## With

# 简介

设计对象类时，我们可以为对象类新增两个方法，一个是\_\_enter(self)\_\_,一个是\_\_exit\_\_(self, exc\_type, exc\_val, exc\_tb)。

\_\_enter(self)\_\_

负责返回一个值，该返回值将赋值给as子句后面的var\_name，通常返回对象自己，即“self”。函数优先于with后面的“代码块”(statements1,statements2,……)被执行。

\_\_exit\_\_(self, exc\_type, exc\_val, exc\_tb)

执行完with后面的代码块后自动调用该函数。with语句后面的“代码块”中有异常(不包括因调用某函数,由被调用函数内部抛出的异常) ，会把异常类型，异常值，异常跟踪信息分别赋值给函数参数exc\_type, exc\_val, exc\_tb，没有异常的情况下，exc\_type, exc\_val, exc\_tb值都为None。另外，如果该函数返回True、1类值的Boolean真值，那么将忽略“代码块”中的异常，停止执行“代码块”中剩余语句，但是会继续执行“代码块”后面的语句；如果函数返回类似0，False类的Boolean假值、或者没返回值，将抛出“代码块”中的异常，那么在没有捕获异常的情况下，中断“代码块”及“代码块”之后语句的执行

with xxx as var\_name：

# 代码块开始

statements1

statements2

……

# 代码块结束

# 代码快后面的语句

statements after code block

# 代码演示1

*#!/usr/bin/env python*

*# -\*- coding:utf-8 -\*-*

\_\_author\_\_ = **'laiyu'**

**class**User(object):

**def**\_\_init\_\_(self, username, password):

self.\_username =  username

self.\_password = password

    @property

**def**username(self):

**return**self.\_username

    @username.setter

**def**username(self, username):

self.\_username = username

    @property

**def**password(self):

**return**self.\_password

    @password.setter

**def**password(self, password):

self.\_password = password

**def**\_\_enter\_\_(self):

**print**(**'auto do something before statements body of with executed'**)

**return**self

**def**\_\_exit\_\_(self, exc\_type, exc\_val, exc\_tb):

**print**(**'auto do something after statements body of with executed'**)

**if**\_\_name\_\_ == **'\_\_main\_\_'**:

    boy = User(**'shouke'**, **'shouke2014'**)

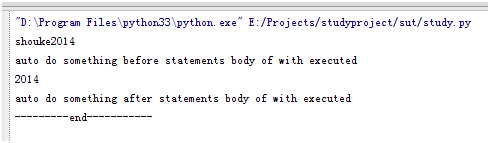
**print**(boy.password)

**with**User(**'shouke'**, **'2014'**) **as**user:

**print**(user.password)

**print**(**'---------end-----------'**)

运行结果



如上，代码块运行前调用了\_\_enter\_\_函数，代码块运行完，自动调用了\_\_exit\_\_函数

# 

# 代码演示2

更改上述部分代码如下，继续运行

**def**\_\_exit\_\_(self, exc\_type, exc\_val, exc\_tb):

**print**(**'auto do something after statements body of with executed'**)

**print**(**'exc\_type:'**, exc\_type)

**print**(**'exc\_val:'**, exc\_val)

**print**(**'exc\_tb:'**, exc\_tb)

**return**False

**if**\_\_name\_\_ == **'\_\_main\_\_'**:

    boy = User(**'shouke'**, **'shouke2014'**)

**print**(boy.password)

**with**User(**'shouke'**, **'2014'**) **as**user:

**print**(user.password)

12/0

**print**(**'after execption'**)

**print**(**'---------end-----------'**)

运行结果

**对比实验：**

在上述的基础上继续修改代码

**def**\_\_exit\_\_(self, exc\_type, exc\_val, exc\_tb):

**print**(**'auto do something after statements body of with executed'**)

**print**(**'exc\_type:'**, exc\_type)

**print**(**'exc\_val:'**, exc\_val)

**print**(**'exc\_tb:'**, exc\_tb)

**return**True

**if**\_\_name\_\_ == **'\_\_main\_\_'**:

    boy = User(**'shouke'**, **'shouke2014'**)

**print**(boy.password)

**with**User(**'shouke'**, **'2014'**) **as**user:

**print**(user.password)

12/0

**print**(**'after execption'**)

**print**(**'---------end-----------'**)

运行结果：

注意：

1、抛异常后，代码块中剩余的语句没有再继续运行

2、如果在上述的基础上，把代码中的 12/0剪切后放到password(self)中 ，抛出异常的异常信息是不会传递给\_\_exit\_\_函数的

@property

**def**password(self):

12/0

**return**self.\_password

运行结果如下，

