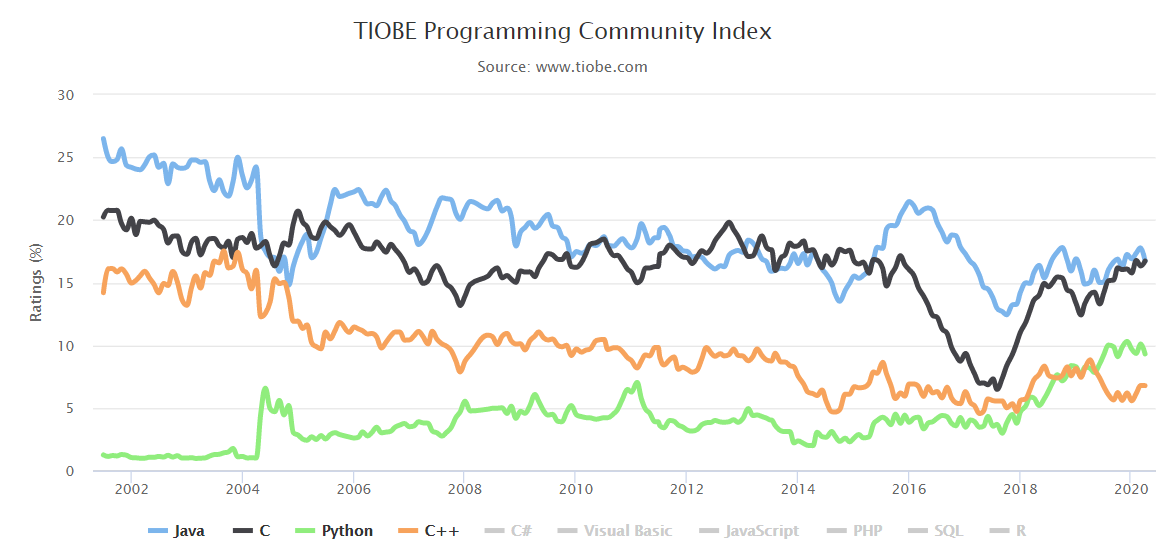
Road to Learn Python

# Intro

## What is Python

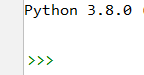
## Why using Python

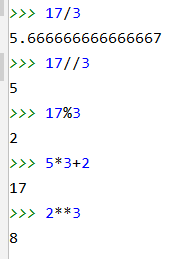


## How to use Python

### 作为计算器

通过执行Python进入解释器界面



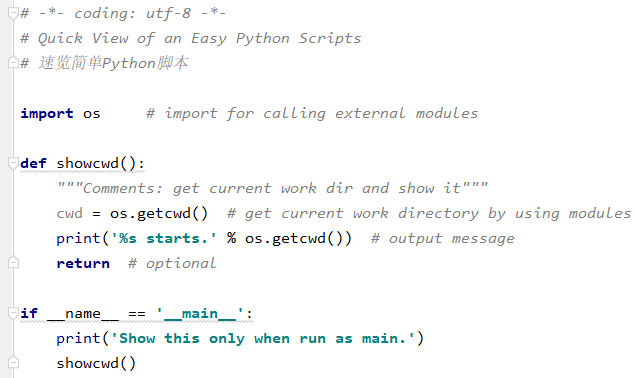


### 运算符说明

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **运算符说明** | **Python运算符** | **优先级** | **结合性** | **运算优先级** |
| 小括号 | ( ) | 19 | 无 | 最高 |
| 索引运算符 | x[i] 或 x[i1: i2 [:i3]] | 18 | 左 |  |
| 属性访问 | x.attribute | 17 | 左 |  |
| 乘方 | \*\* | 16 | 左 |  |
| 按位取反 | ~ | 15 | 右 |  |
| 符号运算符 | +（正号）、-（负号） | 14 | 右 |  |
| 乘除 | \*、/、//、% | 13 | 左 |  |
| 加减 | +、- | 12 | 左 |  |
| 位移 | >>、<< | 11 | 左 |  |
| 按位与 | & | 10 | 右 |  |
| 按位异或 | ^ | 9 | 左 |  |
| 按位或 | | | 8 | 左 |  |
| 比较运算符 | ==、!=、>、>=、<、<= | 7 | 左 |  |
| is 运算符 | is、is not | 6 | 左 |  |
| in 运算符 | in、not in | 5 | 左 |  |
| 逻辑非 | not | 4 | 右 |  |
| 逻辑与 | and | 3 | 左 |  |
| 逻辑或 | or | 2 | 左 |  |
| 逗号运算符 | exp1, exp2 | 1 | 左 |  |

