循环

循环

要计算1+2+3，我们可以直接写表达式：

>>> 1 + 2 + 3

6

要计算1+2+3+...+10，勉强也能写出来。

但是，要计算1+2+3+...+10000，直接写表达式就不可能了。

为了让计算机能计算成千上万次的重复运算，我们就需要循环语句。

Python的循环有两种，一种是for...in循环，依次把list或tuple中的每个元素迭代出来，看例子：

names = ['Michael', 'Bob', 'Tracy']

**for** name **in** names:

print(name)

执行这段代码，会依次打印names的每一个元素：

Michael

Bob

Tracy

所以for x in ...循环就是把每个元素代入变量x，然后执行缩进块的语句。

再比如我们想计算1-10的整数之和，可以用一个sum变量做累加：

sum = 0

**for** x **in** [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]:

sum = sum + x

print(sum)

如果要计算1-100的整数之和，从1写到100有点困难，幸好Python提供一个range()函数，可以生成一个整数序列，再通过list()函数可以转换为list。比如range(5)生成的序列是从0开始小于5的整数：

>>> list(range(5))

[0, 1, 2, 3, 4]

range(101)就可以生成0-100的整数序列，计算如下：

sum = 0

**for** x **in** range(101):

sum = sum + x

print(sum)

请自行运行上述代码，看看结果是不是当年高斯同学心算出的5050。

第二种循环是while循环，只要条件满足，就不断循环，条件不满足时退出循环。比如我们要计算100以内所有奇数之和，可以用while循环实现：

sum = 0

n = 99

**while** n > 0:

sum = sum + n

n = n - 2

**print**(sum)

在循环内部变量n不断自减，直到变为-1时，不再满足while条件，循环退出。

练习

请利用循环依次对list中的每个名字打印出Hello, xxx!：

窗体顶端

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

L = ['Bart', 'Lisa', 'Adam']

窗体底端

小结

循环是让计算机做重复任务的有效的方法，有些时候，如果代码写得有问题，会让程序陷入“死循环”，也就是永远循环下去。这时可以用Ctrl+C退出程序，或者强制结束Python进程。

请试写一个死循环程序。