

[Home](https://learning.ccbp.in/)Code Playground

Nakka Vijay Kumar

**NK**

**Cheat Sheet**

Cheat Sheet

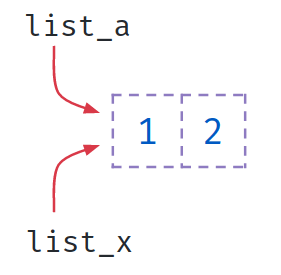
**Passing Mutable Objects**

The same object in the memory is referred by both

list\_a

and

list\_x



**Code**



1

2

3

4

5

6

def add\_item(list\_x):

list\_x += [3]

list\_a = [1,2]

add\_item(list\_a)

print(list\_a)

PYTHON

**Output**



[1, 2, 3]

**Code**



1

2

3

4

5

6

def add\_item(list\_x):

list\_x = list\_x + [3]

list\_a = [1,2]

add\_item(list\_a)

print(list\_a)

PYTHON

**Output**



[1, 2]

Default args are evaluated only once when the function is defined, not each time the function is called.

**Code**



1

2

3

4

5

6

7

def add\_item(list\_x=[]):

list\_x += [3]

print(list\_x)

add\_item()

add\_item([1,2])

add\_item()

PYTHON

**Output**



[3]

[1, 2, 3]

[3, 3]

**Built-in functions**

Built-in functions are readily available for reuse.

We are already using functions which are pre-defined in Python

* print()
* int()
* str()
* len()

Finding Minimum

min()

returns the smallest item in a sequence or smallest of two or more arguments.



1

2

min(sequence)

min(arg1, arg2, arg3 ...)

PYTHON

*Example - 1*

**Code**



1

2

smallest= min(3,5,4)

print(smallest)

PYTHON

Output



3

*Example - 2*

**Code**



smallest = min([1,-2,4,2])

print(smallest)

**Output**



-2

Minimum of Strings

min(str\_1, str\_2)

Strings are compared character by character using unicode values.

* P - 80(unicode)
* J - 74(unicode)

**Code**



1

2

smallest = min("Python", "Java")

print(smallest)

PYTHON

**Output**



Java

Finding Maximum

max()

returns the largest item in a sequence or largest of two or more arguments.



1

2

max(sequence)

max(arg1, arg2, arg3 ...)

PYTHON

*Example - 1*

**Code**



1

2

largest = max(3,5,4)

print(largest)

PYTHON

Ouput



5

*Example - 2*

**Code**



1

2

largest = max([1,-2,4,2])

print(largest)

PYTHON

**Output**



4

Finding Sum

sum(sequence)

returns sum of items in a sequence.

**Code**



1

2

sum\_of\_numbers = sum([1,-2,4,2])

print(sum\_of\_numbers)

PYTHON

**Output**



5

Ordering List Items

sorted(sequence)

returns a new sequence with all the items in the given sequence ordered in increasing order.

**Code**



1

2

3

list\_a = [3, 5, 2, 1, 4, 6]

list\_x = sorted(list\_a)

print(list\_x)

PYTHON

**Output**



[1, 2, 3, 4, 5, 6]

Ordering List Items - Reverse

sorted(sequence, reverse=True)

returns a new sequence with all the items in the given sequence ordered in decreasing order.

**Code**



1

2

3

list\_a = [3, 5, 2, 1, 4, 6]

list\_x = sorted(list\_a, reverse=True)

print(list\_x)

PYTHON

**Output**



[6, 5, 4, 3, 2, 1]

Notes

Discussions

Notes

NEW NOTE

CANCELSAVE

[**Download App**](https://play.google.com/store/apps/details?id=co.ibhubs.chat)

**Personal Manager**



1



[Home](https://learning.ccbp.in/)Code Playground

Nakka Vijay Kumar

**NK**

**Cheat Sheet**

Cheat Sheet

# Function Call Stack & Recursion

### Stack

Stack is a data structure that stores items in an Last-In/First-Out manner.



