python -m venv venv  
source venv/Scripts/activate  
pip install -r requirements.txt  
  
python --version  
python --V  
  
pip list  
  
pip install robotframework  
pip install robotframework==2.9.2  
pip install --upgrade robotframework  
pip uninstall robotframework  
  
pip show robotframework  
pip check robotframework

# Any pytest file should start with test\_ or end with test\_  
# pytest method name should start with test.

# Any code should be wrapped in method only.

# Method name should be sensible

# -k stands for method names execution, -s logs in output, -v stands for more info metadata

# You can run specific file with py.test <filename>

# you can mark (tag) tests @pytest.mark.smoke and then run with -m  
# you can skip tests with @pytest.mark.skip  
# @pytest.mark.xfail  
# fixtures are used as setup and tear down methods for test cases- conftest file to generalize fixture  
# fixture and make it available to all test cases (fixture name into parameters of method)  
# data driven and parameterization can be done with return statements in tuple format  
# when you define fixture scope to class only, it will run once before class is initiated and at the end

*"""  
--------  
Log levels  
--------  
DEBUG  
INFO  
WARNING  
ERROR  
CRITICAL  
"""*

--------------------------------------------

Run all the pytest tests using command line:

--------------------------------------------

py.test

py.test -v

py.test -v -s

To run the all tests:

py.test -v -s

To run specific test:

py.test -v -s test\_home\_page\_title.py

To run tests that have the name specified:

py.test -k creditcard -v -s

To run the tests marked:

py.test -m smoke -v -s

To pass the browser name in the command line:

py.test -–browser\_name firefox

To generate the html report:

py.test -–html=report.html

py.test -s -v -n=3 -–html=reports/report.html tests/test\_login\_page\_title.py -–browser chrome

To run the last failed tests:

py.test -v -s test\_rerun.py –lf

To run the failed tests first:

py.test -v -s test\_rerun.py –ff

To re-run the failed tests:

py.test -s -v tests/test\_rerun.py --reruns 5

To re-run the failed tests with delay:

py.test -s -v tests/test\_rerun.py --reruns 5 –reruns-delay 1

To re-run the tests failed with particular error:

py.test -s -v tests/test\_rerun.py –only-rerun AssertionError

To re-run the tests failed with multiple particular error:

py.test -s -v tests/test\_rerun.py –only-rerun AssertionError –only-rerun ValueError

To report the test failures instantly:

py.test -s -v tests/test\_rerun.py –instafail

To capture the screen shot on failure:

pytest --screenshot=on --screenshot\_path=on -s -v tests/test\_home\_page\_title.py

To put the logger logs to html report:

py.test -s -v --html=reports/report.html --capture=tee-sys

To generate the junit xml report:

py.test –browser chrome --html=reports/report.html -v –-junitxml=”reports/result.xml”

To test the runs parallelly:

py.test -n 2 -k title

To test disable warnings:

py.test -s -v -m regression --disable-warnings

To start the Jenkins server:

java -jar Jenkins.war -httpPort=9090java -jar Jenkins.war -httpPort=8080

Docker commands:  
cd C:\PycharmProjects\pomproject  
docker build -t pomproject:1.0 .  
docker run -ti pomproject:1.0 /bin/sh  
export

docker run -e GROUP=”smoke” pomproject:1.0

docker run pomproject:1.0

List all the docker images:

docker images

List all the docker containers:

docker container ps

Stop a docker container:

docker container stop <container\_id>

Remove a docker container:

docker container rm <container\_id>

Remove all Docker images at once:  
$images = docker images -a -q

foreach ($image in $images) { docker image rm $image -f }