

الدرس دا هنتكلم فيه عن ال **Decorators**، هي بتكون عبارة عن فانكشن بتقبل فانكشن فى ال arguments بتوعها وبتستخدم علشان اطبع خرج فانكشن ب decoration معين واعمل تحسين لل behavior بتاع الفانكشن دى (اللى هي انا دخلتها ك argument لفانكشن ال decoration).

بنشأ ال decorator عن طريق اني اعمل فانكشن واعرف فيها parameter على سبيل المثال اسمه func وجوا الفانكشن بعمل nested function استدعي جواها الفانكشن اللى هي فى ال parameter بتاع ال decorator واعمل ال decoration سواء لو هطبع كلام قبل او بعد الفانكشن او ايا كان ال decoration اللى انا هعمله، بعد كدا بخلي قيمة ال return بتاع ال decorator هي ال nested function. (الكود احسن من الكلام).

بعد كدا اى فانكشن عايز انفذ عليها ال decoration دا بدخلها ك argument جوا فانكشن ال decorator او الطريقه الافضل واللى هنستخدمها وهي ان وانا بعرف الفانكشن اللى عايز اعملها decoration بكتب فوقها @myDecorator او ايا كان اسم فانكشن ال decorator بتاعتي، وبعد كدا لما اجي اعمل call للفانكشن هيطبعهاالى متنفذ عليها ال decoration اللى عملته.

```
# -----
# -- Decorators => Intro --
# -----
# [1] Sometimes Called Meta Programming
# [2] Everything in Python is Object Even Functions
# [3] Decorator Take A Function and Add Some Functionality and Return It
# [4] Decorator Wrap Other Function and Enhance Their Behavior
# [5] Decorator is Higher Order Function (Function Accept Function As Parameter)
# -----

def myDecorator(func): # Decorator

    def nestedFunc(): # Any Name Its Just For Decoration

        print("Before") # Message From Decorator

        func() # Execute Function

        print("After") # Message From Decorator

    return nestedFunc # Return All Data

@myDecorator
def sayHello():

    print("Hello From Say Hello Function")

@myDecorator
def sayHowAreYou():

    print("Hello From Say How Are You Function")

afterDecoration = myDecorator(sayHello)
```

```
afterDecoration()  
  
sayHello()  
  
print("#" * 50)  
  
sayHowAreYou()
```

#### CODE

```
ocuments/Python Course/first.py"  
Before  
Before  
Hello From Say Hello Function  
After  
After  
Before  
Hello From Say Hello Function  
After  
#####  
Before  
Hello From Say How Are You Function  
After
```

#### OUTPUT