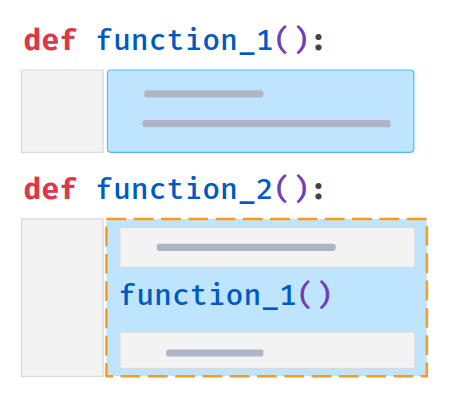
## **Calling a Function**

Calling

function\_1()

inside

function\_2()



#### Code



1

2

3

4

5

6

7

8

def get\_largest\_sqr(list\_x):

len\_list = len(list\_x)

for i in range(len\_list):

x = list\_x[i] list\_x[i] = x \* x largest = max(list\_x)

return largest

list\_a = [1,-3,2]

result = get\_largest\_sqr(list\_a)

print(result)

PYTHON

#### Output



9

In the above code calling functions are

len()

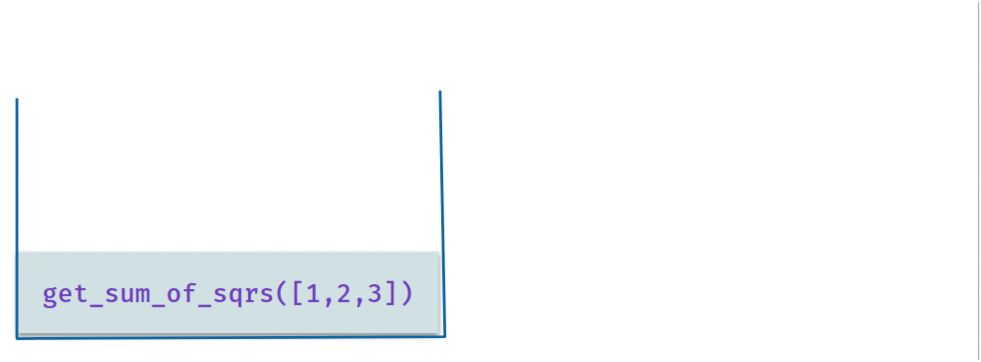
and

max()

inside

get\_largest\_sqr()

### Sum of Squares of List Items



#### Code



1

2

3

4

5

6

7

8

9

10

11

def get\_sqrd\_val(x):

return (x \* x)

def get\_sum\_of\_sqrs(list\_a):

sqrs\_sum = 0

for i in list\_a:

sqrs\_sum += get\_sqrd\_val(i)

return sqrs\_sum

list\_a = [1, 2, 3]

sum\_of\_sqrs = get\_sum\_of\_sqrs(list\_a)

PYTHON

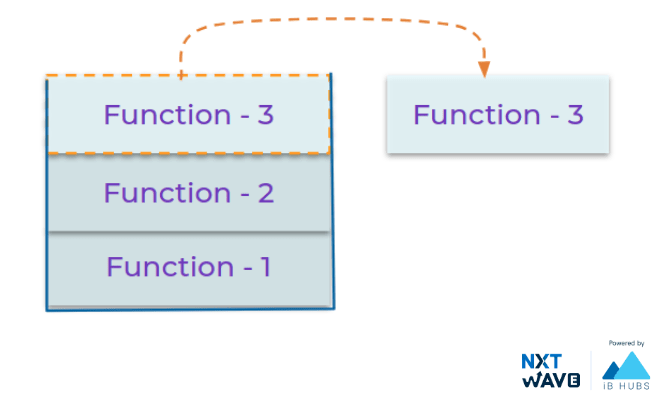
#### Output



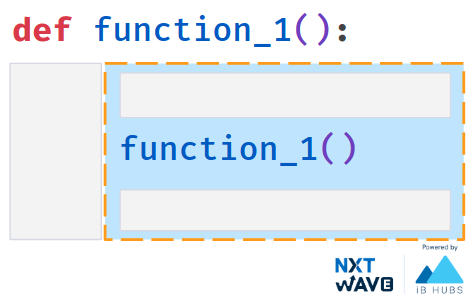
14

# Function Call Stack

Function Call Stack keeps track of function calls in progress



## **Recursion**

A function calling itself is called a **Recursion** 

Let's understand recursion with a simple example of multiplying N numbers

### Multiply N Numbers



1

2

3

4

5

6

7

def factorial(n): # Recursive Function

if n == 1: # Base Case

return 1

return n \* factorial(n - 1) # Recursion

num = int(input())

result = factorial(num)

print(result)

