Overview

We've discussed the importance of testing and how to get `pytest` set up. Now it's time to actually build some test files.

Important Information

Project Structure for Testing

A common and clear structure for your project is having your main code in one folder and your tests in a separate folder.

Example:

```
project_folder
|
├── functions.py
└── tests
    └── test_functions.py
```

pytest Naming Rules

`pytest` relies heavily on **naming conventions**. If these rules are not followed, pytest will not find your tests.

Test File Names

- Test files **must start with** `test_`
- Example:
- `test_functions.py`
- `test_math.py`

Test Function Names

- Test functions **must start with** `test_`
- Example:

```
def test_addition():
    ...
```

If either the file name or function name does not start with `test_`, pytest will ignore it.

Writing a Simple Test

Suppose you have a function in `functions.py`:

```
def add(a, b):
    return a + b
```

You can test it in `test_functions.py`:

```
from functions import add

def test_add():
    result = add(2, 3)
    assert result == 5
```

The `assert` Statement

`assert` is the core of pytest testing.

```
# Use the assert keyword, and then follow it by a condition
assert condition
```

If the condition is `True`, the test passes. If the condition is `False`, the test fails.

Example:

```
# 1 + 1 DOES equal 2, which means this condition will be True
# The test will report as passed
assert add(1, 1) == 2

# 4 + 13 DOES NOT equal 5, which means this condition will be False
# The test will report as failed
assert add(4, 13) == 5
```

When testing with `pytest` you are **not printing results**. `pytest` checks the condition for you and reports if it is `True` or `False`.

Testing Multiple Cases

Good tests often include multiple scenarios. Each test runs independently.

```
def test_add_with_zero():
    assert add(5, 0) == 5

def test_add_with_negatives():
    assert add(-2, -3) == -5
```

These tests increase confidence that the function works in more situations.