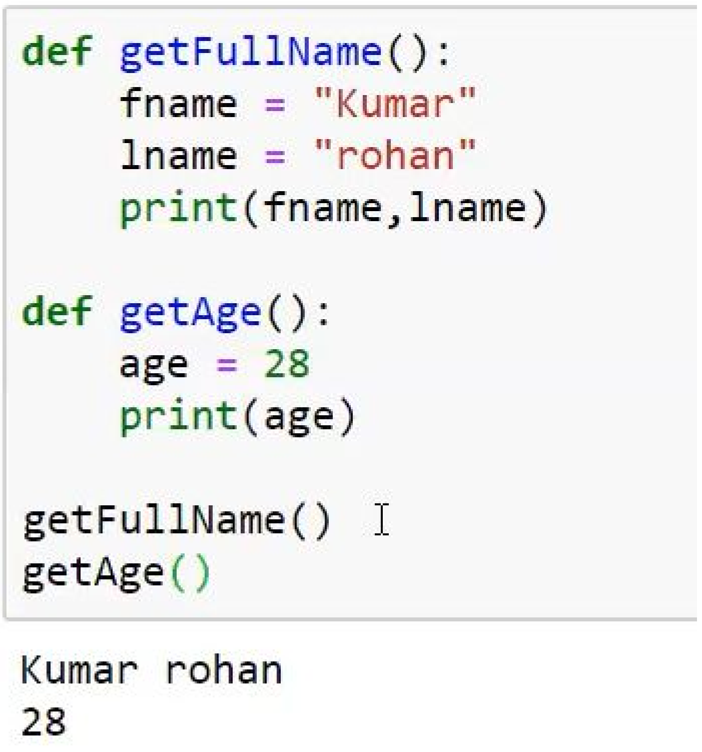
Date : 1st - 09 - 2020
Morning Session : 9am – 11.00 PM
By ~ Rohan Kumar

Topics: Python Functions

Functions:A function is a set of statements that take inputs, do some specific computation and produce output. The idea is to put some commonly or repeatedly done tasks together and make a function, so that instead of writing the same code again and again for different inputs, we can call the function.

Two steps for function:

1) Define the function.
2) Call the function.



```python
def getFullName():
    fname = "Kumar"
    lname = "rohan"
    print(fname,lname)

def getAge():
    age = 28
    print(age)

getFullName()
getAge()
```

```
Kumar rohan
28
```

What order function gets called?

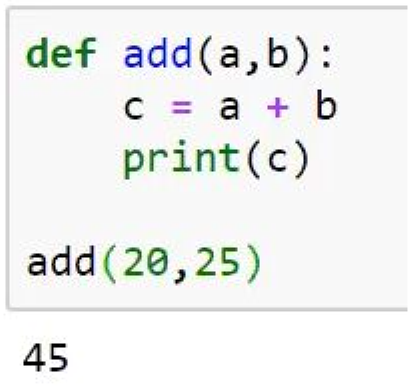
Top to down.

Where ever we find () it is function

Python provides **built-in functions** like print(), etc.

