Monday, April 30, 2018

Exceptions

Content

Checked Exceptions

- Occurs at compile time. (Syntax)

例如随便打一个

>>> a

会标红

Unchecked Exceptions

- Occurs at the time your program is executed. (Logic/Bad Access)

例如 ArrayIndexOutOfBoundsException

会在compile的时候报错

如何引发异常：

- raise语句

Syntax: raise + 一个Exception的子类/实例

例如 raise ArithmeticError

如何创建自定义异常类：

- 和创建其他类一样，但必须直接或间接继承Exception

例如 class MyException(Exception):pass

捕获异常：

- try/except 语句

例如：

try: …

except Exception: …

异常从函数向外传播到调用函数的地方。如果在这里也没有被捕获，异常将想程序的顶层传播。

将异常捕获后，可以用raise继续将它往上传播。

可以用raise…from…语句来提供自己的异常上下文，也可以用None来禁用上下文

例如： raise ValueError from None

捕获多个异常时：

可以用多个expect，也可以用一个except将所有exceptions放在( )里，但这样引发的将会是一样的处理。

如果不处理函数中的异常，它会向上一直传播知道主程序（全局作用域）

捕获异常对象：

例如 except Exception as e: …

直接写 except: … 可以一次捕获所有的异常（但不建议这样做）。更好的方法是写：

except Exception as e: …

没有异常：

在except: … 后加一个 else: … 是没有引发异常时执行的语句。

Finally：

不管try中发生什么，都会引发finally

异常之禅：

有时候可以用条件语句来处理异常情况，但是这样可读性没这么好，

try/except的效率更高，也可以用来检查对象是否有特定的属性。

因此尽可能用try/except代替if/else

警告：

使用warnings module 中的 warn function

from warnings import warn

例如

>> warn(‘ Stop!’)

UserWarning: Stop

warn(‘Stop')

使用warnings module 中的filterwarnings 来抑制发出特定的警告，并指定采取的措施。如’error’ 或 ’ignore’

比如

>> filterwarnings('error')

>> warn(‘Stop’)

warn(‘Stop')

UserWarning: Stop

>> filterwarnings('ignore')

>> warn(‘Stop’)

不会输出东西