Python Testing Tutorials

Read/Watch the required knowledge, and complete each section’s activity

Contents

[1: Installing and testing Pytest 1](#_Toc112248195)

[2: Running multiple tests and the first Fixture 2](#_Toc112248196)

[3: Oh no! more Fixtures 3](#_Toc112248197)

[4: Mocking Time 4](#_Toc112248198)

[5: Covering up the cracks 5](#_Toc112248199)

[6: Back to Git Hooks with Pytest 6](#_Toc112248200)

# 1: Installing and testing Pytest

**Required Knowledge:**  
Watch Pytest 101 Video  
<https://www.youtube.com/watch?v=etosV2IWBF0>

[optional] Alt in-depth Watch the awesome full bells and whistles pytest video from pyconf  
<https://www.youtube.com/watch?v=ofPHJrAOaTE>   
  
Read through pytest docs

<https://peps.python.org/pep-0008/>

Read through Zen of Unit Testing  
<http://codesqueeze.com/the-zen-of-unit-testing/>

**Activities:**

Install pytest locally on your machine

Create and run the “test\_sample.py” from pytest getting started

<https://docs.pytest.org/en/7.1.x/>

Well done you wrote your first test 😊

# 2: Running multiple tests and the first Fixture

**Required Knowledge:**

*none*

**Activities:**

Go back to pytest’s getting started page

<https://docs.pytest.org/en/7.1.x/getting-started.html#get-started>

Complete following sections  
- Assert that a certain exception is raised

- Group multiple tests in a class

- Request a unique temporary directory for functional tests

Go through and run examples locally for the pytest “Howto assert” guide

<https://docs.pytest.org/en/7.1.x/how-to/assert.html>

# 3: Oh no! more Fixtures

**Required Knowledge:**  
**Read through the meaning and description of a test fixture**

<https://en.wikipedia.org/wiki/Test_fixture>

**Read through the various built in fixture types available (aka capsys / monkeyptach etc)**<https://docs.pytest.org/en/7.1.x/reference/fixtures.html#fixtures>

**Activity:**  
Go through and run examples locally for the pytest “Howto fixtures” guide

<https://docs.pytest.org/en/7.1.x/how-to/fixtures.html>

# 4: Mocking Time

**Required Knowledge:**  
**Read through the meaning and description of a test mock**

<https://en.wikipedia.org/wiki/Mock_object>

**Read through excellent mocking guide in Python**<https://realpython.com/python-mock-library/>

***Lightly*-Read through the various built in mocking available  
(warning it’s crazy indepth so would only skim to get a feeling of what’s possible)**<https://docs.python.org/3/library/unittest.mock-examples.html>

**Activity:**  
Go through and run examples locally for the pytest “Howto mock patch” guide

<https://docs.pytest.org/en/7.1.x/how-to/monkeypatch.html>

**Bonus Activity:**

Create a simple function that says what day of the week it is  
  
try to write a test that mock out datetime, so you can test the function working on a specific date

# 5: Covering up the cracks

**Required Knowledge:**  
**Read through the meaning and description of a test coverage**

<https://en.wikipedia.org/wiki/Code_coverage>

**Read through excellent pytest-cov**<https://pytest-cov.readthedocs.io/en/latest/config.html>

**Read through Zen of Coverage**

<https://kykim.github.io/blog/2012/07/24/code-coverage-and-the-zen-master/>

**Activity:**  
1) Load up project used to test run fib

2) Create unit tests to test your fib function  
  
Example unit tests:  
fib(1) = 1   
fib(2) = 2   
fib(5) = 8   
fib(“some-string”) = error

3) Make sure testing is working

4) Add pytest + pytest-cov into your requirements.txt  
<https://pypi.org/project/pytest-cov/>   
<https://pypi.org/project/pytest/>

5) Run the coverage generator  
  
hint: normal command  
pytest tests/ -vv --cov=<SOURCE CODE> --cov-branch --cov-report=term-missing --cov-report html: coverage\_html  
  
6) Look at html coverage and see if there is any logic missed, and fill in the gaps 😊

# 6: Back to Git Hooks with Pytest

**Required Knowledge:**  
Re-Read sections on git hooks  
(covered in GIT basics sections)

<https://www.atlassian.com/git/tutorials/git-hooks>   
<https://git-scm.com/book/en/v2/Customizing-Git-Git-Hooks>

**Read up on precommit ( python git hook manager )**

<https://pre-commit.com/>

**Activity:**  
Load up project used to test run fib   
  
Add precommit to the project’s requirements.txt

Add pytest to automatically run, when doing a git commit, using precommit

Check your .git/hooks folder, and test the hooks are working

Well done, you’ve done typical production testing git hook, one many projects use to run unit tests 😊