In [1]:
```
def fun():
    print("Welcome to GFG")
```

In [3]:
```
def fun():
    print("Welcome to GFG")
fun()
```

```
Welcome to GFG
```

In [5]:
```
def evenOdd(x: int) ->str:
    if (x % 2 == 0):
        return "Even"
    else:
        return "Odd"

print(evenOdd(16))
print(evenOdd(7))
```

```
Even
Odd
```

In [7]:
```
def evenOdd(x):
    if (x % 2 == 0):
        return "Even"
    else:
        return "Odd"

print(evenOdd(16))
print(evenOdd(7))
```

```
Even
Odd
```

Default Arguments:

In [9]:
```
def myFun(x, y=50):
    print("x: ", x)
    print("y: ", y)


myFun(10)
```

```
x:  10
y:  50
```

Keyword Arguments:

In [12]:
```
def student(fname, lname):
    print(fname, lname)

student(fname='Geeks', lname='Practice')
student(lname='Practice', fname='Geeks')
```

```
Geeks Practice
Geeks Practice
```

Postional Arguments:

In [16]:
```
def nameAge(name, age):
    print("Hi, I am", name)
    print("My age is ", age)


print("Case-1:")
nameAge("Suraj", 27)

print("\nCase-2:")
nameAge(27, "Suraj")
```

```
Case-1:
Hi, I am Suraj
My age is  27

Case-2:
Hi, I am 27
My age is  Suraj
```

Arbitary keyword Arguments:

Example:Variable length non-keywords argument

In [20]:
```
def myFun(*argv):
    for arg in argv:
        print(arg)

myFun('Hello', 'Welcome', 'to', 'GeeksforGeeks')
```

```
Hello
Welcome
to
GeeksforGeeks
```

Example:Variable length keywords argument

In [25]:
```
def myFun(**kwargs):
    for key, value in kwargs.items():
        print("%s == %s" % (key, value))


myFun(first='Geeks', mid='for', last='Geeks')
```

```
first == Geeks
mid == for
last == Geeks
```

DocString

Example: Adding Docstring to the function

In [29]:
```
def evenOdd(x):
    """Function to check if the number is even or odd"""

    if (x % 2 == 0):
        print("even")
    else:
        print("odd")
```

```
print(evenOdd.__doc__)
```

```
Function to check if the number is even or odd
```

Python Function within Functions:

In [32]:
```
def f1():
    s = 'I love GeeksforGeeks'

    def f2():
        print(s)

    f2()

f1()
```

```
I love GeeksforGeeks
```

Anonymous FUnction in Python:

In [38]:
```
def cube(x):
    return x*x*x

cube_l = lambda x : x*x*x

print(cube(7))
print(cube_l(7))
```

```
343
343
```

Example: Python Function Return Statement

In [41]:
```
def square_value(num):
    """This function returns the square
    value of the entered number"""
    return num**2

print(square_value(2))
print(square_value(-4))
```

```
4
16
```

Pass by Reference and Pass by Value:

In [44]:
```
def myFun(x):
    x[0] = 20

lst = [10, 11, 12, 13, 14, 15]
myFun(lst)
print(lst)
```

```
[20, 11, 12, 13, 14, 15]
```

In [46]:
```
def myFun(x):
    x = [20, 30, 40]


lst = [10, 11, 12, 13, 14, 15]
```

```
myFun(lst)
print(lst)
```

```
[10, 11, 12, 13, 14, 15]
```

In [48]:
```
def myFun(x):
    x = 20

x = 10
myFun(x)
print(x)
```

```
10
```

In [50]:
```
def swap(x, y):
    temp = x
    x = y
    y = temp

x = 2
y = 3
swap(x, y)
print(x)
print(y)
```

```
2
3
```

Recursive Functions in Python

In [53]:
```
def factorial(n):
    if n == 0:
        return 1
    else:
        return n * factorial(n - 1)

print(factorial(4))
```

```
24
```

In [ ]:
```

```