PYTHON

**Base Case**

A recursive function terminates when base condition is met

#### Input



3

#### Output



6

### Without Base case

#### Code



1

2

3

4

5

def factorial(n):

return n \* factorial(n - 1)

num = int(input())

result = factorial(num)

print(result)

PYTHON

#### Input



3

#### Output



RecursionError: maximum recursion depth exceeded

Notes

Discussions

Notes

NEW NOTE

CANCELSAVE

[**Download App**](https://play.google.com/store/apps/details?id=co.ibhubs.chat)

**Personal Manager**



1



[Home](https://learning.ccbp.in/)Code Playground

Nakka Vijay Kumar

**NK**

**Cheat Sheet**

Cheat Sheet

**List Methods**

Python provides list methods that allow us to work with lists.

Let’s learn few among them

* append()
* extend()
* insert()
* pop()
* clear()
* remove()
* sort()
* index()

Append

list.append(value)

Adds an element to the end of the list.

**Code**



1

2

3

4

list\_a = []

for x in range(1,4):

list\_a.append(x)

print(list\_a)

PYTHON

**Output**



[1, 2, 3]

Extend

list\_a.extend(list\_b)

Adds all the elements of a sequence to the end of the list.

**Code**



1

2

3

4

list\_a = [1, 2, 3]

list\_b = [4, 5, 6]

list\_a.extend(list\_b)

print(list\_a)

PYTHON

**Output**



[1, 2, 3, 4, 5, 6]

Insert

list.insert(index,value)

Element is inserted to the list at specified index.

**Code**



1

2

3

list\_a = [1, 2, 3]

list\_a.insert(1,4)

print(list\_a)

PYTHON

**Output**



[1, 4, 2, 3]

Pop

list.pop()

Removes last element.

**Code**



1

2

3

list\_a = [1, 2, 3]

list\_a.pop()

print(list\_a)

PYTHON

**Output**



[1, 2]

Remove

list.remove(value)

Removes the first matching element from the list.

**Code**



1

2

3

list\_a = [1, 3, 2, 3]

list\_a.remove(3)

print(list\_a)

PYTHON

**Output**



[1, 2, 3]

Clear

list.clear()

Removes all the items from the list.

**Code**



1

2

3

list\_a = [1, 2, 3]

list\_a.clear()

print(list\_a)

PYTHON

**Output**



[]

Index

list.index(value)

Returns the index at the first occurrence of the specified value.

**Code**



1

2

3

list\_a = [1, 3, 2, 3]

index =list\_a.index(3)

print(index)

PYTHON

**Output**



1

Count

list.count(value)

Returns the number of elements with the specified value.

**Code**



1

2

3

list\_a = [1, 2, 3]

count = list\_a.count(2)

print(count)

PYTHON

**Output**



1

Sort

list.sort()

Sorts the list.

**Code**



1

2

3

list\_a = [1, 3, 2]

list\_a.sort()

print(list\_a)

PYTHON

**Output**



[1, 2, 3]

Sort & Sorted

sort()

Modifies the list

**Code**



1

2

3

list\_a = [1, 3, 2]

list\_a.sort()

print(list\_a)

PYTHON

**Output**



[1, 2, 3]

sorted()

Creates a new sorted list

**Code**



1

2

3

list\_a = [1, 3, 2]

sorted(list\_a)

print(list\_a)

PYTHON

**Output**



[1, 3, 2]

Notes

Discussions

Notes

NEW NOTE

CANCELSAVE

[**Download App**](https://play.google.com/store/apps/details?id=co.ibhubs.chat)

**Personal Manager**



1