

Python - Classes Wrapper

Uma função em Python é um objeto de primeira ordem. Uma função pode ter outra função como argumento e envolver outra definição de função dentro dela. Isso ajuda a modificar uma função sem realmente alterá-la. Essas funções são chamadas de decoradores.

Este recurso também está disponível para encerrar uma aula. Esta técnica é usada para gerenciar a classe após ela ser instanciada, envolvendo sua lógica dentro de um decorador.

Exemplo

```
def decorator_function(Wrapped):  
    class Wrapper:  
        def __init__(self,x):  
            self.wrap = Wrapped(x)  
        def print_name(self):  
            return self.wrap.name  
    return Wrapper  
  
@decorator_function  
class Wrapped:  
    def __init__(self,x):  
        self.name = x  
  
obj = Wrapped('TutorialsPoint')  
print(obj.print_name())
```

Aqui, **Wrapped** é o nome da classe a ser empacotada. É passado como argumento para uma função. Dentro da função, temos uma classe Wrapper, modificamos seu comportamento com os atributos da classe passada e retornamos a classe modificada. A classe retornada é instanciada e seu método agora pode ser chamado.

Quando você executa este código, ele produzirá a seguinte **saída** -

```
TutorialsPoint
```