Python Decorators

- **Decorators** are a very powerful and useful tool in Python since it allows programmers to modify the **behavior of function or class**.
- Decorators allow us to **wrap another function** in order to extend the behavior of the wrapped function, **without permanently modifying it**.
- **<u>Important Notes:</u>**
    - ***Decorators****: are special types of functions which helps to extends the functionality of the function by adding some new action on it.*
    - *It will always have a* ***parameter****.*
    - *It will always have an* ***inner function.***
    - *It* ***will return*** *the inner function.*
- **Consider the following example:**

```
-   def myFunct(func):
-       def inner():
-           print("Python")
-           func()
-       return inner
-
-   @myFunct
-   def function1():
-       print("Hello")
-
-   function1()
```

- **Output:**

```
-   Python
-   Hello
```

- **Consider following example for parameter passing:**

```
-   def myFunct(func):
-       def inner(x,y):
-           print("Addition of two number=",x+y)
-           func(x,y)
-       return inner
-
-   @myFunct
-   def function1(a,b):
-       print("Substraction of two number=",a-b)
-   function1(12,34)
```

- **Output:**

```
-   Addition of two number= 46
-   Substraction of two number= -22
-
```