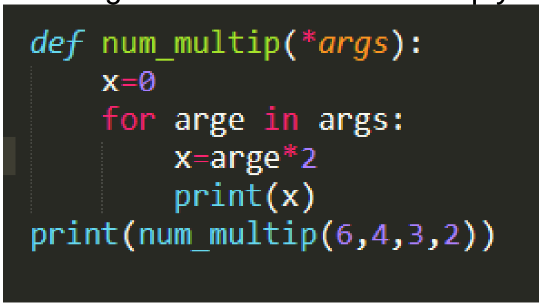
Source=https://www.geeksforgeeks.org/args-kwargs-python/

Code is written by me.

*ARGS

- What **args* allows you to do is take in more arguments than the number of formal arguments that you previously defined. With **args*, any number of extra arguments can be tacked on to your current formal parameters (including zero extra arguments).
- For example : we want to make a multiply function that takes any number of arguments and able to multiply them all together. It can be done using *args.



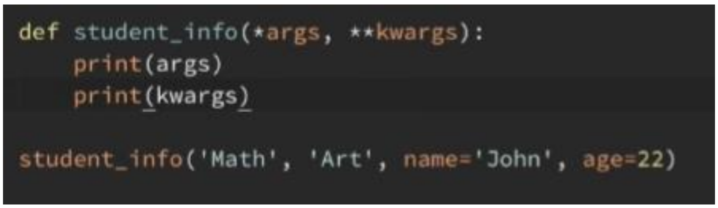
```
def num_multip(*args):
    x=0
    for arge in args:
        x=arge*2
        print(x)
print(num_multip(6,4,3,2))
```

***I forgot to write there return x instead of print x**

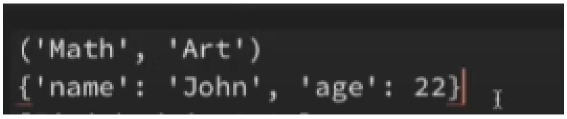
**kwargs= keyword (dictionaries etc…)

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 because the double star allows us to pass through keyword arguments (and any number of them).

- A keyword argument is where you provide a name to the variable as you pass it into the function.
- One can think of the *kwargs* as being a dictionary that maps each keyword to the value that we pass alongside it. That is why when we iterate over the *kwargs* there doesn't seem to be any order in which they were printed out.



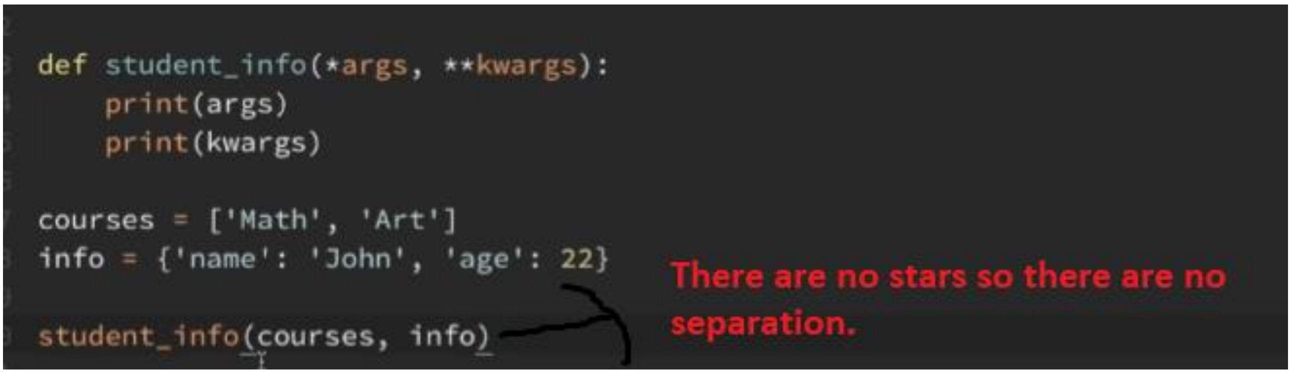
```
def student_info(*args, **kwargs):
    print(args)
    print(kwargs)

student_info('Math', 'Art', name='John', age=22)
```



```
('Math', 'Art')
{'name': 'John', 'age': 22}
```

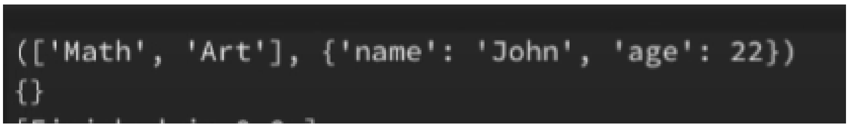
(source,Corey Schafer ,Python tutorial,Youtube)

What we see here print args gives us Math and Art. But kwargs gives us key and values**



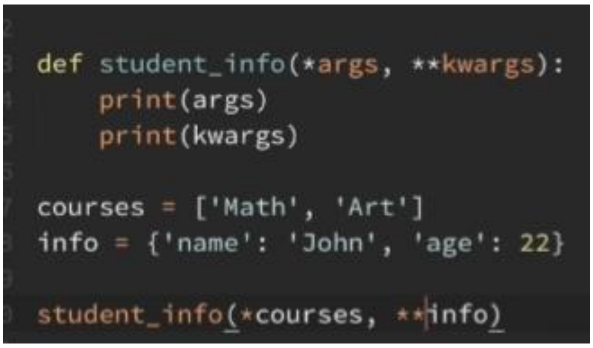
```
def student_info(*args, **kwargs):
    print(args)
    print(kwargs)

courses = ['Math', 'Art']
info = {'name': 'John', 'age': 22}

student_info(courses, info)
```

There are no stars so there are no separation.

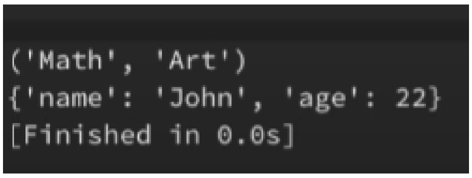


```
(['Math', 'Art'], {'name': 'John', 'age': 22})
{}
```

When we add to * and **



```
def student_info(*args, **kwargs):
    print(args)
    print(kwargs)

courses = ['Math', 'Art']
info = {'name': 'John', 'age': 22}

student_info(*courses, **info)
```



```
('Math', 'Art')
{'name': 'John', 'age': 22}
[Finished in 0.0s]
```

Now what we see separation here because