On code quality

June 9, 2022

1 Vim

The attached configuration in .vimrc is minimalistic and pragmatic:

:imap jj ¡Esc¿ set expandtab ts=4 sw=4 ai

imap replaces esc with a double pressing of j. *Expandtab* converts tabs in N blank spaces, ts=4 set N=4. *ai* is auto-indention with width sw=4. Typing :retab replaces the current tabs with whitespaces.

2 Pylint and Pytest

Writing clean code is essential: from an algorithmic viewpoint (performance) anf for readability and maintenance.

The tool *pylint* gives a good insight on your code quality, and its straightforward use via *python3 -m pylint pythonfile.py*.

Every time you implement a new function, in principle you should also write a test to check that this function works as expected. For example you can test against some trivial conditions, easy cases or extreme values on which the code has a known output. Writing tests can seem to be time consuming, but it is **very** convenient in the long run: every time you modify some parts of your code, if the tests are still successful, you have some chance to believe you didn't damage your work.

In python you can use *pytest*. Running it on a script will just run all functions whose name start with the word 'test' followed by an underscore. Remember that each function of such a kind should end with an assert condition. See the attached code for a minimalistic example.

3 Install a Python module locally

$python3 -m pip install –user MODULENAME

4 Saying 'yes' to rm for all the files

$yes — rm -R folder