Day19

封装

Interface

私有属性和私有方法（类内以\_\_下划线开头的标识符）

多态

动态（Python只能实现动态）

静态

多继承

Class tank(ship,car)

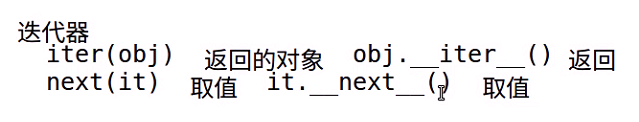
Pass

MRO(Method Resolution Order)

Tank.\_\_mro\_\_

函数重写：

在自定义的类内加入相应的方法可以让自定义类型的对象使用内见函数进行操作



Day20

异常（高级）：

With 语句

语法：

With 表达式1 [as 变量1]，表达式2 [as 变量2]….

语句块

作用：

使用对资源进行访问你的场合，确保使用过程中不管是否发生异常都会执行必须的‘清理’操作，并释放资源。

如：文件代开后自动关闭，线程中锁的自动获取和释放（线程后面学）

说明：

as子句中的变量用于绑定表达式执行后生成的对象

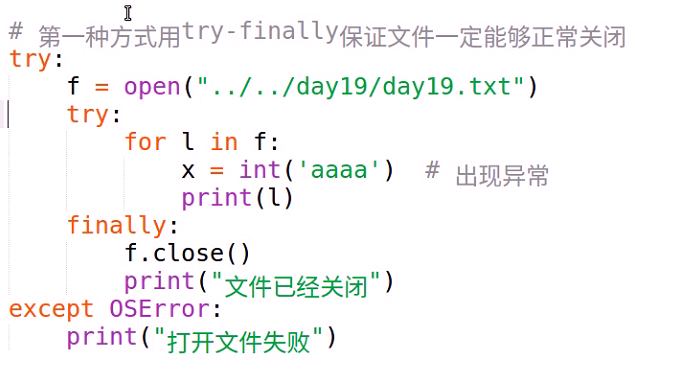
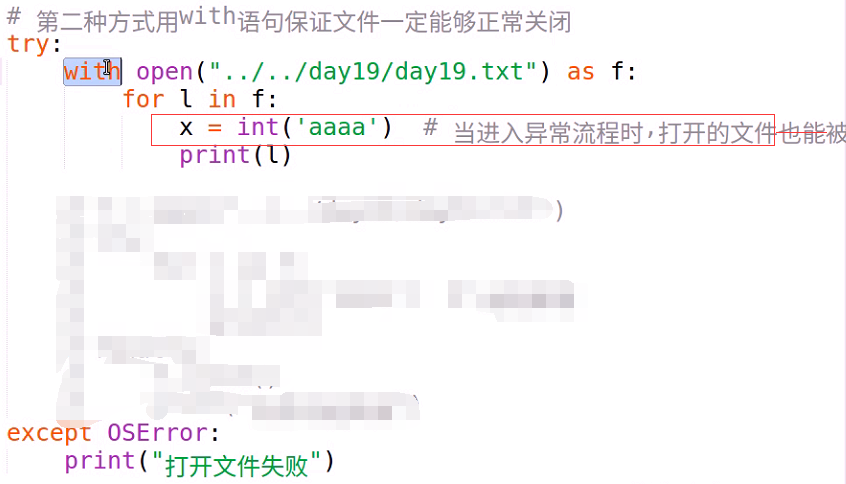
with语句并不会改变异常的状态

#异常相关语句：

#Try-exxept

#Try-finally.

