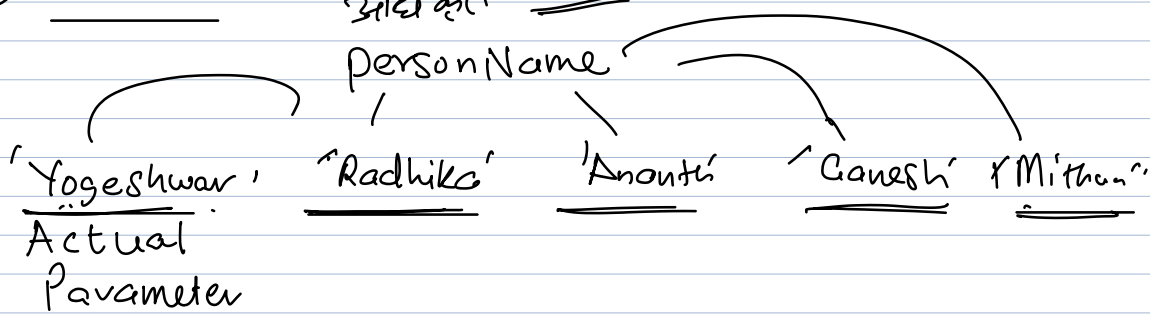
```
def DisplayBirthdayWishes(personName):
    print('Happy birthday,')
    print(personName, '!')

DisplayBirthdayWishes('Yogeshwar')
```

formal parameter

---

(1) Parameter : An entity which influences an outcome.

(2) formal : औपचारिक



Actual Parameter

(3) function define केलेला : formal parameter.

function ला call केलेला : Actual parameter.

(4) At the time of function call, before executing the steps in the function, each formal parameter is bound with corresponding actual parameter.

THIS PROCESS OF BINDING A FORMAL PARAMETER WITH ITS ACTUAL PARAMETER IS CALLED AS PARAMETER PASSING.

```
def DisplayBirthdayWishes(personName):
    print('Happy Birthday,')
    print(personName, '!')

DisplayBirthdayWishes('Yogeshwar').
```

on reading def stmt Python understands that a new function is getting define. Python allocates a memory block in program memory to save the steps of function and names that memory block with function name, in our case it is DisplayBirthdayWishes.

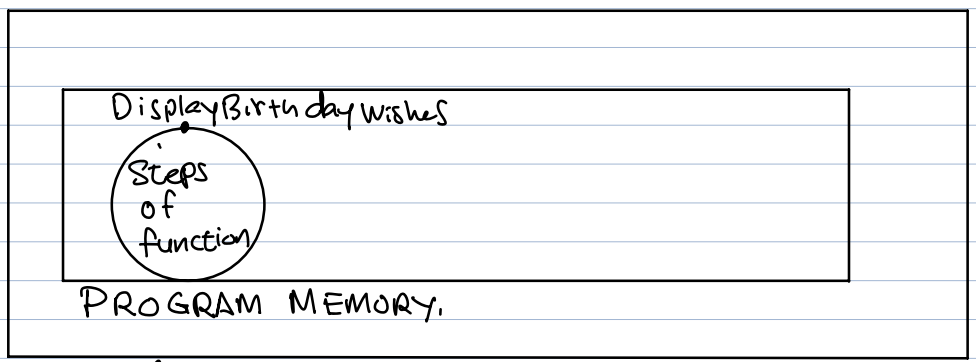
[personName = 'Yogeshwar']

when Python encounters a function call, it first does a parameter passing. In our case, actual parameter which is a string object 'Yogeshwar' will be assigned or passed to a formal parameter personName & then two steps will be executed.

01-def-statement-demo.py

---

run module → python.exe 01-def-statement-demo.py



RAM (System Memory)

DisplayBirthdayWishes('Yogeshwar'). Non-keyword Syntax

DisplayBirthdayWishes(personName = 'Yogeshwar'). keyword Syntax

```
def testFunction(a, b, c):
    print(a, b, c)

testFunction(10, 20, 30)

testFunction(a=10, b=20, c=30)
```

a = 10
b = 20
c = 30

testFunction(b=20, c=30, a=10)

---

testFunction(10, 20, c=30)

testFunction(a=10, 20, 30) ✗
