



课题名称	第 10 章 文件与文件路径操作	计划学时	4 学时
内容分析	在 Python 程序中为避免出现各种各样以外的问题，Python 提供异常处理机制，异常处理机制以处理或修复程序中可能出现的错误，提供诊断信息，帮助开发人员尽快解决问题，恢复程序的正常运行。		
教学目标及基本要求	<ol style="list-style-type: none"> <li>1. 理解异常的概念</li> <li>2. 掌握捕获并处理异常的方式</li> <li>3. 掌握 raise 和 assert 语句</li> <li>4. 掌握自定义异常</li> <li>5. 掌握 with 语句的使用</li> <li>6. 了解上下文管理器</li> </ol>		
教学重点	<ol style="list-style-type: none"> <li>1. try-except 语句</li> <li>2. 捕获异常信息</li> <li>3. else 子句</li> <li>4. finally 子句</li> <li>5. raise 语句</li> <li>6. assert 断言语句</li> </ol>		
教学难点	<ol style="list-style-type: none"> <li>1. 异常的传递</li> <li>2. assert 断言语句</li> <li>3. 自定义异常</li> <li>4. 上下文管理器</li> <li>5. 自定义上下文管理器</li> </ol>		
教学方式	教学采用教师课堂讲授为主，使用教学 PPT 讲解		

教 学 过 程	<p style="text-align: center;"><b>第一课时</b></p> <p style="text-align: center;">(错误和异常概述、try-except 语句、捕获异常信息、else 子句、finally 子句)</p> <p><b>一、创设情境，引出什么是异常</b></p> <p>(1) 教师通过提出需求，引出什么是异常。</p> <p>(2) 明确学习目标</p> <ul style="list-style-type: none"><li>● 要求学生了解错误和异常概述</li><li>● 要求学生掌握 try-except 语句</li><li>● 要求学生掌握捕获异常信息</li><li>● 要求学生掌握 else 子句</li><li>● 要求学生掌握 finally 子句</li></ul> <p><b>二、进行重点知识的讲解</b></p> <p>(1) 教师根据课件，讲述错误和异常的概述。</p> <p>Python 程序中最常见的错误为语法错误。语法错误又称为解析错误，它是指开发人员编写了不符合 Python 语法规则的代码所引起的错误。含有语法错误的程序会抛出异常，无法运行。</p> <p>(2) 教师根据课件，讲述 try-except 的使用方法。</p> <p>try-except 语句用于捕获程序运行时的异常，其语法格式为：</p> <p>try:</p> <p style="padding-left: 2em;">可能出错的代码</p> <p>except [错误类型]:</p> <p style="padding-left: 2em;">错误处理语句</p> <p>try-except 语句的执行过程为：1) 先执行 try 子句，即 try 与 except 之间的代码；2) 若 try 子句中未产生异常，忽略 except 子句代码；3) 若 try 子句产生异常，则忽略 try 子句的剩余代码，执行 except 子句代码。</p> <p>(3) 教师根据课件，讲述如何捕获异常信息。</p> <p>try-except 语句可以捕获单个异常、多个异常信息或全部异常，同时也支持使用关键字 as 为异常类起别名。</p> <p>(4) 教师根据课件，讲述 try-except-else 语句的使用。</p>
------------------	---

`try-except` 语句可以与 `else` 子句联合使用，该子句放在 `except` 语句之后，表示当 `try` 子句没有出现错误时应执行的代码。其格式为：

`try:`

可能出错的语句

`except:`

出错后执行的语句

`else:`

未出错时的执行语句

(5) 教师根据课件，讲述 `finally` 语句的使用。

`finally` 子句与 `try-except` 语句连用时，无论 `try-except` 是否捕获到异常，`finally` 子句后的代码都要执行，其语法格式为：

`try:`

可能出错的语句

`except:`

出错后执行语句

`finally:`

无论是否出错都会执行的语句

### 三、归纳总结，布置作业/随堂练习

(1) 回顾上课前的学习目标，并对本节课的内容进行总结。

教师总结本节课需要掌握的知识点，包括错误和异常的概述、`try-except`、捕获异常信息、`else` 子句、`finally` 子句。

(2) 布置随堂练习，检查学生掌握情况。

根据博学谷和随堂练习资源，给学生布置随堂练习，检测学生的掌握程度，并对学生出现的问题进行解决。

(3) 使用博学谷系统下发课后作业。

### 第二课时

(`raise` 语句、异常的传递、`assert` 断言语句、自定义异常)

#### 一、回顾上节课的内容，继续讲解本课时的知识

(1) 教师对学生们的疑问进行统一答疑。

(2) 回顾总结上节课内容，继续介绍本课时的内容。

上节课我们学习了错误和异常概述、try-except 语句、捕获异常信息、else 子句、finally 子句，本节课将带领大家学习 raise 语句、异常的传递、assert 断言语句、自定义异常。