With.py

f.close会自动调用

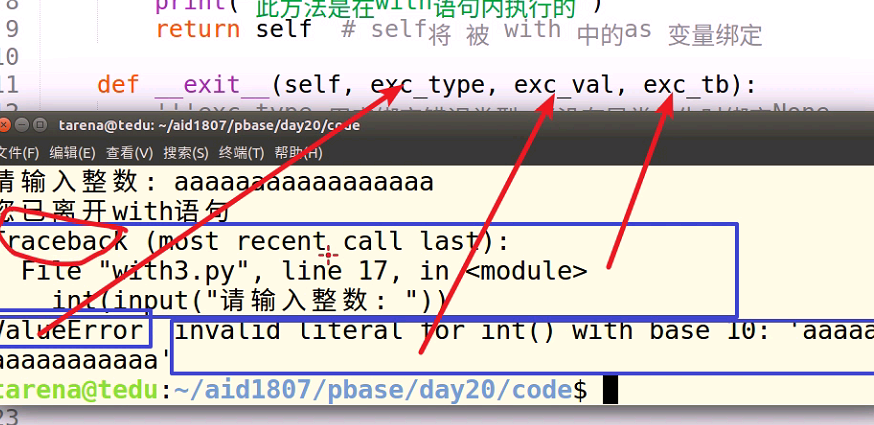
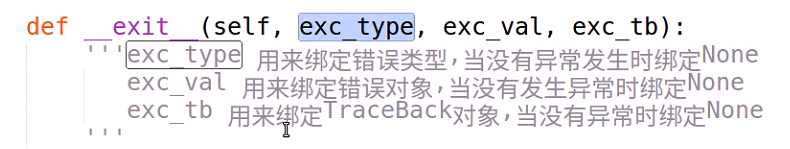
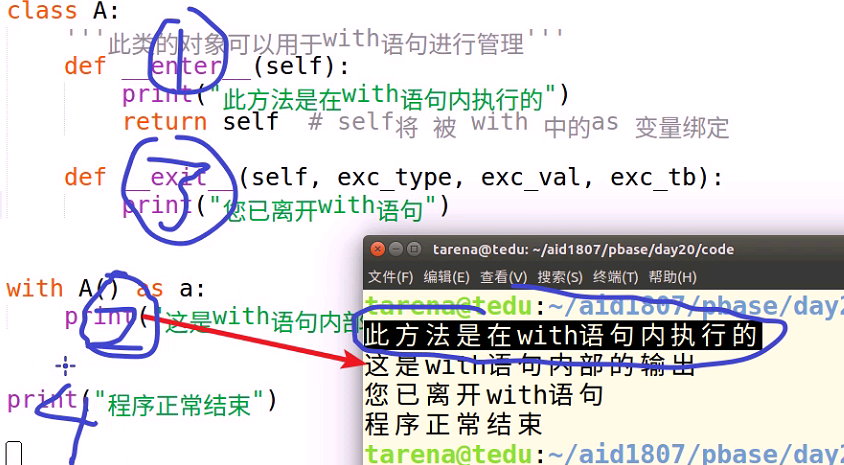
资源管理器：

