

Python异常 处理

python的运行时错误成为异常

语法错误：软件的结构上有错误而导致不能被解释器解释或不能被编译器编译

逻辑错误：由于不完整或不合法输入所致，也可能是逻辑无法生成、计算或输出结构需要的过程无法执行等

python异常是一个对象，表示错误或意外情况

在Python检查到一个错误是，将触发一个异常

python可以通过异常传导机制传递一个对象，发出一个异常情况出现的信号

程序员也可以在代码中手动触发异常

错误处理

python的默认处理：停止程序，打印错误消息

使用try语句处理异常并从异常中回复

事件通知

用于发出有效状态信号

特殊情况处理

无法调整代码去处理的场景

终止行为

try/finally语句可确保执行必须的结束处理机制

非常规控制流程

异常是一种高级跳转（goto）机制

异常通过try语句来检测

任何在try语句块里的代码都会被监控，以检查有无异常发生

try语句主要有两种形式

try-except:检测和处理异常

可以有多个except

支持使用else子句处理没有探测异常的执行的代码

try-finally:金检查异常并做一些必要的清理工作

仅能一个finally

try语句复合形式

try-except语句

语法:

```
try:
    f = open('/tmp/a.txt','r')
except IOError, e:
    print 'Could not open file:',e
```

```
try:
    try_suite
except Exception1[,reason]:
    suite_exception1
except Exception2[,reason]:
    suite_
...
except:
    suite_
else:
    else_suite
```

except 分句个数没有限制，但else只有一个
没有异常发生时， else分句才会执行

没有符合except分句时，异常会向上传递到程序中的之前进入try中或者到进程顶层

try-finally语句

无论异常是否发生，finally子句都会执行

常用于定义必须进行的清理动作，如关闭文件，断开服务器

```
try:
    try_suite
finally:
    finally——suite
```

```
try:
    f = open('/tmp/a.txt','r')
    f.write("hello world")
finally:
    f.close()
```

```
try:
    while True:
        d = raw_input("an interger:")
        f = raw_input("another interger:")
        if d == 'quit':break
        print d / f
except ZeroDivisionError, e:
    print "Not 0" e
except ValueError:
    print "Not string"
except:
    print "unkown error"
```

KeyError、IndexError、ZeroDivisionError、IOError

try-except-else-finally语句

```
try:
    try_suite
except Exception1:
    suites_exception1
...
else:
    else_suite
finally:
    finally_suite
```

```
try:
    try:
        try_suite
    except
    except
    else
finally
```

自定义异常

raise语句可显示出发异常

```
raise ValueError,"string..."
```

```
def sel(seq1,seq2):
    if not seq1 or not seq2:
        raise ValueError,"seq not must be empty."
    print [(x,y) for x in seq1 for y in seq2]
```

python 异常是内置经典类Exception的子类实例

Baseexceptiong Exception

ArithmeticError

OverflowError,ZeroDivisionError,FloatingPointError

LookupError

IndexError,KeyError

EnvironmentError

IOError,OSError,WindowsError

标准异常类

AssertionError

断言语句是不

AttributeError

属性引用或赋值失效

FloatingPointError

浮点数运算是不

IOError

I/O操作失败

ImportError

找不到模块

IndentaionError

缩进错误

IndexError

索引超出范围

KeyError

键不在映射中

KeyboardInterrupt

用户按了中断键 (ctrl+c、delete、ctrl+Break)

MemoryError

运算耗尽内存

NameError

应用了一个不存在的变量名

OSError

由os模块函数引发的异常，用来指示平台相关错误

OverflowError

整数运算结果太大导致溢出

SyntaxError

语法错误

SystemError

python 本身或某些扩展模块中的内部错误

TypeError

对某对象执行了不支持的操作

UnboundLocalError

引用未绑定值得本地变量

UnicodeError

在Unicode的字符串之间进行转化时发生的错误

ValueError

不适当的值

WindosError

模块os中的函数引发的异常，用来指示与windows相关的错误

ZeroDivisionError

除数为0

自定义异常类

assert 实现：手动触发异常

```
assert codition[,expression]
```

```
if __debug__:
```

```
    if not _codition:
```

```
        raise AsserttionError ,expression
```