L))

print ("Sort with len :", sorted(L, key = len))

**Examples -3 – sort based on user defined function**

def func(x):

    return x % 7

L = [15, 3, 11, 7]

print ("Normal sort:", sorted(L))

print ("Sorted with key:", sorted (L, key = func))

**Answer**

def last(n): return n[-1]

def sort\_list\_last(tuples):

return sorted(tuples, key=last)

print(sort\_list\_last([(2, 5), (1, 2), (4, 4), (2, 3), (2, 1)]))

1. Write a Python program to remove duplicates from a list.

a = [10,20,30,20,10,50,60,40,80,50,40]

dup\_items = set()

uniq\_items = []

for x in a:

if x not in dup\_items:

uniq\_items.append(x)

dup\_items.add(x)

print(dup\_items)

1. Write a Python program to check a list is empty or not.

l = []

if not l:

print("List is empty")

1. Write a Python program to clone or copy a list.

original\_list = [10, 22, 44, 23, 4]

new\_list = list(original\_list)

print(original\_list)

print(new\_list)

**Alternate Answer**

original\_list = [10, 22, 44, 23, 4]

new\_list = original\_list.copy()

print(original\_list)

print(new\_list)

1. Write a Python program to find the list of words that are longer than n from a given list of words.

def long\_words(n, str):

word\_len = []

txt = str.split(" ")

for x in txt:

if len(x) > n:

word\_len.append(x)

return word\_len

print(long\_words(3, "The quick brown fox jumps over the lazy dog"))

1. Write a Python function that takes two lists and returns True if they have at least one common member.

def common\_data(list1, list2):

result = False

for x in list1:

for y in list2:

if x == y:

result = True

return result

print(common\_data([1,2,3,4,5], [5,6,7,8,9]))

print(common\_data([1,2,3,4,5], [6,7,8,9]))

1. Write a Python program to print a specified list after removing the 0th, 4th and 5th elements.    
   Sample List : ['Red', 'Green', 'White', 'Black', 'Pink', 'Yellow']  
   Expected Output : ['Green', 'White', 'Black']

**List comprehensions** are a concise way to create lists. They are used to create a new list by iterating over another list.

Enumerate() method adds a counter to an iterable and returns it in a form of enumerate object. This enumerate object can then be used directly in for loops or be converted into a list of tuples using list() method.

color = ['Red', 'Green', 'White', 'Black', 'Pink', 'Yellow']

color = [x for (i,x) in enumerate(color) if i not in (0,4,5)]

print(color)

# enumerate Example

l1 = ["eat","sleep","repeat"]

s1 = "geek"

# creating enumerate objects

obj1 = enumerate(l1)

obj2 = enumerate(s1)

print (obj1)

print (obj2)

print (list(enumerate(l1)))

print (list(enumerate(s1)))

# changing start index to 2 from 0

print (list(enumerate(s1,2)))