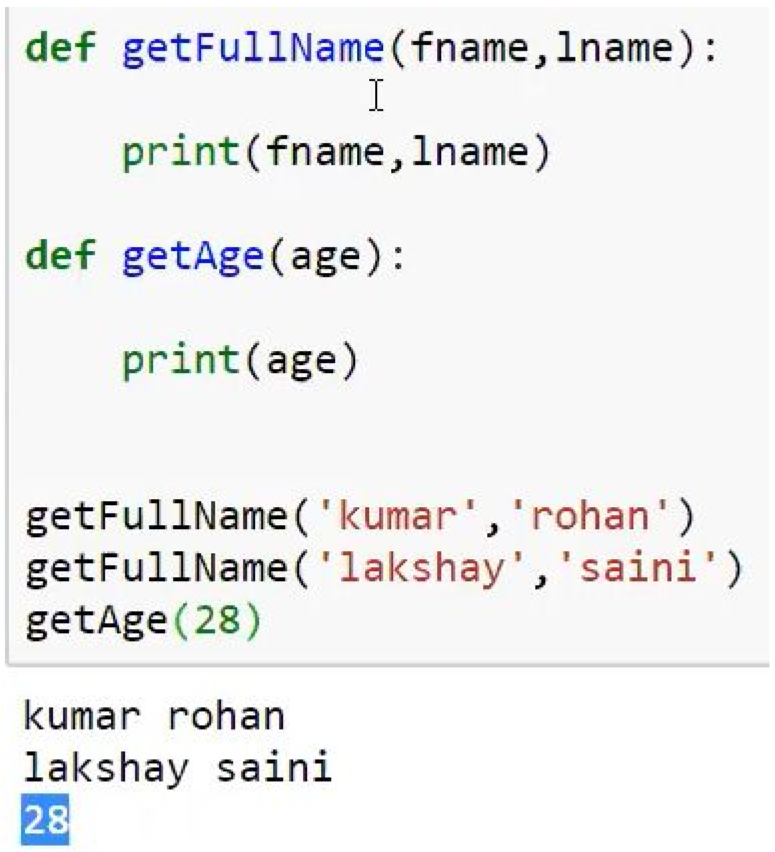
but we can also create your own functions. These functions are called **user-defined functions.**

Parameters/Argument: Information can be passed into functions as arguments. Arguments are specified after the function name, inside the parentheses. You can add as many arguments as you want, just separate them with a comma.



```
def add(a,b):
    c = a + b
    print(c)

add(20,25)
```

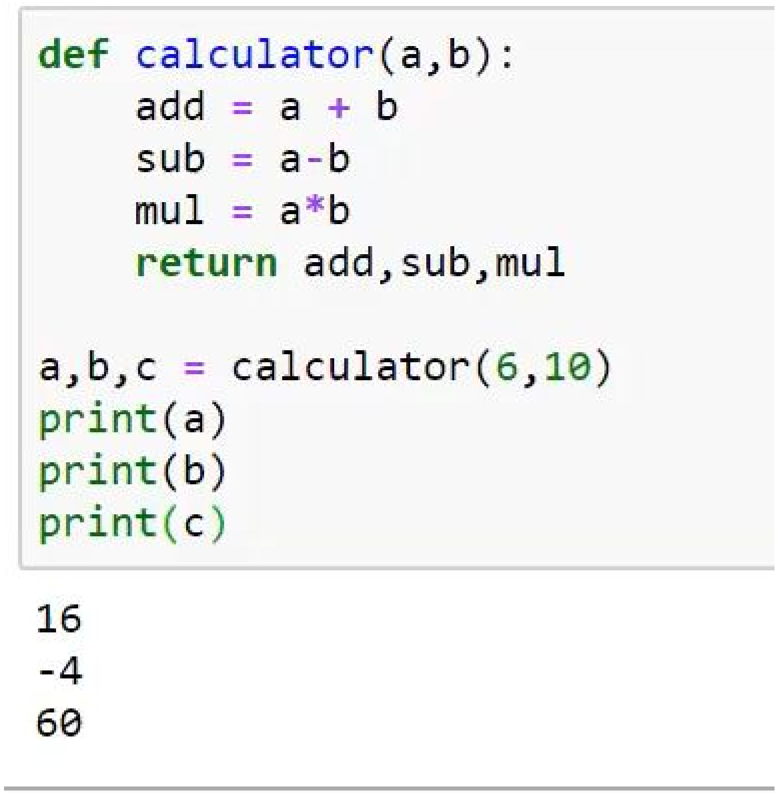
```
45
```



```
def getFullName(fname,lname):

    print(fname,lname)

def getAge(age):

    print(age)


getFullName('kumar','rohan')
getFullName('lakshay','saini')
getAge(28)
```

```
kumar rohan
lakshay saini
28
```

When a function does not return anything it means it returns none.



```
def calculator(a,b):
    add = a + b
    sub = a-b
    mul = a*b
    return add,sub,mul

a,b,c = calculator(6,10)
print(a)
print(b)
print(c)
```

```
16
-4
60
```

We can return multiple values

Pass by Reference or pass by value:

Pass the by value: passing the value to the function

Pass by Reference: passing memory address to the function.

in Python every variable name is a reference. When we pass a variable to a function, a new reference to the object is created. When we pass a reference and change the received reference to something else, the connection between passed and received parameters is broken.