1、类内有\_\_enter\_\_和\_\_exit\_\_实例方法的类创建对象被称为环境换利器

2、能够用with语句进行管理的对象必须是环境管理器

3、\_\_enter\_\_方法将在进入with语句时被调用，由as变量绑定返回的对象

4、\_\_exit\_\_方法将在离开with语句时被自动调用，且可以通过参数来判断，离开with语句是否有异常发生。



运算符重载

什么是运算符重载

让自定义的类生成的对象（实例）能够使用运行算进行操作

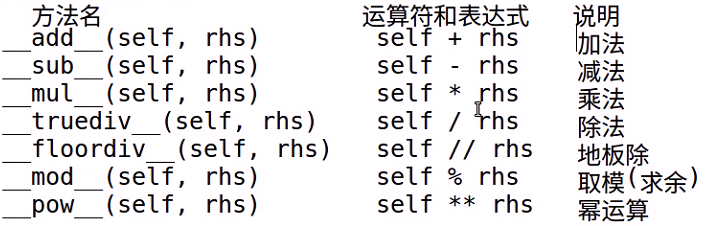
作用：、

让自定义类的实例像内见对象一样进行运算符操作

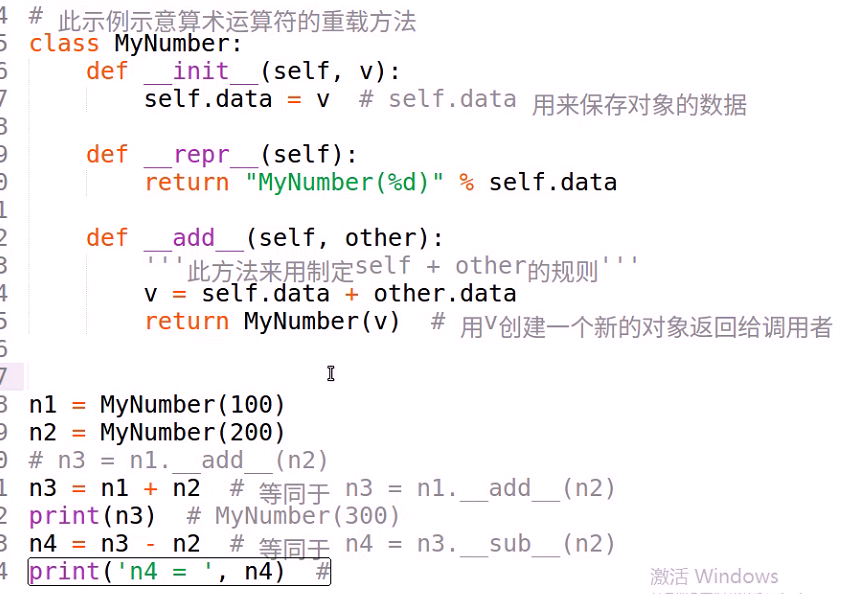
让程序简洁易读

对自定义对象将运算符赋予新的规则

算数运算符的重载：



示例：



Rhs(right hand side)

说明：

运算符重载方法的参数已经有固定的含义，不建议改变原的运算符的含义及参数的意义

二元运算符的重载方法格式：

Def \_\_xxx\_\_(self,other):

语句块

练习：

实现两个自定义列表的相加

Class mylist:

Def \_\_init\_\_(self,iterable=()):

Self.data=list(iterable)

def \_\_add\_\_(self,other):

v=self.data +other.data

return mylist(v)

L1=MyList([1,2,3])

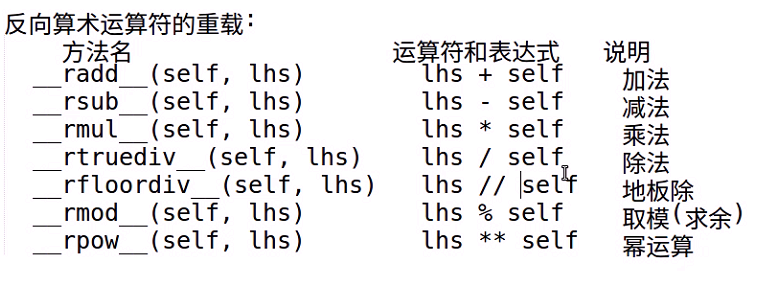
L2=MyList([4,5,6])

L3=L1+L2

Print（L3）

反向算数运算符的重载

当运算符的左侧为内建类型时，右侧为自定义类型进行算数运算符运算时会出现TyoeError错误，因无法修改内建类型的代码实现运算符重载，此时需要使用反向算数运算符重载。

