args and **kwargs in python

▫ *args and **kwargs are special keyword which allows function to take variable length argument.

▫ *args passes variable number of non-keyworded arguments and on which operation of the tuple can be performed.* args receives arguments as a tuple.

▫ **kwargs passes variable number of keyword arguments dictionary to function on which operation of a dictionary can be performed.** kwargs receives arguments as a dictionary.

▫ *args and **kwargs make the function flexible.

*args (Arguments or non keyword arguments)

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.

In [1]:
```
#example
def multiply(*args):
    for i in args:
        c = 3*i
        print(c)

multiply(1,2,3,4,5)
```

```
3
6
9
12
15
```

In [2]:
```
def sentence(*words):
    for i in words:
        print(i,'')

sentence('arguments', 'and', 'keyword', 'arguments')
```

```
arguments
and
keyword
arguments
```

In [3]:
```
def example(arg1, *args):
    print("First argument :", arg1)
    for arg in args:
        print("Next argument through *args :", arg)
```

```
example('Hello', 'Welcome', 'to', 'Python')
```

```
First argument : Hello
Next argument through *args : Welcome
Next argument through *args : to
Next argument through *args : Python
```

In [4]:
```
# defining fruits class
class fruits():
    # args receives unlimited no. of arguments as an array
    def __init__(self, *args):
        # access args index like array does
        self.color = args[0]
        self.taste = args[1]


# creating objects of fruits class
kiwi = fruits('green', 'sour')
apple = fruits('red', 'sweet')

# printing the color and taste of the fruits
print('The color of apple is',apple.color)
print('The taste of kiwi is {}'.format(kiwi.taste))
```

```
The color of apple is red
The taste of kiwi is sour
```

**kwargs (Keyword Arguments)

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).

In [5]:
```
#example
def example(**kwargs):
    for key, value in kwargs.items():
        print(key,value)

example(a=10, b=20, c=30, d=40)
```

```
a 10
b 20
c 30
d 40
```

In [6]:
```
def abc(arg1, **words):
    print('Palindrome', arg1)
    for i, j in words.items():
        print("%s == %s" % (i, j))


abc("Words", MOM='MOM', radar='radar', civic='civic')
```

```
Palindrome Words
MOM == MOM
radar == radar
civic == civic
```

In [7]:

```python
def intro(**data):
    print("\nData type of argument:",type(data))

    for key, value in data.items():
        print("{} is {}".format(key,value))

intro(Firstname="abc", Lastname="def", Age=22, Phone=1234567890)
intro(Firstname="ghi", Lastname="jkl", Email="ghijkl@nomail.com", Country="India", Age
```

```
Data type of argument: <class 'dict'>
Firstname is abc
Lastname is def
Age is 22
Phone is 1234567890

Data type of argument: <class 'dict'>
Firstname is ghi
Lastname is jkl
Email is ghijkl@nomail.com
Country is India
Age is 23
Phone is 9876543210
```

In [8]:

```python
# defining fruits class
class fruits():
    # args receives unlimited no. of arguments as an array
    def __init__(self, **kwargs):
        # access args index like array does
        self.color = kwargs['c']
        self.taste = kwargs['t']


# creating objects of fruits class
kiwi = fruits(c='green', t='sour')
apple = fruits(c='red', t='sweet')
# printing the color and taste of the fruits
print('The color of apple is',apple.color)
print('The taste of kiwi is {}'.format(kiwi.taste))
```

```
The color of apple is red
The taste of kiwi is sour
```

In [9]:

```python
def example(*args, **kwargs):
    print("args: ", args)
    print("kwargs: ", kwargs)


# Now we can use both *args ,**kwargs
# to pass arguments to this function :
example('hello', 'world', first="hello", last="world")
```

```
args:  ('hello', 'world')
kwargs:  {'first': 'hello', 'last': 'world'}
```