



Testy

Testy

- **Testy** w programowaniu to proces weryfikacji, czy kod działa zgodnie z oczekiwaniami. Testowanie polega na uruchamianiu fragmentów kodu w kontrolowanych warunkach, aby upewnić się, że funkcje, moduły lub całe aplikacje spełniają swoje założenia.

Rodzaje testów

- Testy jednostkowe (Unit Tests)
- Testy integracyjne (Integration Tests)
- Testy systemowe (System Tests)
- Testy regresji (Regression Tests)
- Testy wydajnościowe (Performance Tests)
- Testy bezpieczeństwa (Security Tests)

Testy jednostkowe

- Testy jednostkowe to testy w których najmniejsza testowalna część aplikacji (np. funkcja, metoda lub klasa) jest uruchamiana w izolacji. Ich celem jest zapewnienie, że dana jednostka działa zgodnie z oczekiwaniami w różnych scenariuszach.

Cechy testów jednostkowych

- Izolacja
- Automatyzacja
- Szybkość
- Powtarzalność

Biblioteka unittest

unittest to wbudowany framework do testowania jednostkowego. Jest standardowym narzędziem w Pythonie, dostępne bez instalowania zewnętrznych pakietów.

Zalety

- Nie wymaga instalowania zewnętrznych bibliotek.
- Jest integralną częścią standardowej biblioteki Pythona.
- Oferuje wszystkie podstawowe mechanizmy testowania jednostkowego.

Wady

- Testy muszą być napisane w ramach klas dziedziczących po `unittest.TestCase`.
- Wymaga więcej kodu niż nowsze frameworki, np. `pytest`.
- Mniej wygodne dla testów dynamicznych i bardziej złożonych przypadków testowych.

Biblioteka pytest

pytest to nowoczesny framework do testowania w Pythonie.

Zalety

- Nie wymaga klas, testy mogą być napisane jako zwykłe funkcje.
- Przyjazna i bardziej czytelna składnia.
- Obsługa parametrów testowych, integracji z CI/CD i bardziej elastyczne asercje.
- Bogaty ekosystem wtyczek, np. pytest-cov do mierzenia pokrycia kodu.
- Bardziej intuicyjny w przypadku nowych projektów.

Wady

- Wymaga instalacji (pip install pytest).
- Może być trudniejszy do opanowania dla początkujących użytkowników w porównaniu do prostych testów w unittest.