

黑马程序员

《Python 快速编程入门（第2版）》

教学设计

课程名称：Python 快速编程入门

授课年级：2020 年级

授课学期：2020 学年第二学期

教师姓名：某某老师

2020 年 09 月 09 日

课题名称	第3章 流程控制	计划学时	5 课时
内容分析	<p>程序中的语句默认自上而下顺序执行，但通过一些特定的语句可以更改语句的执行顺序，使之产生跳跃、回溯等现象，进而实现流程控制。Python 中用于实现流程控制的特定语句分为条件语句、循环语句和跳转语句，本章将结合这些特定语句介绍与流程控制相关的知识。</p>		
教学目标及基本要求	<ol style="list-style-type: none"> 1. 理解条件语句的结构，掌握条件语句的使用 2. 理解循环语句的结构，掌握循环语句的使用 3. 掌握跳转语句的使用 		
教学重点	<ol style="list-style-type: none"> 1. 理解条件语句的结构，掌握条件语句的使用 2. 理解循环语句的结构，掌握循环语句的使用 3. 掌握跳转语句的使用 		
教学难点	<ol style="list-style-type: none"> 1. 理解条件语句的结构，掌握条件语句的使用 2. 理解循环语句的结构，掌握循环语句的使用 		
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;">第一课时 （条件语句）</p> <p>一、创设情境，导入条件语句的概念</p> <p>（1）教师通过列举一个生活实例，引出程序的条件语句。</p> <p>现实生活中，大家在 12306 网站购票时需要先验证身份，验证通过后可进入购票页面，验证失败则需重新验证。程序中的语句都是自上而下执行的，它可以增加条件语句以产生分支，进而有选择地执行不同分支的语句。</p> <p>（2）明确学习目标</p> <ul style="list-style-type: none"> ● 要求学生理解条件语句的结构 ● 要求学生掌握条件语句的使用 <p>二、进行重点知识的讲解</p> <p>（1）教师根据课件，介绍 if 语句的语法格式和执行流程。</p> <p>if 语句是最简单的条件语句，该语句由关键字 if、判断条件和冒号组成，if 语句和从属于该语句的代码段可组成选择结构。执行 if 语句时，若 if 语句的</p>		

判断条件成立（判断条件的布尔值为 True），执行之后的代码段；若 if 语句的判断条件不成立（判断条件的布尔值为 False），跳出选择结构，继续向下执行。

（2）教师根据课件，介绍 if 语句的用法，并使用代码进行演示。

（3）教师根据课件，介绍 if 语句的局限性，引出 if-else 语句。

if 语句只能处理满足条件的情况，但一些场景不仅需要处理满足条件的情况，也需要对不满足条件的情况做特殊处理。因此，Python 提供了可以同时处理满足和不满足条件的 if-else 语句。

（4）教师根据课件，介绍 if-else 语句的语法格式和执行流程。

执行 if-else 语句时，若判断条件成立，执行 if 语句之后的代码段；若判断条件不成立，执行 else 语句之后的代码段。

（5）教师根据课件，介绍 if-else 语句的用法，并使用代码进行演示。

（6）教师根据课件，介绍 if-else 语句的局限性，引出 if-elif-else 语句。

根据上个小节的考试成绩评估程序可知，该程序只能评估考试及格和不及格的情况，但实际评估成绩时会划分为优良中差四个等级，if-else 局限于两个分支，像这种出现多个分级的情况无法通过 if-else 语句进行处理。为处理类似上述情景的一个事项的多种情况，Python 提供了可产生多个分支的 if-elif-else 语句。

（7）教师根据课件，介绍 if-elif-else 语句的语法格式和执行流程。

执行 if-elif-else 语句时，若 if 条件成立，执行 if 语句之后的代码段 1；若 if 条件不成立，判断 elif 语句的判断条件 2：条件 2 成立则执行 elif 语句之后的代码段 2，否则继续向下执行。以此类推，直至所有的判断条件均不成立，执行 else 语句之后的代码段。

