

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

I live in Belmont, in the San Francisco Bay Area Peninsula. I wish to acknowledge it as the ancestral homeland of the Ramaytush Ohlone people.

0.1 The Good (3m)

0.1.1 Redefine function

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
```

0.1.2 Test works

Test works!

collected 2 items

```
test.py ..  
2 passed
```

0.1.3 Pylint finds it

PyLint: The Good

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

0.2 The Bad

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-methods)

```

0.3 The Ugly (3m)

PyLint: The Ugly

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

0.4 Summary (3m)

0.4.1 Pin

Pin PyLint

```

# pyproject.toml

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

```

Pin PyLint

```

# 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",
        )

```

0.5 Default deny

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 checkers)
        "src",
    )
```

0.5.1 A Few of My Favorite Things

A few of my favorite things

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

PyLint

- Keep the good: CI Checkers
- Lose the bad: (default deny)
- Avoid the ugly: pin