

Tests & Qualité — Python

Écrire des tests rapides, fiables et lisibles avec `pytest`



Changelog — V0.0.1

- 10/03/2026 04:58 — Création du deck qualité Python avec `pytest`, couverture et règles de CI locale.

Objectif qualité

- Tester le comportement, pas l'implémentation interne
- Garder un cycle `test -> code -> validation`
- Produire des erreurs lisibles et localisables rapidement

Exemple `pytest`

```
def normalize_login(value: str) -> str:  
    return value.strip().lower()  
  
def test_normalize_login_removes_spaces() -> None:  
    assert normalize_login(" Alice ") == "alice"
```

- Un test = un comportement observé
- Le nom du test doit décrire l'intention

Paramétrer proprement

```
import pytest

@pytest.mark.parametrize(
    ("raw", "expected"),
    [(" Alice ", "alice"), ("BOB", "bob"), (" carol", "carol")],
)
def test_normalize_login_cases(raw: str, expected: str) -> None:
    assert normalize_login(raw) == expected
```

- Réduire la duplication
- Couvrir plusieurs cas sans multiplier les fonctions inutiles

Fixtures & isolation

- Une fixture prépare un contexte réutilisable
- Éviter les états globaux et les fichiers persistants non nettoyés
- Préférer `tmp_path` pour les tests filesystem

```
def test_write_report(tmp_path):  
    output = tmp_path / "report.txt"  
    output.write_text("ok", encoding="utf-8")  
    assert output.read_text(encoding="utf-8") == "ok"
```

Couverture & lint

- La couverture n'est pas un objectif en soi, c'est un signal
- Coupler `pytest`, `coverage`, `ruff` et `mypy`
- Faire échouer tôt quand un garde-fou tombe

```
pytest -q
coverage run -m pytest
coverage report --fail-under=85
ruff check .
mypy src
```

Pipeline local recommandé

1. Écrire ou adapter le test.
2. Implémenter le code minimum.
3. Relancer `pytest`.
4. Vérifier lint + typage.
5. Corriger avant commit.

Anti-patterns à éviter

- Tests qui dépendent de l'ordre d'exécution
- Assertions trop larges ou trop floues
- Mocks partout alors qu'un appel simple suffit
- Suites longues qui empêchent la boucle courte de feedback