try 测试

reason 原因

except 除了

Exception 例外 [异常的父类]

Filename 文件名

遇到的常见的异常：

SyntaxError 语法错误

NameError 未声明/初始化对象

TypeError 对类型无效的操作

KeyError 字典中查找不存在的键

IOError 输入/输出操作失败

IndexError 序列中没有此索引

ImportError 导入模块/对象失败

异常的捕抓：（第四种方法最好用）

◎ try......except

寻找异常遵循从上而下寻找

把异常变量赋值给了reason，reason并非一成不变，可进行更改。

一、同一个类型的异常出现才能捕抓。

try:

print (a)

except NameError as reason: #as把异常的原因赋值给reason

print (reason)

结果：name 'a' is not defined

二、不同类型进行捕抓是捕抓不到的，会报错。

try:

print (1+’1’)

except NameError as reason:

print (reason)

结果：

Traceback (most recent call last):

File "C:/Users/Win8/Desktop/Error.py", line 2, in <module>

print (1+'1')

TypeError: unsupported operand type(s) for +: 'int' and 'str'

三、兼容方法

try:

print (1+'1')

except (NameError,TypeError) as reason: #把不同异常写在括号里

print (reason)

结果：unsupported operand type(s) for +: 'int' and 'str'

四、在父类异常中直接找出子类异常 #直接用Exception异常的父类

try:

print (1+’1’)

except Exception as reason:

print (reason)

结果：unsupported operand type(s) for +: 'int' and 'str'

例子：

try:

filename = 'm.py'

f = open(filename)

except Exception as reason:

xx = input('文件不存在，是否创建这个文件？（Y/N）')

if xx == 'Y':

f = open(filename,'w+')

print ('已为您创建了文件%s,现在可以对f文件对象进行读写的操作'% filename)

文件不存在，是否创建这个文件？（Y/N）Y

已为您创建了文件m.py,现在可以对f文件对象进行读写的操作

>>> f

<\_io.TextIOWrapper name='m.py' mode='w+' encoding='cp936'>

异常处理：

注意：中间的except，else，finally都是可选的但至少有一个，不然try就没有意义了，根据实际中的需求来选择。

◎ try......except

#如果测试语句中发生异常时执行

try:

print (a)

except:

print('异常')

结果：异常 #原本是报错的

◎ try......else

else如果测试语句中没有发生异常时执行

try:

print (1)

except Exception as reason:

print(reason)

else:

print('没异常')

结果：没异常

◎ try......finally

finally后面的接的语句就是无论异常是否发生都会执行的语句

try:

print (a)

except NameError as reason:

print (reason)

finally:

print('xxxxx')

结果

name 'a' is not defined

xxxxx

触发异常：

◎ raise [Exception]

使用raise语句自己触发异常，并且raise后面的语句不再执行

inputValue =input("请输入一个整数类型数据 :")

if type(inputValue) != type(1):

raise ValueError

else:

print inputValue

断言：

◎ assert 条件

若真则返回None，若假则会抛出AssertionError异常。

>>> 3>8

False

>>> assert 3>8

Traceback (most recent call last):

File "<pyshell#2>", line 1, in <module>

assert 3>8

AssertionError