1. **Python3 错误和异常**

**第一部分**

**一、基本概念**

Python有两种错误很容易辨认：语法错误和异常。Python 的语法错误或者称之为解析错误。

**1、1语法错误的例子：**

#coding=utf-8

while True

print("OK")

break

上面的代码少些：或者把True写成true,再比如代码的缩进错了。

**1.2 异常**

即便Python程序的语法是正确的，在运行它的时候，也有可能发生错误。运行期检测到的错误被称为异常。

大多数的异常都不会被程序处理，都以错误信息的形式展现在这里:

>>> 20 \* (1/0)

Traceback (most recent call last):File "<stdin>", line 1, in ?ZeroDivisionError: division by zero

>>> 8+ spam\*3

Traceback (most recent call last):File "<stdin>", line 1, in ?NameError: name 'spam' is not defined

>>> '4' + 2

Traceback (most recent call last):File "<stdin>", line 1, in ?TypeError: Can't convert 'int' object to str implicitly

异常以不同的类型出现，这些类型都作为信息的一部分打印出来: 例子中的类型有 ZeroDivisionError，NameError 和 TypeError。

错误信息的前面部分显示了异常发生的上下文，并以调用栈的形式显示具体信息。

**二、异常处理**

**2.1、语法**

try:

#可能出现异常的代码块

#处理异常

except 异常类型

异常类型语句

#可选语句，try无异常，才会执行else

else:

#可选语句，不管有没有异常，都会执行finally

finally:

首先，先执行try子句块（在关键字try和关键字except之间的语句块）

 如果没有异常发生，忽略except子句，try子句执行后结束，else语句和finally语句都会执行。

 如果在执行try子句块的过程中发生了异常，那么try子句块余下的部分将被忽略。如果异常的类型和 except 之后的名称相符，那么对应的except子句将被执行。finally语句会执行，再接着往下执行代码。

 如果一个异常没有与任何的except匹配，那么这个异常将会传递给上层的try中。

例子1：

#coding=utf-8

#使用try语句捕获异常

try:

#可能出现异常的代码块

x = int(input("Please enter a number: "))

print(x)

#处理异常

except ValueError:

print("Oops! That was no valid number,Try again")

#可选语句，try无异常，才会执行else

else:

print("else")

#可选语句，不管有没有异常，都会执行finally

finally:

print("finally")

**2.2 获取错误类型**

**import** sys  
**try**:  
 s=1/0  
**except**:  
 print(sys.exc\_info()[0])

输出结果就是错误类型

**第二部分**

一个 try 语句可能包含多个except子句，分别来处理不同的特定的异常。最多只有一个分支会被执行。处理程序将只针对对应的try子句中的异常进行处理，而不是其他的 try 的处理程序中的异常。

一个except子句可以同时处理多个异常，这些异常将被放在一个括号里成为一个元组，例如:

except (RuntimeError, TypeError, NameError):pass

最后一个except子句可以忽略异常的名称，它将被当作通配符使用。你可以使用这种方法打印一个错误信息。

例子2：

#coding=utf-8

import sys

#使用try语句捕获异常

try:

#可能出现异常的代码块

x = int(input("Please enter a number: "))

print(10/x)

print(x+'20')

#处理异 常

except ValueError:

print("输入不是一个有效的数据:")

except ZeroDivisionError:

print("除数不能为0")

except:

print("Unexpected error:", sys.exc\_info()[0])

pass

#可选语句，try无异常，才会执行else

else:

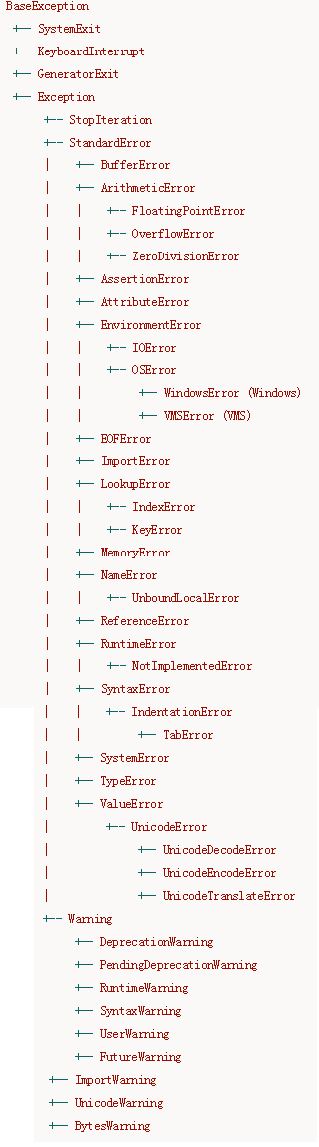
print("else")

#可选语句，不管有没有异常，都会执行finally

finally:

print("finally")

**三、Python中异常的层次结构**



想要捕获所有的异常，可以直接捕获 Exception 即可：

try:

...

except Exception as e:

...

log('Reason:', e)

这个将会捕获除了 SystemExit 、 KeyboardInterrupt 和 GeneratorExit

之外的所有异常。 如果你还想捕获这三个异常，将 Exception 改成 BaseException 即可