Examples and customization tricks

Here is a (growing) list of examples. Contact us if you need more examples or have questions. Also take a look at the comprehensive documentation which contains many example snippets as well. Also, pytest on stackoverflow.com often comes with example answers.

For basic examples, see

- Installation and Getting Started for basic introductory examples
- Asserting with the assert statement for basic assertion examples
- pytest fixtures: explicit, modular, scalable for basic fixture/setup examples
- Parametrizing fixtures and test functions for basic test function parametrization
- unittest.TestCase Support for basic unittest integration
- Running tests written for nose for basic nosetests integration

The following examples aim at various use cases you might encounter.

- Demo of Python failure reports with pytest
- Basic patterns and examples
  - Pass different values to a test function, depending on command line options
  - Dynamically adding command line options
  - Control skipping of tests according to command line option
  - Writing well integrated assertion helpers
  - Detect if running from within a pytest run
  - Adding info to test report header
  - profiling test duration
  - incremental testing - test steps
  - Package/Directory-level fixtures (setups)
  - post-process test reports / failures
  - Making test result information available in fixtures
  - **`PYTEST_CURRENT_TEST`** environment variable
  - Freezing pytest
- Parametrizing tests
  - Generating parameters combinations, depending on command line
  - Different options for test IDs
  - A quick port of “testscenarios”
  - Deferring the setup of parametrized resources
  - Apply indirect on particular arguments
  - Parametrizing test methods through per-class configuration
  - Indirect parametrization with multiple fixtures
  - Indirect parametrization of optional implementations/imports
- Working with custom markers
  - Marking test functions and selecting them for a run
  - Selecting tests based on their node ID
  - Using **`-k expr`** to select tests based on their name
  - Registering markers
  - Marking whole classes or modules
  - Marking individual tests when using parametrize
  - Custom marker and command line option to control test runs
  - Passing a callable to custom markers
  - Reading markers which were set from multiple places
  - marking platform specific tests with pytest
  - Automatically adding markers based on test names
- A session-fixture which can look at all collected tests