**Fixtures**: Functions That Run Before (Some Of Them After) The Test Functions.

The Most Value Of Them: Using For DB Initialization, WebDriver Initialization, And …etc.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**Note**: If We Set The Fixture Inside Any File, It Then Will Run For That File Only.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**Note**: If We Set Fixtures Inside The conftest.py-File It Will Be Then Available To All Directories And Sub-Directory.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

import pytest;

@pytest.fixture()

def setup\_names():

    print("=" \* 15);

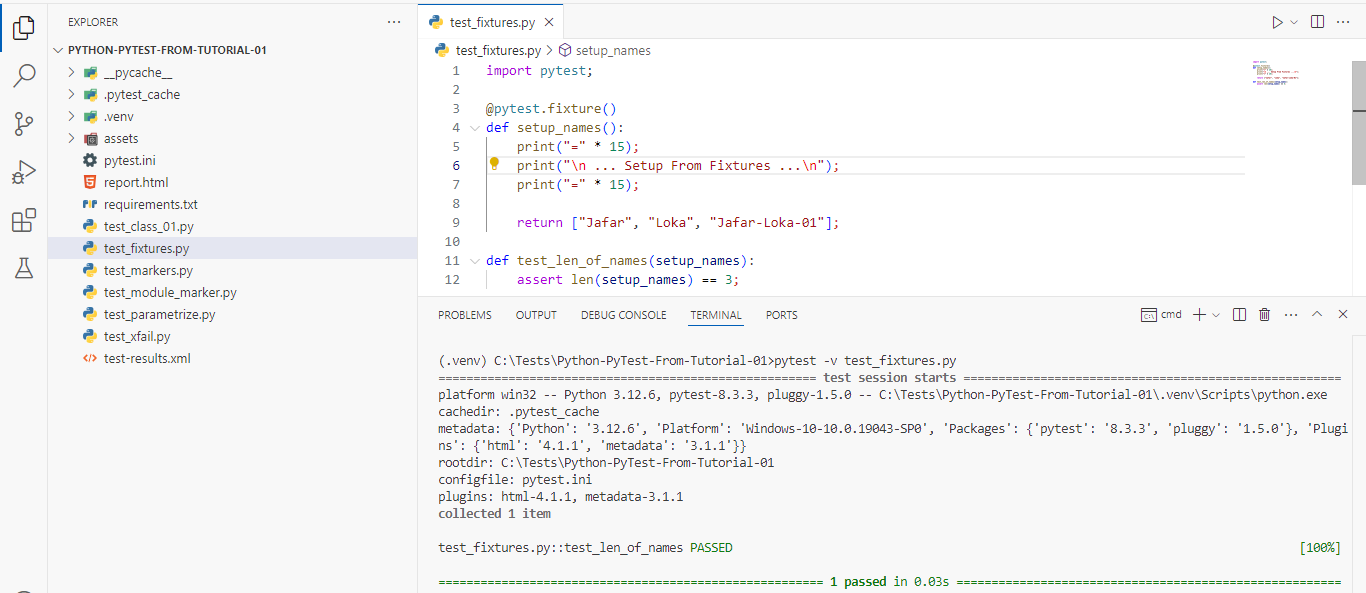
    print("\n ... Setup From Fixtures ...\n");

    print("=" \* 15);

    return ["Jafar", "Loka", "Jafar-Loka-01"];

def test\_len\_of\_names(setup\_names):

    assert len(setup\_names) == 3;



\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

If We Use Fixtures With Markers, Then The Returned Value From Fixtures Can’t Be Used.

We Can Use Fixtures Using Markers, By This Way:

@pytest.mark.usefixtures("setup\_names")

def test\_use\_fixtures():

    assert 1 == 1;



\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

If We Want To Close The:

* Connection To DB.
* The Driver Page.
* The File.
* Any Thing That We Returned From Fixture

Then We Must Use Yield.

days1 = ["Sat", "Sun", "Mon"];

days2 = ["Tue", "Wed", "Thu"];

@pytest.fixture()

def setup\_days1():

    wk1 = days1.copy();

    wk1.append('Fri');

    yield wk1;

