Conception Orientée Objet, Outils

Université Toulouse Paul Sabatier - KEAR9RA1

Guilhem Saurel

Available at

`https://homepages.laas.fr/gsaurel/talks/`
`conception-orientee-objet-outils.pdf`

Under License

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

Source

```
https://gitlab.laas.fr/gsaurel/talks :
  conception-orientee-objet-outils.md
```
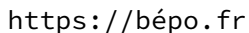
Discussions

https://im.laas.fr/#/room/#conception-orientee-objet:laas.fr

Vous

- hauteurs
- position
- distances

- hauteurs
- position
- distances

Adaptez votre environement

`https://bépo.fr`

Style

Code is read much more often than it is written

— Guido Van Rossum

```
https://peps.python.org/pep-0008/
print ( "hello" )
```

```
https://peps.python.org/pep-0008/
print ( "hello" )

└>$ flake8 hello-spaces.py
hello-spaces.py:1:6: E211 whitespace before '('
hello-spaces.py:1:8: E201 whitespace after '('
hello-spaces.py:1:16: E202 whitespace before ')'
```

```
https://github.com/psf/black

↳$ black hello-spaces.py
reformatted hello-spaces.py

All done!
1 file reformatted.
```

```
https://github.com/psf/black

└─>$ black hello-spaces.py
reformatted hello-spaces.py

All done!
1 file reformatted.
```
```diff
-print ( "hello" )
+print("hello")
```

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

```
└>$ clang-format -i QGVScene.cpp

-for( auto node : _nodes ) {
-   node->update_layout();
-}
+for (auto *node : _nodes) node->update_layout();
```

Analyse Statique

```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]))
```

Analyse Statique

```
└>$ 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()
```

En C++: `clang-tidy`

Méta

Bring your own ;)

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

```
└─>$ cat .pre-commit-config.yaml
```

```yaml
repos:
-   repo: https://github.com/psf/black
    rev: 22.8.0
    hooks:
    -   id: black
-   repo: https://github.com/pre-commit/mirrors-clang-fo
    rev: v14.0.6
    hooks:
    -   id: clang-format
```

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

```
└─>$ pre-commit run -a
```

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

└>$ pre-commit run -a

└>$ pre-commit install

https://pre-commit.ci/