

警报！异常来袭





1. 探索新知

1.1

什么是异常

异常

程序在运行中，如果python遇到错误，会停止程序的执行，并提示一些错误信息，这就是异常



抛出异常

程序停止执行，并提示错误信息的动作 称为抛出异常



下面给大家介绍一些常见的异常

1.2

常见异常类型

SyntaxError 语法错误

1. `SyntaxError: invalid syntax` 无效语法

- 遗漏了标点符号，比如漏了冒号

```
>>> for i in range(3)
SyntaxError: invalid syntax
```

- 关键字拼写错误或遗漏

```
>>> for i range(3):
SyntaxError: invalid syntax
```

- 关键字拼写错误或遗漏

```
>>> in = 3
SyntaxError: invalid syntax
```

2. SyntaxError: invalid character in identifier

标识符中有无效字符

- 代码中出现了无法识别的字符，检查是否有多余字符或者中文字符

```
>>> for i in range (3) :
SyntaxError: invalid character in identifier
```

```
>>> for i in range(3) :
SyntaxError: invalid character in identifier
```

3. SyntaxError: EOL while scanning string literal

检查到不完整的字符串

- 这种情况通常是因为遗漏了字符串两边的引号，或者混用了引号（即字符串两边的引号不统一）

```
>>> print('nihao')
SyntaxError: EOL while scanning string literal
```

```
>>> print('nihao"')
SyntaxError: EOL while scanning string literal
```

NameError 变量名错误

- 访问一个还没有定义或初始化的变量、函数、类或其他对象

```
>>> a += 1
Traceback (most recent call last):
  File "<pyshell#20>", line 1, in <module>
    a += 1
NameError: name 'a' is not defined
```

```
>>> f(5)
Traceback (most recent call last):
  File "<pyshell#21>", line 1, in <module>
    f(5)
NameError: name 'f' is not defined
```

TypeError 类型错误

- 在使用函数或操作时，对象的类型不正确，导致无法进行相应的操作或函数调用

```
>>> a = 1
>>> b = "hello"
>>> print(a+b)
Traceback (most recent call last):
  File "<pyshell#2>", line 1, in <module>
    print(a+b)
TypeError: unsupported operand type(s) for +: 'int' and 'str'
```

“+”不支持不同类型的数据进行运算

```
>>> def f(a,b):
    return a+b

>>> f(3)
Traceback (most recent call last):
  File "<pyshell#9>", line 1, in <module>
    f(3)
TypeError: f() missing 1 required positional argument: 'b'
```

函数缺少一个必须的位置参数

IndexError 索引错误

- 当尝试访问列表或元组中不存在的元素时，会抛出indexError异常

```
>>> a = [1, 2, 3, 4]
>>> print(a[4])
Traceback (most recent call last):
  File "<pyshell#11>", line 1, in <module>
    print(a[4])
IndexError: list index out of range
```

列表索引超出范围

ValueError 值错误

- 当你的操作值本身合理，但其具体数值不满足函数或操作的要求时，ValueError就会出现

```
>>> float("z3")
Traceback (most recent call last):
  File "<pyshell#0>", line 1, in <module>
    float("z3")
ValueError: could not convert string to float: 'z3'
```

无法将字符串转换为浮点数：“z3”

KeyError 键错误

- 当我们以一个不存在的键访问字典中的元素时，就会抛出KeyError异常

```
>>> a = {'李明': '黎明', '柯艾': '可爱'}
>>> a['遛遛']
Traceback (most recent call last):
  File "<pyshell#4>", line 1, in <module>
    a['遛遛']
KeyError: '遛遛'
```

键错误

1.2

常见异常类型

异常类型	
SyntaxError	语法错误
TypeError	类型错误
ValueError	值错误
NameError	变量名错误
IndexError	索引错误
KeyError	键错误
ZeroDivisionError	除数为0错误
FileNotFoundError	表示在尝试打开或操作文件时找不到指定的文件
ImportError	当尝试导入不存在的模块或包时引发
AttributeError	当尝试访问对象不存在的属性时引发
KeyboardInterrupt	当用户中断程序执行时引发

1.3

异常处理

程序在运行过程中遇到错误往往会终止执行。如果遇到错误不想终止程序，则需要对异常进行捕获并处理

1. try-except语句

```
try:  
    # <可能抛出异常的代码>  
except:  
    # <处理异常的代码>
```

```
try:  
    num = int(input('请输入数字: '))  
except:  
    print('请输入正确的数字')
```

