فى الدرس اللى فات اتكلمنا عن ان ال decorator بياخد الفانكشن ك argument وقبل م ينفذها ممكن انه ينفذ عليها task معينه، بس كان الفانكشن بتاعتنا عباره عن فانكشن بسيطه ملهاش parameters، ف لو عايز اعمل decoration على فانكشنز ليها parameters فبحط ال parameters بتوعها ك ل ال nested function .

حاجه مهمه كمان ان ال decorator مش بس بيطبعلي حاجه قبل تنفيذ الفانكشن وبعدها، لا كنا اتكلمنا الدرس اللى فات عن انه بيحسن من ال behavior بتاع الفانكشن، يعنى على سبيل المثال فى الفانكشن اللى ف الكود خليناه قبل م يبدأ ينفذ الفانكشنز يعمل على ال arguments اللى داخلينله check اذا كانت الارقام اقل من صفر او لا وبناءا على دا ممكن اكتبله تحذير او حتى اعمل break من الفانكشن.

ممكن كمان ان يكون عندي اكتر من decorator وانفذهم على نفس الفانكشن وكل واحد بيعمل وظيفه معينه، زي م موجود فى اخر جزئيه فى الكود.

```
# ---------------------------------------------
# -- Decorators => Function With Parameters --
# ---------------------------------------------

def myDecorator(func):  # Decorator

  def nestedFunc(num1, num2):  # Any Name Its Just For Decoration

    if num1 < 0 or num2 < 0:

      print("Beware One Of The Numbers Is Less Than Zero")

    func(num1, num2)  # Execute Function

  return nestedFunc  # Return All Data

def myDecoratorTwo(func):  # Decorator

  def nestedFunc(num1, num2):  # Any Name Its Just For Decoration

    print("Coming From Decorator Two")

    func(num1, num2)  # Execute Function

  return nestedFunc  # Return All Data

@myDecorator
@myDecoratorTwo

def calculate(n1, n2):

  print(n1 + n2)

calculate(-5, 90)
```

CODE

```
Beware One Of The Numbers Is Less Than Zero
Coming From Decorator Two
85
```

OUTPUT