

软件测试Python课程V5.0



黑马程序员-软件测试



- 1. 掌握如何处理程序中的异常
- 2. 掌握对JSON数据的操作



- ◆ 异常
- ◆ 文件操作
- ◆ JSON操作



- 异常的概念
- 捕获异常
- 异常的传递
- 抛出异常

学习目标

- 1. 理解异常的概念
- 2. 掌握如何捕获异常
- 3. 掌握异常的传递
- 4. 知道如何抛出异常

黑马程序员-软件测试



异常的概念

- 程序在运行时,如果Python解释器遇到到一个错误,则会停止程序的执行,并且提示一些错误信息,这就是**异常**.
- 程序停止执行并且提示错误信息这个动作,通常称之为: 抛出 (raise) 异常



提示:

- 程序开发时,很难将所有的特殊情况都处理的面面俱到 ,通过 异常捕获可以针对突发事件做集中的处理,从而 保证程序的稳定性和健壮性
- 在自动化测试过程中,也可以借助捕获异常操作,完成类似:用例执行报错时截图、打印日志信息等操作



捕获异常

基本语法格式

try:

尝试执行的代码

except:

#出现异常时执行的代码

说明:

- try: 尝试, 下方编写要尝试代码, 不确定是否能够正常执行的代码
- except: 如果有异常捕获,下方编写捕获到异常,处理失败的代码



捕获异常案例

需求:

- 1. 获取用户从键盘输入的数据
- 2. 转换数据类型为整数
- 3. 数据转换类型正确时, 输出数据内容
- 4. 数据转换类型错误时, 提示输入正确数据





捕获异常案例



```
def input_num():
    """定义输入数据函数"""

try:
    num = int(input('请输入数字:'))
    print('获取的数据为: {}'.format(num))
    except:
    print('请输入正确数据!')

if __name__ == '__main__':
    input_num()
```



捕获特定异常类型

语法格式

try:

尝试执行的代码

except 异常类型:

#出现异常时执行的代码

注意:

- 1. 捕获特定异常类型,能够实现仅在特定异常类型出现时,代码再做对应的处理
- 2. 具体的异常类型可以先行从代码执行的报错信息中获取,在设置捕获异常操作



捕获特定异常类型案例

需求:

- 1. 提示用户输入一个整数
- 2. 使用 8 除以用户输入的整数并且输出



案例分析:

- 用户输入的可能不是一个整数
- 0不能做除数



捕获特定异常类型案例



```
def input_num():
  """定义输入数据函数"""
 try:
    num = int(input("请输入整数: "))
   result = 8 / num
    print(result)
  except ValueError:
    print("请输入正确的整数")
  except ZeroDivisionError:
    print("除 0 错误")
if __name__ == '__main__':
  input_num()
```



捕获未知类型异常

语法格式

try:

尝试执行的代码

except Exception as e:

#出现异常时执行的代码

print('异常信息为: {}'.format(e))

说明:

- 如果希望程序无论出现任何错误,都不会因为Python解释器抛出异常而被终止,可以捕获 Exception
- except Exception as e: e表示捕获到的异常对象,记录异常的错误信息,e为惯用变量名,可以自定义



捕获未知类型异常案例

需求:

- 1. 提示用户输入一个整数
- 2. 使用 8 除以用户输入的整数并且输出
- 3. 要求: 为了简化代码,统一捕获所有异常类型





捕获未知类型异常案例



```
def input_num():
    """定义输入数据函数"""

try:
    num = int(input("请输入整数: "))
    result = 8 / num
    print(result)
    except Exception as e:
        print("出错了: {}".format(e))

if __name__ == '__main__':
    input_num()
```



异常捕获完整语法

```
try:
 # 尝试执行的代码
 pass
except 错误类型1:
 #针对错误类型1,对应的代码处理
 pass
except 错误类型2:
 #针对错误类型2,对应的代码处理
 pass
except Exception as e:
 # 打印错误信息
 print(e)
finally:
 #无论是否有异常,都会执行的代码
 pass
```

