

python in action

Lesson 8 - 异常处理, 读写文件

凤凰山 github.com/gxtrobot/pyinaction

2019-12

Lesson 8 - 异常处理, 读写文件

课程内容

今天的课程将介绍 Python 的异常处理方式, 以及文件的读写操作, 并利用新学的知识开发一个有用的程序, 可以处理文件里保存的一系列数字.

课程目标

- 了解 python 的异常处理方式
- 了解文件的读和写
- 了解并使用几种不同的文件读取方式
- 完成一个完整的程序, 处理一个包含一个班级分数的文件, 并进行统计打印相关信息

```
def div(a, b):  
    try:  
        return a / b  
    except Exception as ex:  
        print('error', ex)
```

系统抛出异常

- try 下面的语句块包括可能抛出异常的语句
- except 后跟上要捕获的异常, as 为别名, 下面的语句块包含异常处理语句块
- try 语句块里发生任何异常, 就会跳到 except 语句块, 如果抛出的异常在 except 捕获异常的列表中, 就会被处理, 系统不会再报错, 如果没有被捕获处理, 系统会报错
- 一个 try 语句块后面可以跟多个 except 语句块, 系统按顺序查找匹配的异常, 找到就进入处理语句块开始处理异常
- 一个 except 可以捕获多个异常, 用',' 分开: except (ValueError, OSError):
- except 后面不跟任何异常名, 默认捕获所有异常 except:

常见的系统异常

- `AttributeError` - 对象没有该属性
- `IndexError` - 列表没有该索引对应元素
- `KeyError` - 字典没有该元素
- `ValueError` - 值错误
- `ZeroDivisionError` - 除法, 0 为分母错误
- `OSError` - 系统错误
- `Exception` - 所有错误的基类, 可以捕获所有异常

IDLE 玩一玩

```
a = [1,2,3]
a[10]
b = {'name':'jack', 'age':30}
b['job']
1/0
int('a')
try:
    a[10]
except Exception
    pass
```

reporting_exec.py

```
def report(*args):  
    count = 0  
    total = 0  
    avg = 0  
    for v in args:  
        try:  
            value = int(v)  
            if value < 0:  
                raise ValueError('{},must above 0 '.  
                                format(v))  
            count += 1  
            total += int(v)  
        except Exception as ex:  
            print(ex)
```

```
avg = total / count  
print('count:', count)  
print('total:', total)  
print('avg:', avg)
```

用户主动抛出异常

- 一般用于业务相关的原因，比如期望一个非负数，传入负数，主动抛出异常
- `raise` 关键字后跟异常对象 `raise ValueError('必须传入非负数')`
- 所有异常类，创建对象时接收一个字符串对象，用于说明异常原因，打印异常是会被打印

文件的打开

- `open` 函数处理文件的操作，包括读和写，`open(file, mode)`
- `file` 参数为文件的路径，可以是相对路径或绝对路径
- `mode` 参数为读写模式，可包含多个以下字符组合
 - `r` - 读取模式，默认
 - `w` - 写入模式，每次清空
 - `a` - 末尾追加写入模式，不清空
 - `b` - 二进制文件模式，比如写入图片
 - `t` - 文本模式，默认
 - `r+` - 同时读和写
- `open` 函数返回一个文件对象

文件的读取

```
f = open('a.txt')  
f.read()  
f.close()
```

- `open(file)` - 函数返回的文件对象可以进行内容的读取
- `read` - 函数读取所有内容, 返回一个字符串
- `readlines` - 函数读取行, 返回一个字符串列表, 每一个元素为一行, 含换行符
- `readline` - 读取一行, 返回一个字符串, 含换行符
- 读取时如果返回空字符串, 说明已经到了文件末尾, 没有更多内容
- 每次读取会记住当前读取的位置, 下次读取会继续
- 操作结束后要调用 `close` 函数, 关闭文件

文件的写入

```
f = open('b.txt', 'w')  
f.write('abc\n')  
f.writelines(['def\n'])  
f.close()
```

- `open(file, 'w')` - 函数返回的文件对象可以进行内容的写入
- `write(text)` - 函数写入一个字符串, 返回写入字符个数
- `writelines(lines)` - 函数写入一个列表, 每个元素为一行
- 操作结束后要调用 `close` 函数, 关闭文件

IDLE 玩一玩

```
f = open('b.txt', 'w')
f.write('abc\n')
f.write('haha')
f.writelines(['def\n'])
f.close()
```

```
f = open('b.txt')
f.read()
f.close()
```

```
f = open('b.txt', 'a')
f.write('hello world')
f.close()
```

写一个小程序

reporting_scores.py 基于第 6 课的 reporting.py 改写成可以读取一个文件 (scores.txt) 里的所有数字，并处理打印模块

- 将 `reporting_scores.py` 加入一个异常处理，主动抛出异常，如果分数大于 100 或者小于 0
- 将 `report_scores.py` 的报告内容追加写入文件中 (`scores_report.txt`)

