

序列的增量赋值

不要去控制台运行代码，想一下下面的表达式会产生什么结果：

```
>>> t = (1, 2, [30, 40])
>>> t[2] += [50, 60]
```

1. t 变成 (1, 2, [30, 40, 50, 60])
2. 因为 tuple 不支持对它的元素赋值，所以会抛出 `TypeError` 异常。
3. 以上两个都不是。
4. 1 和 2 都是对的。

增量赋值运算符 `+=` 和 `*=` 的表现取决于它们的第一个操作对象。

`+=` 背后的特殊方法是 `__iadd__`（用于“就地加法”）。如果一个类没有实现这个方法的话，Python会退一步调用 `__add__`。

```
>>> a += b
```

如果 `a` 实现了 `__iadd__` 方法，就会调用这个方法。同时对可变序列（如list、bytearray和array.array）来说，`a` 会就地改动，就像调用了 `a.extend(b)` 一样。否则将等同于 `a = a + b`。

变量名会不会关联到新的对象，完全取决于这个类型有没有实现 `__iadd__` 这个方法。

总体来讲，可变序列一般都实现了 `__iadd__` 方法，因此 `+=` 是就地加法。而不可变序列根本就不支持这个操作，对这个方法的实现就无从谈起。

对不可变序列进行重复拼接操作的话，效率会很低，因为每一次都有一个新对象，解释器需要把原来对象中的元素先复制到新的对象里，然后再追加新的元素。

2个提醒：

1. 不要把可变对象放在元组里面。
2. 增量赋值不是一个原子操作。