

**标签：**本程序所称“标签”，指的是电子表格文件（如 xls 等）中与列名 A、B、C……类似的区分各列含义的词语，如单位、学校、姓名、性别等，有时可能也称之为项目或项目名称。按指定标签行将电子表格文件（如 xls 等）数据加载到内存中，内存表中便有了标签名作为各列列名。本程序中**标签的值或内容**用**实心中括号**包围标签名表示，如：**【姓名】**、**【实发金额（元）】**等。

**电子表格文件中标签所在行的各标签均不能处在合并单元格中！**

（注：本程序所称标签、列、项目、字段等均为同义词语。）

=====

### 条件表达式说明（Python 语言）

**条件表达式**中的英文空格、英文单双引号（需成对使用）、英文括号、英文逗号等都是重要的运算符，请注意与中文符号的区分！

在 Python 中，**英文单引号和双引号**的使用没有本质区别，它们可以互换使用（**但需成对使用**），没有语义上的区别。但是，为了代码的一致性和可读性，通常推荐使用一种风格，并在整个代码中保持这种风格。通常，Python 社区推荐使用双引号（"）来表示字符串，主要是因为它与 C 语言的习惯一致，并且在某些情况下，字符串中包含单引号是更自然的。

使用单双引号混合，如果需要在字符串中包含引号：

```
quote_string = "He said, 'Hello world'."
```

如果需要在字符串中包含双引号，可以使用单引号：

```
quote_string = 'She said, "Hello world".'
```

也可使用转义符\" 或 \"

```
quote_string = "He said, \"Hello world\"."
```

```
quote_string = 'She said, \'Hello world\'.'
```

在程序中，文本一般称做字符或字符串，字符串必须由英文双引号或单引号所包含，如 10.23 为数值，"10.23"为字符串。

**本程序条件表达式**中的变量名或函数名一般使用全部小写字母，使用**实心中括号【性别】**表示标签的值或内容。

### 条件表达式 if （三目运算符）

三目运算符，又称条件运算符，是计算机语言（c, c++, java 等）的重要组成部分。它是唯一有 3 个操作数的运算符，有时又称为三元运算符。

Python 中的三目运算符（形式上不同于其它语言）：

返回 True 执行 if 表达式 else 返回 False 执行

分三部分：左边“返回 True 执行”表示条件表达式为真时的执行过程或结果；中部“if 表达式 else”为条件表达式，如果“表达式”成立则执行左边的部分，否则执行右边的部分；右边“返回 False 执行”表示条件表达式为假时的执行过程或结果。

如：判断男或女（如果【性别】为 1，则结果是男，否则是女）

“男” if 【性别】=="1" else “女”

需要注意的是，除 if、else 和 == 这三个词语外，其它的词语则需要根据所要表达的类型（数值或字符）来确定是否加英文引号，如果是数值类型的（如 1），则可加可不加英文引号，如果是字符类型的则必须加英文引号，否则达不到预期的结果。本例中 1 可加可不加，程序会自动判断，【性别】也可加可不加，因为【性别】的内容是按 1 或不是 1 处理的。但下例就需要仔细斟酌了（如果【性别】为男，则结果是 1，否则是 0）：

“1” if “【性别】”=="男" else “0”

左右的 1 或 0 可加可不加英文引号，但中部的条件中男是必须加英文引号的，而【性别】的内容可能是男，它不带英文引号，因此，它也要加英文引号才匹配的，否则，不加英文引号的男会按变量处理报错的，而“男”是按字符男处理的。

如：判断是否及格：

“及格” if 【分数】>=60 else “不及格”

如：获取两值中的较大值：

【分数 1】 if 【分数 1】>= 【分数 2】 else 【分数 2】

Python 三目运算符支持嵌套，如此可以构成更加复杂的表达式。在嵌套时需要注意 if 和 else 的配对，例如：

a if a>b else c if c>d else d

应该理解为（即：条件运算符是右结合的）：

a if a>b else ( c if c>d else d )

=====

## 条件表达式说明

条件表达式 if-else（三目运算符）

程序中出现的条件表达式说明：

1 if 条件 else 0（取 1, 为真时）

其中“条件”是需要在其下面的文字框内输入的。

如输入 “【性别】”=="男" 表示：

1 if “【性别】”=="男" else 0

结果是 取性别是男的（真）

取真时一般用于查找符合条件的项目。

1 if 条件 else 0 （取 0, 为假时）

取假时，一般用于验证查错。

如验证性别时，可以输入：“【性别】”=="男" or "【性别】"=="女"  
是男或女为真，否则为假。

包含 "孟津区" in "【学校名称】"、

截取 "【身份证号】"[0:6]、

替换 "【学校名称】".replace("孟津区", "洛阳市") 洛阳市将替换掉孟津区 注意：替换将被执行！

注意：包含、截取、替换等均为字符操作，一定要加引号！

一些数值型标签的比较条件（不要引号）：

300>【应发金额】 or 【应发金额】>=600

300<=【】<600

【】%2==0 能被 2 整除

【】%3!=0 不能被 3 整除

【】 in [500, 625, 250, 312.5] 同按数组功能

【】 not in [500, 625, 250, 312.5] 不在数组中

字符型标签的比较条件（一定要引号）：

"【性别】"=="女"

"【】"==" " 为空

"【】"==" " 为一个空格

"【性别】" in ["男", 625, "女", 312.5] 数组元素可以是不同类型的，但非数值必须加引号！

and 逻辑与（并且）、or 逻辑或（或者）、not 逻辑非（取反）