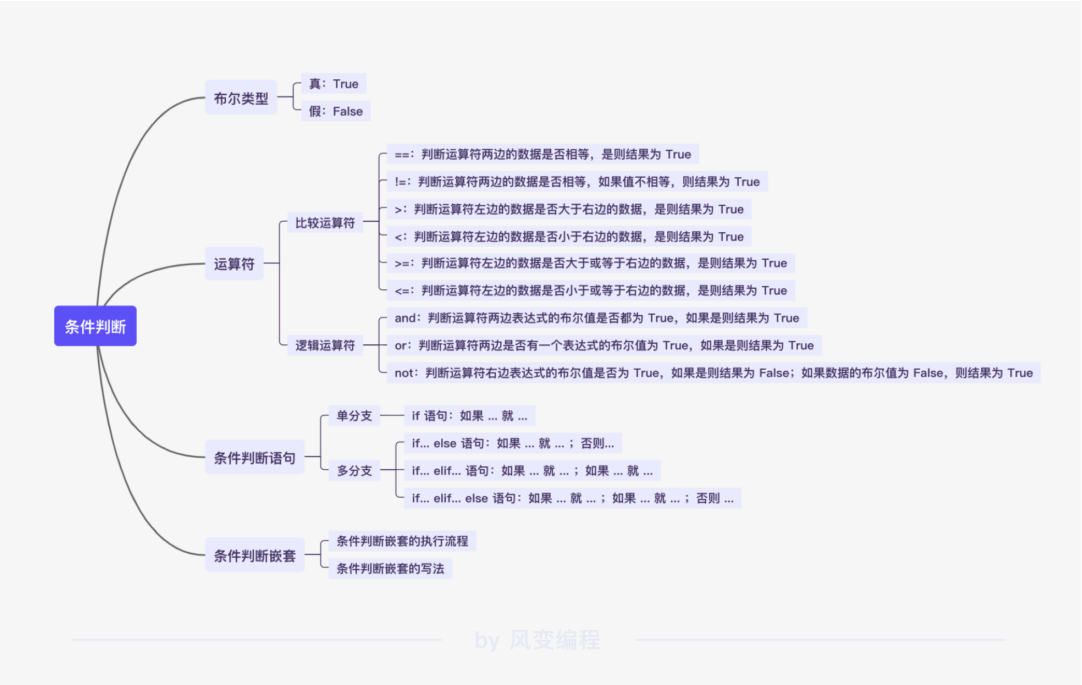


# 基础语法第3关笔记



## 一、控制流

程序的功能是解决问题。问题解决的流程并不总是线性的：有时需要做选择，有时需要重复做某件事。Python 中解决问题的流程称之为控制流。控制流包含：顺序结构、分支结构和循环结构。

### 1.顺序结构

程序从上到下执行

### 2.循环结构

程序重复执行，如for循环。

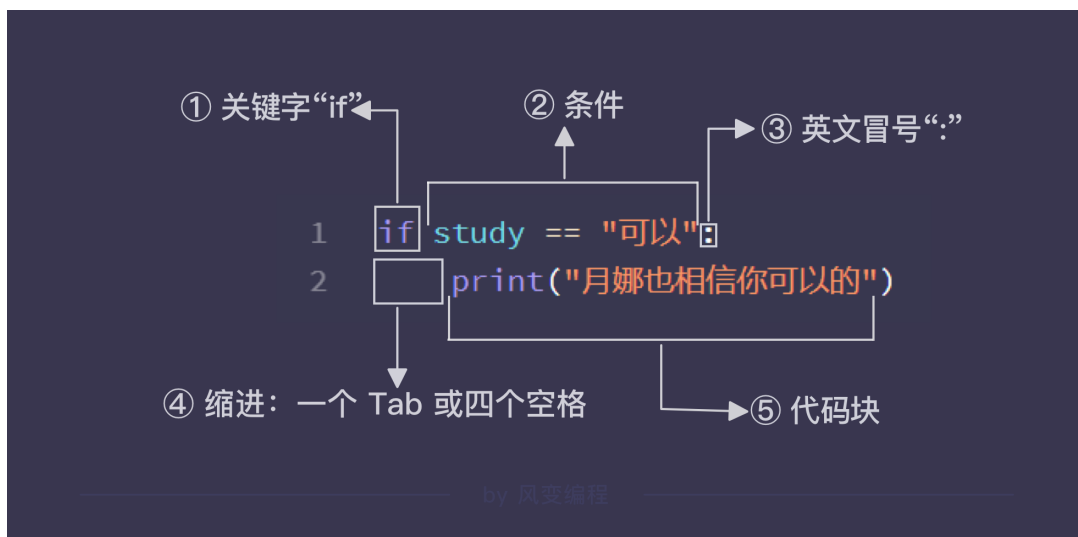
### 3.分支结构

#### • if...else...单分支结构

if...else...的意思就是“如果.....就.....， 否则.....”，如果 if 的条件为“真”，就执行 if 后面的语句。

一个 if 语句包含有五个要素：

- ① 关键词“if”；
- ② “条件”；
- ③ 英文冒号“:”；
- ④ 缩进；
- ⑤ 代码块



- 第一个注意点是关键词 `if` 后有个空格，才能加上条件，否则程序会报错；
- 第二个注意点是条件后面的冒号必须是英文冒号，否则程序会报错，并提示 “SyntaxError: invalid character in identifier”（标识符中有无效字符）；
- 第三个注意点是缩进，一个缩进是四个空格或一个 Tab 键，代码块前一定要加缩进，而且缩进不要混用 Tab 和空格。
- 第四个注意点是代码块，它和我们之前在 for 循环中了解到的代码块是一样的。相对于 for 循环的首行有相同缩进的代码是代码块。那这里相对于 if 语句的首行有相同缩进的代码，也是代码块。

