What is mocking?

Mocking

- Creating a fake object that represents the “real” object
- Used in testing environments
- Allows more control over your code’s behavior
- Python mock objects provide deeper insight into your code
    - When functions were called
    - How many times they were called
    - What arguments were passed

The Python Mock Library

unittest.mock

- Built in to Python 3.3+
- Provides the Mock() class
- Provides the patch() method

- Now let’s code! ▶

Common Problems

Changes to Interfaces and Misspellings

- Creating a fake object that represents the “real” object
- Used in testing environments
- Allows more control over your code’s behavior
- Python mock objects provide deeper insight into your code

Congratulations! 🥳

You are now able to

- Use `Mock` to imitate objects in your tests
- Check usage data to understand how to use your objects
- Customize your mock objects’ return values and side effects
- `patch()` objects throughout your codebase
- See and avoid problems with using Python mock objects