**Keyword-Only Arguments**

Keyword-only arguments mean whenever we pass the arguments(or value) by

their parameter names at the time of calling the function in Python

in which if you change the position of arguments then there will be

no change in the output.

Benefits

you will get the correct output because the order of argument doesn’t matter

provided the logic of your code is correct

def nameAge(name, age):

print("Hi, I am", name)

print("My age is ", age)

nameAge(name="Prince", age=20)

**Positional Arguments**

During a function call, values passed through arguments should be in

the order of parameters in the function definition. This is called positional

arguments.

But in the case of positional arguments, you will get more than one output

on changing the order of the arguments.

def add(a,b,c):

return (a+b+c)

print (add(10,20,30))

#Output:60

def nameAge(name, age):

print("Hi, I am", name)

print("My age is ", age)

**# You will get correct output because argument is given in order**

print("Case-1:")

nameAge("Prince", 20)

**# You will get incorrect output because argument is not in order**

print("\nCase-2:")

nameAge(20, "Prince")

**Variable Keywords** = **\*args** and **\*\*kwargs** in Python

\*args (Non-Keyword Arguments)

\*\*kwargs (Keyword Arguments)

What is Python \*args?

The special syntax \*args in function definitions in Python is used to

pass a variable number of arguments to a function. It is used to pass a

non-keyworded, variable-length argument list.

def myFun(\*argv):

for arg in argv:

print(arg)

myFun('Hello', 'Welcome', 'to', 'GeeksforGeeks')

What is Python \*\*kwargs?

The special syntax \*\*kwargs in function definitions in Python is used to

pass a keyworded, variable-length argument list. We use the name kwargs

with the double star. The reason is that the double star allows us to pass

through keyword arguments (and any number of them).

kwargs as being a dictionary that maps each keyword to the value that

we pass alongside it.

def myFun(\*\*kwargs):

for key, value in kwargs.items():

print("%s == %s" % (key, value))

# Driver code

myFun(first='Geeks', mid='for', last='Geeks')