1. `-v` (Verbose Output)

Provides detailed output for each test.

Example:

```bash
pytest -v
```

Output:

```plaintext
test_example.py::test_add PASSED
test_example.py::test_subtract FAILED
```

---

2. `-q` (Quiet Mode)

Reduces the verbosity of the output.

Example:

```bash
pytest -q
```

Output:

```plaintext
..F
```

---

3. `--maxfail`

Stops test execution after a specified number of failures.

Example:

```bash
pytest --maxfail=2
```

Stops after two test failures.

1. `-v` (Verbose Mode)

4. `--tb` (Traceback Style)

Controls the traceback output format when a test fails. Common options:

- `short`: Minimal traceback.
- `line`: Single line per failure.
- `no`: No traceback.

Example:

```bash
pytest --tb=short
```

Output:

```plaintext
E       AssertionError: Expected 3, got 2
test_example.py:12: AssertionError
```

5. `-k` (Keyword Expression)

Runs only tests that match the given keyword expression.

Example:

```bash
pytest -k "add or subtract"
```

Runs tests with names containing `add` or `subtract`.

6. `-m` (Run Specific Markers)

Runs only tests with specific markers.

Example:

```python
# test_example.py
import pytest

@pytest.mark.slow
def test_slow_function():
    pass

@pytest.mark.fast
def test_fast_function():
    pass
```

Run only `slow` tests:

```bash
pytest -m "slow"
```

7. `--disable-warnings`

Disables warning messages during test execution.

Example:

```bash
pytest --disable-warnings
```

8. `--capture`

Controls the capture mode for test output:

- `no`: Disables output capturing.
- `fd`: Captures `stdout`/`stderr` (default).
- `sys`: Captures Python `sys.stdout`/`sys.stderr`.

Example:

```bash
pytest --capture=no
```

Displays all print statements during test execution.

---

9. `--durations`

Displays the slowest tests after execution.

Example:

```bash
pytest --durations=3
```

Displays the top 3 slowest tests.

Output:

```plaintext
3.00s test_example.py::test_slow_function
2.00s test_example.py::test_another_function
```

---

10. `--junitxml`

Generates a JUnit XML report for integration with CI tools.

Example:

```bash
pytest --junitxml=report.xml
```

---

11. `--html` (Requires `pytest-html`)

Generates an HTML report.

Example:

```bash
bash                                    Copy code

pytest --html=report.html
```

12. `-x` (Exit on First Failure)

Stops the test session immediately after the first failure.

Example:

```bash
bash                                    Copy code

pytest -x
```

13. `--lf` (Re-run Last Failed Tests)

Runs only the tests that failed in the last session.

Example:

```bash
bash                                    Copy code

pytest --lf
```

14. `--ff` (Re-run Last Run Tests First)

Reorders the tests to run the previously failed tests first.

Example:

```bash
bash                                    Copy code

pytest --ff
```

15. `--collect-only`

Lists all the test cases without executing them.

Example:

```bash
pytest --collect-only
```

16. `-n` (Parallel Execution with `pytest-xdist`)

pip install pytest.xdist

Executes tests in parallel across multiple processes.

Example:

```bash
pytest -n 4
```

Runs tests in 4 parallel processes.

17. `--markers`

Lists all available markers in the project.

Example:

```bash
pytest --markers
```

18. `--help`

Displays the complete list of pytest options.

Example:

```bash
pytest --help
```

Cheat Sheet

Flag	Description
-v	Verbose output
-q	Quiet mode
--maxfail	Stop after specified failures
--tb	Modify traceback output
-k	Run tests matching keywords
-m	Run tests with specific markers
--disable-warnings	Disable warnings
--capture	Control output capturing
--durations	Show slowest tests
--junitxml	Generate JUnit XML report
--html	Generate HTML report (with pytest-html)
-x	Stop on first failure
--lf	Re-run last failed tests
--ff	Re-run last tests first
--collect-only	List all tests without running them
-n	Run tests in parallel