Abdul:

Functions:

* print(“hello”)
* Input(“ “)
* range(1,5)
* list(
* len(
* int(“25”)
* max(l1)

built in function

def print (<input parameters>):

return result (output)

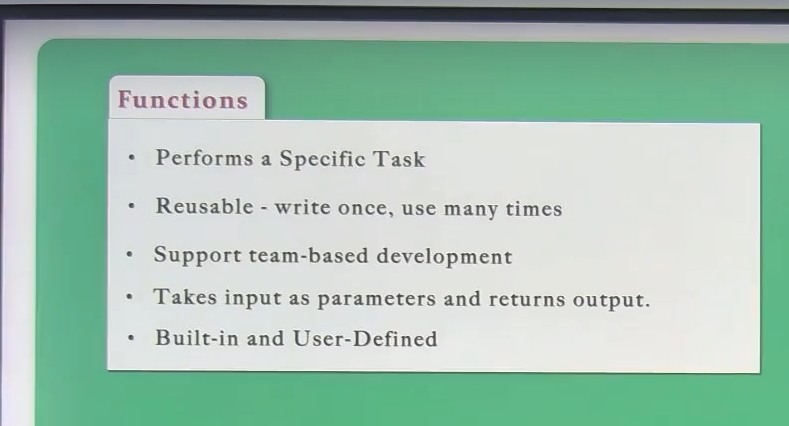
def range(start, stop, step):

return result

* Function is a piece of code which performs specific tasks
* Own functions

def fun1(<parameters>):

return result



How to write a function:

User defined functions:

1. Header
2. Return
3. Calling
4. Parameters

def fun\_name(<parameters>): (input)

statement 1

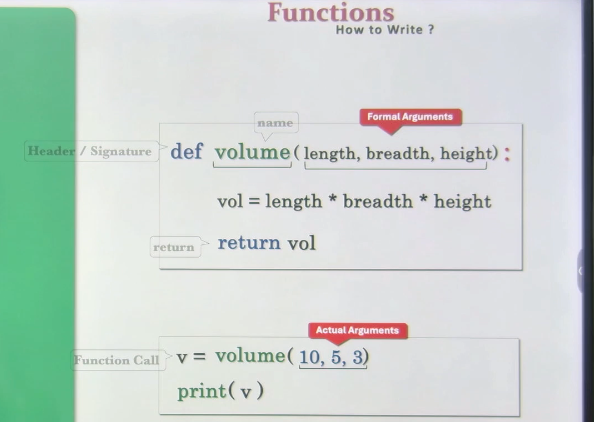
statement 2

return value (output)

def volume(length, breadth, height):

    vol=length\*breadth\*height

    return vol



v=volume(10,5,3)

print(v)

def volume(length, breadth, height):

    vol=length\*breadth\*height

    return vol

if \_\_name\_\_== '\_\_main\_\_':

    v=volume(10,5,3)

    print(v)

    #150

A screenshot of a computer

AI-generated content may be incorrect.

Call by reference 🡪C

Parameter passing🡪 Python

Positional vs Keyword Arguments:

A screen shot of a computer

AI-generated content may be incorrect.

#Positional Parameters

def volume(length,breadth,height):

    print(length,breadth,height)

    vol=length\*breadth\*height

    return vol

v=volume(10,5,3)

print(v)

#Result:

    #10,5,3

    # 150

#Result

    #5,10,3

    #150

Pass in same order

Keyword based:

def volume (length, breadth, height):

    print(length is : {length)

    print(breadth is : {breadth}’)

    print(height is {height}’)

1. v=volume(length=10,breadth=5,height=3)

#print(v)

# length is : 10

#breadth is : 5

#height is 3

# Pass in any order

1. #v=volume(breadth=10,height=3,length=5)

#print(v)

# lenghth is : 5

# breadth is : 10

# height is 3

# Pass in any order

1. v=volume(5,length=10,height=3)

print(v)

    #  TypeError : volume () got multiple values for argument 'length'

    # 5 goes in length

    # Mixed arguments

    # Either positional or keyword, but not both

v=volume(length=10,5,3)

print(v)

# #SyntaxError: positional argument follows keyword argument

#posional left , keyword should be right

e.

v=volume(10,5,height=3)

print(v)

    #length is : 10

    #breadth is : 5

    #height is 3

A screenshot of a computer

AI-generated content may be incorrect.

#def volume(l,b,h)

#a.  v=volume(10,b=5,h=3) --Right

#b   v=volume(10,b=5,3)   -- Keyword should be right hand side

#c   v=volume(l=10,5,k=3)  --> k and keyword should be right

#d   v=volume(10,5,b=3)  ---> Wrong. positional on Right

                              # multiple times of “h”

Default arguments:

List

* Index(element,start,end)

A screenshot of a computer screen

AI-generated content may be incorrect.

