### Responder a una prueba fallida

¿Qué haces cuando falla una prueba? Suponiendo que estés comprobando las condiciones correctas, una prueba superada significa que la función se comporta correctamente y una prueba fallida significa que hay un error en el nuevo código que has escrito. Así que cuando falle una prueba, no cambies la prueba. Si lo haces, puede que tus pruebas pasen, pero cualquier código que llame a tu función como lo hace la prueba dejará de funcionar de repente. En lugar de eso, arregla el código que está haciendo que falle la prueba. Examina los cambios que acabas de hacer en la función, y averigua cómo esos cambios rompieron el comportamiento deseado.

En este caso, get\_formatted\_name() solía requerir sólo dos parámetros: un nombre y un apellido. Ahora requiere un nombre, un segundo nombre y un apellido. La adición de ese parámetro obligatorio del segundo nombre rompió el comportamiento original de get\_formatted\_name(). La mejor opción en este caso es hacer que el segundo nombre sea opcional. Una vez que lo hagamos, nuestra prueba para nombres como Janis Joplin debería volver a pasar, y también deberíamos poder aceptar segundos nombres. Modifiquemos get\_formatted\_name() para que los segundos nombres sean opcionales y volvamos a ejecutar el caso de prueba. Si se supera, pasaremos a asegurarnos de que la función gestiona correctamente los segundos nombres.

Para que los segundos nombres sean opcionales, movemos el parámetro middle al final de la lista de parámetros en la definición de la función y le damos un valor por defecto vacío. También añadimos una prueba if que construye el nombre completo correctamente, dependiendo de si se proporciona un segundo nombre:

**name\_function.py**

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

En esta nueva versión de get\_formatted\_name(), el segundo nombre es opcional. Si se pasa un segundo nombre a la función, el nombre completo contendrá un nombre, un segundo nombre y un apellido. En caso contrario, el nombre completo constará sólo de nombre y apellidos. Ahora la función debería funcionar para ambos tipos de nombres. Para averiguar si la función sigue funcionando para nombres como Janis Joplin, vamos a ejecutar de nuevo la prueba:

$ pytest  
========================= test session starts =========================  
--snip--  
test\_name\_function.py . [100%]  
========================== 1 passed in 0.00s ==========================

Ahora la prueba pasa. Esto es ideal; significa que la función vuelve a funcionar para nombres como Janis Joplin, sin que tengamos que probar la función manualmente. Arreglar nuestra función fue más fácil porque la prueba fallida nos ayudó a identificar cómo el nuevo código rompía el comportamiento existente.

[anterior](c11_12.html)[Subtema 13 de 22: (Ver todo)](c11.html)[siguiente](c11_14.html)