（8）教师根据课件，介绍 if-elif-else 语句的用法，并使用代码进行演示。

（9）教师通过列举一个生活实例，引出嵌套的概念。

大家在某些火车站乘坐高铁出行时需要历经检票和安检两道程序：检票符合条件后方可进入安检程序，安检符合条件后方可进站乘坐列车。这个场景中虽然涉及两个判断条件，但这两个条件并非选择关系，而是嵌套关系，两个条件逐一判断，只有全部满足之时才能继续后续操作。

(10) 教师根据课件，介绍 if 嵌套的内容，并使用代码进行演示。

三、归纳总结，布置课后作业

(1) 回顾上课前的学习目标，对本节课知识点进行总结。

教师带领学生总结本节课需要掌握的条件语句，包括 if 语句、if-else 语句、if-elif-else 语句和 if 嵌套。

(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

(实训案例，while 语句、for 语句)

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) 回顾上个课时所学习的内容，继续介绍本课时的内容。

上节课主要为大家介绍了条件语句的内容，接下来，本节课先为大家演示两个精彩的实例，再介绍循环语句的部分内容，包括 while 语句、for 语句。

(3) 明确学习目标

- 要求学生掌握循环语句的使用

二、进行重点知识的讲解

(1) 教师根据课件，介绍 3.2.1 的实例，并使用代码进行演示。

(2) 教师根据课件，介绍 3.2.2 的实例，并使用代码进行演示。

(3) 教师通过列举一个生活实例，引出程序的循环语句。

现实生活中存在着很多重复的事情，例如，地球一直围绕着太阳不停地旋转；月球始终围绕地球旋转；每年都会经历四季的更替；每天都是从白天到黑夜的过程……程序开发中同样可能现代码的重复执行，Python 提供了循环语句，使用该语句能以简洁的代码实现重复操作。

(4) 教师根据课件，介绍 while 语句的语法格式和执行流程。

执行 while 语句时，若循环条件的值为 True，则执行之后的代码段，执行完代码段之后再次判断循环条件，如此往复，直至循环条件的值为 False 时循环终

止，执行循环之后的代码。

(5) 教师根据课件，介绍 while 语句的用法，并使用代码进行演示。

(6) 教师根据课件，介绍 for 语句的语法格式和执行流程。

(7) 教师根据课件，介绍 for 语句的用法，并使用代码进行演示。

三、归纳总结，布置作业

(1) 回顾上课前的学习目标，对本节课知识点进行总结。

教师带领学生总结本节课需要掌握的循环语句，包括 while 语句和 for 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

(循环嵌套，实训案例)

一、回顾上节课内容，继续讲解本节课的内容

(1) 教师对学生们的疑问进行统一答疑。

(2) 回顾上个课时所学习的内容，继续介绍本课时的内容。

上节课主要为大家介绍了 while 语句和 for 语句，接下来，本节课先为大家介绍循环嵌套的内容，再结合循环的内容演示两个精彩的实例。

(3) 明确学习目标

- 要求学生掌握循环语句的使用

二、进行重点知识的讲解

(1) 教师根据课件，引出循环嵌套的内容。

循环之间可以互相嵌套，进而实现更为复杂的逻辑。循环嵌套按不同的循环语句可以分为 while 循环嵌套和 for 循环嵌套。

(2) 教师根据课件，介绍 while 循环嵌套的语法格式和执行流程。

while 循环嵌套是指 while 语句中嵌套了 while 或 for 语句。以 while 语句中嵌套 while 语句为例，执行 while 循环嵌套时，若外层循环的循环条件 1 的值为 True，则执行代码段 1，并对内层循环的循环条件 2 进行判断：值为 True 则执行代码段 2，值为 False 则结束内层循环。内层循环执行完毕后继续判断外层循环

