
title: "Python 第二课" tags: [Python, if, else, elif, list, del, remove(), sort(), sorted()] categories: 资料

先说一句...

- 第三周的社团活动可能会有一个德国人来参加，因此会用中文和英语讲 Python。
- 如果你需要下载或打印这篇文章，[点击这里下载 PDF 版本](#)。

这一周的总结

第二周，学习了判断语句，`if`、`if-else`和`if-elif-else`三种结构。初识列表，了解列表的初始化、删除和添加元素。

if 语句

作用

进行条件测试。

格式

条件测试表达式

每一个条件测试表达式（又称布尔表达式）的值只可能是正确，即`True`；或错误，即`False`二者中一种（注意大小写）。条件测试表达式的基本格式是：

```
var == value
```

其中`var`为一变量名，`value`“`==`”被称为相等运算符。显然，这个运算符使得只有待测试值是变量的值时，整个表达式的值才为`True`；反之则为`False`。

```
name = "lumiX"  
if name == "lumiX":  
    print("Nice!")
```

注意第三行的缩进：“`:`”的下一行在开始处需要**四个空格**（或一个制表符——按`Tab`键输入）

直接给值

由于一个条件测试表达式等价于`True`或`False`，原表达式可以用这两种值的一种或代表这两种值的一种的变量名替代，即：

```
if False:
    print("It's false!")
```

```
expr = True
if expr:
    print("It's true!")
```

其他条件运算符

这些运算符的用法同“==”。

| 符号 | 含义 |
|----|------|
| != | 不等于 |
| > | 大于 |
| < | 小于 |
| >= | 大于等于 |
| <= | 小于等于 |

显然，它们可以用于判断数字（整数、浮点数）之间的大小关系。那么像

```
if "a" > "b":
```

比较字符串符合 Python 的语法要求吗？

关键词and和or

和数学中的复合命题含义相同。这两个关键字决定了其左右两个表达式复合到一起的新表达式的值的判断。格式：

```
if expr_a and expr_b:
    # 省略剩下循环语句
```

和

```
if expr_a or expr_b:
    # 省略剩下循环语句
```

`and`左右两边的表达式值均为`True`时，复合表达式的值才为`True`；否则就是`False`。`or`左右至少一边的表达式值为`True`时，符合表达式的值就为`True`；反之则是`False`。

更复杂的结构

`if-else` 结构

```
age = 17
if age == 18:
    print("Hey! Happy Comming-of-Age Day!")
else:
    print("Your age isn't correct for the celebration!")
```

`else`后不再需要条件表达式。同样，两个`:`后的第二行都需要缩进。

`if-elif-else` 结构

顾名思义，`elif`指的是“else if”，即“其他如果”。`if`和`else`之间可以存在单个或多个`elif`。

条件的判断从前到后，即未通过第一个`if`之后才能进行第一个`elif`判断；任意一个`if`或`elif`的判断一旦被满足，程序的执行就会跳过整个`if-elif-else`结构的剩下部分。

```
score = 50
if score = 100:
    print("full marks")
elif score < 100 and score >= 90:
    print("nice")
elif score < 90 and score >= 80:
    print("so so")
else:
    print("improvement needed")
```

列表 (list)

定义

在 Python 中，变量指的是有序的、可变的元素集合。（同一个列表中）元素可为多种数据类型，如数字和字符串。

声明

类似数字和字符串的声明，列表的声明使用`=`赋值符号。以逗号（英语，半角）将元素间隔开；一头一尾用方括号包裹。声明一个全由字符串组成的列表：

```
topics = ['maker', 'arduino', 'ml', 'nl']
```

声明一个多种元素数据类型的列表：

```
mess = ['lumix', 4, 10.00, True]
```

在 Python 中，值只能为True或False的变量称为布尔型变量。因此，数组mess中的元素有字符串、数字（整数与浮点数）和布尔三种变量类型。

输出

输出整个列表

函数print()接受直接将列表作为参数。

```
print(themes)
```

引用元素

基本格式为list[num]。其中list是一已声明的列表名；num是所引用元素在该列表中的索引（角标）。

Python 中的角标从0开始。使用list[-n]引用列表list中倒数第n个元素。

将值'arduino'赋给变量name：

```
topics = ['maker', 'arduino', 'ml', 'nl']  
name = topic[1]
```

打印出第三个元素（角标为2）：

```
print(topics[2])
```

在这里倒着引用

```
print(topics[-2])
```

等价。

删除元素

del

Python 中，可以用语句del删除列表中的元素。之所以被叫做语句，是因为它没有像函数一样把参数框起来的括号。删除topics中的最后一项：

```
del topics[3]
```

remove()

注意remove方法的格式和参数。

```
topics.remove('ml')
```

排序

使用方法sort()。

```
topics.sort()
```

另外...（首尾照应）

- 请**认真看完这篇并亲自写一遍并执行了一下所有程序**的你在这个页面的评论区回复一个“名字+已完成”。（“幸好翻到了这里”，你庆幸道。😊）
- 请**大家尽量自己亲自写一遍上面的代码并执行**。编程如同刷题，不过手、当“云玩家”真的很难进步！有任何问题，**都可以**必须在下面的评论区或是 Lumix 2019 QQ群里提出来！

下周一请尽量电脑。大家周末快乐！😊