(3) 明确学习目标

- 要求学生掌握 raise 语句
- 要求学生掌握异常的传递
- 要求学生掌握 assert 断言语句
- 要求学生了解自定义异常

## 二、进行重点知识的讲解

(1) 教师根据课件，讲解 raise 语句的使用。

Python 程序中的异常不仅可以自动触发异常，而且还可以由开发人员使用关键字 raise 抛出异常，其私用方法大致可分为 1) 由异常类名引发异常；2) 由异常对象引发异常；3) 由程序中出现的异常引发异常。

(2) 教师根据课件，讲解异常的传递。

如果程序中的异常没有被处理，默认情况下会将该异常传递给上一级，如果上一级仍然没有处理，那么会继续向上传递，直至异常被处理或程序崩溃。

(3) 教师根据课件，讲解 assert 断言语句的使用。

assert 断言语句用于判定一个表达式是否为真，如果表达式为 True，不做任何操作，否则引发 AssertionError 异常，其语法格式为:assert 表达式[,参数]，表达式 assert 语句的判定对象，参数通常是一个自定义异常或用于显示异常描述信息的字符串。

(4) 教师根据课件，讲解自定义异常的使用。

Python 中定义了大量的异常类，虽然这些异常类可以描述编程时出现的绝大部分情况，但仍难以涵盖所有可能出现的异常。Python 允许开发人员自定义异常。自定义的异常类方法很简单，只需创建一个类，让它继承 Exception 类或其他异常类即可。

## 三、归纳总结，布置作业

- (1) 回顾学习目标，总结本节课所学知识包括 raise 语句、异常的传递、assert 断言语句。
- (2) 布置随堂练习，检查学生掌握情况。

根据博学谷和随堂练习资源，给学生布置随堂练习，检测学生的掌握程度，并对学生出现的问题进行解决。

(3) 使用博学谷系统下发课后作业。

### 第三课时

#### (with 语句、上下文管理器、自定义上下文管理、实例 1：身份证归属地查询添加异常)

##### 一、回顾上节课内容，继续介绍本课时的内容

- (1) 教师对学生们的疑问进行统一答疑。
- (2) 教师通过提问学生问题，由上一课时引出本课时要讲解的内容。
- (3) 明确学习目标
  - 要求学生掌握 with 语句的使用
  - 要求学生了解上下文管理
  - 要求学生了解自如何自定义上下文管理器
  - 要求学生了解实例 1：身份证归属地添加异常实现过程

##### 二、进行重点知识的讲解

(1) 教师根据课件，讲解 with 语句的使用方法。

with 语句适用于对资源进行访问的场合，无论资源在使用过程中是否发生异常，都可以使用 with 语句保证执行释放资源操作。with 语句语法格式为：

with 上下文表达式 [as 资源对象]:

语句体

with 语法中的上下文表达式返回一个上下文管理器对象，如果指定了 as 子句，将上下文管理器对象的\_\_enter\_\_()方法的返回值赋值给资源对象。资源对象可以是单个变量，也可以是元组。

(2) 教师根据课件，讲解什么是上下文管理器。

with 语句之所以能够自动关闭资源，是因为它使用了一种名为上下文管理的技术管理资源，上下文管理协议包括了\_\_enter\_\_()和\_\_exit\_\_()方法，支持该协议的对象均要实现这两个方法。

(3) 教师根据课件，讲解自定义上下文管理器。

教师根据教材资源实现自定义上下文管理，并为学生讲解实现过程。

	<p>(4) 教师根据课件，讲解实例 1：身份证归属地查询添加异常。</p> <p>教师根据教材资源实现实例 1：身份证归属地添加异常实例，并为学生讲解其实现过程。</p> <p><b>三、归纳总结，布置作业</b></p> <p>(1) 回顾学习目标，总结本节课所学知识包括 with 语句、上下文管理器、自定义上下文管理、实例 1：身份证归属地添加异常。</p> <p>(2) 使用博学谷系统下发课后作业。</p> <p style="text-align: center;"><b>第四课时</b> <b>(上机练习)</b></p> <p>上机练习主要针对本章中需要重点掌握的知识点，以及在程序中容易出错的内容进行练习，通过上机练习可以考察同学对知识点的掌握情况，对代码的熟练程度。</p> <p><b>上机一：（练习教材示例代码以及实例 1：身份证归属地查询添加异常）</b></p> <p><b>形式：独立完成</b></p> <p><b>要求：</b></p> <p>(1) 要求学生能够熟练掌握教材中示例代码。</p> <p>(2) 要求学生能够自己实现实例 1：身份证归属地查询添加异常。</p>
<p>思考题 和习题</p>	<p>见教材第 10 章配套的习题</p>
<p>教 学 后 记</p>	

