Department of Computer Science and Engineering
PES University, Bangalore, India

Lecture Notes
Python for Computational Problem Solving
UE23CS151A

Lecture #53 & #54

Functions: Variable number of arguments and Key value pair as arguments. Combination of these

By,
Prof. Sindhu R Pai,
Anchor, PCPS - 2023
Assistant Professor
Dept. of CSE, PESU

Verified by,
PCPS Team - 2023

Many Thanks to
Dr. Shylaja S S (Director, CCBD and CDSAML Research Centers, Former Chairperson, CSE, PES University)
Prof. Chitra G M(Asst. Prof, Dept. of CSE, PCPS Anchor – 2022)

More on Functions

Variable number of positional arguments: *arg [Any variable can be used with *]

If the user wants to create a function which can take any number of positional arguments, what should be the parameter of the function? This function must be able to handle any number of positional arguments from 0 to any. This is similar to the print function which can take any number of arguments. * handles the variable number of positional arguments. **arg is of type tuple inside the function.**

Example code 1:

```
def f1(*arg):
        print(arg, type(arg))    #arg is of type tuple
        for i in arg:
                print(i)
f1(2,1,6,4,9,7)   #All three function calls are valid
f1()
f1(2,4)
```

```
C:\Users\Dell>python notes_functions.py
(2, 1, 6, 4, 9, 7) <class 'tuple'>
2
1
6
4
9
7
() <class 'tuple'>
(2, 4) <class 'tuple'>
2
4

C:\Users\Dell>
```

Variable number of keyword arguments: *kwarg [Any variable can be used with **]

If the user wants to create a function which can take any number of keyword arguments, what should be the parameter of the function? This function must be able to handle any number of keyword arguments from 0 to any. ** handles the variable number of keyword arguments. **kwarg is of type dictionary inside the function.**

Example code 2:

```
def f1(**kwarg):
        print(kwarg, type(kwarg))     #kwarg is of type dict
        for i in kwarg:
                print(i,end="")
        print()
        for i in kwarg.keys():
                print(i,end="")
        print()
```

```
C:\Users\Dell>python notes_functions.py
{'a': 2, 'b': 1, 'c': 6} <class 'dict'>
a b c
a b c
2 1 6
C:\Users\Dell>
```

```
        for i in kwarg.values():
                print(i,end="")
f1(a=2,b=1,c=6)
```

Think about it: Can we call the above function this way? f1(a = 2,b = 1,c = 6, b = 90)

Now, if we have a combination of variable number of arguments and variable number of keyword arguments? How do we handle this?

Example code 3:

```
def f1(*arg,**kwarg):
        print(arg)
        print(kwarg)
f1(56,33,34,4,6,a=21,b=31,c=61)
```

```
C:\Users\Dell>python notes_functions.py
(56, 33, 34, 4, 6)
{'a': 21, 'b': 31, 'c': 61}

C:\Users\Dell>
```

Can we send few arguments to parameters explicitly and then other variable args and kwargs?

Example code 4:

```
def f1(x,y,*arg,**kwarg):
        print(x+y)
        print(arg)
        print(kwarg)
f1(2,3,56,33,34,4,6,a=21,b=31,c=61)
```

```
C:\Users\Dell>python notes_functions.py
5
(56, 33, 34, 4, 6)
{'a': 21, 'b': 31, 'c': 61}

C:\Users\Dell>
```

-END-