

# Untitled6

May 11, 2023

## 1 Decorators

[ ]:

```
[2]: def test():  
      print(4+5)
```

```
[3]: test()
```

9

```
[4]: def test1():  
  
      print("this is the start of my fun")  
      print(4+5)  
      print("this is the end of my fun")
```

```
[5]: test1()
```

this is the start of my fun

9

this is the end of my fun

[ ]:

##a design pattern in Python that allows a user to add new functionality to an existing object without modifying its structure

[ ]:

```
[7]: def deco(func):  
  
      def inner_deco():  
          print("this is the start of my fun")  
          func()  
          print("this is the end of my fun")
```

```
return inner_deco
```

```
[9]: @deco
def test2():
    print(4+5)
```

```
[10]: test2()
```

```
this is the start of my fun
9
this is the end of my fun
```

```
[ ]:
```

```
[11]: import time
def timer_test(func):
    def timer_test_inner():
        start = time.time()
        func()
        end = time.time()
        print(end-start)

    return timer_test_inner
```

```
[13]: @timer_test
def test4():
    print(45+67)
```

```
[14]: test4()
```

```
112
4.649162292480469e-05
```

```
[ ]:
```

```
[25]: @timer_test
def test6():
    for i in range (10000000):

        pass
```

```
[26]: test6()
```

```
0.24210667610168457
```

```
[ ]:
```

[ ]: