4. : Assertions and Test Discovery

- **Assertions:**
  - Way of validating the tests
  - Use operators like ==, !=, <=, >= for asserting.
  - You can use the above operators for validating strings and numbers.
  - assert 1, means assert True
  - assert 0, means assert False. This is going to fail your test.
  - In Python, multiplication, division has higher precedence than add, subtract.
  - assert with 'in' operator to validate if a value is within a tuple, list or string.
  - Good Test writing Practice: Try to add only 1 assert in single test.

- **Test Discovery:**
  - Given no arguments, pytest looks at your current directory and all subdirectories for test files and runs the test code it finds.
  - If you give pytest a filename, a directory name, or a list of those, it looks there instead of the current directory.
  - Each directory listed on the command line is recursively traversed to look for test code.

4. : Assertions and Test Discovery

- **__init__.py:**
  - Have __init__.py in folder to make it a python pkg.
  - Benefit is you can have same module-name in different pkgs.

- **pytest.raises:**
  - Assertions about expected exceptions, i.e. you are expecting the test to throw an exception, and you want to test that condition.
  - In this case, if the test condition throws an exception then it passed, else a failed test.
  - Useful for negative test scenarios.