

# 异常处理

# 1

异常处理

—— 异常

# 什么是异常

- 当python检测到一个错误时，解释器就会指出当前流已经无法继续执行下去，这时候就出现了异常
- 异常是因为程序出现了错误而在正常控制流以外采取的行为
- 这个行为又分为两个阶段：
  - 首先是引起异常发生的错误
  - 然后是检测（和采取可能的措施）阶段

# python中的异常

- 当程序运行时，因为遇到未解的错误而导致中止运行，便会出现traceback消息，打印异常

异常	描述
NameError	未声明/初始化对象
IndexError	序列中没有没有此索引
SyntaxError	语法错误
KeyboardInterrupt	用户中断执行
EOFError	没有内建输入，到达EOF标记
IOError	输入/输出操作失败

# try-except语句

- 定义了进行异常监控的一段代码，并且提供了处理异常的机制

```
try:
    try_suite #监控这里的异常
except Exception[as reason]:
    except_suite #异常处理代码
```

```
>>> try:
...     f = open('foo.txt')
... except IOError:
...     print('No such file')
...
No such file
```

## 带有多多个expect的try语句

- 可以把多个except语句连接在一起，处理一个try块中可能发生的多种异常

```
>>> try:
...     data = int(input('input a number: '))
... except KeyboardInterrupt:
...     print 'user cancelled'
... except ValueError:
...     print('you must input a number!')
...
input a number: hello
you must input a number!
```

# 异常参数

- 异常也可以有参数，异常引发后它会被传递给异常处理器
- 当异常被引发后参数是作为附加帮助信息传递给异常处理器的

```
>>> try:
...     10 / 0
... except ZeroDivisionError as e:
...     print('error', e)
...
error division by zero
```

- 简化除法判断
  1. 提示用户输入一个数字作为除数
  2. 如果用户按下Ctrl+C或Ctrl+D则退出程序
  3. 如果用户输入非数字字符，提示用户应该输入数字
  4. 如果用户输入0，提示用户0不能作为除数



# else子句

- 在try范围中没有异常被检测到时，执行else子句
- 在else范围中的任何代码运行前，try范围中的所有代码必须完全成功

```
>>> try:
...     result = 100 / int(input("number: "))
... except Exception as e:
...     print('Error:', e)
... else:
...     print(result)
...
number: 10
10.0
```

# finally子句

- finally子句是无论异常是否发生，是否捕捉都会执行的一段代码
- 如果打开文件后，因为发生异常导致文件没有关闭，可能会发生数据损坏。使用finally可以保证文件总是能正常的关闭

# 2

## 异常处理

### —— 触发异常

# raise语句

- 要想引发异常，最简单的形式就是输入关键字raise，后面跟要引发的异常的名称
- 执行raise语句时，Python会创建指定的异常类的一个对象
- raise语句还可指定对异常对象进行初始化的参数

# 断言

- 断言是一句必须等价于布尔值为真的判定
- 此外，发生异常也意味着表达式为假

```
>>> assert 10 > 100, "Wrong"
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
AssertionError: Wrong
```



更多精彩...



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