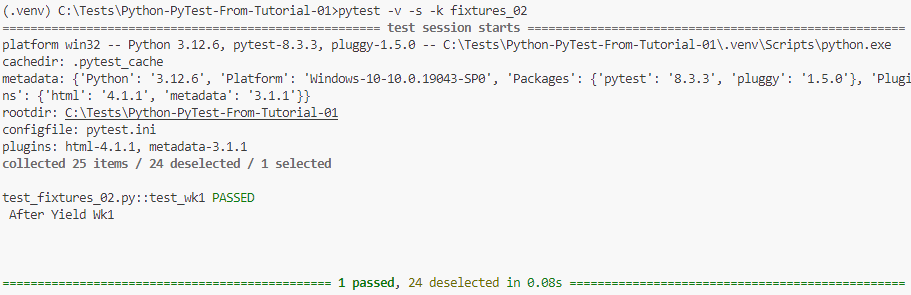
    print("\n After Yield Wk1 \n");

    # wk1.clear();

    wk1.pop();

def test\_wk1(setup\_days1):

    assert len(setup\_days1) == 4;



\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

We Can Also, Use Multiple Fixtures For The Same Test:

def test\_wk1\_wk2(setup\_days1, setup\_days2):

    assert len(setup\_days1) + len(setup\_days2) == 7;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

We Can Use This Way To Handle DB Connection Open, File Open, …etc.

@pytest.fixture()

def setup\_days1():

    wk1 = days1.copy();

    wk1.append('Fri');

    yield wk1;

    wk1.pop();

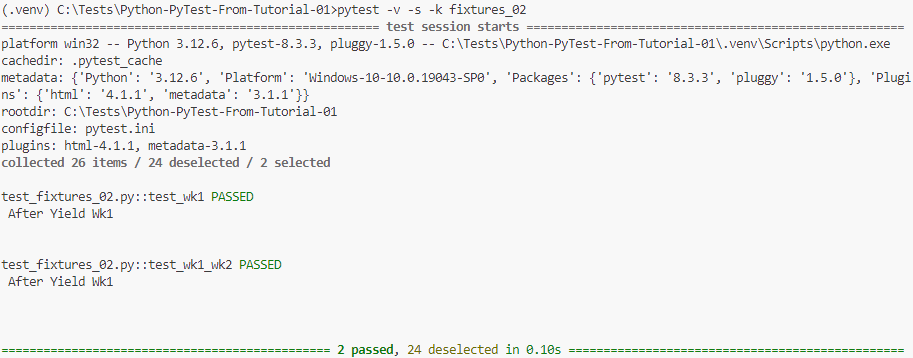
@pytest.fixture()

def setup\_days2():

    wk2 = days2.copy();

    yield wk2;

    wk2.clear();



\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

The conftest.py File Used To Share The Fixtures Across The Test Files.

We Can Set Multiple conftest.py Files In All Subdirectories.

It Shouldn’t Import The conftest.py-Files From The Test Files.

The conftest.py File Is Python Module.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**Note**: If We Used Files Using Fixtures And Generators (Yield), We Must Close Them Before Remove Them From Directories.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

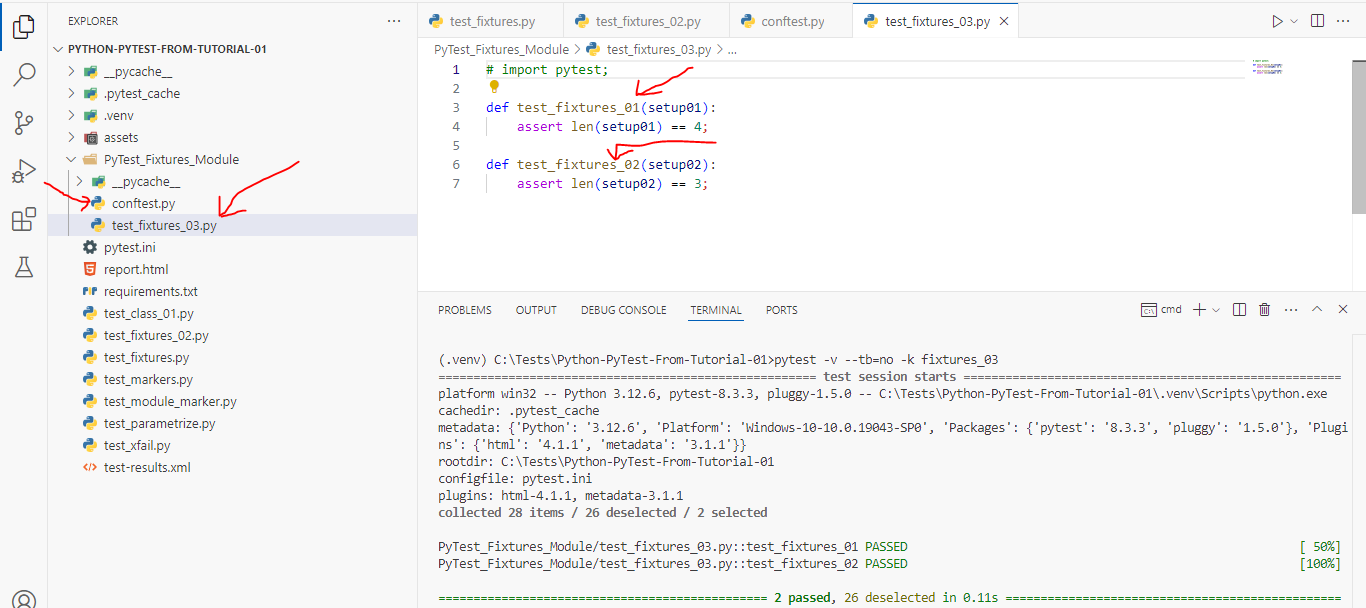
To Create Global Variables For Fixtures, To Use Them We Can Use: *pytest\_configure.py*:

