Desconocido

### Aritmética de listas

Podemos unir listas sumándolas, igual que sumamos números, utilizando el signo más ( + ). Por ejemplo, supongamos que tenemos dos listas: list1 , que contiene los números del 1 al 4, y list2 , que contiene algunas palabras. Podemos sumarlas utilizando el signo + , así:

>>> list1 = [1, 2, 3, 4]  
>>> list2 = ['I', 'tripped', 'over', 'and', 'hit', 'the', 'floor']  
>>> print(list1 + list2)  
[1, 2, 3, 4, 'I', 'tripped', 'over', 'and', 'hit', 'the', 'floor']

También podemos sumar las dos listas y asignar el resultado a otra variable:

>>> list1 = [1, 2, 3, 4]  
>>> list2 = ['I', 'ate', 'chocolate', 'and', 'I', 'want', 'more']  
>>> list3 = list1 + list2  
>>> print(list3)  
[1, 2, 3, 4, 'I', 'ate', 'chocolate', 'and', 'I', 'want', 'more']

Y podemos multiplicar una lista por un número. Por ejemplo, para multiplicar list1 por 5, escribimos list1 \* 5 :

>>> list1 = [1, 2]  
>>> print(list1 \* 5)  
[1, 2, 1, 2, 1, 2, 1, 2, 1, 2]

Esto le dice a Python que repita list1 cinco veces, dando como resultado 1, 2, 1, 2, 1, 2, 1, 2, 1, 2 . En cambio, la división ( / ) y la resta ( - ) sólo dan errores, como en estos ejemplos:

>>> list1 / 20  
Traceback (most recent call last):  
File "<pyshell>", line 1, in <module>  
 list1 / 20  
TypeError: unsupported operand type(s) for /: 'list' and 'int'  
  
>>> list1 - 20  
Traceback (most recent call last):  
File "<pyshell>", line 1, in <module>  
 list1 - 20  
TypeError: unsupported operand type(s) for -: 'list' and 'int'

Pero, ¿por qué? Bueno, unir listas con + y repetir listas con \* son operaciones bastante sencillas. Estos conceptos también tienen sentido en el mundo real. Por ejemplo, si te diera dos listas de la compra en papel y te dijera: "Suma estas dos listas", podrías escribir todos los elementos en otra hoja de papel en orden, de punta a punta. Lo mismo ocurriría si te dijera: "Multiplica esta lista por 3". Podrías imaginarte escribiendo una lista con todos los elementos de la lista tres veces en otra hoja de papel.

Pero, ¿cómo dividirías una lista? Por ejemplo, piensa cómo dividirías en dos una lista de seis números (del 1 al 6). Aquí tienes tres formas diferentes:

[1, 2, 3] [4, 5, 6]  
[1] [2, 3, 4, 5, 6]  
[1, 2, 3, 4] [5, 6]

¿Dividiríamos la lista por la mitad, la dividiríamos después del primer elemento, o simplemente elegiríamos un lugar al azar y la dividiríamos ahí? No hay una respuesta sencilla, y cuando le pides a Python que divida una lista, tampoco sabe qué hacer. Por eso responde con un error.



Por la misma razón, no puedes añadir nada que no sea una lista a una lista. Por ejemplo, esto es lo que ocurre cuando intentamos añadir el número 50 a list1 :

>>> list1 + 50  
Traceback (most recent call last):  
 File "<pyshell>", line 1, in <module>  
 list1 + 50  
TypeError: can only concatenate list (not "int") to list

¿Por qué se produce un error? Bien, ¿qué significa añadir 50 a una lista? ¿Significa añadir 50 a cada elemento? ¿Y si los elementos no son números? ¿Significa añadir el número 50 al final o al principio de la lista?

En programación informática, los comandos deben funcionar exactamente igual cada vez que los introduces. Tu ordenador sólo ve las cosas en blanco y negro. Pídele que tome una decisión complicada, y se echará las manos a la cabeza con errores.

[anterior](ch03_9.html)[Subtema 10 de 14: (Ver todo)](ch03.html)[siguiente](ch03_11.html)