فى الدرس دا هنطبق فكرة ال packing وال unpacking على ال decorator وهناخد مثال عملي بسيط على حاجه اقدر انى اعملها باستخدام ال decorators.

فى الدرس اللى فات كنا بنتغلب على مشكلة الدرس اللى قبله اللى هي اني اطبق ال decorator على فانكشن بتاخد عدد معين من ال parameters وقلنا علشان نعمل دا بنحط ال parameters جوا ال nested function جوا ال decorator، طيب فى حالة لو انا مش عارف عدد ال arguments اللى هتجيلي من ال user فى الحاله دى بستخدم طريقة ال packing وال unpacking اللى كانت بتخليني ادخل للفانكشن اى عدد من ال arguments.

ف فى حالة ال decorator بحط ال parameter فى ال nested function بطريقة ال packing اللى هو المتغير بيكون قبله * وبعدين علشان انفذ تاسك على ال arguments بعمل عليهم loop واشوف عايز انفذ عليهم ايه.

فى الجزء التانى من الدرس مثال بيحسب الوقت اللى بتستغرقه الفانكشن في التنفيذ، فعلشان هنستخدم الوقت استدعينا ال time module.

عرفنا ال decorator بتاعنا على انه يجيب الوقت قبل تنفيذ الفانكشن والوقت بعد تنفيذها ويطبع الوقت بعد ناقص الوقت قبل.

```
# ----------------------------------------
# -- Decorators => Practical Speed Test --
# ----------------------------------------

from time import time

def myDecorator(func):  # Decorator

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

    for number in numbers:

      if number < 0:

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

    func(*numbers)  # Execute Function

  return nestedFunc  # Return All Data

@myDecorator

def calculate(n1, n2, n3, n4):

  print(n1 + n2 + n3 + n4)

calculate(-5, 90, 50, 150)
```

```python
def speedTest(func):

  def wrapper():

    start = time()

    func()

    end = time()

    print(f"Function Running Time Is: {end - start}")

  return wrapper

@speedTest

def bigLoop():

  for number in range(1, 10):

    print(number)

bigLoop()
```

CODE

```
Beware One Of The Numbers Is Less Than Zero
285
1
2
3
4
5
6
7
8
9
Function Running Time Is: 0.0009949207305908203
```

OUTPUT