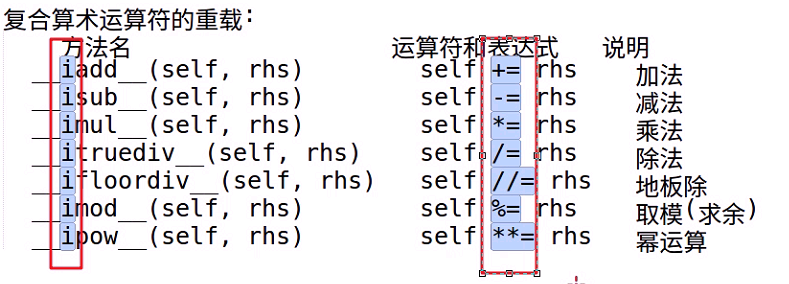




复合赋值算术运算符的重载

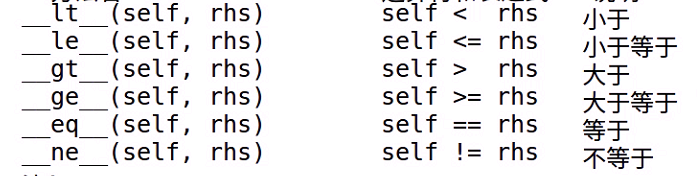
以复合赋值算数运算符x+=y为例，此运算符会优先调用x.\_\_iadd\_\_(y)方法

如果没有\_\_iadd\_\_方法时会将复合赋值运算拆解为：x=x+y然后调用x=x.\_add\_\_(y)方法，如果再不存在\_\_add\_\_方法则会触发TypeError类型的错误异常



比较运算符的重载

方法名 运算符和表达式 说明



注：比较运算符通常返回布尔值True或False

位相关运算符重载

方法名 运算符和表达式 说明

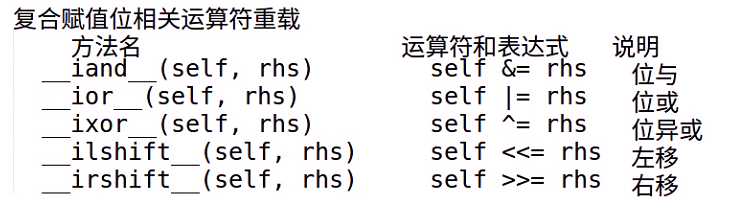
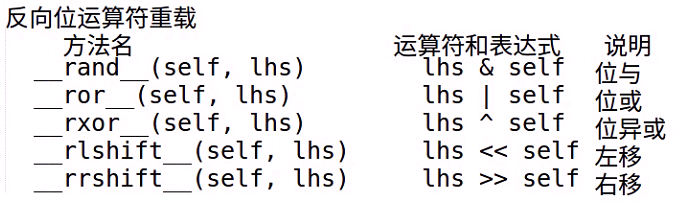
\_\_and\_\_(self,rhs) self & rhs 位与

\_\_or\_\_(self,rhs) self | rhs 位或

\_\_xor\_\_(self,rhs) self ^ rhs 位异或

\_\_lshift\_\_(self,rhs) self << rhs 左移

\_\_rshift\_\_(self,rhs) self >> rhs 右移



一元运算符的重载：

方法名 运算符和表达式 说明

\_\_neg\_\_(self) -self 负号

\_\_pos\_\_(self) +self 正号

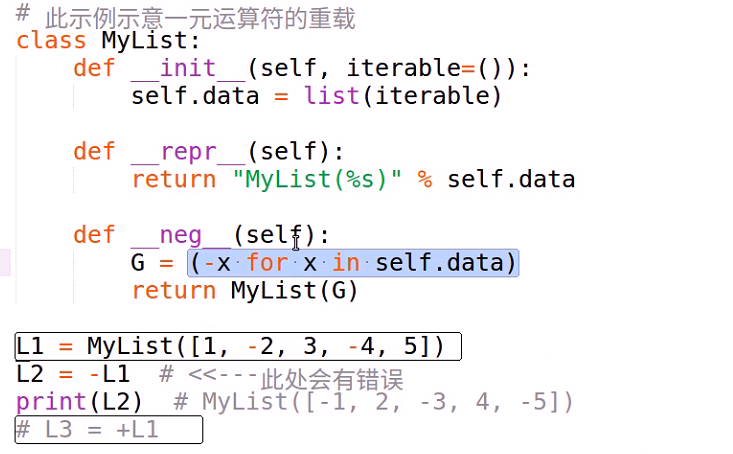
\_\_invert\_\_(self) ~self 取反

语法：

Class 类名：

Def \_\_xxx\_\_(self):

….



