Python Testing Example pytest

PyTest is a testing framework that allows users to write test codes using Python programming language.

Installation

Run the following command in the command line or terminal.

```
pip install pytest
```

Project Structure

This project has two modules: * `src.calc_func` contains math functions. * `src.calc_class` contains a basic Calculator class.

Test modules are placed under the `tests` directory. Note that `tests` is *not* a Python package, It doesn't need to be shipped as part of final package but generally used for interal testing.

Running Tests

- `python -m pytest` to discover and run all tests from the current directory
- `python -m pytest -v` to explicitly print the result of each test as it is run
- `python -m pytest tests/test_calc_func.py` to run only the math function tests
- `python -m pytest tests/test_calc_class.py` to run only the Calculator class tests
- `python -m pytest -k dividetestsgroup` to run a group of testcases that are marked with the groupname as dividetestsgroup
- `python -m pytest -h` for command line help

It is also possible to run pytest directly with the "pytest" or "py.test" command, instead of using the longer "python -m pytest" module form. However, the shorter command does *not* append the current directory path to *PYTHONPATH*.

Coverage Report

Install the `coverage` package using the following command:

```
pip install coverage
```

Running the coverage reports:

- `coverage run --source=src/ -m pytest -v` to run the coverage on the selected folder, the tests will automatically run since `pytest` option is

given. It does not display any output, you must run `coverage report` to view the report.

- `coverage report` to view the report of the last coverage run.
- `coverage html` to view the report in html format.