### 编程示例

* Simple Demo



标准输出：



* What can this do？



# Start Program

## 导入 import

编程时，有些功能没必须自己实现，可以借助 Python 现有的标准库或者其他人提供的第三方库。

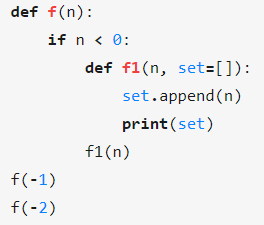
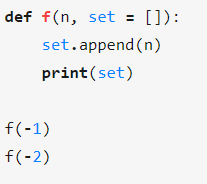
## def

关键字 def 引入一个函数 定义。

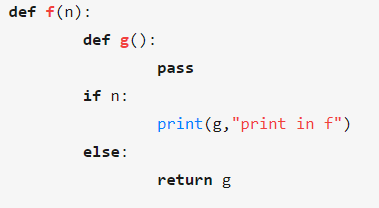
def 函数名（参数1，参数2，…）

**思考：**Python是解释性语言，是不是在每次调用def时都重新def一遍？

例1、

 V.S. 

例2、



## 调用

**if \_\_name\_\_=="\_\_main\_\_"**

Python文件通常会直接执行，或者import到其他文件中。文件作为脚本直接执行才会被执行；import 到其他脚本中是不会被执行的。

## 执行

$ python ./your\_script.py

# Data Types

## 序列 Sequence

定义：有序存放的一组数据，这组数据由一个个独立的元素组成。

包含：列表、元组、字符串、buffer对象和 xrange 对象等

特征：

1. 元素有序存放，且每个元素的位置都有一个数字标识，第一个为0，最后一个是-1，倒数第二个是-2，以此类推，元素的位置也叫索引，有时也称下标。

>>>print s1[-2]

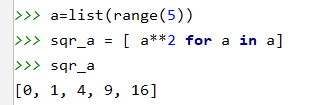
1. 通过索引操作；索引操作指能够根据索引修改或查找元素的值，也能够根据元素的值查找元素的索引。

>>> s1[0] = 3

1. 序列能够进行切片操作、加法运算、乘法运算。切片操作指能够获取序列中任意一段的数据。
2. 序列支持检查元素。能够通过len/max/min等计算出序列类型的长度、最大值和最小值。

### 列表 List

列表推导式：



内置方法：

|  |  |
| --- | --- |
| **Method** | **Explaination** |
| list.**append**(x) | 在列表的末尾添加一个元素。相当于 a[len(a):] = [x] 。 |
| list.**extend**(iterable) | 使用可迭代对象中的所有元素来扩展列表。相当于 a[len(a):] = iterable 。 |
| list.**insert**(i, x) | 在给定的位置插入一个元素。第一个参数是要插入的元素的索引，所以 a.insert(0, x) 插入列表头部， a.insert(len(a), x) 等同于 a.append(x) 。 |
| list.**remove**(x) | 移除列表中第一个值为 x 的元素。如果没有这样的元素，则抛出 [ValueError](https://docs.python.org/zh-cn/3.7/library/exceptions.html" \l "ValueError" \o "ValueError) 异常。 |
| list.**pop**([i]) | 删除列表中给定位置的元素并返回它。如果没有给定位置，a.pop() 将会删除并返回列表中的最后一个元素。（ 方法签名中 i 两边的方括号表示这个参数是可选的，而不是要你输入方括号。你会在 Python 参考库中经常看到这种表示方法)。 |
| list.**clear**() | 移除列表中的所有元素。等价于``del a[:]`` |
| list.**index**(x[, start[, end]]) | 返回列表中第一个值为 x 的元素的从零开始的索引。如果没有这样的元素将会抛出 [ValueError](https://docs.python.org/zh-cn/3.7/library/exceptions.html" \l "ValueError" \o "ValueError) 异常。  可选参数 start 和 end 是切片符号，用于将搜索限制为列表的特定子序列。返回的索引是相对于整个序列的开始计算的，而不是 start 参数。 |
| list.**count**(x) | 返回元素 x 在列表中出现的次数。 |
| list.**sort** (key=None, reverse=False) | 对列表中的元素进行排序（参数可用于自定义排序，解释请参见 [sorted()](https://docs.python.org/zh-cn/3.7/library/functions.html#sorted)）。 |
| list.**reverse**() | 翻转列表中的元素 |
| list.**copy**() | 返回列表的一个浅拷贝，等价于 a[:]。 |

