



Python 程序设计基础

Python Programming



如何处理异常

➤ 使用 if 语句处理异常。

```
number1, number2 = eval(input("请输入以逗号分隔的两个整数: "))
if number2 != 0:
    print(number1, '/', number2, '=', number1 / number2)
else:
    print("除数不能为0!")
```

请输入以逗号分隔的两个整数: 1,2
1 / 2 = 0.5

请输入以逗号分隔的两个整数: 1,0
除数不能为0!

- 使用 if 语句来检查除数 number2 是否为 0，若不为 0 输出除法结果，否则输出错误信息，不会产生异常。



如何处理异常

➡ 使用 try-except 语句处理异常。

■ try-except 语句的语法如下：

try:

< 语句块 1>

except < 异常类型>:

< 语句块 2>

< 语句块 1> 是程序正常执行的内容，但包含了可能产生异常的代码。当异常出现时，< 语句块 1> 中产生异常的语句下面的剩余代码被跳过。若该异常匹配 < 异常类型>，那么执行 < 语句块 2>，< 语句块 2> 是处理异常的代码。若该异常不匹配 < 异常类型>，那么这是一个未处理的异常，程序终止运行。



如何处理异常

```
number1, number2 = eval(input("请输入以逗号分隔的两个整数: "))
try:
    print(number1, '/', number2, '=', number1 / number2)
except ZeroDivisionError:
    print("除数不能为0!")
```

请输入以逗号分隔的两个整数: 1,2
1 / 2 = 0.5

请输入以逗号分隔的两个整数: 1,0
除数不能为0!

- 若除数 `number2` 为 0，抛出 `ZeroDivisionError` 异常，若异常被捕获，处理异常，这里仅是输出错误信息。若没有抛出异常，则输出除法结果。若抛出的异常没有被捕获，则程序终止运行。
- `try-except` 语句会消耗更多的资源。对于简单的错误检查，应该使用 `if` 语句。