Test cases

Aftabur Rahman (Aftab)

CSD325-308A

Module 7.2 Assignment

11/27/2025

"""

city\_functions.py

Function: city\_country Accepts city and country names (required) and population and language (optional). Returns a formatted string. """

def city\_country(city, country, population, language=None): """ Returns a string formatted as 'City, Country - population xxx' or 'City, Country - population xxx, Language'

The population parameter is required. The language parameter is optional.  
"""  
# Format population with commas for readability  
if isinstance(population, int):  
 pop\_formatted = f"{population:,}"  
else:  
 pop\_formatted = str(population)  
  
output\_string = f"{city.title()}, {country.title()} - population {pop\_formatted}"  
  
# Add language if provided  
if language is not None:  
 output\_string += f", {language.title()}"  
  
return output\_string

# **--- Required Demonstration Calls ---**

if **name** == '**main**': print("--- Demonstration Calls ---")

# Call 1: City, Country, Population (no language)  
result1 = city\_country('santiago', 'chile', 5000000)  
print(f"Call 1 (City, Country, Population): {result1}")  
  
# Call 2: City, Country, Population (no language)  
result2 = city\_country('tokyo', 'japan', 13960000)  
print(f"Call 2 (City, Country, Population): {result2}")  
  
# Call 3: City, Country, Population, Language  
result3 = city\_country('paris', 'france', 2140000, 'french')  
print(f"Call 3 (City, Country, Population, Language): {result3}")  
  
print("-" \* 30)

<https://replit.com/@afrahman/My-first-Python-programm#city_functions.py:47:19>

""" test\_cities.py

Tests for the city\_country function in city\_functions.py. """ import unittest from city\_functions import city\_country

class CitiesTestCase(unittest.TestCase): """Tests for 'city\_functions.py'."""

def test\_city\_country\_simple(self):  
 """Test for the 'City, Country - population xxx' format."""  
 formatted\_name = city\_country('santiago', 'chile', 5000000)  
 self.assertEqual(formatted\_name, 'Santiago, Chile - population 5,000,000')  
  
def test\_city\_country\_population(self):  
 """Test for the 'City, Country - population xxx' format."""  
 formatted\_name = city\_country('tokyo', 'japan', 13960000)  
 expected\_output = 'Tokyo, Japan - population 13,960,000'  
 self.assertEqual(formatted\_name, expected\_output)  
  
def test\_city\_country\_all\_params(self):  
 """Test for the 'City, Country - population xxx, Language' format."""  
 formatted\_name = city\_country('paris', 'france', 2140000, 'french')  
 expected\_output = 'Paris, France - population 2,140,000, French'  
 self.assertEqual(formatted\_name, expected\_output)  
  
def test\_city\_country\_only\_language(self):  
 """Test for the case where only City, Country, and Language are provided."""  
 formatted\_name = city\_country('madrid', 'spain', 6400000, 'spanish')  
 expected\_output = 'Madrid, Spain - population 6,400,000, Spanish'  
 self.assertEqual(formatted\_name, expected\_output)

if **name** == '**main**': unittest.main()

<https://replit.com/@afrahman/My-first-Python-programm#test_cities.py:36:19>











