2.19.2 Supported nose Idioms

- setup and teardown at module/class/method level
- SkipTest exceptions and markers
- setup/teardown decorators
- `__test__` attribute on modules/classes/functions
- general usage of nose utilities

2.19.3 Unsupported idioms / known issues

- unittest-style `setUp, tearDown, setUpClass, tearDownClass` are recognized only on `unittest.TestCase` classes but not on plain classes. `nose` supports these methods also on plain classes but pytest deliberately does not. As nose and pytest already both support `setup_class, teardown_class, setup_method, teardown_method` it doesn't seem useful to duplicate the unittest-API like nose does. If you however rather think pytest should support the unittest-spelling on plain classes please post to this issue.
- nose imports test modules with the same import path (e.g. `tests.test_mode`) but different file system paths (e.g. `tests/test_mode.py` and `other/tests/test_mode.py`) by extending sys.path/import semantics. pytest does not do that but there is discussion in #268 for adding some support. Note that nose2 choose to avoid this sys.path/import hackery.

  If you place a conftest.py file in the root directory of your project (as determined by pytest) pytest will run tests "nose style" against the code below that directory by adding it to your `sys.path` instead of running against your installed code.

  You may find yourself wanting to do this if you ran `python setup.py install` to set up your project, as opposed to `python setup.py develop` or any of the package manager equivalents. Installing with develop in a virtual environment like tox is recommended over this pattern.
- nose-style doctests are not collected and executed correctly, also doctest fixtures don't work.
- no nose-configuration is recognized.
- `yield`-based methods are unsupported as of pytest 4.1.0. They are fundamentally incompatible with pytest because they don't support fixtures properly since collection and test execution are separated.

Here is a table comparing the default supported naming conventions for both nose and pytest.

Convention	nose	pytest
Ⓜ test*.py	✓	
Ⓜ test_*.py	✓	✓
Ⓜ *_test.py		✓
Ⓜ *_tests.py		
Ⓒ *(unittest.TestCase)	✓	✓
ⓜ test_*	✓	✓
Ⓒ Test*		✓
ⓜ test_*		✓
ⓕ test_*		✓

Symbols are described below