的循环条件 1，如此往复，直至循环条件 1 的值为 `False` 时结束循环。

(3) 教师根据课件，介绍 `while` 循环嵌套的用法，并使用代码进行演示。

(4) 教师根据课件，介绍 `for` 循环嵌套的语法格式和执行流程。

`for` 循环嵌套是指 `for` 语句中嵌套了 `while` 或 `for` 语句。以 `for` 语句中嵌套 `for` 语句为例，执行 `for` 循环嵌套时，程序首先会访问外层循环中目标对象的首个元素、执行代码段 1、访问内层循环目标对象的首个元素、执行代码段 2，然后访问内层循环中的下一个元素、执行代码段 2……如此往复，直至访问完内层循环的目标对象后结束内层循环，转而继续访问外层循环中的下一个元素，访问完外层循环的目标对象后结束外层循环。因此，外层循环每执行一次，都会执行一轮内层循环。

(5) 教师根据课件，介绍 `for` 循环嵌套的用法，并使用代码进行演示。

(6) 教师根据课件，介绍 3.4.1 的实例，并使用代码进行演示。

(7) 教师根据课件，介绍 3.4.2 的实例，并使用代码进行演示。

三、归纳总结，布置作业

(1) 回顾上课前的学习目标，对本节课知识点进行总结。

教师带领学生总结本节课需要掌握的循环嵌套。

(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。

第四课时

(跳转语句，阶段案例——房贷计算器)

一、回顾上节课内容，继续讲解本节课的内容

(1) 教师讲解上节课有难度的作业，并对学生的疑问进行统一答疑。

(2) 回顾上个课时所学习的内容，继续介绍本课时内容。

上节课主要为大家介绍了循环嵌套的内容，接下来，本节课先为大家介绍跳转语句的内容，再演示阶段案例。

(3) 明确学习目标

- 要求学生掌握跳转语句的使用

二、进行重点知识的讲解

(1) 教师通过列举一个生活实例，引出程序的跳转语句。

大家在使用音乐播放器听歌时，有时会以循环模式进行播放，中途若遇到不想听的歌曲，可以切换到下一首歌曲。Python 提供了控制循环的跳转语句：`break` 和 `continue`。

(2) 教师根据课件，介绍 `break` 语句的用法，并使用代码进行演示。

`break` 语句用于结束循环，若循环中使用了 `break` 语句，程序执行到 `break` 语句时会结束循环；若循环嵌套使用了 `break` 语句，程序执行到 `break` 语句时会结束本层循环。

(3) 教师根据课件，介绍 `continue` 语句的用法，并使用代码进行演示。

`continue` 语句用于在满足条件的情况下跳出本次循环，该语句通常也与 `if` 语句配合使用。

(4) 教师根据课件，介绍 3.6 的阶段案例，并使用代码进行演示。

三、归纳总结，布置作业

(1) 回顾上课前的学习目标，对本节课知识点进行总结。

教师带领学生总结本节课需要掌握的跳转语句，包括 `break` 语句、`continue` 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。

第五课时

(上机练习)

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。

上机一：（考察知识点为 `if-else` 语句）

形式：独立完成

题目：

	<p>已知一个三位数，若它的各位数字的 3 次幂之和等于它本身，这个数就是一个水仙花数。请按照以下要求操作，判断用户输入的三位数是否为水仙花数。</p> <p>(1) 接收用户输入的三位数；</p> <p>(2) 取出该三位数中各位的数字；</p> <p>(3) 通过 if-else 语句判断各位数字的 3 次幂之和是否等于该三位数。</p> <p>上机二：（考察知识点为 if 语句嵌套）</p> <p>形式：单独完成</p> <p>题目：</p> <p>请按照以下要求操作。</p> <p>(1) 通过变量 num1 和 num2 接收用户输入的两个数值；</p> <p>(2) 定义一个保存四则运算符的变量 calculate；</p> <p>(3) 如果 calculate 为 “+”，计算 num1 和 num2 的和；如果为 “-”，计算它们的差；如果为 “*”，计算它们的乘积；如果 calculate 为 “/”，计算它们的差，并必须保证 num2 的值不能为 0。</p> <p>上机三：（考察知识点为循环嵌套）</p> <p>形式：单独完成</p> <p>题目：</p> <p>打印一个九九乘法表，该表中各等式的格式为 “乘数*乘数=积”。</p>
思考题和习题	见教材第 3 章配套的习题
教学后记	