

PCS112 异常

概述

使用 `try...except` 可以捕捉各种异常类型。Python 中有很多内置的异常类型，具体可以参见下面的资料。

- Python 异常类型: <http://www.python.org/doc/current/lib/module-exceptions.html>
精巧地址: <http://bit.ly/r17AI>

应用

捕捉异常

首先举个小例子。

```
try:
    code_block()
    # ...
except SomeException, e:
    do_something_with_exception(e)
except (Exception1, Exception2), e:
    do_something_with_exception(e)
except:
    do_something_with_other_exceptions()
else:
    do_something_when_success()
```

```
finally:
    do_something()
```

try...except 可以带一个 else 子句，该子句只能出现在所有 except 子句之后。当 try 语句没有抛出异常时，需要执行一些代码，可以使用这个子句。无论 try 子句中有没有发生异常，finally 子句都一定会被执行。如果发生异常，在 finally 子句执行完后它会被重新抛出。try 子句经由 break 或 return 退出也一样会执行 finally 子句。

抛出异常

使用 raise 来抛出异常。

```
try:
    raise NameError, 'Hi There'
except NameError, a:
    print 'An exception flew by!'
    print type(a)
```

raise 函式的第一个参数是异常名，第二个是这个异常的实例，它存储在 instance.args 的参数中。和 except NameError, a: 中的第二个参数意思差不多。

自定义异常

异常类中可以定义任何其他类中可以定义的东西，但是通常为了保持简单，只在其中加入几个属性信息，以供异常处理句柄提取。

```
class MyError(Exception):
    def __init__(self, value):
        self.value = value
    def __str__(self):
        return repr(self.value)
try:
    raise MyError(2*2)
except MyError, e:
    print 'My exception occurred, value:', e.value
```

如果一个新创建的模块中需要抛出几种不同的错误时，一个通常的做法是为该模块定义一个异常基类，然后针对不同的错误类型派生出对应的异常子类。

小结

本文介绍了 Python 中如何捕捉异常、抛出异常和如何自定义异常来方便编程。在下面的参考链接中可以找到更多关于异常的知识。

- 异常处理: <http://wiki.woodpecker.org.cn/moin/ObpLovelyPython/LpyQLearn-3-process#id2>
精巧地址: <http://bit.ly/1o9j0x>
- try ... except ... finally / raise:
<http://wiki.woodpecker.org.cn/moin/PyAbsolutelyZipManual#head-263e21b75a15eebdaa6574be53cfbddeb7a1231>
精巧地址: <http://bit.ly/1BfHoi>
- Python 学习笔记: <http://www.blogjava.net/JAVA-HE/archive/2008/04/10/191867.html>
精巧地址: <http://bit.ly/JCokE>