请输入数字:test
请输入正确的数字

1.3

异常处理

2. 捕获指定异常

```
try:  
    # <可能抛出异常的代码>  
except 异常1:  
    # <处理异常的代码>
```

```
try:  
    num_str = "Hello" # 这是一个字符串, 不是一个有效的整数  
    # 尝试将字符串转换为整数, 可能会抛出 ValueError 异常  
    num = int(num_str)  
except ValueError:  
    print("发生了值错误 (ValueError), 无法将字符串转换为整数")
```

3. 捕获多种异常

在python中, 一个try子句也可以对应多个except子句, 每个except子句分别用来处理不同的异常

```
try:  
    # <可能抛出异常的代码>  
except 异常1:  
    # <处理异常的代码>  
except 异常2:  
    # <处理异常的代码>
```

```
try:  
    x = 1/0  
except ZeroDivisionError:  
    print("捕获除数为0错误")  
except TypeError:  
    print("捕获类型错误")
```

```
try:  
    # <可能抛出异常的代码>  
except 异常1:  
    # <处理异常的代码>  
except:  
    # <处理除异常1外其他异常>
```

```
try:  
    x = a/b  
except ZeroDivisionError:  
    print("捕获除数为0错误")  
except:  
    print("捕获其他错误")
```

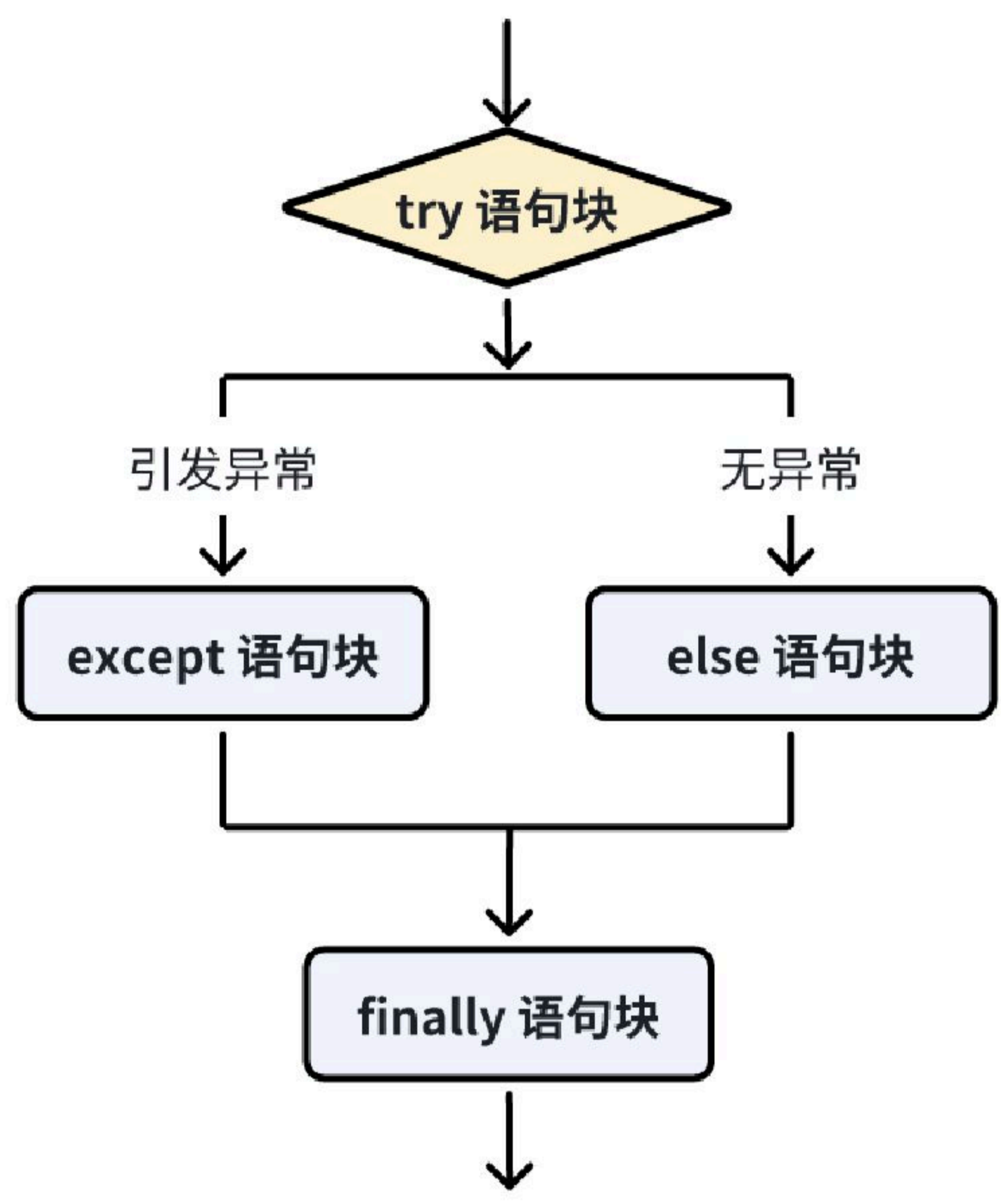
不指定异常类型, 默认捕获所有异常

1.3

异常处理

4. try-except-else-finally

```
try:  
    # <可能抛出异常的代码>  
except:  
    # <处理异常的代码>  
else:  
    # <无异常时执行的代码>  
finally:  
    # <有无异常都执行的代码>
```



示例，输入数字，如果输入的不是数字，则捕获异常并处理

```
try:  
    num = int(input('请输入数字: '))  
except:  
    print('请输入正确的数字')  
else:  
    print('num =', num)  
finally:  
    print('程序测试结束')
```

显示结果，如果输入非数字，会提示“请输入正确的数字”

---测试1---

请输入数字:test
请输入正确的数字
程序测试结束

---测试2---

请输入数字:83
num = 83
程序测试结束

5. 获取异常信息

```
try:
    # <可能抛出异常的代码>
except:
    # <处理异常的代码>
else:
    # <无异常时执行的代码>
finally:
    # <有无异常都执行的代码>
```

```
try:
    a += 1
except NameError as s: # 处理指定异常