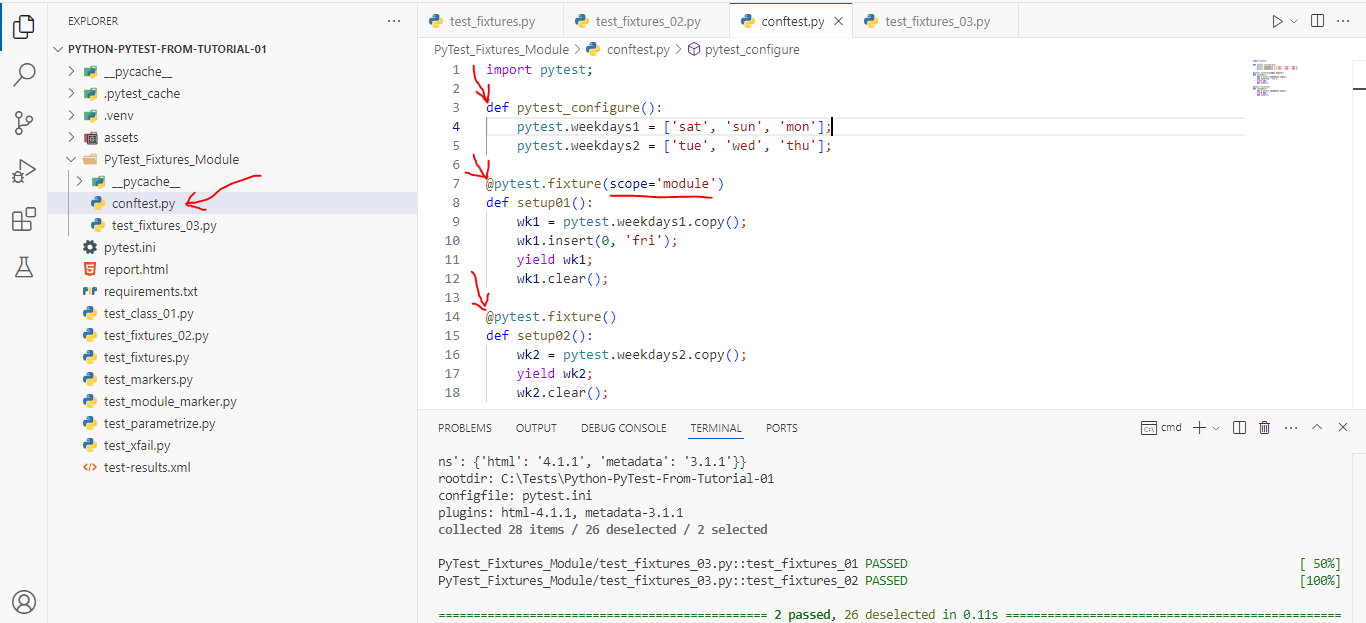
def pytest\_configure():

