**1、现在有一个包含 N 个元素的元组或者是序列，怎样将它里面的值解压后同时赋值？**

data = [ 'ACME', 50, 91.1, (2012, 12, 21) ]

name, shares, price, date = data

此外括字符串，文件对象，迭代器和生成器也可以这样操作

str = 'xyz'

a, b, c = str # 等号左边的对象个数必须和str的字符数一样多

有时候，你可能只想解压一部分，丢弃其他的值。对于这种情况 Python 并没有提供特殊的语法。但是你可以使用任意变量名去占位，到时候丢掉这些变量就行了。

data = [ 'ACME', 50, 91.1, (2012, 12, 21) ]

\_, shares, price, \_ = data

实际上 \_ 也是也是一个对象名，根据命名规则命名为 \_ 并不违反规定，甚至可以 print(\_)，这里紧作为站位，非常形象

注意：你必须保证你选用的那些占位变量名在其他地方没被使用到。

（*可迭代对象通过******\_\_iter\_\_******⽅法向我们提供⼀个迭代器，我们在迭代⼀个可迭代对象的时候，实际上就是先获取该对象提供的⼀个迭代器，然后通过这个迭代器来依次读取对象中的每⼀个数据。那么也就是说，⼀个具备了******\_\_iter\_\_******⽅法的对象，就是⼀个可迭代对象。*）

1. 如果一个可迭代对象的元素个数超过变量个数时，会抛出一个 ValueError 。那么怎样才能从这个可迭代对象中解压出 N 个元素出来？

def drop\_first\_last(grades):

first, \*middle, last = grades

return avg(middle)

也可以用同样的原理： first second \*other