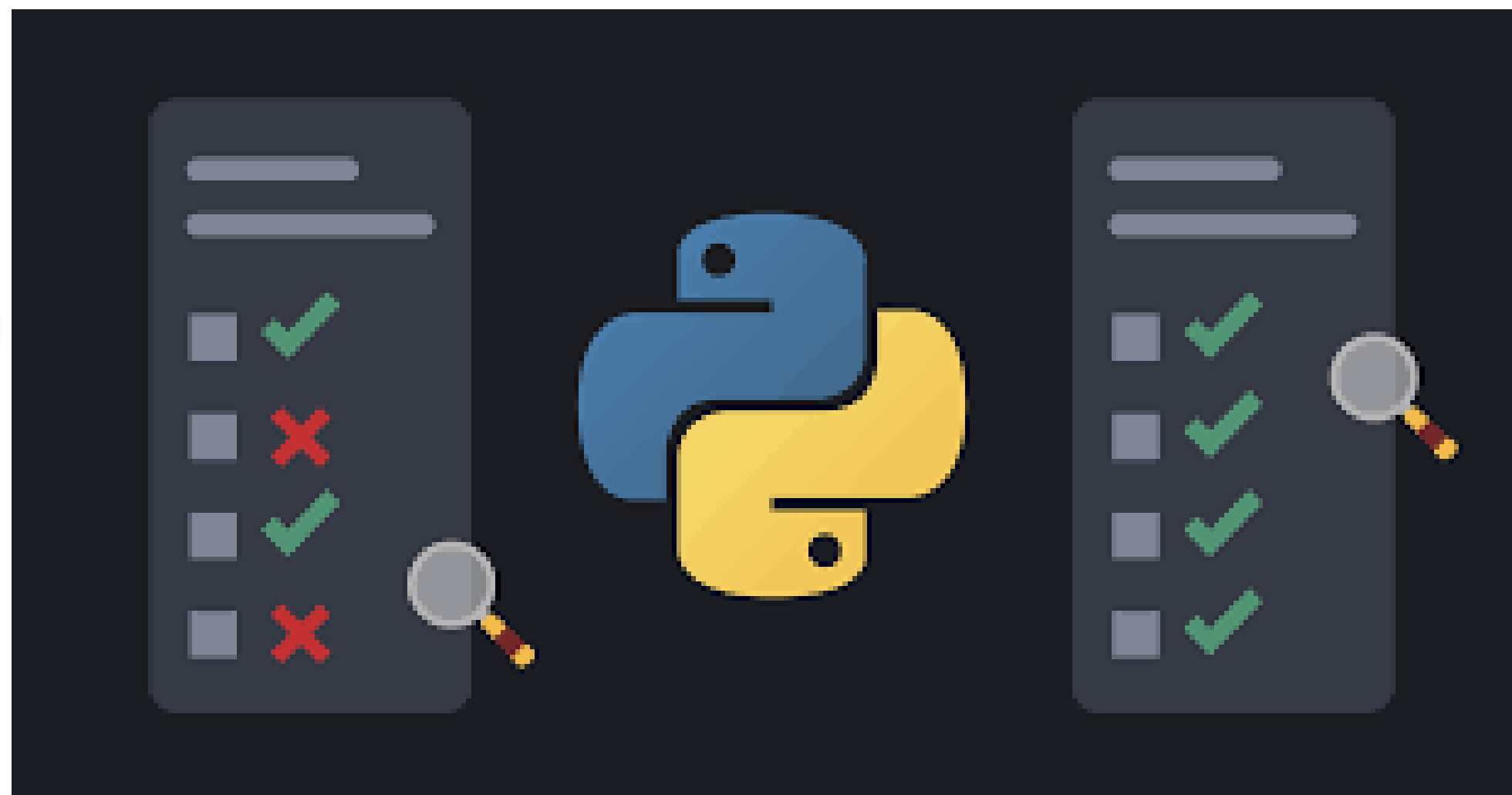



Unittest





Стандартна бібліотека Python для модульного тестування, яка реалізує підхід **xUnit**. Вона дозволяє створювати, організовувати та запускати тести для перевірки роботи вашого коду. Ось детальний опис основних можливостей unittest, його компонентів і приклади використання.

xUnit — це набір правил і принципів для створення модульних тестів, які стали стандартом у багатьох мовах програмування. Це не конкретна бібліотека чи фреймворк, а підхід до побудови тестових інструментів.

У реалізаціях xUnit тести зазвичай організовані у класи.

Кожен клас відповідає певній групі тестів, а його методи представляють окремі тестові випадки.

Як працює структура xUnit

Тестовий клас

Клас організовує логічно пов'язані тести.

У Python це клас, який успадковується від `unittest.TestCase`.

Методи класу

Методи в класі, які починаються зі слова `test`, є окремими тестами.

Кожен метод перевіряє одну функцію або поведінку.

Фікстури

setUpClass: Викликається один раз перед усіма тестами в класі.

setUp: Викликається перед кожним тестом.

tearDown: Викликається після кожного тесту.

tearDownClass: Викликається один раз після всіх тестів у класі.



Типовий життєвий цикл виконання тестів

Для класу тестів у xUnit виконується наступний порядок:

setUpClass (один раз для класу).

setUp (перед кожним тестом).

Виконання тесту (test_* метод).

tearDown (після кожного тесту).

setUp (перед кожним тестом).

Виконання тесту (test_* метод).

tearDown (після кожного тесту).

tearDownClass (після виконання всіх тестів у класі).



TestCase

TestSuite

<code>addTest(test)</code>	Додає один тест (або TestSuite) до тест-сюїту.
<code>addTests(tests)</code>	Додає кілька тестів (ітерується по об'єкту tests).
<code>run(result)</code>	Виконує всі тести в тест-сюїті.
<code>countTestCases()</code>	Повертає загальну кількість тестів у тест-сюїті.
<code>__iter__()</code>	Дозволяє ітерацію по всіх тестах, доданих до тест-сюїту.



TextTestRunner

```
unittest.TextTestRunner(stream=None, descriptions=True, verbosity=1, resultclass=None)
```

```
runner = unittest.TextTestRunner()  
result = runner.run(my_test_suite)
```

SkipTest

self.skipTest(Причина пропуску)

Метод класу TestCase, який явно пропускає тест із вказаною причиною.

@unittest.skip(reason)

Декоратор для пропуску тесту.

@unittest.skipIf(condition, reason)

Декоратор для пропуску тесту за настання умови.

@unittest.skipUnless(condition, reason)

Декоратор для пропуску тесту за ненастання умови.

@unittest.expectedFailure

Декоратор для очікуваного провалу тесту.



Mock



Імпорт

Функції з файлу main.py до test.py

Імпорт із тієї ж директорії

```
from main import some_function
```

Імпорт із вкладених директорій

```
from app.main import some_function
```

Імпорт із сусідніх директорій

```
import sys sys.path.append("/path/to/your/module")  
from main import some_function
```

```
import sys  
sys.path.append("/home/user/projects/my_project")  
from app.module import some_function
```