

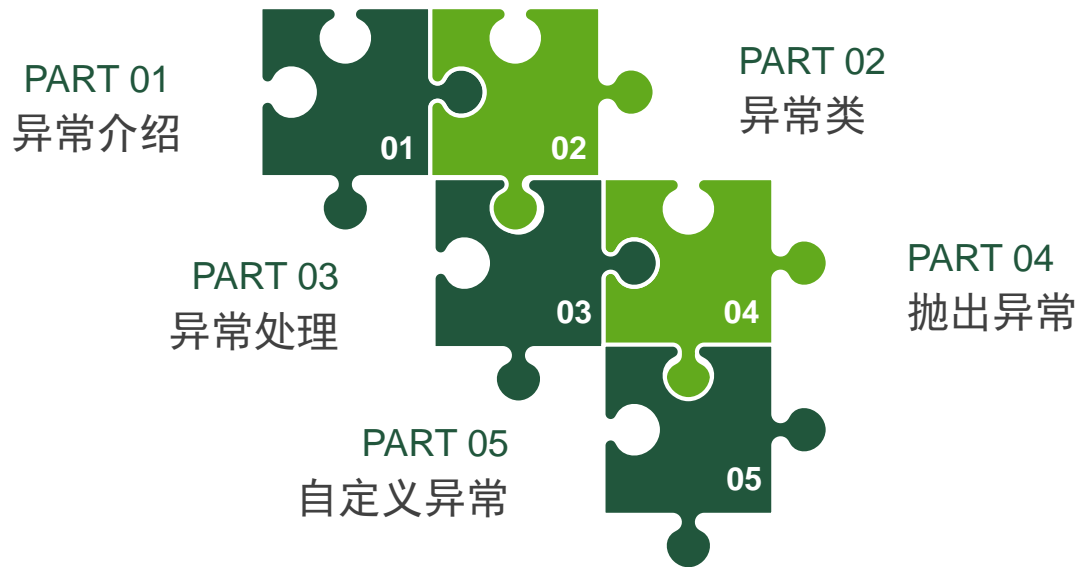


## 第9章 异常

主讲人：刘潇潇

# 目录

## Contents



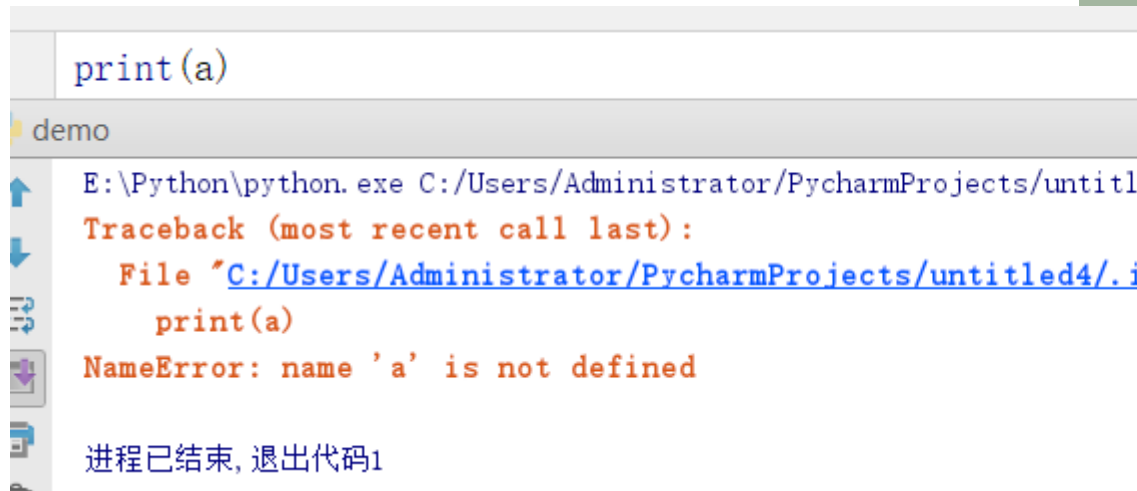


# 异常介绍

在Python中，程序在执行的过程中产生的错误称为异常，比如列表索引越界、打开不存在的文件等。

## ● 异常介绍

在Python中，程序在执行的过程中产生的错误称为异常，比如列表索引越界、打开不存在的文件等。



The screenshot shows a code editor with the file `demo` containing the line `print(a)`. The console output shows the command `E:\Python\python.exe C:/Users/Administrator/PycharmProjects/untitled4/.i` followed by a `Traceback (most recent call last):` and the error `File "C:/Users/Administrator/PycharmProjects/untitled4/.i" line 1, in <module> print(a) NameError: name 'a' is not defined`. At the bottom, a status bar indicates `进程已结束, 退出代码1`.

```
print(a)
```

demo

E:\Python\python.exe C:/Users/Administrator/PycharmProjects/untitled4/.i

Traceback (most recent call last):

