Decorator

Mohammad Tasin (Tasin Coder)

Agenda

- Decorator

Decorator

- Decorators are implemented using the @ symbol, followed by the name of the decorator function.
- The decorator function returns a wrapper function that wraps the original function.
- The wrapper function calls the original function and can add additional behavior before or after the function is executed.
- Decorator A function that takes another function as an argument and extends its behavior

Normal Function

```
def greet():
    print("TasiNCoder", end=' ')
```

Decorator Function

```python
def my_decorator(func):
    def wrapper():
        print("Add Additional features")
        print("Hello", end=' ')
        func()
        print("sir")
    return wrapper
```

```python
def my_decorator(func):
    def wrapper():
        print("Add Additional features")
        print("Hello", end=' ')
        func()
        print("sir")
    return wrapper
@my_decorator
def greet():
    print("TasiNCoder", end=' ')
greet()
```