链表推导式   
  
链表推导式提供了一个创建链表的简单途径，无需使用 map()， filter() 以及 lambda。返回链表的定义通常要比创建这些链表更清晰。每一个链表推导式包括在一个for语句之后的表达式，零或多个for或[if语句](https://www.baidu.com/s?wd=if%E8%AF%AD%E5%8F%A5&tn=SE_PcZhidaonwhc_ngpagmjz&rsv_dl=gh_pc_zhidao" \t "_blank)。返回值是由for或if子句之后的表达式得到的元素组成的链表。如果想要得到一个元组，必须要加上括号。  
  
>>> freshfruit = [' banana', ' loganberry ', 'passion fruit ']  
>>> [weapon.strip() for weapon in freshfruit]  
['banana', 'loganberry', 'passion fruit']  
>>> vec = [2, 4, 6]  
>>> [3\*x for x in vec]  
[6, 12, 18]  
>>> [3\*x for x in vec if x > 3]  
[12, 18]  
>>> [3\*x for x in vec if x < 2]  
[]  
>>> [[x,x\*\*2] for x in vec]  
[[2, 4], [4, 16], [6, 36]]  
>>> [x, x\*\*2 for x in vec] # error - parens required for tuples  
File "<stdin>", line 1, in ?  
[x, x\*\*2 for x in vec]  
^  
SyntaxError: invalid syntax  
>>> [(x, x\*\*2) for x in vec]  
[(2, 4), (4, 16), (6, 36)]  
>>> vec1 = [2, 4, 6]  
>>> vec2 = [4, 3, -9]  
>>> [x\*y for x in vec1 for y in vec2]  
[8, 6, -18, 16, 12, -36, 24, 18, -54]  
>>> [x+y for x in vec1 for y in vec2]  
[6, 5, -7, 8, 7, -5, 10, 9, -3]  
>>> [vec1[i]\*vec2[i] for i in range(len(vec1))]  
[8, 12, -54]  
为使链表推导式匹配[for循环](https://www.baidu.com/s?wd=for%E5%BE%AA%E7%8E%AF&tn=SE_PcZhidaonwhc_ngpagmjz&rsv_dl=gh_pc_zhidao)的行为，可以在推导之外保留循环变量：  
  
>>> x = 100 # this gets overwritten  
>>> [x\*\*3 for x in range(5)]  
[0, 1, 8, 27, 64]  
>>> x # the final value for range(5)