变量：变量是可以被赋值的标签

字符串：（string）就是一系列字符，引号可以是双引号，也可以是单引号，三个引号可以创建跨行字符串，一个引号的写法则不能跨行（补充：一个引号的写法也可以跨行，只需要在每行的末尾加入表示连接的字符（\）即可，但这种写法并不会保留换行，而三个引号的写法将会保留括起的所有内容，两者均会包括开头和末尾添加的额外空白）

字符串

upper()：将字符串全部改为大写  
 lower()：将字符串全部改为小写。  
 title()：以首字母大写的方式显示每个单词

f--字符串字符串：在 f-字符串中，使用花括号来引用代码中定义的变 量f 是 format 的简写

特殊的字符来添加空白：•\t：在字符串中表示制表符（看着比空格要长 一些）

\n：在字符串中表示换行符

\t：在字符串中表示制表符（长空白）

可以使用以下方法来删除空白：

•lstrip()：移除左端的空白•

rstrip()：移除右端的空白

•strip() ：移除两端的空白  
 删除空白的方法不会改变原字符串

字符串的简单拼接：使用 + 号来拼接字符串

使用 \* 号来生成重复的字符串

同时给多个变量赋值:

x, y, z = 0, 0, 0

(书写的顺序就是分配的顺序)

整数和浮点数：

加减乘除：加（+）减（-）乘（\*）除（/）

（只要运算过程中有浮点数，结果就会是浮点数）  
 （除法的结果总是浮点数）

双斜杠（//）：表示整数除法

双乘号（\*\*）：表示乘方运算

（浮点数与浮点数间也是可以做乘方运算）

布尔值（BooleanBoolean）

用于表示逻辑真或假

•True： 表示真

•False：表示假

常量（constantconstant）

常量：程序内一直保持不变的变量

Python 没有内置的常量支持，我们给出的是一个约定俗成的惯例

•使用全大写字母（单词由下划线分割）来将某个变量视为常量

**MAX\_CONNECTIONS = 5000**

注释（CommentComment）

注释：让你能够使用自然语言在程序中添加说明

•Python 中使用（#）标识注释

•解释器会忽略注释的内容

精度丢失原因：

浮点数是以二进制形式存储的，将浮点数转化为二进制过程中，也会出现无限循环的情况，造成结果的不准确

方法（method）：实例名.方法名(...)  
函数（function）：函数名(...)

可以把方法视为特殊的函数

列表

列表（list）由一系列按特定顺序排列的元素组成

使用方括号来标识，用逗号分隔其中的元素，元素之间可以没有任何关系

索引（index）：元素在列表中的位置，用于访问列表中的元素

负数索引：表示列表中倒数第几个元素的位置

• lst.append(val) ：在列表的末尾添加新元素。

• lst.insert(idx, val)：在列表的指定位置上添加新元素。

• del lst[idx] ：删除列表中指定索引的元素

• lst.pop([idx]) -> val：删除并返回列表中指定索引（默认为末尾） 的元素，弹出（pop）的值能被接着使用

lst.remove(val)：删除列表中匹配到的第一个指定元素。

lst.sort() ：永久修改原列表，对其中的元素进行排序。

• sorted(lst) -> lst'：返回排序后的列表的副本。

# 列表排序（指定倒序）nums.sort(reverse=True)

lst.reverse()：永久修改原列表，对其中的元素进行翻转

len() 函数可以快速获悉列表的长度（len() 是一个 Python 的内置函数

）

在循环中使用单数和复数形式的名称，可帮助你判断代码段处理的是单个列表元素还是整个列表。

range([start,] end [,step])：生成可迭代的数值列表的表示

我们还可以利用列表推导式直接生成列表：运行的结果是相同的：squares = [value\*\*2 for value in range(1, 11)]

max(lst)：取数值列表中的最大值。

•min(lst)：取数值列表中的最小值。

•sum(lst)：对数值列表执行求和计算  
  
：处理列表中部分元素的语法我们可以通过在索引中添加冒号（:）来获取部分列表

如果没有指定起始索引，Python 将自动从列表开头开始

切片操作总是返回列表的拷贝

经过切片操作，虽然两个列表看起来已经相互独立，但要注意这里的拷贝实际上是浅拷贝

不使用切片，实际上不能起到拷贝后相互独立的效果

元组（tupletuple）

元组：不可变的列表，使用圆括号而不是方括号来标识

也可以通过使用 for 循环来遍历元组中的元素

If

car = 'Audi'

print(car.lower() == 'audi')

代码使用了常见小技巧：在比较时忽略大小写

•and、or 关键字：用于布尔运算，分别表示和/或。

•not 关键字：用于布尔运算，表示非。

•in 关键字：用于检查特定值是否在集合中。

条件测试：布尔运算布尔表达式：条件测试的别名，结果不是 True 就是 False布尔值变量：存储布尔表达式结果的变量

字典

字典（dictionary）是一系列键值对，使用花括号来标识。•每个键都与一个值关联，可以使用键来访问与之关联的值。•键在字典中是唯一的，键和值之间用冒号分割  
 使用 del 语句将指定字典名中要删除的键相应的键值对彻底删除：  
 del alien\_0['points']

如果指定的键不存在，之前的访问方式将会给出键错误（KeyError)

我们可以使用 get() 方法来避免错误的发生

•d.get(key, [default])：根据键获取值

•d.items() ：返回所有键值对的元组视图。

•d.keys() ：返回所有键的列表视图。

•d.values()：返回所有值的列表视图

嵌套：字典列表

在字典中存储列表

在字典中存储字典

用户输入和 whilewhile 循环循

求模运算符（%）：将两个数相除并返回余数。

for 循环用于针对集合中的每个元素执行一个代码块，而 while 循环则不断地运行，直到指定的条件不再满足为止。

break 关键字：立即退出循环

continue 关键字：跳过当前的循环，返回到循环的开头

函数

函数（function）：带名字的代码块，用于完成具体的工作要执行函数定义的特定任务，可调用（call）该函数

对象