说明:

● finally: 无论是否出现异常都会执行



异常捕获完整语法案例

需求:

- 1. 获取用户输入的数字
- 2. 判断获取的数字是否整数
- 3. 如果不是整数,提示输入错误
- 4. 如果是整数,则进一步判断是奇数还是偶数
- 5. 最终提示: 程序运行结束





异常捕获完整语法案例



```
def input_num():
  """定义输入数据函数"""
 try:
   num = int(input('请输入数字:'))
  except ValueError:
    print('输入错误!')
 else:
   if num % 2 == 0:
     print('获取的数据为: {}, 是偶数!'.format(num))
    else:
     print('获取的数据为: {}, 是奇数!'.format(num))
 finally:
   print('程序运行结束!')
if __name__ == '__main__':
 input_num()
```



异常的传递

- 异常的传递: 当函数/方法执行出现异常,会将异常传递给函数/方法的调用一方
- 如果 传递到主程序,仍然 没有异常处理,程序才会被终止



- 1. 保持代码整洁,不想增加大量的异常捕获代码
- 2. 需要让外部知道出现了异常
- 3. 异常比较严重,自己处理不了





异常传递案例

需求:

- 1. 定义函数 demo1() 提示用户输入一个整数并且返回
- 2. 定义函数 demo2() 调用 demo1()
- 3. 在主程序中调用 demo2()





异常传递案例



```
def demo1():
  return int(input("请输入一个整数:"))
def demo2():
  return demo1()
if __name__ == '__main__':
 try:
    print(demo2())
  except ValueError:
    print("请输入正确的整数")
  except Exception as e:
    print('异常错误信息为: {}'.format(e))
```



抛出异常

应用场景:

- 在开发中,除了 代码执行出错 Python 解释器会 抛出 异常之外
- 还可以根据 应用程序 特有的业务需求 主动抛出异常

案例:提示用户输入密码,如果长度少于8,抛出异常





抛出异常

- Python 中提供了一个 Exception 异常类
- 在开发时,如果满足特定业务需求时希望抛出异常,可以:
 - 1. 创建一个 Exception类 的 对象
 - 2. 使用 raise 关键字 抛出 异常对象

创建异常对象 ex = Exception("异常描述信息")

抛出异常对象 raise ex



黑马程序员-软件测试



抛出异常案例

需求:

- 1. 定义 input_password 函数,提示用户输入密码
- 2. 如果用户输入长度 < 8, 抛出异常
- 3. 如果用户输入长度 >=8, 返回输入的密码





抛出异常案例

案例演练

```
def input_password():
 #1. 提示用户输入密码
 pwd = input("请输入密码:")
 # 2. 判断密码长度,如果长度 >= 8,返回用户输入的密码
 if len(pwd) >= 8:
   return pwd
 #3. 密码长度不够,需要抛出异常
 # 1> 创建异常对象 - 使用异常的错误信息字符串作为参数
 ex = Exception("密码长度不够")
 #2>抛出异常对象
 raise ex
try:
 user pwd = input password()
 print(user_pwd)
except Exception as result:
 print(f"发现错误: {result}")
                  黑马程序员-软件测试
```





- 1. 什么是异常?
- 2. 如何捕获异常?
- 3. 如果没有捕获异常, 异常会传递给谁?
- 4. 如何主动抛出一个异常?