|  |  |
| --- | --- |
| 练习题3-1 |  |
| 成绩统计 |  |
| 实现堆栈功能 |  |
| 矩阵行列转换 |  |

### 切片 Slice

基本样式[下限:上限:步长]

|  |  |
| --- | --- |
| Seq[0] | 序列从0开始 |
| Seq[-1] | 序列倒数第三个元素 |
| Seq[0:-1] | 范围引用，从0到倒2，上限本身不包括 |
| Seq[0:] | 范围引用，从0到最后 |
| Seq[0:5:2] | 从下标0到下标4 (下标5不包括在内)，每隔2取一个元素 （下标为0，2，4的元素） |
| Seq[-1:0:-1] | 从最后到下标1 |
| Seq[-1:0:-1] | 最后到下标0 |

### 元组 Tuple

区别：

1. 列表输出时被方括号包围，元组在输出时被圆括号包围
2. 元组是不可变值immutable，通常包含不同种类元素。字符串是一种元组

### 字符串 String

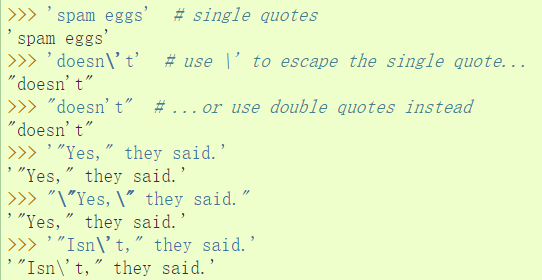
#### 1 Intro

以单引号（'...'），双引号（"..."）形式创建的数据类型。

String是一种特殊的Tuple，immutable



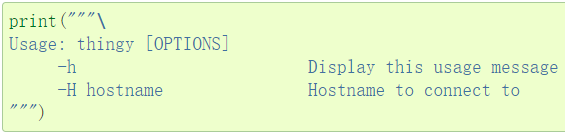
* 基本形式



* 原始字符串



* 多行字符串



#### 2 方法

* 拼接

**>>>** 3 \* 'un' + 'ium'

'unununium'

**>>>** text = ('Put several strings within parentheses '

**...**  'to have them joined together.')

**>>>** text

'Put several strings within parentheses to have them joined together.'

* 索引

**>>>** word = 'Python'

**>>>** word[0] *# character in position 0*

'P'

**>>>** word[5] *# character in position 5*

'n'

* 类C格式化

print "My name is %s and weight is %d kg!" % ('Zara', 21)

* 格式化 format

>>> "{1} {0} {1}".format("hello", "world") # 'world hello world'

>>> "{:.2s} ".format("hello") # 'he'

* 字符串比较

>>> if( "H" in "wwwwH") : print "H 在变量 a 中"