File "C:/Users/Administrator/PycharmProjects/untitled4/.i" line 1, in <module>

print(a)

NameError: name 'a' is not defined

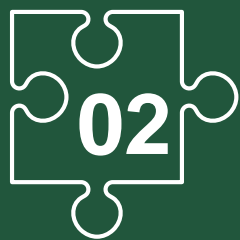
进程已结束, 退出代码1

## ● 异常介绍

```
NameError: name 'a' is not defined
```

```
FileNotFoundError: [Errno 2] No such file or directory: '123.txt'
```

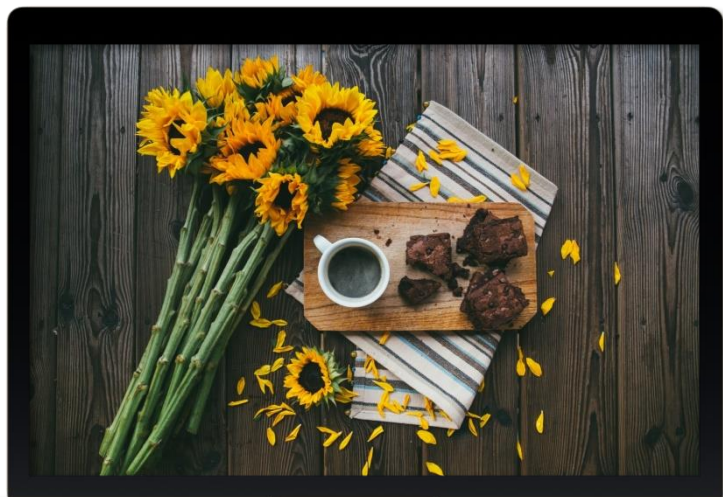
第1个异常的类型为NameError（名称），描述信息为a没有定义；  
第2个异常为FileNotFoundError，描述信息为没有找到123.txt文件



异常类



## 异常类



- 所有异常都是基类Exception的成员，它们都定义在exceptions模块中。
- 如果这个异常对象没有进行处理和捕捉，程序就会用所谓的回溯（traceback，一种错误信息）终止执行，这些信息包括错误的名称（例如NameError）、原因和错误发生的行号。

## NameError

尝试访问一个未声明的变量，会引发NameError

```
print(foo)
```

demo

```
E:\Python\python.exe C:/Users/Administrator/PycharmProjects/untitled4/.idea/demo.py
```

```
Traceback (most recent call last):
```

```
File "C:/Users/Administrator/PycharmProjects/untitled4/.idea/demo.py", line 1, in <module>
```

```
    print(foo)
```

```
NameError: name 'foo' is not defined
```



## ZeroDivisionError

当除数为零的时候，会引发ZeroDivisionError异常。

```
x = 1/0
```

emo

```
E:\Python\python.exe C:/Users/Administrator/PycharmProjects/untitled4/.idea/demo.py
```

```
Traceback (most recent call last):
```

```
File "C:/Users/Administrator/PycharmProjects/untitled4/.idea/demo.py", line 1, in <module>
```

```
x = 1/0
```

```
ZeroDivisionError: division by zero
```

## SyntaxError

当解释器发现语法错误时，会引发SyntaxError异常

```
for i in list_
print(i)
```

demo

```
E:\Python\python.exe C:/Users/Administrator/PycharmProjec
File "C:/Users/Administrator/PycharmProjects/unt:
    for i in list
              ^
SyntaxError: invalid syntax
```

## IndexError

当使用序列中不存在的索引时，会引发IndexError异常

```
num_list = [0, 1, 2]  
print(num_list[3])
```

emo

```
E:\Python\python.exe C:/Users/Administrator/PycharmPr
```

```
Traceback (most recent call last):
```

```
File "C:/Users/Administrator/PycharmProjects/1
```

```
    print(num_list[3])
```

```
IndexError: list index out of range
```

## KeyError

当使用映射中不存在的键时，会引发KeyError异常。

```
fruits_color = {'potato': 'yellow', 'tomato': 'red'}  
print(fruits_color['photo'])
```

demo

```
E:\Python\python.exe C:/Users/Administrator/PycharmProjects/untitled4/.idea/c  
Traceback (most recent call last):  
  File "C:/Users/Administrator/PycharmProjects/untitled4/.idea/demo.  
    print(fruits_color['photo'])  
KeyError: 'photo'
```

