

# Introduction to pytest



# pytest

- Defacto standard testing tool for Python
- unittest is in the standard library, pytest is better
- pytest is a command line tool for collecting and running tests
- Also a framework for writing tests
- Extendable by plugins (not very hard to write), for example pytest-asyncio for async testing
- Very widely used, lots of documentation and videos
- The purpose of testing is to **verify behaviour**
- <https://pytest.org/>

# Creating a Test Suite

- Install pytest into a virtual environment
  - pipenv is commonly used to manage environments and dependencies
- Test collection is done with a naming convention:
  - Write tests as functions in files called "*test\_something.py*" (etc)
  - Functions should be named "*test\_something*" as well
- They probably live in a project directory called "tests"
- Use the assert statement to verify something
- Run the tests with pytest
- The test fails with a useful error message if an assert fails or something goes wrong

# Test Functions

```
def test_function():  
    result = 1 + 2  
    assert result == 3
```

```
def test_failing_test():  
    result = 1 + 2  
    assert result == 4
```

```
$ pytest
```

# Test Run

```
(pytest) michael@lappy:~/code/talks/pytest$ pytest
===== test session starts =====
platform linux -- Python 3.10.12, pytest-8.2.0, pluggy-1.5.0
rootdir: /home/michael/code/talks/pytest
collected 2 items

test_first.py .F [100%]

===== FAILURES =====
_____ test_failing_test _____

    def test_failing_test():
        result = 1 + 2
>       assert result == 4
E       assert 3 == 4

test_first.py:9: AssertionError
===== short test summary info =====
FAILED test_first.py::test_failing_test - assert 3 == 4
===== 1 failed, 1 passed in 0.02s =====
(pytest) michael@lappy:~/code/talks/pytest$
```