Testing in Python

# What is Testing?

Testing is the process of evaluating your code to ensure it behaves as expected. It helps in identifying bugs and verifying that the application meets its requirements.

# Types of Testing

1. Manual Testing

Performed by a human without using tools.  
Time-consuming and error-prone.

1. Automated Testing

Done using scripts and tools.  
Faster, repeatable, and more reliable.

# Common Testing Types in Python

1. Unit Testing

Tests individual components (functions/methods).  
Tools: unittest, pytest

2. Integration Testing

Tests how different modules work together.  
Tools: pytest, unittest, nose2

3. Functional Testing

Validates the software against functional requirements.  
Tools: pytest, behave (for BDD)

4. System Testing

Tests the complete integrated system.  
Ensures all modules work together correctly.

5. Acceptance Testing

Verifies the system meets business requirements.  
Usually done by clients.

6. Regression Testing

Ensures new code changes don’t break existing features.

7. Smoke Testing

Basic tests to check whether the application is stable.

8. Performance Testing

Checks speed, scalability, and stability.  
Tools: locust, JMeter

9. Load Testing

Checks how the system handles a large number of users.

10. Stress Testing

Tests the limits of the system under extreme conditions.

# Popular Python Testing Frameworks

* unittest – Built-in, xUnit-style
* pytest – Powerful and popular third-party tool
* nose2 – Successor to nose
* doctest – Tests embedded in documentation
* tox – Automates testing in multiple environments