

# Decoratori\_Funzioni

February 19, 2020

```
[31]: def decoratore(func):  
      def wrapper():  
          print("io sono prima della funzione")  
          func(input("come ti chiami?\n"))  
          print("io sono dopo la funzione")  
      return wrapper()
```

```
[32]: @decoratore  
def amico(nome):  
    print("ciao amico " + str(nome))
```

```
io sono prima della funzione  
ciao amico Alan  
io sono dopo la funzione
```

```
[61]: amico("Eleonora") # senza decoratore
```

```
ciao amico Eleonora
```

```
[1]: import time
```

```
[32]: def calc_square(numbers):  
      start = time.time()  
      result = []  
      for number in numbers:  
          result.append(number * number)  
      end = time.time()  
      print("calc_square_took " + str((end - start)*1000) + " mil_second")  
      return result
```

```
[33]: def calc_cube(numbers):  
      result = []  
      start = time.time()  
      for number in numbers:  
          result.append(number * number * number)  
      end = time.time()  
      print("calc_cube_took " + str((end - start)*1000) + " mil_second")
```

```
return result
```

```
[34]: array = range(1, 10000)
```

```
[35]: out = calc_square(array)
```

```
calc_square took 1.7163753509521484 mil_second
```

In questo esempio ci sono 2 problemi: \* abbiamo del codice che si *ripete* in entrambe le funzioni \* le funzioni dovrebbero restituire quadrato e cubo ma noi ritorniamo un TEMPO

La soluzione è usare i **Decoratori**...

```
[59]: def time_it(func):
      def wrapper(*args, **kwargs):
          start = time.time()
          func(*args, **kwargs)
          end = time.time()
          print(func.__name__ + " took " + str((end - start)*1000) + " mil_seconds")

      return wrapper
```

```
[60]: @time_it
      def calc_square(numbers):
          start = time.time()
          result = []
          for number in numbers:
              result.append(number * number)
          end = time.time()
          return result
```

```
[61]: @time_it
      def calc_cube(numbers):
          result = []
          start = time.time()
          for number in numbers:
              result.append(number * number * number)
          end = time.time()
          return result
```

```
[62]: array = range(1, 10000)
```

```
[63]: risultato = calc_square(array)
```

```
calc_square took 0.7786750793457031 mil_seconds
```

```
[64]: risultatoCubo = calc_cube(array)
```

```
calc_cube took 2.429962158203125 mil_seconds
```

[ ]: