



# Python 程序设计基础

## Python Programming



## 进一步讨论异常处理

➡ 一个 try 语句可以有多个 except 语句来处理不同的异常，还可以有可选的 else 语句和 finally 语句

try:

< 语句块 1>

except < 异常类型 1>

< 语句块 2>

...

except < 异常类型 n>

< 语句块 n+1>

except:

< 语句块 n+2>

else:

当一个异常出现时，它会被顺序检查是否匹配 try 语句后的 except 语句中的异常类型，若匹配 < 异常类型 1>，那么匹配该异常的 < 语句块 2> 将被执行，剩下的 except 语句将会被忽略。依此类推。若都不匹配 except 语句中的异常类型，则最后一个 except 语句的 < 语句块 n+2> 将被执行

可选的 else 语句，若没有异常发生，执行完 < 语句块 1> 后，将会执行 else 语句的 < 语句块 n+3>。

可选的 finally 语句，不管是否发生异常，都会执行 finally 语句的 < 语句块 n+4>。



## 进一步讨论异常处理

```
def main():  
    try:  
        number1, number2 = eval(input("请输入以逗号分隔的两个整数: "))  
        print(number1, '/', number2, '=', number1 / number2)  
    except ZeroDivisionError:  
        print("除数为0!")  
    except SyntaxError:  
        print("输入整数时缺少逗号分隔!")  
    except:  
        print("其他输入错误!")  
    else:  
        print("没有异常发生, 执行else语句!")  
    finally:  
        print("执行finally语句!")  
  
main()
```

请输入以逗号分隔的两个整数: 3,4  
3 / 4 = 0.75  
没有异常发生, 执行else语句  
执行finally语句

请输入以逗号分隔的两个整数: 2,0  
除数为0!  
执行finally语句

请输入以逗号分隔的两个整数: 2 3  
输入整数时缺少逗号分隔  
执行finally语句

请输入以逗号分隔的两个整数: 1,b  
其他输入错误  
执行finally语句



## 进一步讨论异常处理

- 当有多个 `except` 语句时，`except` 语句的书写顺序非常重要，因为 Python 是按这个顺序来寻找异常处理的。如果一个异常父类的 `except` 语句出现在异常子类的 `except` 语句之前，那么这个异常子类的 `except` 语句将永远不会被执行。

```
try:
    number1, number2 = eval(input("请输入以逗号分隔的两个整数: "))
    print(number1, '/', number2, '=', number1 / number2)
except Exception:
    print("错误!")
except ZeroDivisionError:
    print("除数为0!")
except SyntaxError:
    print("输入整数时缺少逗号分隔!")
```

`except ZeroDivisionError:` 和 `except SyntaxError:` 永远不会执行。因为 `ZeroDivisionError` 和 `SyntaxError` 都是 `Exception` 的子类。

请输入以逗号分隔的两个整数: 1,0  
错误!



## 进一步讨论异常处理

- 还可以使用 `except:` 来捕获所有异常，但通常不建议这样做。

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

请输入以逗号分隔的两个整数: 1,0  
错误!



## 进一步讨论异常处理

- 除了用多个 `except` 语句外，还可以在一个 `except` 语句的后面放多个异常类型。但通常不建议这样做。
- 注意： `except` 语句的后面如果有多个异常类型，则一定要用圆括号括起来。

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

请输入以逗号分隔的两个整数: 1,0  
错误!