### Definir una tupla

Una tupla tiene el mismo aspecto que una lista, salvo que utilizas paréntesis en lugar de corchetes. Una vez definida una tupla, puedes acceder a los elementos individuales utilizando el índice de cada elemento, igual que harías con una lista.

Por ejemplo, si tenemos un rectángulo que siempre debe tener un tamaño determinado, podemos asegurarnos de que su tamaño no cambie poniendo las dimensiones en una tupla:

**dimensions.py**

dimensions = (200, 50)  
print(dimensions[0])  
print(dimensions[1])

Definimos la tupla dimensions, utilizando paréntesis en lugar de corchetes. A continuación, imprimimos cada elemento de la tupla individualmente, utilizando la misma sintaxis que hemos estado utilizando para acceder a los elementos de una lista:

200  
50

Veamos qué ocurre si intentamos cambiar uno de los elementos de la tupla dimensions:

dimensions = (200, 50)  
dimensions[0] = 250

Este código intenta cambiar el valor de la primera dimensión, pero Python devuelve un error de tipo. Como estamos intentando modificar una tupla, cosa que no se puede hacer con ese tipo de objeto, Python nos dice que no podemos asignar un nuevo valor a un elemento de una tupla:

Traceback (most recent call last):  
 File "dimensions.py", line 2, in <module>  
 dimensions[0] = 250  
TypeError: 'tuple' object does not support item assignment

Esto es beneficioso porque queremos que Python lance un error cuando una línea de código intente cambiar las dimensiones del rectángulo.

[anterior](c04_27.html)[Subtema 28 de 41: (Ver todo)](c04.html)[siguiente](c04_29.html)