* 内置函数

|  |  |
| --- | --- |
| **方法** | **描述** |
| [string.capitalize()](https://www.runoob.com/python/att-string-capitalize.html) | 把字符串的第一个字符大写 |
| [string.center(width)](https://www.runoob.com/python/att-string-center.html) | 返回一个原字符串居中,并使用空格填充至长度 width 的新字符串 |
| [**string.count(str, beg=0, end=len(string))**](https://www.runoob.com/python/att-string-count.html) | 返回 str 在 string 里面出现的次数，如果 beg 或者 end 指定则返回指定范围内 str 出现的次数 |
| [string.decode(encoding='UTF-8', errors='strict')](https://www.runoob.com/python/att-string-decode.html) | 以 encoding 指定的编码格式解码 string，如果出错默认报一个 ValueError 的 异 常 ， 除非 errors 指 定 的 是 'ignore' 或 者'replace' |
| [string.encode(encoding='UTF-8', errors='strict')](https://www.runoob.com/python/att-string-encode.html) | 以 encoding 指定的编码格式编码 string，如果出错默认报一个ValueError 的异常，除非 errors 指定的是'ignore'或者'replace' |
| [**string.endswith(obj, beg=0, end=len(string))**](https://www.runoob.com/python/att-string-endswith.html) | 检查字符串是否以 obj 结束，如果beg 或者 end 指定则检查指定的范围内是否以 obj 结束，如果是，返回 True,否则返回 False. |
| [string.expandtabs(tabsize=8)](https://www.runoob.com/python/att-string-expandtabs.html) | 把字符串 string 中的 tab 符号转为空格，tab 符号默认的空格数是 8。 |
| [**string.find(str, beg=0, end=len(string))**](https://www.runoob.com/python/att-string-find.html) | 检测 str 是否包含在 string 中，如果 beg 和 end 指定范围，则检查是否包含在指定范围内，如果是返回开始的索引值，否则返回-1 |
| [**string.format()**](https://www.runoob.com/python/att-string-format.html) | 格式化字符串 |
| [**string.index(str, beg=0, end=len(string))**](https://www.runoob.com/python/att-string-index.html) | 跟find()方法一样，只不过如果str不在 string中会报一个异常. |
| [string.isalnum()](https://www.runoob.com/python/att-string-isalnum.html) | 如果 string 至少有一个字符并且所有字符都是字母或数字则返  回 True,否则返回 False |
| [string.isalpha()](https://www.runoob.com/python/att-string-isalpha.html) | 如果 string 至少有一个字符并且所有字符都是字母则返回 True,  否则返回 False |
| [string.isdecimal()](https://www.runoob.com/python/att-string-isdecimal.html) | 如果 string 只包含十进制数字则返回 True 否则返回 False. |
| [string.isdigit()](https://www.runoob.com/python/att-string-isdigit.html) | 如果 string 只包含数字则返回 True 否则返回 False. |
| [string.islower()](https://www.runoob.com/python/att-string-islower.html) | 如果 string 中包含至少一个区分大小写的字符，并且所有这些(区分大小写的)字符都是小写，则返回 True，否则返回 False |
| [string.isnumeric()](https://www.runoob.com/python/att-string-isnumeric.html) | 如果 string 中只包含数字字符，则返回 True，否则返回 False |
| [string.isspace()](https://www.runoob.com/python/att-string-isspace.html) | 如果 string 中只包含空格，则返回 True，否则返回 False. |
| [string.istitle()](https://www.runoob.com/python/att-string-istitle.html) | 如果 string 是标题化的(见 title())则返回 True，否则返回 False |
| [string.isupper()](https://www.runoob.com/python/att-string-isupper.html) | 如果 string 中包含至少一个区分大小写的字符，并且所有这些(区分大小写的)字符都是大写，则返回 True，否则返回 False |
| [**string.join(seq)**](https://www.runoob.com/python/att-string-join.html) | 以 string 作为分隔符，将 seq 中所有的元素(的字符串表示)合并为一个新的字符串 |
| [string.ljust(width)](https://www.runoob.com/python/att-string-ljust.html) | 返回一个原字符串左对齐,并使用空格填充至长度 width 的新字符串 |
| [string.lower()](https://www.runoob.com/python/att-string-lower.html) | 转换 string 中所有大写字符为小写. |
| [string.lstrip()](https://www.runoob.com/python/att-string-lstrip.html) | 截掉 string 左边的空格 |
| [string.maketrans(intab, outtab])](https://www.runoob.com/python/att-string-maketrans.html) | maketrans() 方法用于创建字符映射的转换表，对于接受两个参数的最简单的调用方式，第一个参数是字符串，表示需要转换的字符，第二个参数也是字符串表示转换的目标。 |
| [max(str)](https://www.runoob.com/python/att-string-max.html) | 返回字符串 str 中最大的字母。 |
| [min(str)](https://www.runoob.com/python/att-string-min.html) | 返回字符串 str 中最小的字母。 |
| [**string.partition(str)**](https://www.runoob.com/python/att-string-partition.html) | 有点像 find()和 split()的结合体,从 str 出现的第一个位置起,把 字 符 串 string 分 成 一 个 3 元 素 的 元 组 (string\_pre\_str,str,string\_post\_str),如果 string 中不包含str 则 string\_pre\_str == string. |
| [**string.replace(str1, str2,  num=string.count(str1))**](https://www.runoob.com/python/att-string-replace.html) | 把 string 中的 str1 替换成 str2,如果 num 指定，则替换不超过 num 次. |
| [string.rfind(str, beg=0,end=len(string) )](https://www.runoob.com/python/att-string-rfind.html) | 类似于 find()函数，不过是从右边开始查找. |
| [string.rindex( str, beg=0,end=len(string))](https://www.runoob.com/python/att-string-rindex.html) | 类似于 index()，不过是从右边开始. |
| [string.rjust(width)](https://www.runoob.com/python/att-string-rjust.html) | 返回一个原字符串右对齐,并使用空格填充至长度 width 的新字符串 |
| [string.rpartition(str)](https://www.runoob.com/python/att-string-rpartition.html) | 类似于 partition()函数,不过是从右边开始查找 |
| [string.rstrip()](https://www.runoob.com/python/att-string-rstrip.html) | 删除 string 字符串末尾的空格. |
| [**string.split(str="", num=string.count(str))**](https://www.runoob.com/python/att-string-split.html) | 以 str 为分隔符切片 string，如果 num 有指定值，则仅分隔 num+ 个子字符串 |
| [string.splitlines([keepends])](https://www.runoob.com/python/att-string-splitlines.html) | 按照行('\r', '\r\n', \n')分隔，返回一个包含各行作为元素的列表，如果参数 keepends 为 False，不包含换行符，如果为 True，则保留换行符。 |
| [string.startswith(obj, beg=0,end=len(string))](https://www.runoob.com/python/att-string-startswith.html) | 检查字符串是否是以 obj 开头，是则返回 True，否则返回 False。如果beg 和 end 指定值，则在指定范围内检查. |
| [**string.strip([obj])**](https://www.runoob.com/python/att-string-strip.html) | 在 string 上执行 lstrip()和 rstrip() |
| [string.swapcase()](https://www.runoob.com/python/att-string-swapcase.html) | 翻转 string 中的大小写 |
| [string.title()](https://www.runoob.com/python/att-string-title.html) | 返回"标题化"的 string,就是说所有单词都是以大写开始，其余字母均为小写(见 istitle()) |
| [**string.translate(str, del="")**](https://www.runoob.com/python/att-string-translate.html) | 根据 str 给出的表(包含 256 个字符)转换 string 的字符,  要过滤掉的字符放到 del 参数中 |
| [string.upper()](https://www.runoob.com/python/att-string-upper.html) | 转换 string 中的小写字母为大写 |
| [string.zfill(width)](https://www.runoob.com/python/att-string-zfill.html) | 返回长度为 width 的字符串，原字符串 string 右对齐，前面填充0 |