## 捕获简单异常

当使用映射中不存在的键时，会引发KeyError异常。

```
fruits_color = {'potato': 'yellow', 'tomato': 'red'}  
print(fruits_color['photo'])
```

demo

```
E:\Python\python.exe C:/Users/Administrator/PycharmProjects/untitled4/.idea/c  
Traceback (most recent call last):  
  File "C:/Users/Administrator/PycharmProjects/untitled4/.idea/demo.  
    print(fruits_color['photo'])  
KeyError: 'photo'
```

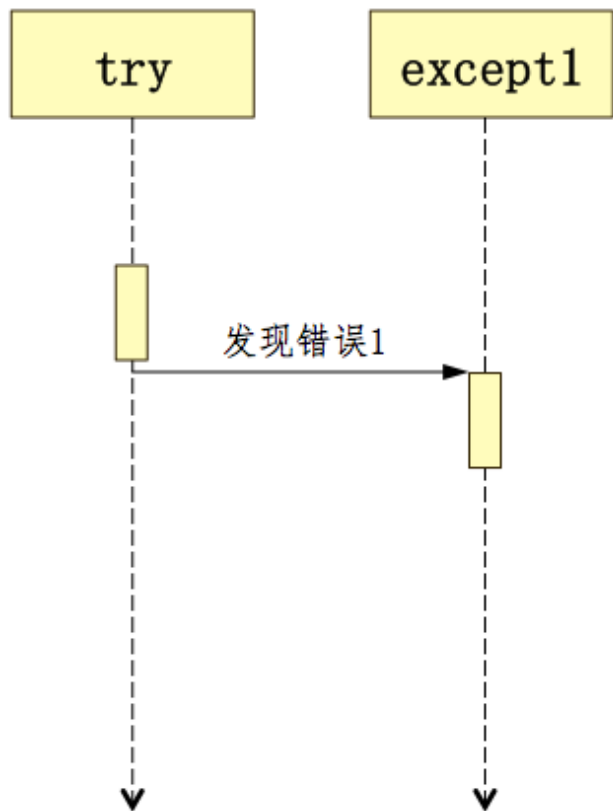


异常处理





## 捕获简单异常



try-except语句定义了监控异常的一段代码，并提供了处理异常的机制。

```
try:  
    # 语句块  
except:  
    # 异常处理代码
```



## 捕获简单异常

try-except语句定义了监控异常的一段代码，并提供了处理异常的机制。

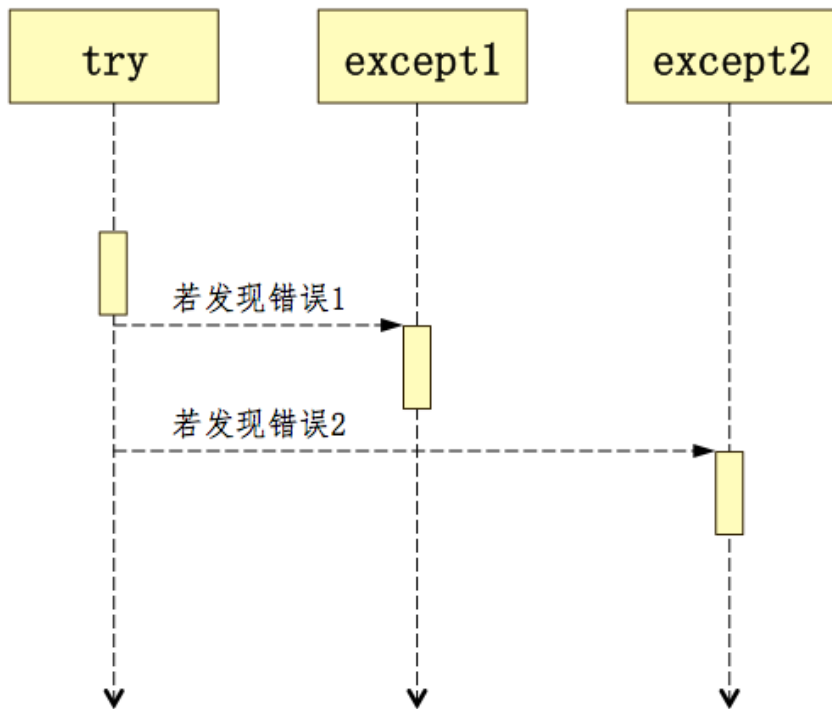
```
try:
    f = open("test")
except:
    print('File not found, pls input right path or file!')
```

demo

```
E:\Python\python.exe C:/Users/Administrator/PycharmProjects/untitled4/.idea/demo.py
File not found, pls input right path or file!
```



