## Команды

## pip freeze > requirements.txt

## Эта команда сохранит все версии пакетов в специальный файл **requirements.txt.**

pip install -r requirements.txt

В свежем окружении все пакеты установлены одной командой!

## pytest scripts/selenium\_scripts # найти все тесты в директории scripts/selenium\_scripts

## pytest test\_user\_interface.py # найти и выполнить все тесты в файле

## pytest scripts/drafts.py::test\_register\_new\_user\_parametrized # найти тест с именем test\_register\_new\_user\_parametrized в указанном файле в указанной директории и выполнить

## - v – verbose – подробный отчёт

## pytest –v test\_abs.py

## Другие полезные команды для манипуляции выводом тестов PyTest можно найти по ссылке: [Useful py.test commands.](https://gist.github.com/amatellanes/12136508b816469678c2) (<https://gist.github.com/amatellanes/12136508b816469678c2>)

указать параметр **-s**, чтобы увидеть текст, который выводится командой print().

pytest -s test\_fixture1.py

## pytest -s -v test\_fixture2.py

## pytest -s -v -m smoke test\_fixture8.py

## pytest -s -v -m "not smoke" test\_fixture8.py

## pytest -s -v -m "smoke or regression" test\_fixture8.py

## pytest -s -v -m "smoke and win10" test\_fixture81.py

## smoke - метка критичных тестов,  которые нужно запускать на каждый коммит разработчиков

## regression – метка регрессионных тестов, которые запускаются только перед релизом

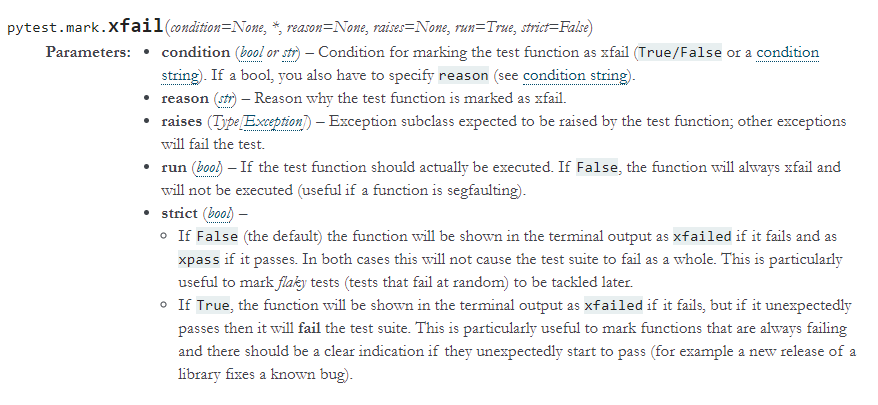
## **@pytest.mark.skip** - метка, которой отмечают в коде, чтобы пропустить тест

## @pytest.mark.xfail

## @pytest.mark.xfail(reason="fixing this bug right now")

## pytest -rx -v test\_fixture10a.py -->> -rx – для xfail+reason

## pytest -rX -v test\_fixture10b.py -->> X в параметре -r, чтобы получить подробную информацию по XPASS-тестам



сondition - (условие) - условие при котором тестовая функция помечается как xfail (не работающая) , принимает либо булево значение либо строку, при булевом значении, нужно будет указать дополнительный параметр reason(причина)

reason - (причина) - причина по которой функция помечена как xfail, принимается строка

raises - (выбрасывания исключений) подкласс исключений которые ожидаются при вызове теста, другие исключения завалят тест

run - (запуск) - параметр отвечающий за запуск теста, при значении False будет всегда xfail и не будет запускаться. принимается булевы значения

strict - (строгость) значение False по умолчанию,  в выводе терминала  будет xfailed, если функция провалила тест и xpass, если прошла. при значении True в выводе терминала будет xfailed, если функция не прошла тест, но если фунцкция неожиданно прошла тест то будет fail

## <https://pytest-docs-ru.readthedocs.io/ru/latest/skipping.html>

## xfail (x): Тест не должен был пройти, был запущен и провалился. XPASS (X): Тест не должен был пройти, был запущен и прошел! Вроде разные понятия, но один параметр xfail с False это запретить

Для себя сделал шпаргалку, что бы все в одном месте по маркировке и пропускам(постарался вместить основное на мой взгляд)  
После маркировок @pytest.mark..., через двоеточие указывал команду для запуска скрипта  
Может кому тоже пригодится..  
  
**Маркировка тестов и пропуск тестов**

**# Запустится только smoke:**  
@pytest.mark.smoke : pytest -s -v -m smoke test\_fixture.py

**# Запуска всех тестов, не отмеченных как smoke:**  
@pytest.mark.smoke : pytest -s -v -m "not smoke" test\_fixture.py

**# Запуск тестов с разными маркировками:**  
@pytest.mark.smoke  
@pytest.mark.regression : pytest -s -v -m "smoke or regression" test\_fixture.py

**# Запуск тестов имеющих несколько маркировок:**  
@pytest.mark.smoke  
@pytest.mark.win10 : pytest -s -v -m "smoke and win10" test\_fixture.py

**# Пропуск тестов:**  
@pytest.mark.skip : pytest -s -v  test\_fixture.py

**# Помечать тест как ожидаемо падающий(пометка:XFAIL):  
# упавший тест теперь отмечен как xfail, но результат прогона тестов помечен как успешный  
# Когда баг починят, мы это узнаем, так как тест будет отмечен как XPASS**  
@pytest.mark.xfail : pytest -rx -v test\_fixture.py

**# reason - Чтобы увидеть это сообщение в консоли, при запуске нужно добавлять параметр pytest -rX**  
@pytest.mark.xfail(reason="fixing this bug right now") : pytest -rX -v test\_fixture.py

# Параметр **strict**  
# Ни XFAIL, ни XPASS по умолчанию не приводят к падению всего набора тестов.  
# Но это можно изменить, установив параметру strict значение True:  
# В этом случае, если тест будет неожиданно пройден (XPASS),  
# то это приведет к падению всего тестового набора  
@pytest.mark.xfail(strict=True) : pytest -rX -v test\_fixture.py