**Pytest Naming Conventions**

When using pytest, it is very important to follow naming conventions.

If we don't follow naming conventions, then pytest will not pick up tests from the file.

File names should start or end with “test”, as in test\_example.py or example\_test.py

Class name should start with “Test”, as in TestExample

Test method names should start with “test\_”, as in test\_example

You can refer to the official documentation for more details:

http://pytest.readthedocs.io/en/reorganize-docs/new-docs/user/naming\_conventions.html

**Pytest Fixture Update**

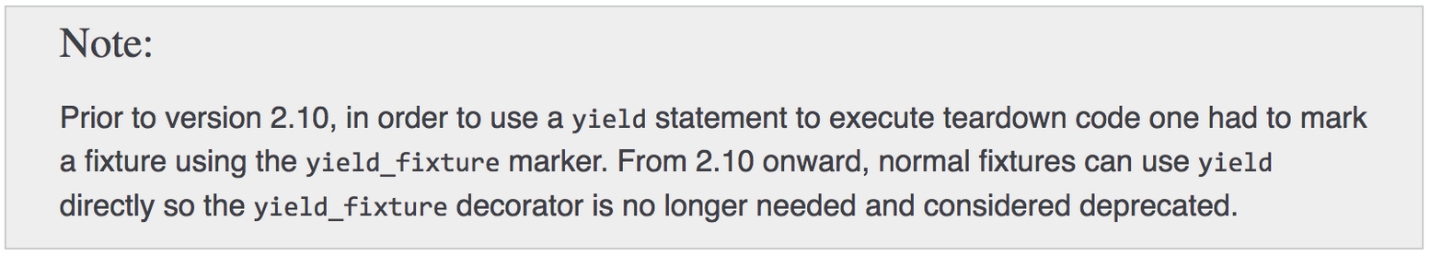
After pytest version 2.10, you do not need @pytest.yield\_fixture explicitly to use yield.

The default @pytest.fixture also supports yield.

It means the code will work if you are using @pytest.yield\_fixture or @pytest.fixture.

This screenshot is from pytest website:

http://doc.pytest.org/en/latest/fixture.html



**Important Note**

In the next lecture, we will be using a wildcard "\*" -> Asterisk for running multiple files with one command.

Windows machine does not recognize "\*" and it cannot be used on that.

If you are using Windows, then please use the full file name, instead of using wildcards.

[Lets](https://www.udemy.com/user/criticalthinking/) — Instructor  
24 days ago

**0**

Hi,  
  
It is confusing about those number and what the stack function does? Need a brief info on that?  
As explained above, the [inspect](https://docs.python.org/3/library/inspect.html#module-inspect) module provides several useful functions to help get information about live objects such as modules, classes, methods, functions, tracebacks, frame objects, and code objects.  
  
Why have you used it?  
inspect.stack()[1][3] -> It is used to get the name of the module which is calling the custom logger.