- ◆ 异常
- ◆ 文件操作
- ◆ JSON操作



- 文件的概念
- 文件的操作方法

学习目标

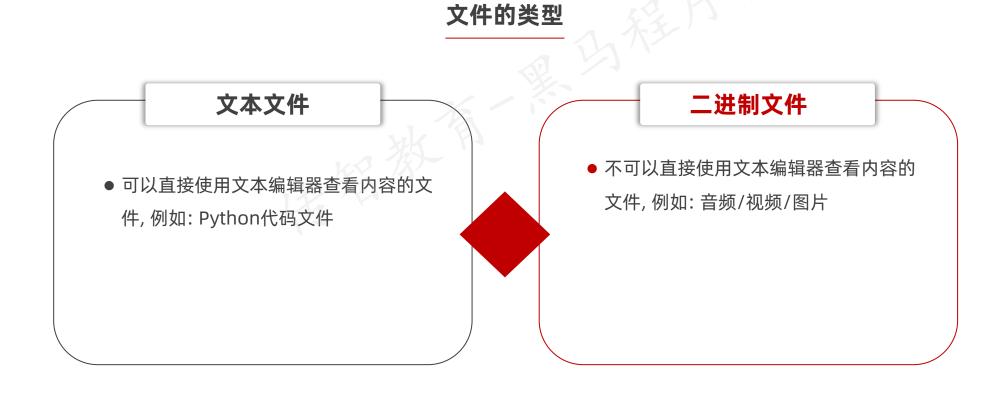
1. 掌握文件的操作方法

黑马程序员-软件测试



文件的概念

文件: 可以存储在长期存储设备上的一段数据即为文件



黑马程序员-软件测试



文件的操作方法

说明:文件的读取与写入,都推荐使用with open()方法,该方法执行结束后,会自动关闭文件

操作文件的套路:

- 1. 打开文件
- 2. 读、写文件
 - 读将文件内容读入内存
 - 写 将内存内容写入文件
- 3. 关闭文件



11111

name: 要访问的文件路径及名称

encoding: 设置字符编码

f: 文件对象, 惯用变量名, 可以自定义

11 11 11

with open(name, encoding='utf8') as f:

pass



文件的操作方法

读取文件内容方法:

- read()方法可以一次性 读入 并 返回 文件的 所有内容
- readline()方法可以一次读取一行内容,方法执行后,会把 文件指针 移动到下一行,准备再次读取

```
with open("README") as f:
# 读取
text = f.read()
print(text)
```

```
with open("README") as file:
while True:
#读取一行内容
text = file.readline()

#判断是否读到内容
if not text:
break

#每读取一行的末尾已经有了一个`\n`
print(text, end="")
```



文件的操作方法

写文件方法:

11 11 11

w: 写入文件方式(默认为 r, 读取文件方式, 可以省略)

write(): 将指定内容写入到文件

11 11 1

with open(name, 'w', encoding='utf8') as f:

f.write('文件内容')

open()函数默认以 只读方式 打开文件,并且返回文件对象



文件的操作方法

打开文件的方式

访问方式	说明
r	以只读方式打开文件。这是默认模式。如果文件不存在,抛出异常
W	以只写方式打开文件。如果文件存在会被覆盖。如果文件不存在,创建新文件
a	以追加方式打开文件。如果该文件已存在,在文件末尾追加内容。如果文件不存在,创建新文件进行写入
rb	以二进制格式打开一个文件用于只读
wb	以二进制格式打开一个文件只用于写入





1. 如何读取文件中的内容?



- ◆ 异常
- ◆ 文件操作
- ◆ JSON操作



- JSON 介绍
- JSON 数据操作

学习目标

- 1. 掌握JSON的语法规范
- 2. 掌握如何实现JSON数据操作

黑马程序员-软件测试

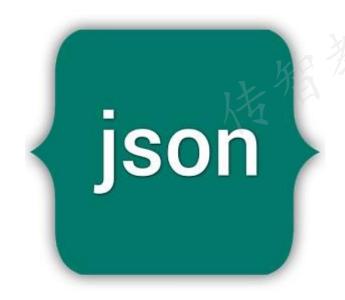


JSON 介绍

- JSON的全称是"JavaScript Object Notation",是JavaScript对象表示法,它是一种基于文本,独立于语言的轻量级数据交换格式。
- 在自动化测试中经常用来存放测试数据,文件后缀名为:.json

提示: 其他常见的数据源文件

格式: txt/excel/csv/xml



```
{
    "name": "小明",
    "gender": "男",
    "age": 88
}
```



