

Py.test Framework

py.test is a test framework that is alternative to Python's standard unittest module. Despite being a fully-featured and extensible test tool, it is of a simple syntax. Creating a test suite is as easy as writing a module with a couple of functions.

Installing

Below is the command to install pytest using pip commands.

'Pip install pytest'

Usage

The immediately noticeable thing is that pytest uses a plain assert statement, which is much easier to remember and use compared to the numerous functions found in unit-test.

```
test_sample.py
```

```
def func(a):
```

```
    return a + 1
```

```
def test_answer():
```

```
    assert func(3) == 5
```

Handling Exceptions

The **pytest.raises** helper which asserts that our function should raise an error if something is not working as expected. We can raise an exception of `TypeError` and send custom messages to the user accordingly.

Advanced Features of Pytest Framework

- They help us set up some helper code that should run before any tests are executed, and are perfect for setting up resources that are needed by the tests.
- Fixture functions are created by marking them with the `@pytest.fixture` decorator. Test functions that require fixtures should accept them as arguments.
- We can reduce the amount of code by using the pytest fixtures
- Utilizing fixtures helps us de-duplicate our code. If you notice a case where a piece of code is used repeatedly in a number of tests, that might be a good candidate to use as a fixture.
- To see all the available fixtures, run the following command: `pytest --fixtures`

Parameterized Test Functions:

- If we have several individual methods in a class, and have to run tests with various combinations of these methods, it could get tedious. But, with the usage of parameterized test functions, this could get easy.
- This enables us to test different scenarios, all in one function. We make use of the `@pytest.mark.parametrize` decorator, where we can specify the names of the arguments that will be passed to the test function, and a list of arguments corresponding to the names.
- To make our tests less repetitive, we can go further and combine test fixtures and parametrize test functions.
- `pytest.fixture()` allows you to define parameterization at the level of fixture functions.