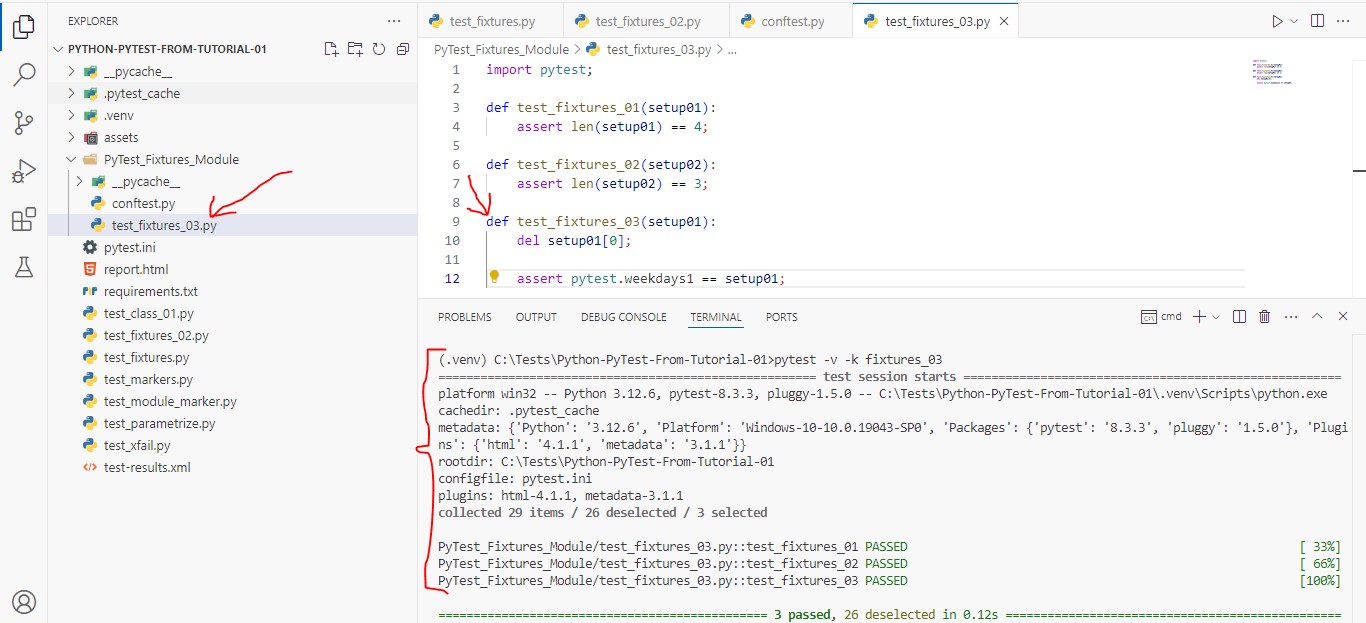
    pytest.weekdays1 = ['sat', 'sun', 'mon'];

    pytest.weekdays2 = ['tue', 'wed', 'thu'];

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*





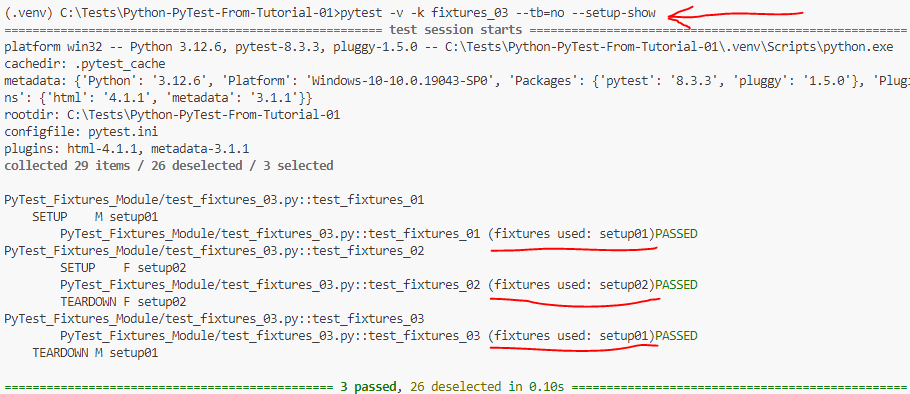


\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**Note**: Fixtures Can Be Overridden From Test Module Level.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

If We Want To Show The Fixtures That We Use We Can Use: --setup-show With Pytest:



\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

We Can Use Fixtures To Intercept The Test Function Calling:

**Note**: Here The Months Are Inside The Test Function That Use This Fixture.

@pytest.fixture()

def setup04(request):

    months = getattr(request.module, "months");

    print("\n The Setup04 Fixture");

    print("\n Fixture Scope: ", str(request.scope));

    print("\n Calling Function: ", request.function.\_\_name\_\_);

