

回顾一下闭包

```
# # 写一个闭包函数，并说明每一行代码的含义。
def outer():# # 一 只检测函数体的语法(工厂合不合格)，不执行函数体代码 (不使用工厂)
    print('外面的函数正在运行')
    # a = 5
    def inner():
        # print(a)
        print('里面的函数正在运行')
    return inner

inner =outer()# 二 执行了outer函数的体代码，1定义了inner函数，2返回了inner函数地址，3.把这个返回的
inner函数地址赋值给inner变量
#
inner() # 三 inner函数地址加括号运行
```

day15.装饰器

装饰器的实现必须遵循两大原则

- 1、不修改被装饰对象的源代码(人的原来的性格，生活方式)
 - 2、不修改被装饰对象的调用方式(人的原来的外貌，名字)
- 装饰器其实就在遵循1和2原则的前提下为被装饰对象添加新功能

```
...
def 装饰器外层函数(被装饰的函数):
    def 装饰器内层函数():
        被装饰的代码
        被装饰的函数()
        被装饰的代码
    return 装饰器内层函数
@装饰器外层函数
def 被装饰的函数():
    被装饰函数体代码
...
```

装饰器是一个特殊的闭包函数

第一题

写一个登录装饰器对一下函数进行装饰，要求输入账号和密码才能运行该函数
(代码写好注释)

```
def run():  
    print('开始执行函数')
```

答案

```
def new_func(func):  
    def inner():  
        username = input('输入账号')  
        password = input('输入密码')  
        if username == 'dahai' and password == '123':  
            print('通过认证')  
            func()  
        else:  
            print('用户或者密码错误')  
    return inner  
  
@new_func  
def run():  
    print('开始执行函数')  
  
run()
```

day16.异常处理与文件读写

题目

第一题（单选题）

以下哪个代码是正确的读取一个文件？

- A. f = open("test.txt", "read")
- B. f = open("r", "test.txt")
- C. f = open("test.txt", "r")
- D. f = open("read", "test.txt")

答案

C

第二题

Python 内置函数 `open` 用来打开或创建文件并返回文件对象

答案

`open`

第三题（单选题）

python中用来抛出异常的关键字是（） A.try B.exception C.raise D.finally

答案

C

选做题

第四题

通过异常捕获获得以下字符串里面的字母字符 `str_1='d52a733i2327ha244i982d23s553b245'`

答案

```
str_1='d52a733i2327ha244i982d23s553b245'
a = ''
for i in str_1:
    # print(i)
    try:
        int(i)
    except Exception as f:
        # print('字母',i)
        a += i
print(a)
```