

Python Tooling

Guilhem Saurel







This presentation

Available at

Under License



https://creativecommons.org/licenses/by-sa/4.0/



This presentation (continued)

Source

Discussions

https://im.laas.fr/#/room/#python-tooling:laas.fr



Outline

- 1 Motivation
- 2 Style
- 3 Packaging
- 4 Tests
- 5 Coverage
- 6 Static analysis
- 7 Meta





The Zen of Python, by Tim Peters (extracts)

- Beautiful is better than ugly.
- Readability counts.
- There should be one- and preferably only one -obvious way to do it.

Motivation 6/4:



Write readable code

Code is read much more often than it is written

Guido



Write readable code

Code is read much more often than it is written

- Guido

- \blacksquare read \rightarrow write
- teamwork
- future self





Style Guide for Python Code

https://peps.python.org/pep-0008/

- indentation: 4 spaces
- maximum line length: 79
- whitespaces
- comments
- naming

Style



pep8 → pycodestyle

https://github.com/PyCQA/pycodestyle



https://github.com/PyCQA/pycodestyle

```
$ pycodestyle --first optparse.py
optparse.py:69:11: E401 multiple imports on one line
optparse.py:77:1: E302 expected 2 blank lines, found 1
optparse.py:88:5: E301 expected 1 blank line, found 0
optparse.py:347:31: E211 whitespace before '('
optparse.py:357:17: E201 whitespace after '{'
optparse.py:472:29: E221 multiple spaces before operator
```



PEP 257 / pydocstyle

https://github.com/PyCQA/pydocstyle

Mostly helps as a reminder to write some doc



flake8

https://github.com/PyCQA/flake8 pycodestyle + pyflakes + mccabe



```
https://github.com/google/yapf
clang-format/gofmt
```

```
x = \{ 'a':37, 'b':42, \}
```

```
'c':927}
```

```
https://github.com/google/yapf
clang-format / gofmt
x = { 'a':37,'b':42,
```

$$x = { 'a': 37, 'b': 42, 'c': 927 }$$



https://github.com/psf/black



Figure 1: black



black main caveat

```
super_long_line.with_small_argument = [0] # some commen
```



black main caveat

black main caveat

super_long_line.with_small_argument = [0]



isort

https://github.com/PyCQA/isort sorts imports.







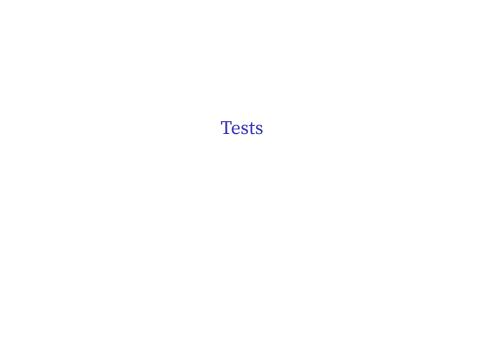
https://github.com/python-poetry/poetry

- poetry init
- pyproject.toml
- minimal python version
- dependencies / dev-dependencies
- virtualenv
- sdist / wheel builder



Others

ref. "Managing Python Packages"



unittest

https://docs.python.org/3/library/unittest.html import unittest class TestStringMethods(unittest.TestCase): def test_upper(self): self.assertEqual('foo'.upper(), 'F00') def test_isupper(self): self.assertTrue('F00'.isupper()) self.assertFalse('Foo'.isupper()) if __name__ == '__main__': unittest.main()

Tests



unittest launch

```
$ python test_string.py
...
```

Ran 2 tests in 0.000s

OK



unittest launch

```
$ python test_string.py
Ran 2 tests in 0.000s
0K
ou
$ python -m unittest
Ran 2 tests in 0.000s
OK
```



https://docs.python.org/3/library/doctest.html

```
def factorial(n):
    """Return the factorial of n, an exact integer \geq 0.
    >>> [factorial(n) for n in range(6)]
    [1, 1, 2, 6, 24, 120]
    >>> factorial(30)
    265252859812191058636308480000000
    >>> factorial(-1)
    Traceback (most recent call last):
    ValueError: n must be >= 0
    11 11 11
```

Tests



```
LAAS
CNRS
```

```
if __name__ == "__main__":
    import doctest
    doctest.testmod()
```



Others

- pytest
- tox





Coverage.py

https://github.com/nedbat/coveragepy



Coverage.py

https://github.com/nedbat/coveragepy

\$ coverage html

sample



online

- https://coveralls.io/
- https://about.codecov.io/



Gitlab

 $Settings \rightarrow CI/CD \rightarrow General\ pipelines \rightarrow Test\ coverage\ parsing$



Mypy: motivation

```
def add(a: int, b: int) -> int:
    """Performs addition on integers.
    >>> add(3, 4)
    11 11 11
    return a + b
if __name__ == "__main__":
    import sys
    print(add(sys.argv[1], sys.argv[2]))
```



Mypy

\$ python add.py 3 4
34

```
https://github.com/asottile/pyupgrade
  class C(Base):
    def f(self):
        super(C, self).f()
+        super().f()
```



Pylint

pass...





Editor / IDE integration

Bring your own;)



https://github.com/pre-commit/pre-commit

```
.pre-commit-config.yaml
repos:
   repo: https://github.com/pre-commit/pre-commit-hooks
   rev: v4.2.0
   hooks:
   - id: check-yaml
   - id: end-of-file-fixer
       id: trailing-whitespace
   repo: https://github.com/psf/black
   rev: 22.3.0
   hooks:
     id: black
```

Meta



LAAS CNRS

pre-commit usage

pre-commit run -a

```
LAAS
CNRS
```

```
pre-commit run -a
pre-commit install
```



```
pre-commit run -a
pre-commit install
demo
```



LAAS CNRS

pre-commit CI

https://pre-commit.ci/



Badges

codecov 94%

pre-commit.ci passed

code style black

coverage 96.00%



Questions?

Thanks for your time:)



Proposition

- flake8
- black
- isort
- pyupgrade
- pre-commit