【白话python连载（5）】python的条件语句--选择的分岔路

* 从小入手-简单的python小应用（学生成绩的实例）

#问题描述：判断不同学生成绩的对应等级。

|  |
| --- |
| *#!/usr/bin/python # -\*- coding: utf-8 -\*-* name **=** input(**'请输入学生的姓名：'**) scores **=** int(input(**'请输入{}的成绩:'**.**format**(name))) **if** scores **>=** 90**:** print(**'{}的等级是：A'**.**format**(name)) **elif** scores **>=** 70**:** print(**'{}的等级是：B'**.**format**(name)) **elif** scores **>=** 60**:** print(**'{}的等级是：C'**.**format**(name)) **else:** print(**'{}的等级是：D'**.**format**(name)) |
| *"D:\anaconda python\python3.6.exe" D:/vernacular/coding\_ziyi/scores.py*  *请输入学生的姓名：吴晓辉*  *请输入吴晓辉的成绩:78*  *吴晓辉的等级是：B*  *Process finished with exit code 0* |

通过条件判断获得学生成绩所对应的等级。

* python让你做出更好的选择

无论解决生活中的难题还是日常生活的方方面面，无时无刻都需要作判断下决策，同样计算机程序可以完成自动化的任务，这也是因为计算机可以自己做判断，根据不同的条件而执行不同的代码，而实现不同的功能，使程序更合理化、人性化。

促使计算机程序完成每个关键操作的核心的就是不同判断条件下执行不同的内容。python中条件语句有四种。

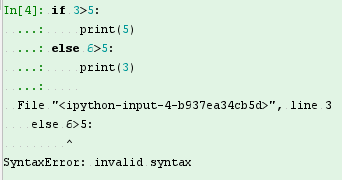
|  |  |  |
| --- | --- | --- |
| 条件语句类型 | 基本表达式 | 实例 |
| 单分支： | if bool\_expression:  suite |  |
| 双分支： | if bool\_expression:  suite  else:  suite |  |
| 多分支： | if bool\_expression:  suite  elif bool\_expression:  suite  ……  else:  suite |  |
| 三元表达式： | expression1 if bool\_expre  else expression2 |  |

最简单的是单分支型，即实现一个判断操作。其次是常规的双分支型操作，也就是非此即彼的二元决策逻辑。更符合实际情况的是类似于上述实例的多分支型操作。对于双分支型的判断内容，可以使用三元表达式来简化代码书写。

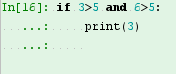
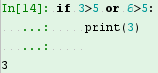
在使用python的判断语句中，要注意的是几个点：

python中判断语句是由上往下依次执行，在同一个判断体中，如果有满足的判断则执行完对应条件下的内容将会直接跳出判断语句，而执行判断体之后的内容。因此在实例中同一个成绩只会得出一个等级判定。

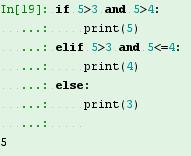
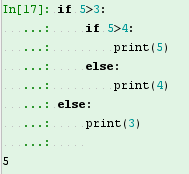
python的else：语句中不能再添加额外的判断条件，默认为if判断条件中的其他所有结果，如果强制添加新的判断条件会报错。这也表明了if语句判断的无二义性。



条件语句的表达式，常常会与逻辑运算结合使用，可以大大扩展python对不同情况的处理。其中and只有在前一个表达式为真的情况下才会处理执行后一个表达式，而or只有在前一个表达式为假的情况下才会处理执行后一个表达式。因此在实际的应用中，为了减少无效的运算，可以尽量将最需要判断的内容至于逻辑表达式的前面。



如果不与逻辑运算结合使用，仍然要判断两个条件时则需要使用if的嵌套方法。即在一个条件体中，嵌入另一个判断需要使用不同的缩进来表示嵌套关系。以下面的实例，输出（3,4,5）中的最大值的条件语句的写法。



总结：

本次主要介绍了python的条件语句，明确了基础的条件语句的书写方式。

重点强调了在使用条件语句中的一些注意点。了解了条件语句的执行顺序、关键字的具体功能、与逻辑表达式结合使用，以及if嵌套使用的方法和作用。

