

# 异常

异常：一个事件，这个事件会在程序中发生，影响程序正常执行  
异常是Python对象，表示一个错误  
当Python发生异常我们需要捕获异常，否则程序终止运行

- 常见的异常

1. `AssertionError`：断言失败
2. `AttributeError`：尝试访问对象的属性或方法时报错
3. `IndexError`：索引越界异常
4. `KeyError`：键异常
5. `NameError`：名字错误
6. `OverflowError`：超出最大值异常
7. `SyntaxError`：语法错误
8. `TypeError`：类型错误
9. `ValueError`：值错误
10. `TabError`：空格和tab不能混用
11. `ZeroDivisionError`：除数为0异常
12. `StopIterationError`：停止迭代
13. `RecursionError`：递归异常
14. `ModuleNotFoundError`：导入模块异常
15. `KeyboardInterrupt`：用户输入中断 ( ctrl C )

- 异常处理

## 1. `try -- except`

语法结构：

```
try:
    代码（可能会出现异常）
except 异常类型 as 变量:
    print('')
```

eg:

```
try:
    f=open('xixi.txt')
    f.read()
    f.close()
except FileNotFoundError as a:
    print('您查找的文件不在哦~~，抱歉~~~~',a)
```

1. 一旦try中有异常，该行后面的代码都不执行
2. `as 变量` 可以省略：  
变量：异常信息
3. 异常类型也可以省略：表示获取所有异常 `Exception`
4. 异常有继承关系，父类异常可以捕获到子类异常

## 2. `try except except`

一个try可以和多个except使用

1. `except`中的代码，只执行一个`except`
2. 可以书写多个`except`，如果其中有一个不写异常类型（默认），它必须放在最后
3. 如果有多个`except`，范围小的放在更靠前的位置
4. 一个`except`后可以书写多个异常，多个异常用元组括起来

```
try:

    f=open('xixi.txt')
    1/0
    print(1111111111111)
    f.read()
    f.close()

except (ZeroDivisionError,FileNotFoundError):
    print('除数不能为0哦~~，抱歉~~~~')
```

### 3. try except finally

语法：

```
try :
    代码
except    :
    print ( )
finally:
    最终执行的代码（无论代码是否出现异常都会执行finally中的代码）
```

```
try:

    f=open('xixi.txt','w')
    print(1111111)
    1/0

except (ZeroDivisionError,FileNotFoundError):
    print('除数不能为0哦~~，抱歉~~~~')

finally:
    print('0000000000000')
    f.close()
```

- raise语句

用于引发一个异常（抛出一个异常）

使用方法：

`raise` 异常类型

#自定义异常

```
class NotTicket(BaseException):
    pass
```

```
try:
    n=input('请输入数字：')
    if n=='1':
        print('有票')
```

```
    else:
        raise NotTicket('[n=%s]车票售完了'%n) #设置异常信息
except NotTicket as s:
    print('没票了',s)
```

#书籍

```
def fun():
    n=input('请输入一个整数：')
    try:
        n=int(n)
        if n==1:
            print('查询所有书籍')
        elif n==2:
            print('根据书名查')
    except ValueError:
        print('您输入的信息有误，请重新输入哦~~~')

fun()
```