#### • if...elif...else...多分支结构

### if... elif... else... 知识总结

1. [if...] 单分支结构：①关键词“if”；②条件；③英文冒号“:”；④缩进；⑤代码块；
2. [if... elif... else...] 多分支结构：如果满足 if 的条件，就执行 if 的代码块；如果满足 elif 的条件，就执行 elif 的代码块；否则就执行 else 的代码块；
3. elif需要一个相同缩进的if，且elif后可不接 else.

by 风变编程

if...elif...else...的意思是“如果...就...；不然...就...；否则...”。（一个分支结构可以存在多个 elif，且 elif 后可以不接上 else。）

elif 语句的语法格式跟 if 语句类似，只需要将关键词换为 elif。但有个地方要注意的是，elif 语句需要与同缩进的 if 语句对齐连用，不能单独使用。

正确写法：

```
point = int(input("知枫 Python 当前的分数为: "))

相同缩进 [ if point >= 90:
            print("可以写进简历")
            elif 60 <= point:
            print("可以写，但是建议先多学点知识哦")
```

正确代码 ✓

错误写法：

```
point = int(input("知枫 Python 当前的分数为: "))

elif 60 <= point:
    print("可以写，但是建议先多学点知识哦")
```

错误代码 ✗

by 风变编程

## • 条件判断嵌套

一个条件判断语句内嵌套多个条件判断语句的形式，就是我们的条件判断嵌套。

### 原代码

```
skill = input('知枫现有的技能: ')
score = int(input('该技能的分数为? '))
```

```
if skill == 'Python':

    if score > 90:
        print('Python 可以写进简历')
    else:
        print('Python 暂不可以写入简历')
```

代码块①

```
elif skill == '吉他':
```

```
    if score > 50:
        print('吉他可以写进简历')
    else:
```

代码块②

```
else:  
    print('吉他暂不可以写入简历')
```

## 概括后的代码

```
skill = input('知枫现有的技能: ')  
score = int(input('该技能的分数为? '))
```

```
if skill == 'Python':
```

(代码块①)

```
elif skill == '吉他':
```

(代码块②)

## 根据条件判断的不同结果，执行不同代码块

```
if score > 90:  
    print('Python 可以写进简历')  
else:  
    print('Python 暂不可以写入简历')
```

代码块①

```
elif skill == '吉他':
```

```
if score > 50:  
    print('吉他可以写进简历')  
else:  
    print('吉他暂不可以写入简历')
```

代码块②

by 风变编程

## 二、布尔类型（布尔值）

布尔类型（布尔值）通常用来表示条件是否成立，只有两个值：True 和 False，也就是“真”和“假”。

这个地方要注意，首字母要大写！

## 三、比较运算符

比较运算符就像是数学上的大于小于号，产生布尔值True/False的结果来告诉我们这两个数比较的结果。

运算符	描述	实例 (例 a=10, b=20)
==	等于：比较两个数据是否相等	(a == b) 结果为 False
!=	不等于：比较两个数据是否不相等	(a != b) 结果为 True
>	大于：比较左边的数据是否大于右边的数据	(a > b) 结果为 False
<	小于：比较左边的数据是否小于右边的数据	(a < b) 结果为 True
>=	大于等于：比较左边的数据是否大于等于右边的数据	(a >= b) 结果为 False
<=	小于等于：比较左边的数据是否小于等于右边的数据	(a <= b) 结果为 True

by 风变编程

#### 四、逻辑运算符

逻辑运算符用于数据的布尔值比较。运算符两边的表达式先运算得出布尔值，再执行逻辑运算符。它有这几种：and（与）、or（或）、not（非）。

可以用这三条口诀来记住它们：

and 口诀：真”与”真为真，假”与”假为假，真”与”假，假”与”真，真假相”与”不为真；

or 口诀：真”或”真为真，假”或”假为假，真”或”假，假”或”真，真假相”或”仍为真；

not 口诀：”非”真反为假，”非”假反为真，真变假，假变真，真假口诀要记清。

运算符	逻辑表达式	描述	实例
and	x and y	判断 x、y 的布尔值是否都为 True，如果是则结果为 True，否则结果为 False	1 == 1 and 2 != '2' 结果为 True
or	x or y	判断 x、y 中是否有一个数据的布尔值为 True，如果是则结果为 True，否则结果为 False	1 == '1' or 2 > 1 结果为 True
not	not x	如果 x 的布尔值为 True，则结果为 False，如果 x 的布尔值是 False，则结果为 True	not 250 < 500 结果为 False

by 风变编程