

Overview

In this lesson, you will learn how to **test functions using pytest**. Testing allows you to automatically check that your functions work correctly and continue working as your program grows. This lesson will focus on setting up pytest, writing test files, following pytest's naming rules, using `assert` statements, and running tests inside **PyCharm**.

Important Information

Testing is a core skill in computer science. Instead of manually running your program and checking output every time, tests allow you to verify that functions behave as expected and catch mistakes early.

In this course, we will use **pytest**, a popular Python testing framework.

What Is pytest?

`pytest` is an external library that allows you to automatically test code after some short set up. `pytest` automatically finds test files in your project folders, runs the test functions that you create, and reports back which tests passed and which tests failed.

`pytest` is not included with Python by default, so it must be installed.

Installing pytest

pytest is installed using `pip`, Python's package manager.

```
pip install pytest
```

After installation, pytest becomes available to your Python environment.

If pytest is installed correctly, running the following command will display the installed version number:

```
pytest --version
```

What Happens When a Test Fails

If a test fails, pytest will show which test failed, show the user the expected value (*the one that the user defined in the condition of the `assert`*), and show the user the actual value (*the value that the function actually output*).

Running Tests in PyCharm

PyCharm has built-in support for pytest.

To run tests:

1. Open the project in PyCharm
2. Locate the `tests` folder
3. **Right-click** on the `tests` folder
4. Select “**Run tests**”

Then PyCharm will run all test files inside the folder, show the results to the user in the terminal, and output key information for any tests that failed.

You can also test a single file by right clicking the specific test file you want to check and run just those tests with the “**Run tests**” menu option.

Common Errors and Fixes

pytest Not Found

- **Cause:** pytest is not installed in the active environment

- **Fix:** Install pytest using `pip` and confirm the correct interpreter is selected in PyCharm

No Tests Are Running

- **Cause:** File name does not start with `test_` or function name does not start with `test_`

- **Fix:** Rename files and functions to follow pytest rules

Import Errors

- **Cause:** Incorrect import path

- **Fix:** Check file locations and ensure your project structure matches your imports

Using `print()` Instead of `assert`

- **Cause:** Misunderstanding how tests work

- **Fix:** Replace prints with `assert` statements

The Importance of Testing

Creating and running tests on your code is crucial to developing problem solving skills. It helps you to think clearly about what your function should do and write more reliable programs. Testing often will also allow you to catch bugs earlier in development. These skills will prepare you for larger projects throughout this course, as well as software development as a career. Testing may feel slower at first, but it saves time and frustration as programs become more complex.