

```
#A decorator takes in a function, adds some functionality and returns it.  
#Functions and methods are called callable as they can be called.  
#Any object which implements the special method __call__() is termed callable.  
#A decorator is a callable that returns a callable.  
#A decorator takes in a function, adds some functionality and returns it.
```

```
def make_pretty(func):  
    def inner():  
        print("I got decorated")  
        func()  
    return inner  
def ordinary():  
    print("I am ordinary")
```

```
ordinary() #outputs "I am ordinary"  
# let's decorate this ordinary function  
pretty = make_pretty(ordinary)  
pretty()  
#outputs:  
#I got decorated  
#I am ordinary
```

```
#make_pretty() is a decorator.  
#pretty = make_pretty(ordinary)  
#The function ordinary() got decorated and  
#the returned function was given the name pretty.
```