# Office Hours 5 Newsletter

Office Hours 5 Newsletter

Date: 10/3/20

TA: Zhenyu Zhao

Topics:

* Semantic versioning
* tox vs. Travis CI in depth
* pytest in the context of tox

Excerpt:

* For general info on setuptools\_scm, check out the official [README.md](https://github.com/pypa/setuptools_scm" \t "/home/dhirendra/Documents\\x/_blank) at: [https://github.com/pypa/setuptools\_scm](https://github.com/pypa/setuptools_scm" \t "/home/dhirendra/Documents\\x/_blank)
* Generally speaking, there are three pieces in tandem to get semantic versioning to work: a) add an argument called use\_scm\_version into setup() in setup.py, and b) src/csci\_utils/\_\_init\_\_.py needs to be updated for retrieving versions at runtime (see previous [URL](https://github.com/pypa/setuptools_scm" \t "/home/dhirendra/Documents\\x/_blank)), and c) “git tag” and “git push --tags” are issued after a commit/push
* In an editable installation environment, the imported package’s metadata needs to be refreshed so “import csci\_utils; print(csci\_utils.\_\_version\_\_)” can get the latest versions. As a result, “pipenv update” needs to issued and it invokes setup() indirectly.
* “python setup.py --version” always returns the newer version since it invokes setup() directly.
* If you see versions like “1.3.dev22”, it means the local changes have not committed and there is no new tag yet.
* In a typical .travis.yml, the test stage launches “pytest” whereas the tox stage launches “tox -v”. See "script:" in .travis.yml
* A Travis CI job associated with the test stage runs “pytest --cov-report xml --cov-report term”. No tox involved here. If there is any error, run “pytest” locally to troubleshoot.
* A Travis CI job associated with the tox stage runs “tox -v”. If there is any error, run “tox” locally to troubleshoot. In your local pipenv environment, you need to first install tox (run “pip install” not “pipenv install” so the tox package doesn’t get added to Pipfile\*). Some useful tox commands are: a) “tox -e py37” runs the default py37 test environment, and b) “tox -al” lists all the test environments. tox follows instructions in tox.ini. Eventually tox launches pytest (see “commands=” in [testenv]).
* You can make stages in .travis.yml conditional so non-master push only triggers test not tox. See Pset1's .travis.yml
* You should always add at least one test to test code; otherwise, tox will fail since tox checks pytest’s return code and it thinks pytest has failed.
* In a typical cookiecutter-generated project directory, there is no pytest.ini so pytest retrieves arguments and options from command-line, tox.ini and setup.cfg in order. See “Initialization: determining rootdir and configfile” in the following [URL](https://docs.pytest.org/en/stable/customize.html" \t "/home/dhirendra/Documents\\x/_blank) on how pytest determines rootdir and configfile at [https://docs.pytest.org/en/stable/customize.html](https://docs.pytest.org/en/stable/customize.html" \t "/home/dhirendra/Documents\\x/_blank)

#\_\_version\_\_ = '0.1.0'  
#from .\_csci\_utils import longest # noqa

[testenv:bootstrap]  
deps =  
 jinja2  
 matrix  
 tox  
skip\_install = true  
commands =  
 python ci/bootstrap.py --no-env  
passenv =  
 \*  
; a generative tox configuration, see: https://tox.readthedocs.io/en/latest/config.html#generative-envlist  
  
[tox]  
envlist =  
 clean,  
 check,  
 {py27,py35,py36,py37,py38,pypy,pypy3},  
 report  
ignore\_basepython\_conflict = true  
  
[testenv]  
basepython =  
 py36: {env:TOXPYTHON:python3.6}  
 py37: {env:TOXPYTHON:python3.7}  
 py38: {env:TOXPYTHON:python3.8}  
 {bootstrap,clean,check,report}: {env:TOXPYTHON:python3}  
setenv =  
 PYTHONPATH={toxinidir}/tests  
 PYTHONUNBUFFERED=yes  
 cover: SETUP\_PY\_EXT\_COVERAGE=yes  
passenv =  
 \*  
usedevelop = false  
deps =  
 pytest  
 pytest-travis-fold  
 pytest-cov  
 Cython  
commands =  
 {posargs:pytest --cov --cov-report=term-missing -vv }  
  
[testenv:check]  
deps =  
 docutils  
 flake8  
 readme-renderer  
 pygments  
 isort  
 setuptools-scm  
skip\_install = true  
commands =  
 python setup.py check --strict --metadata --restructuredtext  
 flake8  
 isort --verbose --check-only --diff --filter-files .  
  
[testenv:cythonize]  
basepython = {env:TOXPYTHON:python}  
deps =  
 Cython  
skip\_install = true  
commands =  
 {posargs:python setup.py clean --all build\_ext --force}  
  
[testenv:report]  
deps =  
 coverage  
 Cython  
skip\_install = true  
commands =  
 coverage report  
 coverage html  
  
[testenv:clean]  
commands = coverage erase  
skip\_install = true  
deps =  
 coverage  
 Cython

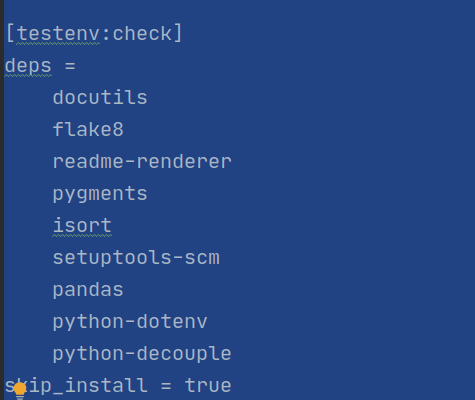
**Notes:**

**Tox setting**

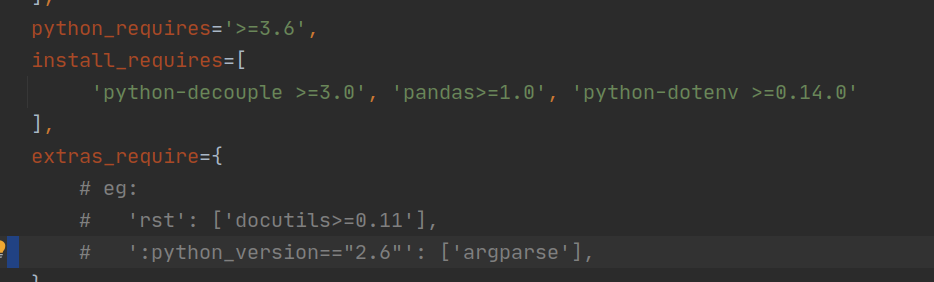
tox doesn't use your Pipfile to build its test virtual environment. It uses its own. You can put pandas into "deps=" in [testenv] in tox.ini.

Alternatively, add pandas into "install\_requires=[]" in setup.py

Setting in the Tox



**Setting in the setup.py**



Yaml file at 9:00 pm 04 October 4, 2020

language: python  
dist: xenial  
python:  
 - 3.7  
cache: false  
env:  
 global:  
 - LD\_PRELOAD=/lib/x86\_64-linux-gnu/libSegFault.so  
 - SEGFAULT\_SIGNALS=all  
 - LANG=en\_US.UTF-8  
matrix:  
 include:  
 - env:  
 - TOXENV=py36  
 python: '3.6'  
 stage: tox  
 - stage: test  
 python: '3.7'  
 install:  
 *# - pip install pipenv* - pipenv install --dev --ignore-pipfile --deploy  
 before\_script: *# code coverage tool* - curl -L https://codeclimate.com/downloads/test-reporter/test-reporter-latest-linux-amd64 > ./cc-test-reporter  
 - chmod +x ./cc-test-reporter  
 - ./cc-test-reporter before-build  
 script: pytest --cov-report xml --cov-report term  
 after\_script:  
 - ./cc-test-reporter after-build --exit-code $TRAVIS\_TEST\_RESULT  
 - env:  
 - TOXENV=py37  
 python: '3.7'  
  
 *#- env:  
 #- TOXENV=py38  
 #python: '3.8'*before\_install:  
 - python --version  
 - uname -a  
 - lsb\_release -a || true  
install:  
 - pip install pipenv  
 - pip install pytest  
 *# - pipenv install pandas  
 # - pipenv install openpyxl  
 # - pipenv install python-dotenv* - pipenv install --system --deploy --ignore-pipfile  
 - python -mpip install --progress-bar=off tox-travis -rci/requirements.txt  
 - virtualenv --version  
 - easy\_install --version  
 - pip --version  
 - tox --version  
script:  
 - tox -r -v  
after\_failure:  
 - cat .tox/log/\*  
 - cat .tox/\*/log/\*  
notifications:  
 email:  
 on\_success: never  
 on\_failure: always  
stages:  
 - test  
 - name: tox  
 if: branch = master