Types Of Formal Parameters:

- **Positional Parameters**
    - **All the formal parameters are acts as positional parameters**
    - **If you define any function with positional parameter then we must pass exactly same no.of Parameters otherwise we will be get an Error**

Example:

```
def myData(name,age):
   print("Name is : ",name)
   print("Age is : ",age)
   print("-"*30)

#calling function
myData("Ramesh",34)
myData("Ravi")   #Error
```

- **Named Parameters**
    - **Actually the formal parameters are also acts named parameter**
    - **In the positional parameter the PVM will always concentrate only on count of the parameter, But it doesn't cares order of the parameter. By change if we change the order of the parameter then result will different**

- To Overcome this problem then we have to pass the values to the formal parameter with their names called “Named Parameter”.

Example:

```
def myData(name,age):
   print("Name is : ",name)
   print("Age is : ",age)
   print("-"*30)

#calling function
myData("Ramesh",34)
myData(age=38,name="sudha")
```

- **Default Parameters**
  - It is Process of defining the formal parameters with some values.
  - These default parameter will also work like an optional parameter
  - These default parameter will takes place whenever we fail to pass the values to formal parameter

Example:

```
def myAdd(x=10,y=20,z=30):
   s=x+y+z
   print("Sum is : ",s)
   print("-"*30)

#calling
myAdd(30,40,50)
myAdd(30,40)
myAdd(30)
myAdd()
```

- **Varargs Parameter**
    - It is the process of declare the formal parameter with * symbol
    - Varargs internally works as tuple collection
    - Varargs can take 0 arguments to N.no.of arguments
    - Varargs can also take with other types of parameter but in this case varargs should be the last parameter

Example:

```
#varargs parameter

import time

def myAdd(*x):
    time.sleep(1)
    print("Type of x : ",type(x))
    print("Data of x : ",x)
    print("-"*30)

#calling
myAdd(10,20)
myAdd(10,20,30,40,50)
myAdd(10)
myAdd()
```

Example 2:

```
#varargs parameter
import time

def myAdd(*x):
    s=0
    for i in x:
        s=s+i
    print("Sum is : ",s)
    print("-"*30)
```

```
#calling
myAdd(10,20,30,40,50)
myAdd(10,20,30)
myAdd(10,20)
myAdd(10)
myAdd()
```

Example 3:

```
import time

def result(name,*marks):
   print("Name is : ",name)
   print("Marks : ",marks)
   print("total is : ",sum(marks))  #sum(iterable) -> int
   print("- "*30)

#calling
result("Sudha",50,40,50)
result("manasa",40,50)
result("bunny",30)
result("nani")
```

➢ Kwargs Parameters

- Kwargs parameters are the process of declaring the formal parameters with ** symbols
- Internally kwargs works as “dict” collections
- It will take 0-items to N-no of.items
- Kwargs can also takes with other type of parameter but in this case also kwargs must be last arguments

Examples:

```
import time

def myStudent(**x):
   for item in x.items():
      time.sleep(1)
      k,d=item  #tuple unpacking
      print(k,d,sep='<<>>')

   print("-" *30)

#calling
myStudent(sno=101)
myStudent(sno=102,sname="Ramesh")
myStudent(sno=103,sname="Madhu",scity="Khammam")
myStudent( )
```