

24/9/25

functions:

functions:

- * A function is a block of reusable code that performs a specific task. functions make our programs more organized, readable and reduce repetition.

Types of function.

① build in function:

already available in Python

type('!', Print(), input('), len(c))

2) user defined functions:

User defined functions are created by the user.

using def keyword | def function(parameters).
as parameters

9

3) lambda function:

anonymous (nameless) functions written
in a single line using the lambda keyword.

In user defined there are four functions.

c) function without input and without return value.

def fun-name(); ~~but function works without statements~~

statements

(1) no output job

call the function:

//function name

e.g; def add1()

$x = \text{int}(\text{input}())$ ~~input for x~~

$y = \text{int}(\text{input}())$ ~~input for y~~

$s = x + y$ ~~x+y = 2~~

print(s) ~~2 output~~

add1()

(~~it's a function "f" input~~)

O/p 3 ~~y input~~

8 ~~target f +~~

(ii) function with input and without return

def fun-name(p₁, p₂, ..., p_n):

statements

(a1...an, n) statement job

e.g; def add2(x, y):

$s = x + y$

print(s)

add2(4, 8)

O/p

(~~it's a function f +~~)

(x+y) job

12

(iii) function without input and with return job
def fun-name():
 statements
 return value.

e.g., def add3():

x = int(input())
y = int(input())

$$x + y = ?$$

return ~~s~~

ts = add3()

print(f"the sum is : {ts}")

O/P
4
5
3
y input

answ: sum is 12

(iv) function with input and with return job
def fun-name (p1, p2..pn):
 statements
 return value.

e.g., def add4(x,y):

$$s = x + y$$

return x, y, z

add(4,6)

print(f" {x} and {y} is {z}")

~~def~~

~~def~~