Working With Functions

1. Simple Function

2. Function with Parameters and return value

3. Calling one function from another function

4. Returning multiple values

5. Function and Arguments

* Default arguments

* Required arguments

* Keyword arguments

* Variable number of arguments

In [1]:

```
# 1. Simple function

# function definition
def hello_world():
    """Prints 'Hello World!' on the console."""
    print("Mahendra Garodi")

# Calling function
hello_world()
```

```
Mahendra Garodi
```

In [2]:

```
# 2. Function with Parameters and return value
def add(x, y):
    """
    Returns sum of x and y
    @param x A number
    @param y A number
    @return Sum of x and y
    """
    return x+y

# Different ways to call a function
answer = add(10, 20)
print("10 + 20 => %d" % answer)
print("30 + 40 => %d" % add(30, 40))
```

```
10 + 20 => 30
30 + 40 => 70
```

In [3]:

```
x= 100
y = 200
print("x + y => %d " % add(x, y))

fullName = add("Mahendra ", "Garodi")
print(fullName)
add("10", 10)
```

```
x + y => 300
Mahendra Garodi

---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-3-6d334b950724> in <module>()
      5 fullName = add("Mahendra ", "Garodi")
      6 print(fullName)
----> 7 add("10", 10)

<ipython-input-2-963b1ba79af0> in add(x, y)
      7     @return Sum of x and y
      8     """
----> 9     return x+y
     10
     11 # Different ways to call a function

TypeError: must be str, not int
```

In [4]:

```
# 3. Calling one function from another function

def double_number(x):
    return add(x, x)

print("Double Number : %d " % double_number(10))
```

```
Double Number : 20
```

In [5]:

```
# 4. Returning multiple values
def get_person_info():
    first_name = "Mahendra"
    last_name = "Garodi"
    age = 27

    return (first_name, last_name, age)

person = get_person_info()
fname, lname, age = get_person_info()
print(person)
print("First Name : %s, Age: %d" % (fname, age))
```

```
('Mahendra', 'Garodi', 27)
First Name : Mahendra, Age: 27
```

In [6]:

```
# 5. Function and Arguments -> Default Argument
def power(number, index=2):
    total = 1
    for i in range(0, index):
        total *= number
    return total

print(power(10,3))
print(power(10))
```

```
1000
100
```

In [7]:

```
# 5. Function and Arguments -> Keyword Argument
def get_full_name(first_name, last_name):
    return first_name + " " + last_name

print(get_full_name("Mahendra", "Garodi"))
print(get_full_name(last_name = "Garodi", first_name = "Mahendra"))
#print(get_full_name(first_name = "Mahendra", "Garodi"))
```

```
Mahendra Garodi
Mahendra Garodi
```

In [8]:

```
# 5. Function and Arguments -> Variable number of arguments
def add_all(*args):
  answer = 0
  for i in args:
    answer += i
  return answer

print(add_all(100, 200, 300))
```

```
600
```

In [9]:

```
# 6. Local Variable vs Global Variable
global_variable = 100

def print_global():
    #global global_variable
    global_variable = 200
    print("Value in function: %d " % global_variable)

print_global()
print("Value outside function: %d" % global_variable)
```

```
Value in function: 200
Value outside function: 100
```

In [10]:

```
# 3. call by reference
def append_number(number_list, number):
    number_list.append(number)

x = []
append_number(x, 10)
append_number(x, 20)
print(x)
```

```
[10, 20]
```

In [11]:

```
# Functions calling other functions

def one_good_turn(n):
  return n + 1

def deserves_another(n):
  return one_good_turn(n) + 2
```