



[Hindi] \*args and \*\*kwargs in python explained | Advanced python tutorials in Hindi



[Overview](#) [Q&A](#) [Downloads](#) [Announcements](#)

## [Hindi] \*args and \*\*kwargs in python explained | Advanced python tutorials in Hindi

What's up guys? In this blog we are going to learn about \*args and \*\*kwargs.

So first of all are the stars more important or the name? Well if answering in single word then **stars**.

Whether you write \*args or \*jokeargs or \*anyrandomthing, if it has one star then it's args. Same with kwargs, you write \*\*kwargs or \*\*jokekwargs or \*\*anyrandomthing, if it has two stars then it's kwargs.

Now you must be thinking **Why writing anything at the end(with the star)? Why can't we just put one star or two if stars are the main thing?**

Well the name is like a variable in which all the data is stored. We can access that data later through that name which we assigned. For example:

```
def listofnumbers(*numbers):  
    print(numbers)
```

In this we can get the whole tuple just by writing "numbers". If the command is:

```
listofnumbers(1,2,3,700,800,100)
```

Output:

```
(1, 2, 3, 700, 800, 100)
```

Output is in tuple. Tuple allows us to do slicing, so we can access data like that too!

### What is this difference between args and kwargs?

In case of args it is stored like a list but it's type is tuple, why tuple? So that nobody could change it's values. If you want to know more about tuple or list slicing then go to this [link](#).

In case of kwargs it is stored like a dictionary. So we can access it like key and value. If you want to know more about dictionary then go to this [link](#).

This is the main basic difference between args and kwargs other than number of stars.

Now I will show you some examples, different ways of using it, different ways of passing data:

### ARGS

```
def listofnumbers(*numbers):  
    j = 0  
    for i in numbers:  
        j = i + j  
    print(f"Sum of all numbers is {j}")
```

```
listofnumbers(1,2,3,700,800,100,390,32809,29380,23908,283703)
```

Output:

```
PS C:\Users\Lovi\Desktop\Blogs> & C:/Users/Lovi/AppData/Local/Programs/Python/Python38/python.exe  
Sum of all numbers is 371796  
PS C:\Users\Lovi\Desktop\Blogs> []
```

As you can see in the code I have no where mentioned the limit of numbers we can pass through that function. It has been possible because of args. Now you don't have to worry about the number of arguments! We can also make a list and pass that. Like this:

```
numberlist = [1,2,3,700,800,100,390,32809,29380,23908,283703]  
listofnumbers(*numberlist)
```

## KWARGS

This is our data:

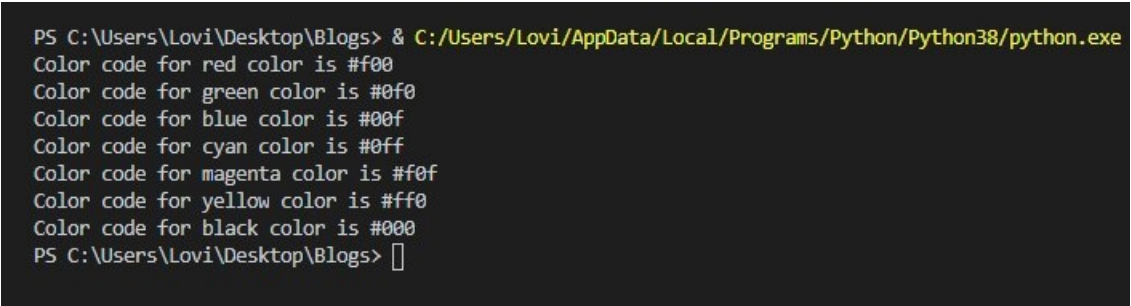
```
dic = {"red": "#f00",
"green": "#0f0",
"blue": "#00f",
"cyan": "#0ff",
"magenta": "#f0f",
"yellow": "#ff0",
"black": "#000",}
```

This is our code:

```
def dicitems(**dic):
    for color, colorcode in dic.items():
        print(f"Color code for {color} color is {colorcode}")

dicitems(**dic)
```

Output:



```
PS C:\Users\Lovi\Desktop\Blogs> & C:/Users/Lovi/AppData/Local/Programs/Python/Python38/python.exe
Color code for red color is #f00
Color code for green color is #0f0
Color code for blue color is #00f
Color code for cyan color is #0ff
Color code for magenta color is #f0f
Color code for yellow color is #ff0
Color code for black color is #000
PS C:\Users\Lovi\Desktop\Blogs> 
```

In this way we can use kwargs. We can get a JSON data and pass through any function with kwargs.

**Note:** Ideally we should pass data in this order -> **NORMAL ARGUMENT > ARGS > KWARGS**

**Why?** Because computer parses them in that order.

**#tut2.py file as described in the video**

[Copy](#)

```
# *args and **kwargs tutorial
# *vars and **kwargs tutorial

def function_1(*argsjoke):

    if(len(argsjoke) ==3):
        print("The name of the student is", argsjoke[0], "and age is",argsjoke[1], "and rollno is",argsjoke[2])
    else:
        print("The name of the student is ", argsjoke[0], "and age is ", argsjoke[1])

def printmarks(**kwargs):
    print(type(kwargs))
    for key, value in kwargs.items():
        print(key, value)

def master(normal, *args, **kwargs):
    print(normal)
    for i in args:
        print(i)
    for key, value in kwargs.items():
        print(key, value)

lis = ["harry", 22, 867656]
# function_1(*lis)
marklist = {"Harry" : 100, "rohan das":97, "Aman Bhateja": 91, "The Programmer":80,
            "Mani Verma":89, "Rajan Gupta":87, "Sanket Wankhede":90, "gaming with hunny": 76}

# printmarks(**marklist)
master("normal arg", *lis, **marklist)
```

[Previous](#)[Next](#)



CodeWithHarry

Copyright © 2022 CodeWithHarry.com