JSON的特点

- 纯文本格式: 不支持音频/视频/图片
- 具有良好的自我描述性,方便阅读与编写
- 具有清晰的层次结构
- 相较于XML文件,能够有效提升网络传输 效率

```
"name": "tom",
"age": 18,
"isMan": true,
"school": null,
"address": {
   "country": "中国",
   "city": "江苏苏州",
    "street": "科技园路"
                                       JSON -> XML
"numbers": [2, 6, 8, 9],
"links": [
        "name": "Baidu",
        "url": "http://www.baidu.com"
   },
        "name": "TaoBao",
        "url": "http://www.taobao.com"
```

```
<?xml version="1.0" encoding="UTF-8" ?>
<person>
   <name>tom</name>
   <age>18</age>
   <isMan>true</isMan>
   <school />
   <address>
       <country>中国</country>
       <city>江苏苏州</city>
       <street>科技园路</street>
   </address>
    <numbers>2</numbers>
   <numbers>6</numbers>
    <numbers>8</numbers>
   <numbers>9</numbers>
   ks>
       k>
           <name>Baidu</name>
           <url>http://www.baidu.com</url>
       </link>
       k>
           <name>TaoBao</name>
           <url>http://www.taobao.com</url>
       </link>
   </links>
</person>
```



JSON 语法

- {}大括号保存对象
- -[]中括号保存数组
- 对象数组可以相互嵌套
- 数据采用键值对表示
- 多个数据由逗号分隔

JSON键 必须是字符串类型, 必须使用英文双引号

JSON值 可以是以下类型:

- 数字(整数或浮点数)
- 字符串(使用双引号)
- 逻辑值(true 和 false)
- 数组 ->[]
- 对象 -> {}
- 空值: null



JSON语法-示例内容

```
"name": "tom",
"age": 18,
"isMan": true,
"school": null,
"address": {
  "country": "中国",
  "city": "北京",
  "street": "长安街"
"numbers": [2, 6, 8, 9],
"links": [
    "name": "Baidu",
    "url": "http://www.baidu.com"
    "name": "TaoBao",
    "url": "http://www.taobao.com"
```

黑马程序员-软件测试



JSON 数据操作



读取JSON文件

导入依赖包 import json

with open('data.json', encoding='UTF-8') as f:

#返回的数据类型为字典或列表

data = json.load(f)



读取JSON文件案例

需求:

1. 通过程序读取下面json文件中的内容,并在控制台打印出所有字段的内容



```
"name": "tom",
"age": 18,
"isMan": true,
"school": null,
"address": {
 "country": "中国",
 "city": "北京",
 "street": "长安街"
"numbers": [2, 6, 8, 9],
"links": [
    "name": "Baidu",
    "url": "http://www.baidu.com"
    "name": "TaoBao",
    "url": "http://www.taobao.com"
                        黑马程序员-软件测试
```



读取JSON文件案例



```
import json
with open("data.json", encoding="UTF-8") as f:
 json data = json.load(f)
  name = json_data.get("name")
  print("name=", name)
  numbers = json_data.get("numbers")
  for n in numbers:
    print(n)
  address = json_data.get("address")
  city = address.get("city")
  print("city=", city)
  links = json_data.get("links")
  for link in links:
    link name = link.get("name")
    link url = link.get("url")
    print(f"link_name={link_name} link_url={link_url}")
```



JSON 数据操作



写入JSON文件

导入依赖包 import json

要写入json文件的内容,可以是字典或列表 param = {'name': 'tom', 'age': 20}

打开写模式

with open('data2.json', 'w', encoding='UTF-8') as f:
 json.dump(param, f)





- 1. JSON中的键是什么类型?
- 2. JSON中的值都有哪些类型?
- 3. 如何读取JSON文件中内容?

异常、文件操作





- 1. 掌握如何处理程序中的异常
- 2. 掌握对JSON数据的操作



传智教育旗下高端IT教育品牌

黑马程序员-软件测试