## 捕获多个异常



处理多个异常的try-except语句格式如下：

```
try:  
    # 语句块  
except 异常名称1:  
    # 异常处理代码1  
except 异常名称2:  
    # 异常处理代码  
...
```



## 捕获简单异常

```
try:
    num1 = input('请输入第1个数字: ')
    num2 = input('请输入第2个数字: ')
    print(int(num1)/int(num2))
except ZeroDivisionError:
    print('第2个数不能为0! ')
except ValueError:
    print('只能输入数字!')
```

mo

E:\Python\python.exe C:/Users/Administrator/PycharmProjects/untitled4/.idea

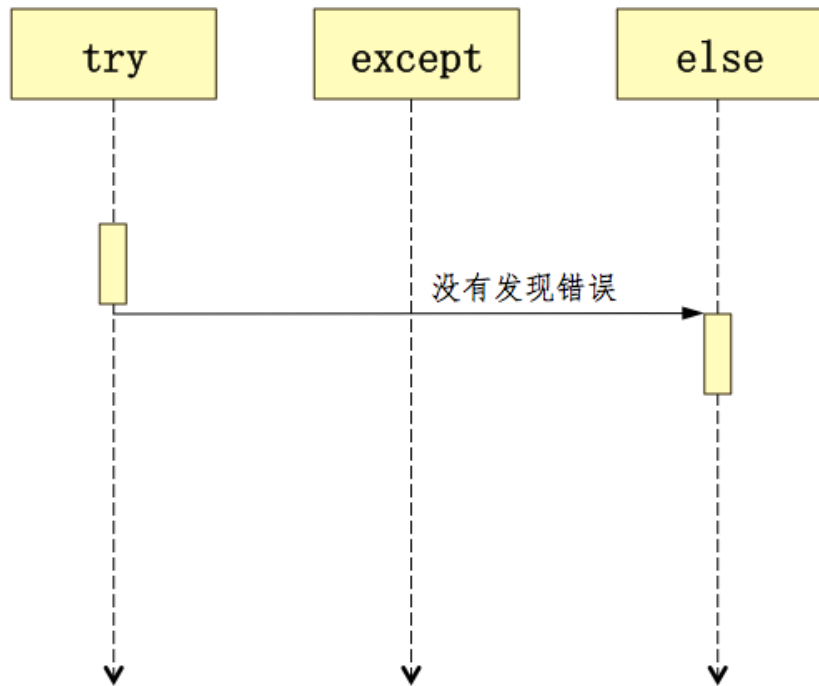
请输入第1个数字: a

请输入第2个数字: 2

只能输入数字!



## 未捕获到异常



如果try语句没有捕获到任何的错误信息，就不再执行任何except语句，而是会执行else语句。



## 未捕获到异常

```
try:
    num1 = input('请输入第1个数字: ')
    num2 = input('请输入第2个数字: ')
    result = int(num1)/int(num2)
except ZeroDivisionError:
    print('第2个数不能为0! ')
except ValueError:
    print('只能输入数字! ')
else:
    print('结果为: ', int(result))
```

emo

E:\Python\python.exe C:/Users/Administrator/PycharmProjects/untit

请输入第1个数字: 4

请输入第2个数字: 5

结果为: 0





抛出异常

## raise语句

### 使用类名引发异常

当raise语句指定异常类名时，  
会创建该类的实例对象，然后  
引发异常。

```
raise  
IndexError
```

```
Traceback (most recent call last):  
  File "D:/异常.py", line 1, in <module>  
    raise IndexError  
IndexError
```

## raise语句

使用异常类的实例引发异常

```
index = IndexError()  
raise index
```

Traceback (most recent call last):  
 File "D:/异常.py", line 2, in <module>  
 raise index  
IndexError

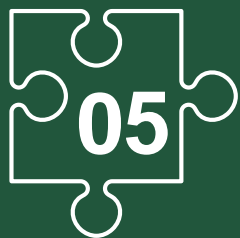
## raise语句

### 传递异常

不带任何参数的raise语句，可以再次引发刚刚发生过的异常，作用就是向外传递异常。

```
try:  
    raise IndexError  
except:  
    print("出错了")  
    raise
```

出错了  
File "D:/异常.py", line 2, in <module>  
 raise IndexError  
IndexError



## 自定义异常

## 自定义异常



- 创建一个继承Exception类的子类，就是自定义异常类。
- 当遇到自己设定的错误时，使用raise语句抛出自定义的异常。





Thank you