Start=0,end=len()🡪 Default args

L1=[10,20,30,20,40,50,60,50,20]

print(L1.index(20))

#1

print(L1.index(20,2))

#3

print(L1.index(20,2,4))

#3

#pop(index)

#pop(len(l)-1)

Default:

Case 1:

def volume(l,b,h):

    v=l\*b\*h

    return v

v=volume(10,5,3)

print(v)

#150

Case 2:

def volume(l,b=1,h=1):

    v=l\*b\*h

    return v

v=volume(10,5)

print(v)

#50

Case 3:

v=volume(10)

print(v)

# #10

Case 4:

v=volume()

#Error

#TypeError: volume() missing 1 required positional argument: 'l'

Case 6:

def volume(l=1,b,h=1):

    v=l\*b\*h

    return v

v=volume(10,5,3)

#SyntaxError: non-default argument follows default argument

Case 7:

def volume(l=1,b=1,h=1):

    v=l\*b\*h

    return v

v=volume()

print(v)

#1

Case 8:

def fun(a,b,c):

    print(a,b,c)

fun(5,10,15)

#5,10,15

Case 8:

def fun(a=1,b=2.5,c="hello"):

    print(a,b,c)

fun(5,10,15)

# 5,10,15

fun()  # 1 2.5 hello

Case 9:

def fun(a=1,b=2.5,c=[1,2,3]):

    print(a,b,c)

fun() #1 2.5 [1, 2, 3]

fun(5,10,15) #5,10,15

fun(5,10,[10,11]) #5 10 [10, 11]

# Function takes any type of argument

# Pass any type of arguments

Case 10:

def func(l=[1,2,3]):

    l.append(len(l)) #[1,2,3,3]

    print(l)

func()

#[1,2,3,3]

func()

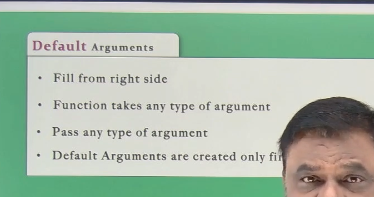
#[1,2,3,3,4]

func([10,11])

#[10,11,2]

func()

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



Positional-Only Arguments:

a,b🡪 Positional

c,d 🡪 Positional and keyword

def fun(a,b,/,c,d):

    print(a,b,c,d)

fun(5,10,c=15,d=20)

fun(a=15,b=10,c=15,d=20)

#TypeError: fun() got some positional-only arguments passed as keyword

fun(5,10,c=15,d=20)

fun(5,10,15,d=20)

fun(5,b=10,c=15,d=20)

##TypeError: fun() got some positional-only arguments passed as keyword

Case 2:

def fun(a,b,c,d,/)--> all are positional only

def fun(a,/,b,c,d)--> only a is poisitional only

def fun(/,a,b,c,d): not allowed

A white paper with black text

AI-generated content may be incorrect.

def fun(a,b,c,d,/):

print(a,b,c,d)

fun(5,b=10,c=15,d=20)

#TypeError

def fun(a,b,c,d,/):

print(a,b,c,d)

fun(5,10,15,20)

5,10,15,20

def fun(/,a,b,c,d):

print(a,b,c,d)

fun(5,10,15,20)

SyntaxError: at least one argument must precede /

Keyword Only:

a,b🡪 Positional/keyword

c,d 🡪 Keyword only

def fun(a,b,\*,c,d):

print(a,b,c,d)

Prog1:

def fun(a,b,\*,c,d):

print(a,b,c,d)

fun(5,10,c=15,d=20)

# 5 10 15 20

A screenshot of a computer

AI-generated content may be incorrect.

def fun(a,b,c,d,\*):

print(a,b,c,d)

fun(5,10,15,20)

SyntaxError: named arguments must follow bare \*

Keyword only

def fun(\*,a,b,c,d):

print(a,b,c,d)

fun(a=5,b=10,c=15,d=20)

# 5 10 15 20

fun(5,10,15,20)🡪 TypeError: fun() takes 0 positional arguments but 4 were given

Positional only and Keyword only:

def fun(a,b,/,c,d,\*,e,f):

print(a,b,c,d,e,f)

A screenshot of a computer

AI-generated content may be incorrect.

A close-up of a computer screen

AI-generated content may be incorrect.

def fun(a,b,c,\*,/,d,e,f)🡪 not allowed

def fun(a,/,b,\*,c):

print(a,b,c)

fun(5,10,15) 🡪 Not allowed

fun(5,10,c=15) 🡪 allowed

fun(5,b=10,c=15) 🡪 Allowed

fun(a=5,b,=10,c=15) 🡪 Not allowed

Variable Length Positional Arguments:

print ()

print(10)

print (10,12,5)

print(10,12.5,3+4j)

Variable length arguments for print function.