Function Arguments

```
def greet(name, msg):
    """
    This function greets to person with the provided message
    """
    print("Hello {0} , {1}".format(name, msg))

#call the function with arguments
greet("satish", "Good Morning")
```

```
Hello satish , Good Morning
```

```
#suppose if we pass one argument

greet("satish") #will get an error
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-3-b48ea98044bf> in <module>()
      1 #suppose if we pass one argument
      2
----> 3 greet("satish") #will get an error

TypeError: greet() missing 1 required positional argument: 'msg'
```

Different Forms of Arguments

1. Default Arguments

We can provide a default value to an argument by using the assignment operator (=).

```
def greet(name, msg="Good Morning"):
    """
    This function greets to person with the provided message
    if message is not provided, it defaults to "Good Morning"
    """
    print("Hello {0} , {1}".format(name, msg))

greet("satish", "Good Night")
```

```
Hello satish , Good Night
```

```
#with out msg argument
greet("satish")

Hello satish , Good Morning
```

Once we have a default argument, all the arguments to its right must also have default values.

def greet(msg="Good Morning", name)

#will get a SyntaxError : non-default argument follows default argument

2. Keyword Arguments

kwargs allows you to pass keyworded variable length of arguments to a function. You should use **kwargs if you want to handle named arguments in a function

Example:

```
def greet(**kwargs):
    """
    This function greets to person with the provided message
    """
    if kwargs:
        print("Hello {0} , {1}".format(kwargs['name'], kwargs['msg']))
greet(name="satish", msg="Good Morning")

Hello satish , Good Morning
```

3. Arbitary Arguments

Sometimes, we do not know in advance the number of arguments that will be passed into a function.Python allows us to handle this kind of situation through function calls with arbitrary number of arguments.

Example:

```
def greet(*names):
    """
    This function greets all persons in the names tuple
    """
    print(names)

    for name in names:
        print("Hello,  {0} ".format(name))
```

```python
greet("satish", "murali", "naveen", "srikanth")
```

```
('satish', 'murali', 'naveen', 'srikanth')
Hello,  satish
Hello,  murali
Hello,  naveen
Hello,  srikanth
```