

在于,可以`持久化保存变量`。

关于持久化保存变量

首先要明白`作用域`这个概念,作用域分为`全局作用域`和`局部作用域`两种,全局作用域指的是全局变量的作用域,一般在外部创建,在程序的整个生命周期内都可以使用,且可以在任意地方使用;局部作用域是指局部变量的作用域,在内部创建,运行时创建,结束时销毁。

```

```
MAX_NUM = 10 # 这里的MAX_NUM就是全局变量,全局可用
def count(a:int):
```

```
 num = 0 # 这里的num就是局部变量,只能在count函数内部使用,且每次运行count函数时创建,count函数结束时销毁
```

```
 if a > MAX_NUM:
 num = a # 修改局部变量,仅在当前有效,第二次运行函数时,变量的值仍然是0
 c = a - MAX_NUM
 c = a + num
 print(f"当前num值为:{num}")
 return c
```

```
print(d)
```

```

上面代码的执行结果为

```
> 15
>
> 20
```

可以看到,当闭包函数中的变量名称和全局变量的名称一致时,内部函数使用的是离它自己最近的变量,而不是全局变量。

这也就涉及到了函数的变量的取值过程,如果我们使用一个变量时,变量没有被定义,那么Python将自动`从内到外`的去寻找这个变量,也就是全局变量是最后的取

值,如果查询到全局变量,且全局变量中也没有这个变量的话,Python将直接产生异常。

闭包总结

通过上面的内容,可以发现闭包的好处主要有两个方面

> 变量可以持久化保存,并不会随着函数的返回而销毁,因为外部函数的返回值是内部函数,所以只要外部函数没有被del掉,变量就可以实现持久化存储(外部函数的变量)

>
> 相较于全局变量的持久化存储,闭包函数中的变量更有安全性,因为不能被随意更改

原文链接: <https://juejin.cn/post/7375030479496331274>