### Añadir nuevas pruebas

Ahora que sabemos que get\_formatted\_name() vuelve a funcionar con nombres sencillos, escribamos una segunda prueba para las personas que incluyen un segundo nombre. Para ello, añadimos otra función de prueba al archivo *test\_name\_function.py*:

**test\_name\_function.py**

from name\_function import get\_formatted\_name  
  
def test\_first\_last\_name():  
 --snip--  
  
def test\_first\_last\_middle\_name():  
 """Do names like 'Wolfgang Amadeus Mozart' work?"""  
❶ formatted\_name = get\_formatted\_name(  
 'wolfgang', 'mozart', 'amadeus')  
❷ assert formatted\_name == 'Wolfgang Amadeus Mozart'

Llamamos a esta nueva función test\_first\_last\_middle\_name(). El nombre de la función debe empezar por test\_ para que la función se ejecute automáticamente cuando ejecutemos pytest. Nombramos la función para que quede claro qué comportamiento de get\_formatted\_name() estamos probando. De este modo, si la prueba falla, sabremos inmediatamente qué tipo de nombres se ven afectados.

Para probar la función, llamamos a get\_formatted\_name() con un nombre, apellido y segundo nombre ❶, y luego hacemos una afirmación ❷ de que el nombre completo devuelto coincide con el nombre completo (nombre, segundo nombre y apellido) que esperamos. Cuando volvemos a ejecutar pytest, ambas pruebas se superan:

$ pytest  
========================= test session starts =========================  
--snip--  
collected 2 items  
  
❶ test\_name\_function.py .. [100%]  
========================== 2 passed in 0.01s ==========================

Los dos puntos ❶ indican que se han superado dos pruebas, lo que también queda claro en la última línea de salida. ¡Esto es estupendo! Ahora sabemos que la función sigue funcionando con nombres como Janis Joplin, y podemos estar seguros de que también funcionará con nombres como Wolfgang Amadeus Mozart.

[anterior](c11_13.html)[Subtema 14 de 22: (Ver todo)](c11.html)[siguiente](c11_15.html)