运算符优先级从低到高：

| **运算符** | **描述** |
| --- | --- |
| lambda | Lambda 表达式 |
| or | 布尔“或” |
| and | 布尔“与” |
| not x | 布尔“非” |
| in，not in | 成员测试 |
| is，is not | 同一性测试 |
| <，<=，>，>=，!=，== | 比较 |
| | | 按位或 |
| ^ | 按位异或 |
| & | 按位与 |
| <<，>> | 移位 |
| +，- | 加法与减法 |
| \*，/，% | 乘法、除法与取余 |
| +x，-x | 正负号 |
| ~x | 按位翻转 |
| \*\* | 指数 |
| x.attribute | 属性参考 |
| x[index] | 下标 |
| x[index:index] | 寻址段 |
| f(arguments...) | 函数调用 |
| (experession,...) | 绑定或元组显示 |
| [expression,...] | 列表显示 |
| {key:datum,...} | 字典显示 |
| 'expression,...' | 字符串转换 |

写注释的方式有两种，一种是单行注释，用 # 开头，另外一种是多行注释，用一对'''包裹起来。

Python 中如何避免中文是乱码

这个问题是一个具有很强操作性的问题。我这里有一个经验总结，分享一下，供参考：

首先，提倡使用 utf-8 编码方案，因为它跨平台不错。

经验一：在开头声明：

# -\*- coding: utf-8 -\*-

有朋友问我-\*-有什么作用，那个就是为了好看，爱美之心人皆有，更何况程序员？当然，也可以写成：

# coding:utf-8

经验二：遇到字符（节）串，立刻转化为 unicode，不要用 str()，直接使用 unicode()

unicode\_str = unicode('中文', encoding='utf-8')

print unicode\_str.encode('utf-8')

经验三：如果对文件操作，打开文件的时候，最好用 codecs.open，替代 open(这个后面会讲到，先放在这里)

import codecs

codecs.open('filename', encoding='utf8')

<https://github.com/qiwsir/ITArticles/blob/master/Python/Python%E7%9A%84%E4%B8%AD%E6%96%87%E6%98%BE%E7%A4%BA%E6%96%B9%E6%B3%95.md>

内建函数 hasattr()判断一个字符串是否是可迭代的

文件操作

Open read readline readlines Seek tell

Modul fileinput

帮助交互界面：

Help()