

<https://www.linkedin.com/in/masoud-faridi-b851aa125/>

What are decorators in Python?

Decorators allow you to change the behavior of a function and, add some functionality to it without modifying the function itself. Consider a situation where we have 100 functions (or more), and we want to know when they were executed, finished, and the time elapsed in executing the functions. We may want to count the number of times a function is called. Does it need to modify all those functions?

In general a decorator function:

- takes a function as an argument
- returns a closure
- the closure usually accepts any combination of parameters
- runs some code in the inner function (closure)
- the closure function calls the original function using the arguments passed to the closure
- returns whatever is returned by that function call

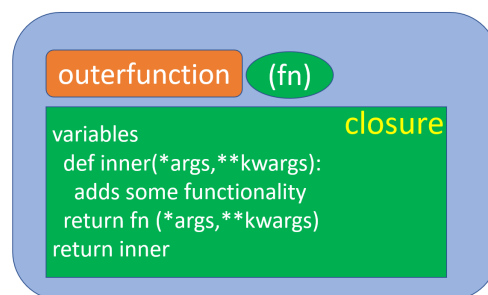


Figure 1: A function decorator in Python is just a function that takes in another function as an argument, extending the decorated function's functionality without changing its structure

## Python Code

```
import time
from datetime import datetime
def summary_fn(fn):
    count=0
    def inner(*args,**kwargs):
        nonlocal count
        count+=1
        start_time = datetime.now().strftime('%Y-%m-%d %H:%M:%S')
        tmp=time.perf_counter()
        ret = fn(*args, **kwargs)
        end_time = datetime.now().strftime('%Y-%m-%d %H:%M:%S')
        total_time = time.perf_counter()-tmp
        print("function started: {}".format(fn.__name__))
        print("start_time :{}".format(start_time))
        print("end_time: {}".format(end_time))
        print("total_time: {}".format(total_time))
        print("count: {}".format(count))
        return ret
    return inner
```

```
def add(a,b):  
    return a+b  
add=summary_fn(add)  
  
@summary_fn  
def pw(a,b):  
    return a**b  
  
from math import sqrt  
sqrt=summary_fn(sqrt)
```

```
add(1,4)
```

```
## function started: add
## start_time :2023-05-11 22:27:16
## end_time: 2023-05-11 22:27:16
## total_time: 5.2999999999858716e-06
## count: 1
## 5
```

```
add(5,2)
```

```
## function started: add
## start_time :2023-05-11 22:27:16
## end_time: 2023-05-11 22:27:16
## total_time: 4.900000000002125e-06
## count: 2
## 7
```

```
add(7,10)
```

```
## function started: add
## start_time :2023-05-11 22:27:16
## end_time: 2023-05-11 22:27:16
## total_time: 4.599999999993498e-06
## count: 3
## 17
```

```
pw(2,3)
```

```
## function started: pw
## start_time :2023-05-11 22:27:16
## end_time: 2023-05-11 22:27:16
## total_time: 5.599999999994498e-06
## count: 1
## 8
```

```
pw(3,2)
```

```
## function started: pw
## start_time :2023-05-11 22:27:16
## end_time: 2023-05-11 22:27:16
## total_time: 4.799999999971494e-06
## count: 2
## 9
```

```
pw(3,3)
```

```
## function started: pw
## start_time :2023-05-11 22:27:16
## end_time: 2023-05-11 22:27:16
## total_time: 5.200000000038507e-06
## count: 3
## 27
```

```
sqrt(4)
```

```
## function started: sqrt
## start_time :2023-05-11 22:27:16
## end_time: 2023-05-11 22:27:16
## total_time: 5.599999999994498e-06
## count: 1
## 2.0
```

```
sqrt(169)
```

```
## function started: sqrt
## start_time :2023-05-11 22:27:16
## end_time: 2023-05-11 22:27:16
## total_time: 4.9000000000298805e-06
## count: 2
## 13.0
```

```
sqrt(16)
```

```
## function started: sqrt
## start_time :2023-05-11 22:27:16
## end_time: 2023-05-11 22:27:16
## total_time: 4.999999999977245e-06
## count: 3
## 4.0
```

Contact me

Contact me at [masoudfaridi@modares.ac.ir](mailto:masoudfaridi@modares.ac.ir) or [masoud1faridi@gmail.com](mailto:masoud1faridi@gmail.com)