Parameters 2 types

1) Actual parameters
2) Formal parameters

When we are defining a function the parameters we use are called formal parameters.

When we are calling a function in the function pass the value the parameter we use is called the actual parameter.

There are 4 types of actual parameters.

1) Position
2) Keyword
3) Default
4) Variable length

Default arguments:

A default argument is a parameter that assumes a default value if a value is not provided in the function call for that argument.

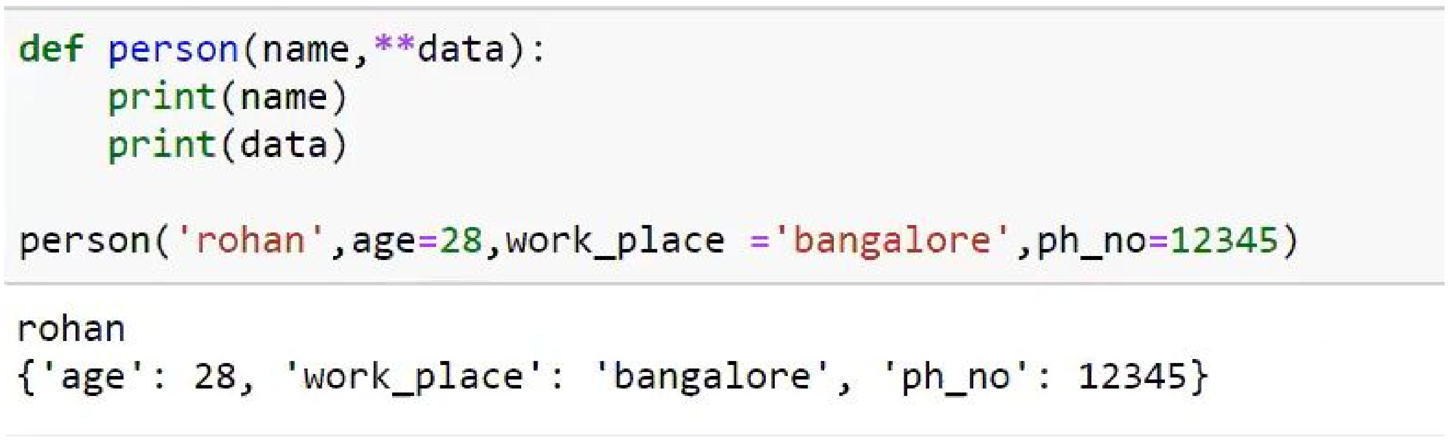
Keyword arguments:

The idea is to allow the caller to specify argument names with values so that the caller does not need to remember the order of parameters.

Variable length arguments:

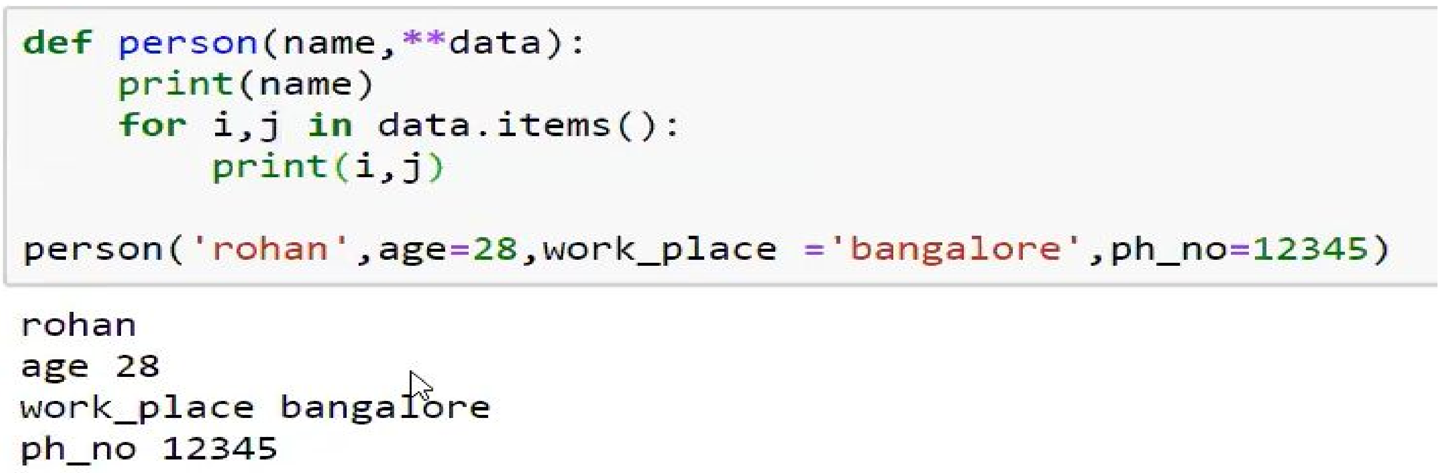
We can have both a normal and keyword variable number of arguments.

**kwargs : keyworded variable length argument () => Variable Lenth



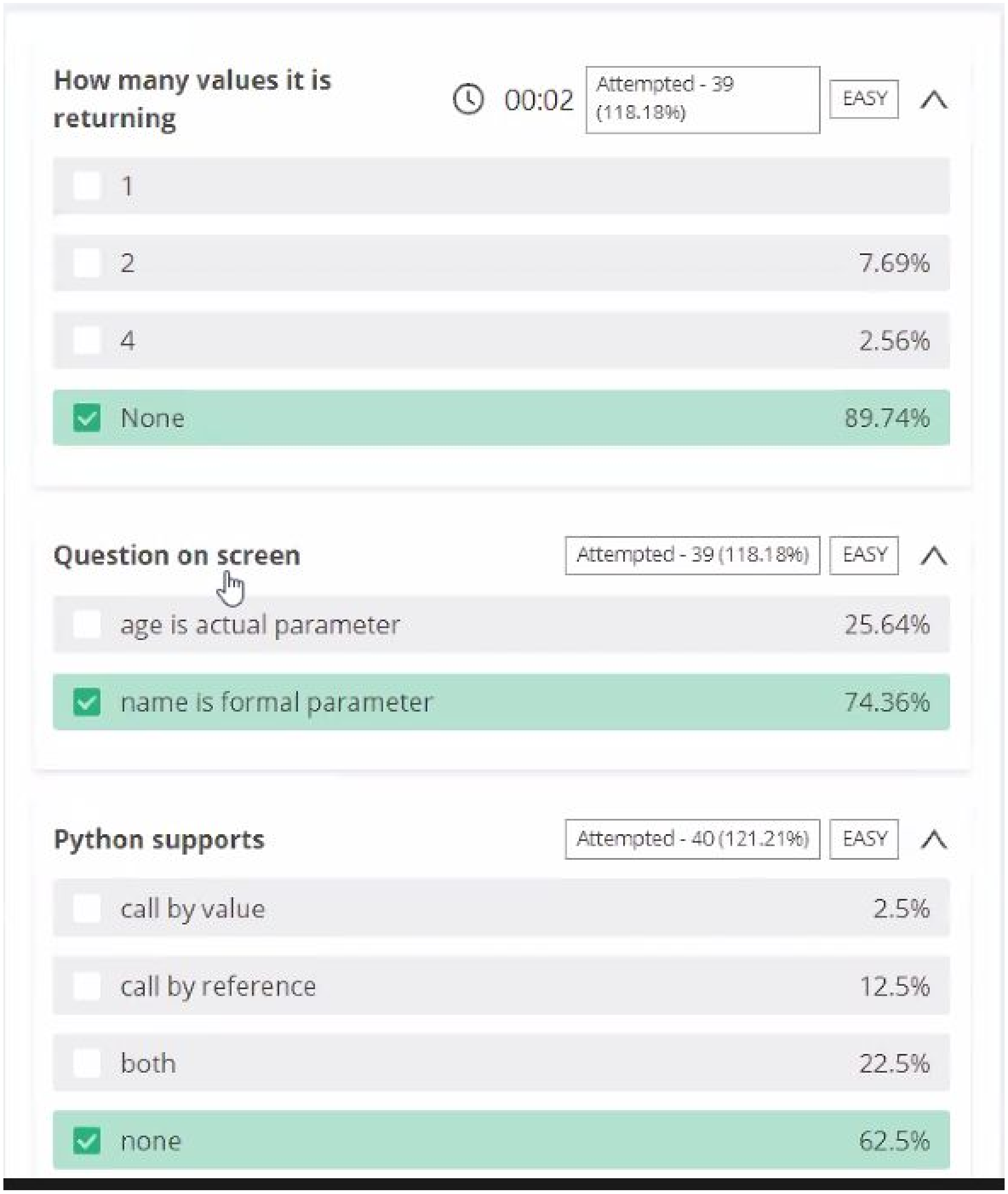
```
def person(name,**data):
    print(name)
    print(data)

person('rohan',age=28,work_place ='bangalore',ph_no=12345)
```

```
rohan
{'age': 28, 'work_place': 'bangalore', 'ph_no': 12345}
```



```
def person(name,**data):
    print(name)
    for i,j in data.items():
        print(i,j)

person('rohan',age=28,work_place ='bangalore',ph_no=12345)
```

```
rohan
age 28
work_place bangalore
ph_no 12345
```



