Desconocido

### Las listas son más potentes que las cadenas

"Ancas de araña, dedo de rana, ala de murciélago, mantequilla de babosa y caspa de serpiente" no es una lista de la compra muy normal (a menos que seas un mago), pero la utilizaremos como primer ejemplo de las diferencias entre cadenas y listas. Podríamos almacenar esta lista de artículos en la variable wizard\_list utilizando una cadena como ésta:



>>> wizard\_list = 'spider legs, toe of frog, bat wing, slug butter, snake dandruff'  
>>> print(wizard\_list)  
spider legs, toe of frog, bat wing, slug butter, snake dandruff

Pero también podríamos crear una *lista* , un objeto Python un tanto mágico que podemos manipular. Así es como se verían estos elementos escritos como una lista:

>>> wizard\_list = ['spider legs', 'toe of frog', 'bat wing',  
 'slug butter', 'snake dandruff']  
>>> print(wizard\_list)  
['spider legs', 'toe of frog', 'bat wing', 'slug butter',  
'snake dandruff']

Para crear una lista hay que teclear un poco más que para crear una cadena, pero una lista es más útil que una cadena porque los elementos de la lista se pueden manipular fácilmente. Podemos imprimir un elemento de la lista introduciendo un número (llamado *posición del índice* ) entre corchetes, así:

>>> print(wizard\_list[2])  
bat wing

Si esperabas que el segundo elemento fuera ‘toe of frog’ , puede que te preguntes por qué se ha impreso ‘bat wing’ . Esto es  porque las listas empiezan en la posición índice 0, de modo que el primer elemento de una lista es 0, el segundo es 1 y el **tercero** es 2. Puede que esto no tenga mucho sentido para los humanos, pero sí para los ordenadores.

También podemos cambiar un elemento de una lista. Quizá nuestro amigo mago nos acaba de decir que tenemos que cogerle lengua de caracol en vez de ala de murciélago. Así es como cambiaríamos el elemento de nuestra lista:

>>> wizard\_list[2] = 'snail tongue'  
>>> print(wizard\_list)  
['spider legs', 'toe of frog', 'snail tongue', 'slug butter',  
'snake dandruff']

Esto cambia el elemento en la posición 2 del índice, que antes era ala de murciélago, a lengua de caracol.

También podemos mostrar una sublista de los elementos de la lista. Para ello, utiliza dos puntos ( : ) dentro de los corchetes. Por ejemplo, escribe lo siguiente para ver los elementos tercero a quinto de nuestra lista (un brillante conjunto de ingredientes para un delicioso bocadillo):



>>> print(wizard\_list[2:5])  
['snail tongue', 'slug butter', 'snake dandruff']

Escribir [2:5] equivale a decir: "Muestra los elementos desde la posición índice 2 hasta (pero sin incluir) la posición índice 5", es decir, los elementos 3, 4 y 5.

Las listas pueden utilizarse para almacenar todo tipo de elementos, como números:

>>> some\_numbers = [1, 2, 5, 10, 20]

También pueden contener cadenas:

>>> some\_strings = ['Which', 'Witch', 'Is', 'Which']

Pueden tener mezclas de números y cadenas:

>>> numbers\_and\_strings = ['Why', 'was', 6, 'afraid', 'of', 7,   
 'because', 7, 8, 9]  
>>> print(numbers\_and\_strings)  
['Why', 'was', 6, 'afraid', 'of', 7, 'because', 7, 8, 9]

Y las listas pueden incluso almacenar otras listas:

>>> numbers = [1, 2, 3, 4]  
>>> strings = ['I', 'kicked', 'my', 'toe', 'and', 'it',   
 'is', 'sore']  
>>> mylist = [numbers, strings]  
>>> print(mylist)  
[[1, 2, 3, 4], ['I', 'kicked', 'my', 'toe', 'and', 'it',  
'is', 'sore']]

Este ejemplo de lista dentro de lista crea tres variables: numbers con cuatro números, strings con ocho cadenas y mylist con numbers y strings . La tercera lista ( mylist ) sólo tiene dos elementos porque es una lista de nombres de variables, no del contenido de las variables.

Podemos intentar imprimir los dos elementos de mylist por separado:

>>> print(mylist[0])  
[1, 2, 3, 4]  
>>> print(mylist[1])  
['I', 'kicked', 'my', 'toe', 'and', 'it', 'is', 'sore']

Aquí podemos ver que mylist[0] contiene la lista de números, y mylist[1] es la lista de cadenas.

[anterior](ch03_6.html)[Subtema 7 de 14: (Ver todo)](ch03.html)[siguiente](ch03_8.html)