PyLint: The Good, The Bad, and the Ugly

Moshe Zadka – https://cobordism.com

Acknowledgement of Country

Belmont (in San Francisco Bay Area Peninsula)
Ancestral homeland of the Ramaytush Ohlone people

Lint

Lint

Find problems

Lint

Find problems
in code

Lint

Find problems
in code
without running it

Linters

Linters

Black: Can fix any problem it finds!

Linters

Black: Can fix any problem it finds!
Flake8: Best practices + plugins

Linters

Black: Can fix any problem it finds!
Flake8: Best practices + plugins
You are probably already using them

So why PyLint?

So why PyLint?

What does it give you?

PyLint

PyLint

...is good actually.

Redefining functions

```
def test_add_small():
    # Math, am I right?
    assert 1 + 1 == 3

def test_add_large():
    assert 5 + 6 == 11

def test_add_small():
    assert 1 + 10 == 11
```

Test works!

```
collected 2 items

test.py ..
2 passed
```

PyLint: The Good

```
test.py:8:0: E0102: function already defined line 1
     (function-redefined)
```

PyLint: The Bad

```
"""Inventory abstractions"""

import attrs

@attrs.define
class Laptop:
    """A laptop"""
    ident: str
    cpu: str
```

PyLint: The Bad

```
$ pylint laptop.py | sed -n '/^laptop/s/[^ ]*: //p'
R0903: Too few public methods (0/2) (too-few-public-
```

PyLint: The Ugly

"People will just disable the whole check if it's too picky"
PyLint issue 6987, July 3rd, 2022

Pin PyLint

Make sure to always use the same version

Pin PyLint

Make sure to always use the same version
...until you choose to upgrade!

Pin PyLint: Loose

```
# pyproject.toml

[project.optional-dependencies]
pylint = ["pylint"]
```

Pin PyLint: Strict

```
# noxfile.py
...
@nox.session(python=VERSIONS[-1])
def refresh_deps(session):
    """Refresh the requirements-*.txt files"""
    session.install("pip-tools")
    for deps in [..., "pylint"]:
        session.run(
            "pip-compile",
            "--extra",
            deps,
            "pyproject.toml",
            "--output-file",
            f"requirements-{deps}.txt",
        )
```

Pin PyLint: Strict

```
# noxfile.py
...
@nox.session(python=VERSIONS[-1])
def refresh_deps(session):
    """Refresh the requirements-*.txt files"""
    session.install("pip-tools")
    for deps in [..., "pylint"]:
        session.run(
            "pip-compile",
            "--extra",
            deps,
            "pyproject.toml",
            "--output-file",
            f"requirements-{deps}.txt",
        )
```

...or use a service.

A few of my favorite things

```
checkers = [
    "missing-class-docstring",
    "missing-function-docstring",
    "missing-module-docstring",
    "function-redefined",
]
```

Default Deny

```
# noxfile.py
...
@nox.session(python="3.10")
def lint(session):
    files = ["src/", "noxfile.py"]
    session.install("-r", "requirements-pylint.txt"
    session.install("-e", ".")
    session.run(
        "pylint",
        "--disable=all",
        *(f"--enable={checker}" for checker in chec
        "src",
    )
```

PyLint

- Keep the good:

PyLint

- Keep the good: CI

PyLint

- Keep the good: CI Checkers

PyLint

- Keep the good: CI Checkers
- Lose the bad:

PyLint

- Keep the good: CI Checkers
- Lose the bad: Default deny

PyLint

- Keep the good: CI Checkers
- Lose the bad: Default deny
- Avoid the ugly:

PyLint

- Keep the good: CI Checkers
- Lose the bad: Default deny
- Avoid the ugly: Pin