How many values it is returning

00:02 Attempted - 39 (118.18%) EASY

1	
2	7.69%
4	2.56%
[✓ None]	89.74%

Question on screen

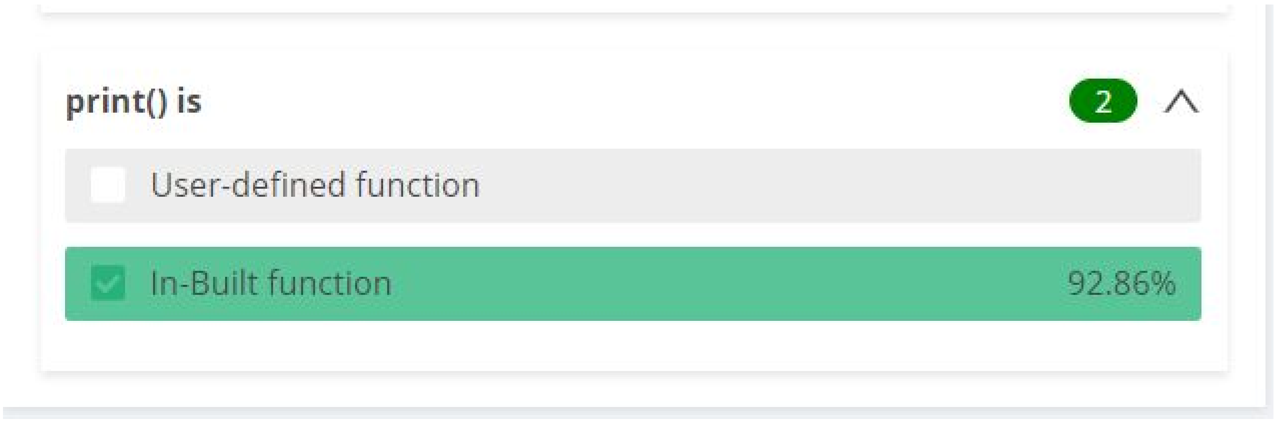
Attempted - 39 (118.18%) EASY

age is actual parameter	25.64%
[✓ name is formal parameter]	74.36%

Python supports

Attempted - 40 (121.21%) EASY

call by value	2.5%
call by reference	12.5%
both	22.5%
[✓ none]	62.5%



print() is

2

User-defined function	
[✓ In-Built function]	92.86%