Python 裝飾器--

Function

Accept function as variation

Return function

e.g. c-funct = var, a-funct = funct; in a-funct, you can choose whether use c-funct or not, or enhance c-funct

# 只有2個函數

def a():

def b():

print(‘hello’)

a() # 在最後呼叫函數即可跑函數裡面的程式碼

b() – just one of a()’s funct, so we can’t use it

def out(func\_args): # 外圍函數

def inter(\*args, \*\*kwargs): # 內嵌函數

return func\_args(\*args, \*\*kwargs)

return inter 在外圍函數返回內嵌函數—因為所有的業務都在內, 不返回則無法調用

1. 將被調函數作參傳入外圍函括弧
2. 將裝飾器與被調函數綁定
3. @ + 裝飾器符號放在被調函數的上一行, 被調函數正常定義, 只需直接調用被執行函數

# 有三個以上的函數

def a(func):

def b(\*args, \*\*kwargs):

func(\*args, \*\*kwargs)

# 在装饰器内部func就是传进来的c( )函数

return b

def c(name): # name為函數c中的參數

print(name)

a(c)(‘dewei’) # a中返回c函數並傳參

@ a # 裝飾器a函數右邊不加括弧 -- 返回a

def c(name): # name為函數c中的參數

print(name)

c(‘dewei’) # c函數傳參 – name = ‘dewei’

在调用c()时，会先执行装饰器clock(func)；

可以将c函数的参数name=’dewei’传给了a( )函数内部的b( )函数(注：在这里c函数有参数name)，func即test函数，

在clocked( )函数中应调用func(\*args,\*\*kwargs)