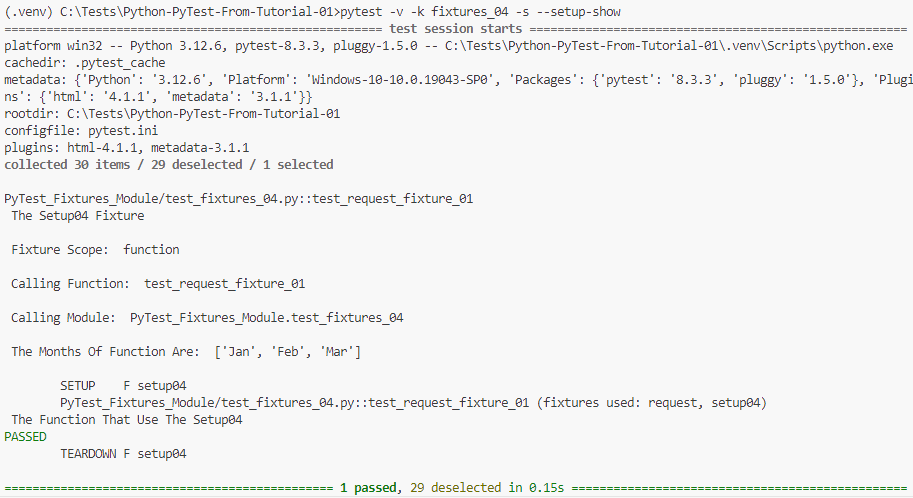
    print("\n Calling Module: ", request.module.\_\_name\_\_);

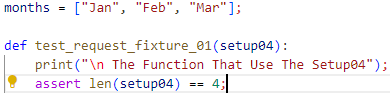
    print("\n The Months Of Function Are: ", months);

    months.append("April");

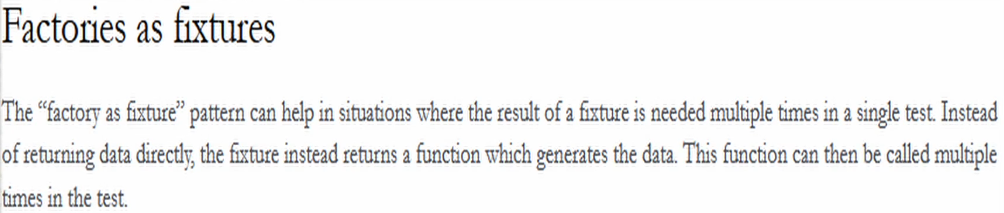
    yield months;

    months.pop();





\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*



\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

In This Way We Can Define Fixture As Factory:

@pytest.fixture()

def setup05():

    def get\_structure(name):

        if name == 'list':

            return [1, 2, 3];

        elif name == 'tuple':

            return (1, 3, 4);

    return get\_structure;

And In This Way We Can Call Our Fixture That Used Factory:

def test\_fixtures\_list\_05(setup05):

    list = setup05('list');

    assert 'list' in str(type(list));

def test\_fixtures\_tuple\_05(setup05):

    tuple = setup05('tuple');

    assert 'tuple' in str(type(tuple))

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**Note (From Me)**: We Can Use Factory As Fixture To Generate The List Of Data That Represent The Test Parametrization Data.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**Note**: This Function Will Be Called 2-Times, For Each Test Cases:

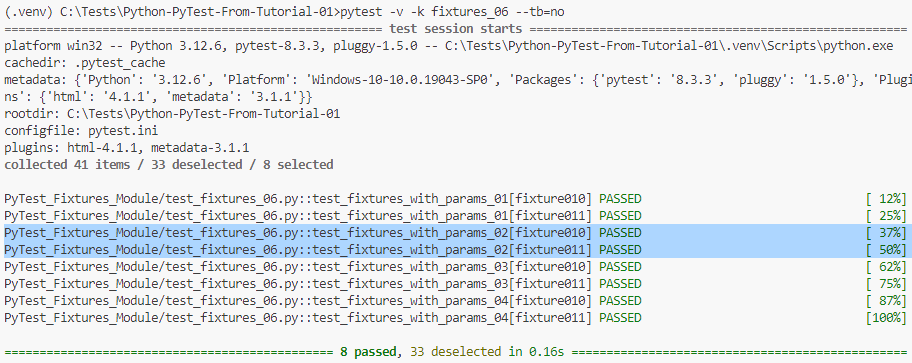
import pytest;

@pytest.fixture(params=[(3, 4), [3, 5]])

def fixture01(request):

    return request.param;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*



\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*