## Probar una función

Para aprender a probar, necesitamos código que probar. He aquí una función sencilla que recibe un nombre y un apellido y devuelve un nombre completo con un formato claro:

**name\_function.py**

def get\_formatted\_name(first, last):  
 """Generate a neatly formatted full name."""  
 full\_name = f"{first} {last}"  
 return full\_name.title()

La función get\_formatted\_name() combina el nombre y los apellidos con un espacio intermedio para completar un nombre completo, y luego escribe en mayúsculas y devuelve el nombre completo. Para comprobar que get\_formatted\_name() funciona, vamos a crear un programa que utilice esta función. El programa *names.py* permite a los usuarios introducir un nombre y un apellido, y ver un nombre completo formateado ordenadamente:

**names.py**

from name\_function import get\_formatted\_name  
  
print("Enter 'q' at any time to quit.")  
while True:  
 first = input("\nPlease give me a first name: ")  
 if first == 'q':  
 break  
 last = input("Please give me a last name: ")  
 if last == 'q':  
 break  
  
 formatted\_name = get\_formatted\_name(first, last)  
 print(f"\tNeatly formatted name: {formatted\_name}.")

Este programa importa get\_formatted\_name() de *name\_function.py*. El usuario puede introducir una serie de nombres y apellidos y ver los nombres completos formateados que se generan:

Enter 'q' at any time to quit.  
  
Please give me a first name: janis  
Please give me a last name: joplin  
 Neatly formatted name: Janis Joplin.  
  
Please give me a first name: bob  
Please give me a last name: dylan  
 Neatly formatted name: Bob Dylan.  
  
Please give me a first name: q

Podemos ver que los nombres generados aquí son correctos. Pero digamos que queremos modificar get\_formatted\_name() para que también pueda manejar segundos nombres. Al hacerlo, queremos asegurarnos de que no rompemos la forma en que la función gestiona los nombres que sólo tienen nombre y apellidos. Podríamos probar nuestro código ejecutando *names.py* e introduciendo un nombre como Janis Joplin cada vez que modificamos get\_formatted\_name(), pero eso se volvería tedioso. Afortunadamente, pytest proporciona una forma eficaz de automatizar la comprobación de la salida de una función. Si automatizamos las pruebas de get\_formatted\_name(), siempre podremos estar seguros de que la función funcionará cuando se le den los tipos de nombres para los que hemos escrito pruebas.

[anterior](c11_6.html)[Subtema 7 de 22: (Ver todo)](c11.html)[siguiente](c11_8.html)