pytest usage Cheat Sheet

by hvid2301 via cheatography.com/186766/cs/39050/

Installation

```
pip install pytest
```

Running pytest

pytest test_mod.py	Run tests in module
pytest testing/	Run multiple modules
pytest test_mod.py::test_func	Run a specific test within a module.
pytest test_mod.py::TestClass::test_method	Run a specific method of a class.

Assertions

pytest allows you to use the standard Python assert for verifying expectations and values in Python tests. For example, you can write the following:

`assert a % 2 == 0`	assert that you have a certain value.
`assert a % 2 == 0, "value was odd, should be even"`	specify a message with the assertion like this

Grouping in tests

@pytest.mark.<markername>	mark it in tests
pytest tests/ -m "sanitytests"	run using -m flag

Disable tests

```
@pytest.mark.skip(
    reason ="no way of currently testing this")
```

Assertion of expected exceptions

Test if the exception is raised

```
def test_zero_division():
  with pytest.raises(ZeroDivisionError):
    1 / 0
```

Test for specific exception info

```
def test_recursion_depth():
  with pytest.ra ise s(R unt ime Error) as excinfo:
    assert " max imu m" in str(ex cin fo.v alue)
```

MultiThreading

```
pip install pytest-xdist
 pytest tests/ -n 3
```

Pytest does not run in parallel by default, therefor we need to enable this behaviour through a plugin.

Fixtures

In testing, a fixture provides a defined, reliable and consistent context for the tests. This could include environment (for example a database configured with known parameters) or content (such as a dataset).

```
@pytest.fixture
 def input_value():
    input = 10
    return input
```

Create fixture in conftest

```
def test_divisible_by_3(input_value):
    assert input_ value % 3 == 0
```

Use it in tests

```
@pytest.fixture
 def sending_user(mail_admin):
     user = mail_admin.create_user()
     yield user
     mail_a dmi n.d ele te_ use r(use
r)
```

using yield to setup and teardown test environment

@pytest.fixture(scope="function")

Pytest fixtures have scope. This scope controls how often the fixture is executed or, in other words, how often the setup and teardown of the fixture is performed.

@pytest.fixture(scope="module")

@pytest.fixture(scope="class")

@pytest.fixture(scope="session")

Data parameterization

The builtin pytest.mark.parametrize decorator enables parametrization of arguments for a test function

Define parametrized tests

```
@pytes t.m ark.pa ram etr ize ("num, output",
 [(1,11),(2,22),(3,35),(4,44)])
def test_m ult ipl ica tio n_1 1(num, output):
    assert 11*num == output
```

Running specific tests of a parametrized test.

```
test_mod.py::test_multiplication_11[2-22]
```

By **hvid2301**
cheatography.com/hvid2301/

Published 1st June, 2023.
Last updated 1st June, 2023.