

Chapter 4 Assignment - Function

Objectives

After completing this assignment, students will be able to:

1. Explain what a function is in Python and why it is used.
2. Identify the structure of a function: definition, body, arguments, and return values.
3. Write Python functions to perform specific tasks.
4. Use arguments and return values effectively to make functions flexible and reusable.

Assignment Instructions

- Implement the function stubs provided below in a file named `src/assignment.py`.
- Do not change the function names or signatures.
- Run the provided pytest test cases to verify your solutions.
- All test cases should pass for full marks.

Function Stubs (in `src/assignment.py`)

```
# 1. A simple function that returns a greeting message
def greet(name: str) -> str:
    # TODO: return "Hello, <name>!"
    pass

# 2. A function to calculate the area of a circle ( $\pi r^2$ )
def circle_area(radius: float) -> float:
    # TODO: return area of the circle
    pass

# 3. A function that converts Celsius to Fahrenheit
def celsius_to_fahrenheit(celsius: float) -> float:
    # TODO: formula: (celsius * 9/5) + 32
    pass

# 4. A function that returns the maximum of three numbers
def max_of_three(a: int, b: int, c: int) -> int:
    # TODO: return the largest number
    pass

# 5. A function that checks if a number is even
def is_even(num: int) -> bool:
    # TODO: return True if even, False otherwise
    pass
```

✓ Test Cases (in tests/test_assignment.py)

```
import math
import pytest
from src.assignment import (
    greet,
    circle_area,
    celsius_to_fahrenheit,
    max_of_three,
    is_even,
)

def test_greet():
    assert greet("Alice") == "Hello, Alice!"
    assert greet("Bob") == "Hello, Bob!"

def test_circle_area():
    assert math.isclose(circle_area(1), math.pi, rel_tol=1e-5)
    assert math.isclose(circle_area(2), 4 * math.pi, rel_tol=1e-5)

def test_celsius_to_fahrenheit():
    assert celsius_to_fahrenheit(0) == 32
    assert celsius_to_fahrenheit(100) == 212

def test_max_of_three():
    assert max_of_three(1, 2, 3) == 3
    assert max_of_three(10, 5, 7) == 10
    assert max_of_three(-1, -5, -3) == -1

def test_is_even():
    assert is_even(2) is True
    assert is_even(3) is False
    assert is_even(0) is True
```

✓ This assignment checks:

- Function definition (objective 1, 2)
- Using arguments (objective 3, 4)
- Returning values and reusability (objective 3, 4)