Exercise: Company Employees – Properties and Functions

Goal

Create a small Python program that manages basic employee data using properties (for data control) and functions (for actions).

Concept Overview

The idea is to show the difference between data (handled by properties) and behavior (handled by functions). Employees have data like name and salary, and perform actions like work() or apply_raise().

Task Description

Create a class called Employee with the following:

- Attributes (private/internal):
    - _name – string
    - _salary – float
- Properties:
    - name → getter returns the employee's name; setter checks that name is not empty
    - salary → getter returns the salary; setter checks that salary is not negative
- Functions (methods):
    - work(self) → prints something like 'Alice is working hard!'
    - apply_raise(self, percent) → increases salary by a given percentage

Expected Output Example

```
emp = Employee("Alice", 4000)
print(emp.name)         # Alice
print(emp.salary)       # 4000.0

emp.work()              # Alice is working hard!

emp.apply_raise(10)
print(emp.salary)       # 4400.0

emp.salary = -2000      #  should raise ValueError
```

My(Trainer) Commentary

In this task:
- The properties (name, salary) handle data control and validation.
- The functions (work, apply_raise) describe what the employee does.
- Together, they demonstrate the difference between data and behavior in an object.

That’s exactly the idea of encapsulation in object-oriented programming.

Optional Extension

Add a subclass Manager that inherits from Employee and overrides the work() method to print, '<name> is managing the team.'