

内部函数

1

内部函数 ——闭包

什么是闭包

- 闭包将内部函数自己的代码和作用域以及外部函数的作用结合起来
- 闭包的词法变量不属于全局名字空间域或者局部的--而属于其他的名字空间，带着“流浪”的作用域
- 闭包对于安装计算，隐藏状态，以及在函数对象和作用域中随意地切换是很有用的
- 闭包也是函数，但是他们能携带一些额外的作用域

闭包实例

- 创建通用的计数器

```
>>> def counter(start=0):  
...     count = start  
...     def incr():  
...         nonlocal count  
...         count += 1  
...         return count  
...     return incr  
>>> a = counter()  
>>> b = counter(10)  
>>> a()  
1  
>>> b()  
11
```

2

内部函数函数

——装饰器

装饰器

- 装饰器是在函数调用之上的修饰
- 这些修饰仅是当声明一个函数或者方法的时候，才会应用的额外调用
- 使用装饰器的情形有：
 - 引入日志
 - 增加计时逻辑来检测性能
 - 给函数加入事务的能力

- 通过装饰器实现以下功能
 1. 函数运行前打印当前时间
 2. 函数运行结束后打印结束时间



更多精彩...



<http://bj.linux.tedu.cn/>
企业QQ：86198501