#### 3 习题

|  |  |
| --- | --- |
| 练习题3-5 |  |
| 去除字符串首位空格 | Trim() |
|  |  |

### 循环遍历 Loop through Sequence

* 在一个序列中循环时：用enumerate（）函数可以将索引和值同时取出

**>>> for** i, v **in** enumerate(['tic', 'tac', 'toe']):

**...**  print(i, v)

* 两个或更多序列中循环时：用zip（）函数将其内元素一一对应

**>>> for** q, a **in** zip(questions, answers):

**...**  print('What is your *{0}*? It is *{1}*.'.format(q, a))

* 逆向循环

**>>> for** i **in** reversed(range(1, 10, 2)):

**...**  print(i)

* 按某个顺序循环

**>>> for** f **in** sorted(set(basket)):

**...**  print(f)

|  |  |
| --- | --- |
| 练习题3-2 |  |
| MEM\_DUMP | 输出： |

## 集合 Set

定义：由不重复元素组成的无序的集。

用法：

成员检测和消除重复元素。

支持像 联合，交集，差集，对称差分等数学运算。

**>>>** a = set('abracadabra')

**>>>** b = set('alacazam')

**>>>** a *# unique letters in a*

{'a', 'r', 'b', 'c', 'd'}

**>>>** a - b *# letters in a but not in b*

{'r', 'd', 'b'}

**>>>** a | b *# letters in a or b or both*

{'a', 'c', 'r', 'd', 'b', 'm', 'z', 'l'}

**>>>** a & b *# letters in both a and b*

{'a', 'c'}

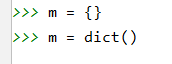
**>>>** a ^ b *# letters in a or b but not both*

{'r', 'd', 'b', 'm', 'z', 'l'}

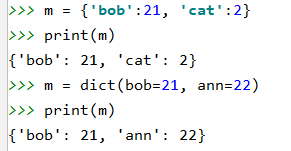
## 字典 Dict

### 初始化

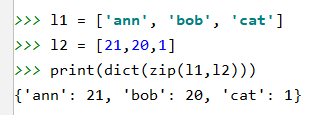
* 空Dict



* 带值初始化



* 两个List转换为Dict



### 循环遍历 Loop through Dict

**>>> for** k, v **in** my\_dict.items():

**...**  print(k, v)

### 其他方法

# Flow Control

## IF语句

**>>> if** x < 0:

**...**  x = 0

**...**  print('Negative changed to zero')

**... elif** x == 0:

**...**  print('Zero')

**... elif** x == 1:

**...**  print('Single')

**... else**:

**...**  print('More')

## for语句

words = ['cat', 'window', 'defenestrate']