In / not in 运算符重载

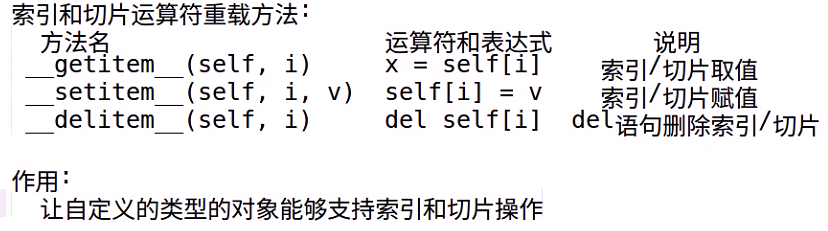
格式：

Def \_\_contains\_\_(self,e):

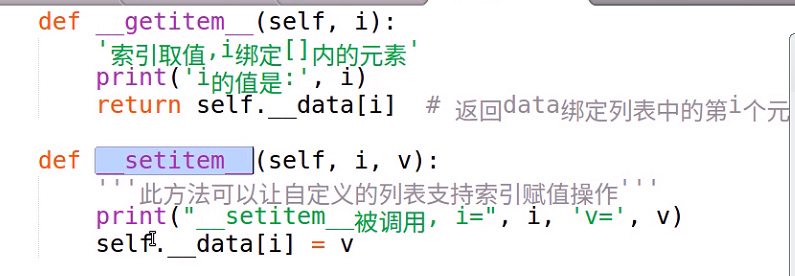
语句

注：in / not in 返回布尔值 True/False 当重载了\_\_contains\_\_后，in和 not in运算符都可以用，not in运算符的返回值与in 相反

示例：







Slice 构造函数

作用：

用于创建一个slice对象，此对于用于切片操作的传值

格式：

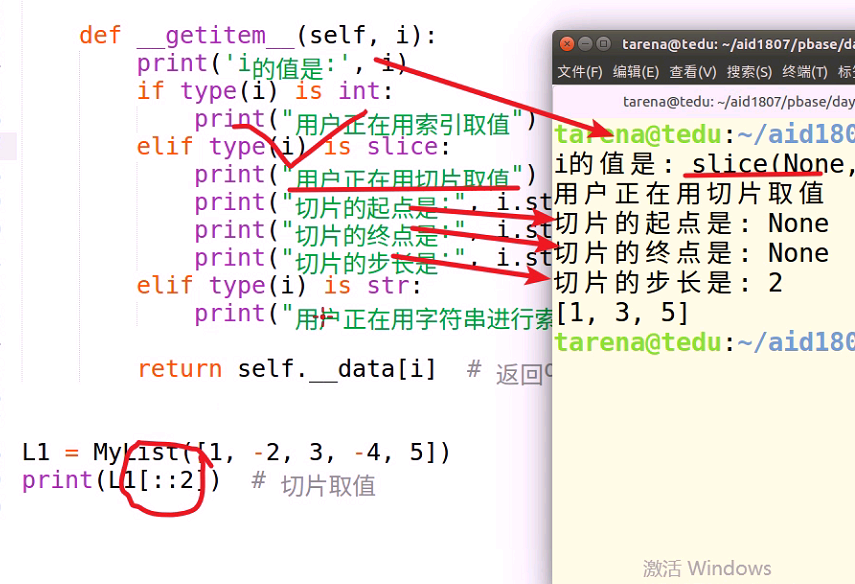
Slice（start=none,stop=none,step=none）

Slice对象的实例属性：

Start 切片的起始值 默认为none

Stop 切片的终止值 默认为none

Step 切片的



特性属性：@property

实现其他语言所有拥有的getter和setter 功能

作用：

用来模拟一个属性

通过@property装饰器，可以对模拟属性的赋值和取值加以控制

