
Software Engineering

軟體工程

自動測試 (Automated Testing)

蘇維宗 (Wei-Tsung Su)
suwt(at)nicu.edu.tw





Revision

| Rev. | Description | Date | Authors |
|------|--|------------|----------|
| v1.0 | Baseline | 2025/08/01 | W.-T. Su |
| v1.1 | 加入 pytest_example 範例與 pytest_assignment 作業 | 2025/12/01 | 蘇維宗 |
| | | | |
| | | | |
| | | | |
| | | | |

自動化測試

- 使用程式碼自動驗證系統行為是否符合預期
- 目的是減少人工測試成本、提高測試的一致性
- 自動化測試在 DevOps 與 CI/CD (Continuous Integration / Continuous Delivery) 流程中扮演關鍵角色，是現代軟體工程的必備能力
- 常見的自動化測試類型：
 - 單元測試(Unit Test)
 - 整合測試(Integration Test)
 - 端對端測試(End-to-end Test)

自動化測試的優點

- 提高測試覆蓋率(coverage)與缺陷偵測率
 - 能在極短時間內執行大量測試案例，進而提高測試覆蓋率並縮短缺陷偵測時間
- 改善系統可維護性與重構品質
 - 自動化測試使開發者敢於重構 (refactoring)與改善架構，提升長期可維護性
- 降低整體開發成本
 - 雖然初期會增加成本，但大量研究指出自動化測試在中長期可以降低整體維運成本



pytest

Pytest

Python 專案的主流自動測試化框架

Pytest

- Pytest 是現代 Python 專案的主流自動測試化框架
 - <https://docs.pytest.org/>
- Pytest 的特性
 - 語法簡潔(可直接使用 assert)
 - 外掛生態豐富(超過 1000 plugins)

Pytest 基本範例

- 待測試程式(`math.py`)

```
1. def add(a, b):  
2.     return a + b
```

- Pytest 測試程式(`test_math.py`)

```
1. from math import add  
3.  
4. def test_add(a, b):  
5.     assert add(2,3) == 5  
6.     assert add(2,-3) == -1
```

- 安裝 Pytest

```
$ pip install pytest
```

- 執行 Pytest 測試

```
$ pytest
```

- 安裝 Pytest HTML 報告插件

```
$ pip install pytest-html
```

- 執行 Pytest 測試並產製報告

```
$ pytest --html=report.html  
--self-contained-html
```

作業說明

- 撰寫 pytest_assignment.zip 專案的測試案例
 - Step 1. 針對每個函數需求撰寫測試案例
 - Step 2. 執行 pytest 自動化測試
 - Step 3. 找出錯誤後修正錯誤
 - Step 4. 重複 Step 2-3 直到所有測試通過
- 作業繳交內容
 - 程式(pytest_assignment.py, score.py)
 - 報告(每個 pytest 找到的錯誤與修正方式、150字以上心得)

Q & A



Computer History Museum, Mt. View, CA