Tooling

for Python & C++

Guilhem Saurel

Available at

https://homepages.laas.fr/gsaurel/talks/
tooling.pdf

Under License

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

Source
https://gitlab.laas.fr/gsaurel/talks : tooling.md

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

Motivation

Code is read much more often than it is written

— Guido Van Rossum

Code is read much more often than it is written

— Guido Van Rossum

- read → write
- teamwork
- future self

Style

Style Guide for Python Code

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

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

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:347:31: E211 whitespace before '('
optparse.py:357:17: E201 whitespace after '{'
optparse.py:472:29: E221 multiple spaces before operator
```

https://github.com/PyCQA/flake8

pycodestyle + pyflakes + mccabe

https://clang.llvm.org/docs/ClangFormat.html

```
$ clang-format -i QGVCore/QGVScene.cpp

@@ -173,14 +173,11 @@
   // Update items layout
-  foreach( QGVNode* node, _nodes ) {
-    node->updateLayout();
-  }
+  foreach (QGVNode *node, _nodes) node->updateLayout();
```

https://github.com/google/yapf

based on clang-format / gofmt, but for python

```python
x = {   'a':37,'b':42,

'c':927}
```

https://github.com/google/yapf

based on clang-format / gofmt, but for python

```python
x = {   'a':37,'b':42,

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

https://github.com/psf/black

Figure 1: black

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

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

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

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

super_long_line.with_small_argument = [
    0
]   # some comment

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

Static analysis

```python
def add(a: int, b: int) -> int:
    """Performs addition on integers.

    >>> add(3, 4)
    7
    """
    return a + b


if __name__ == "__main__":
    import sys

    print(add(sys.argv[1], sys.argv[2]))
```

Static analysis

```
$ python add.py 3 4
34
```

```
$ python add.py 3 4
34

$ mypy add.py
add.py:13: error: Argument 1 to "add" has incompatible
                  type "str"; expected "int"
add.py:13: error: Argument 2 to "add" has incompatible
                  type "str"; expected "int"
Found 2 errors in 1 file (checked 1 source file)
```

https://github.com/asottile/pyupgrade

```
 class C(Base):
     def f(self):
-        super(C, self).f()
+        super().f()
```

https://github.com/asottile/pyupgrade

```
 class C(Base):
     def f(self):
-        super(C, self).f()
+        super().f()
```

For C++: `clang-tidy`

Meta

Bring your own ;)

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

```
$ cat .pre-commit-config.yaml

repos:
-   repo: https://github.com/psf/black
    rev: 22.3.0
    hooks:
    -   id: black
-   repo: https://github.com/pre-commit/mirrors-clang-fo
    rev: v14.0.3
    hooks:
    -   id: clang-format
        args: [-i, --style=Google]
```

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

```
pre-commit run -a

pre-commit install
```

https://pre-commit.ci/

Thanks for your time :)