**>>> for** w **in** words:

**...**  print(w, len(w))

## range()

**>>> for** i **in** range(5):

**...**  print(i)

要以序列的索引来迭代，您可以将 [range()](https://docs.python.org/zh-cn/3.7/library/stdtypes.html#range) 和 [len()](https://docs.python.org/zh-cn/3.7/library/functions.html" \l "len" \o "len) 组合如下:

**>>>** a = ['Mary', 'had', 'a', 'little', 'lamb']

**>>> for** i **in** range(len(a)):

**...**  print(i, a[i])

大多数这类情况下，使用 [enumerate()](https://docs.python.org/zh-cn/3.7/library/functions.html#enumerate) 函数比较方便

## 循环中的break和continue

### Break

用于跳出最近的 for 或 while 循环

**>>> for** n **in** range(2, 10):

**...**  **for** x **in** range(2, n):

**...**  **if** n % x == 0:

**...**  print(n, 'equals', x, '\*', n//x)

**...**  **break**

### Continue

表示继续循环中的下一次迭代

**>>> for** num **in** range(2, 10):

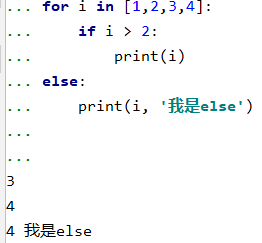
**...**  **if** num % 2 == 0:

**...**  print("Found an even number", num)

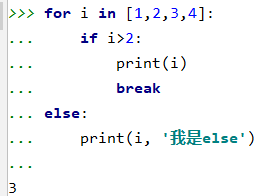
**...**  **continue**

**...**  print("Found a number", num)

### for循环中的else



如何拆散 for 和 else 这对冤家呢？只有当 for 循环被 break 语句中断之后，才会跳过 else 语句



## pass

# Input/Output

## OUTPUT to stdout

|  |  |
| --- | --- |
| 练习题4-1 |  |
| 打印9x9乘法表 |  |
| 杨辉三角 |  |
| Fibonacci |  |

## INPUT from stdin

# Class 类

# import

## 模块Module

有些功能没必须自己实现，可以借助 Python 现有的标准库或者其他人提供的第三方库。

1. import 模块名1 [as 别名1], 模块名2 [as 别名2]，…

使用这种语法格式的 import 语句，会导入指定模块中的所有成员（包括变量、函数、类等）。不仅如此，当需要使用模块中的成员时，需用该模块名（或别名）作为前缀，否则 Python 解释器会报错。

1. from 模块名 import 成员名1 [as 别名1]，成员名2 [as 别名2]，…

使用这种语法格式的 import 语句，只会导入模块中指定的成员，而不是全部成员。同时，当程序中使用该成员时，无需附加任何前缀，直接使用成员名（或别名）即可。

## 模块Module

### 标准模块

### 第三方模块

### 自定义模块

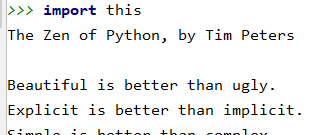
## 包Package

## 运行机制

Python在执行import语句？

# Funny Python

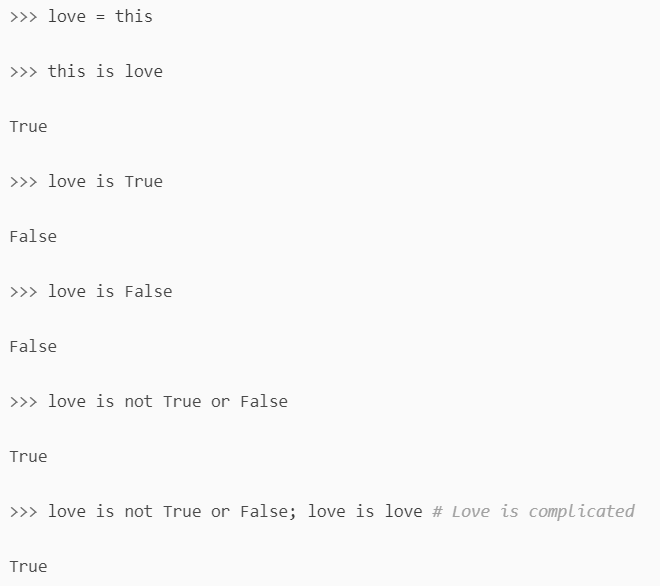
## The Zen of Python



## Quine



## Love is Complicated



# Reference

<https://docs.python.org/zh-cn/3.7/tutorial/index.html>