

A Decorator Is A Function

- ◆ It's a function that creates a new function from an old function

```
def decorator(function: Callable) -> Callable:  
    details go here
```


- >>> from functools import wraps

- >>> def shabby_log(function):

- ... @wraps(function)

- ... def decorated(*args, **kw):

- ... print(f"{function.__name__}(*{args}, **{kw})")

- ... return function(*args, **kw)

- ... return decorated

- ...

- >>> @shabby_log

- ... def example(a, b):

- ... return a+b

- ...

- >>> example(2, 3)

- example(*(2, 3), **{})

- 5

The New Function

Built around the original function