



第一章 Python基础

—— 第12节 Python中的异常处理

讲师：张 涛

1. 什么是异常
2. 异常处理
3. 抛出异常

1. 什么是异常

- 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。**
- 如果异常对象并未被处理或捕获，程序就会用所谓的**回溯**（ traceback，一种错误信息）**终止执行**
- **Python的一些内建异常：**

Exception	常规错误的基类	AttributeError	对象没有这个属性
IOError	输入/输出操作失败	IndexError	序列中没有此索引(index)
KeyError	映射中没有这个键	NameError	未声明/初始化对象 (没有属性)
SyntaxError	Python 语法错误	TypeError	对类型无效的操作
ValueError	传入无效的参数	ZeroDivisionError	除(或取模)零 (所有数据类型)

2. 异常处理

try语句按照如下方式工作；

- 首先，执行try子句（在关键字try和关键字except之间的语句）
- 如果没有异常发生，忽略except子句，try子句执行后结束。
- 如果在执行try子句的过程中发生了异常，那么try子句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。最后执行 try 语句之后的代码。
- 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。
- 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。或者一个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

```
.....  
try:  
    ....  
    可能会有异常抛出的代码  
    ....  
except 异常类名:  
    异常处理代码  
.....
```

3. 抛出异常

- Python 使用 **raise** 语句抛出一个指定的异常
- **raise** 唯一的一个参数指定了要被抛出的异常。它必须是一个异常的实例或者是异常的类（也就是 **Exception** 的子类）。
- 如果你只想知道这是否抛出了一个异常，并不想去处理它，那么一个简单的 **raise** 语句就可以再次把它抛出。

1. 什么是异常
2. 异常处理
3. 抛出异常

- 跟着老师自己动手练一练

EDU

CSDN学院 IT实战派