    print(f"出现异常，进行处理，异常内容：{s}")
```

```
出现异常，进行处理，异常内容：name 'a' is not defined
>>>
```

```
try:
    a += 1
except Exception as e: # 处理其他所有异常
    print(f"出现异常，进行处理，异常内容：{e}")
```

```
出现异常，进行处理，异常内容：name 'a' is not defined
>>>
```



2. 强化练习

1. 关于Python在处理程序异常时，下列说法不正确的是？（ ）

- A. 每一个try模块只能设定一个except模块
- B. 执行except模块部分，可以让程序继续运行
- C. 程序有错误时执行except中的代码，没有错误时执行try中的代码
- D. 异常处理可以弥补程序漏洞，使得程序在一些情况下不会终止运行

2. 在try...except...else...finally异常处理机制中，如果发生异常，则完全不会被执行的代码块是？（ ）

- A. try代码块
- B. except代码块
- C. else代码块
- D. finally代码块

3. 异常处理的作用是不想让程序终止，如果出错了需要特殊处理。下列实现异常处理的语句是？（ ）

- A. if/else
- B. try/except
- C. Warning
- D. Error



2. 强化练习

4. 语句 `float('something')` 抛出的异常名称为? ()

- A. ValueError
- B. ImportError
- C. IndexError
- D. FileNotFoundError

5. 程序如下: 我们输入内容“z2”, 控制台打印的结果是? ()

```
try:
    number = int(input("请输入数字: "))
    print("number:", number)
except Exception as e:
    print("打印异常详情信息:", e)
else:
    print("else")
finally:
    print("finally")
print("end")
```

- A. number: 2
else
finally
end
- B. number: 2
end
- C. 打印异常详情信息: invalid literal for int() with base 10:
finally
end
- D. 打印异常详情信息: invalid literal for int() with base 10:
else
finally
end

3. 术语箱

try 试图, 尝试

except 除...之外

finally 终于, 最后, 最终

SyntaxError 语法错误

4. 课后挑战

编写一个 Python 程序, 要求实现以下功能:

1. 提示用户输入一个数字作为除数 (被除数固定为65), 并尝试计算65除以除数的结果并输出
2. 如果用户输入的是非数字 (如字母、特殊字符等), 则捕获异常并提示“输入的内容有误, 请输入一个数字”
3. 如果用户输入的是数字但为 0 (即除数为 0), 也捕获异常并提示用户“不能输入0作为除数”
4. 如果有其他的异常, 则捕获并输出异常信息

