

CODIO ACTIVITY 2

Exercise 1

Run the code against a variety of linters to test the code quality:

- `pylint code_with_lint.py`
- `pyflakes code_with_lint.py`
- `pycodestyle code_with_lint.py`
- `pydocstyle code_with_lint.py`

Compare the effectiveness of each tool in defining and identifying code quality. What can you conclude about the effectiveness of each approach?

it is tricky to compare the effectiveness of all four linters as they serve different purpose (VanTol, n.d)

Linters	Effectiveness
pylint	checks for code errors and styling.
pyflakes	It focuses on logical code issues and errors, not styling.
pycodestyle	Used to check style conventions from PEP8 but It doesn't check naming conventions and docstrings
pydocstyle	similar to pycodestyle but it checks docstrings against PEP 257 convention rather than PEP8

Screenshots

Pylint

```
SyntaxError: invalid syntax
-----
Command "python setup.py egg_info" failed with error code 1 in /tmp/pip-build-uxkhQt/lazy-object-proxy/
```

There was an error on Codio, unable to run pylint

pyflakes

```
codio@menu-rachel:~/workspace$ pyflakes code_with_lint.py
code_with_lint.py:1:1 'io' imported but unused
code_with_lint.py:2:1 'from math import *' used; unable to detect undefined names
code_with_lint.py:13:5 local variable 'some_global_var' is assigned to but never used
code_with_lint.py:40:44 'pi' may be undefined, or defined from star imports: math
code_with_lint.py:40:9 local variable 'list_comprehension' is assigned to but never used
code_with_lint.py:41:16 local variable 'time' defined in enclosing scope on line 5 referenced before assignment
code_with_lint.py:41:9 local variable 'time' is assigned to but never used
code_with_lint.py:43:9 local variable 'date_and_time' is assigned to but never used
```

pycodestyle

```
codio@menu-rachel:~/workspace$ pycodestyle code_with_lint.py
code_with_lint.py:9:1: E302 expected 2 blank lines, found 1
code_with_lint.py:14:15: E225 missing whitespace around operator
code_with_lint.py:19:1: E302 expected 2 blank lines, found 1
code_with_lint.py:20:80: E501 line too long (80 > 79 characters)
code_with_lint.py:25:10: E711 comparison to None should be 'if cond is not None:'
code_with_lint.py:27:25: E703 statement ends with a semicolon
code_with_lint.py:31:24: E201 whitespace after '('
code_with_lint.py:35:1: E302 expected 2 blank lines, found 1
code_with_lint.py:37:58: E251 unexpected spaces around keyword / parameter equals
code_with_lint.py:37:60: E251 unexpected spaces around keyword / parameter equals
code_with_lint.py:38:28: E221 multiple spaces before operator
code_with_lint.py:38:31: E222 multiple spaces after operator
code_with_lint.py:39:22: E221 multiple spaces before operator
code_with_lint.py:39:31: E222 multiple spaces after operator
code_with_lint.py:40:80: E501 line too long (83 > 79 characters)
code_with_lint.py:44:15: W292 no newline at end of file
```

pydocstyle

```
codio@menu-rachel:~/workspace$ pydocstyle code_with_lint.py
code_with_lint.py:1 at module level:
    D100: Missing docstring in public module
code_with_lint.py:10 in public function `multiply`:
    D401: First line should be in imperative mood; try rephrasing (found 'This')
code_with_lint.py:10 in public function `multiply`:
    D400: First line should end with a period (not 's')
code_with_lint.py:10 in public function `multiply`:
    D200: One-line docstring should fit on one line with quotes (found 3)
code_with_lint.py:20 in public function `is_sum_lucky`:
    D401: First line should be in imperative mood; try rephrasing (found 'This')
code_with_lint.py:20 in public function `is_sum_lucky`:
    D400: First line should end with a period (not 'y')
code_with_lint.py:20 in public function `is_sum_lucky`:
    D205: 1 blank line required between summary line and description (found 0)
code_with_lint.py:35 in public class `SomeClass`:
    D101: Missing docstring in public class
code_with_lint.py:37 in public method `__init__`:
    D107: Missing docstring in __init__
codio@menu-rachel:~/workspace$
```

Reference

VanTol A. (n.d) Available from:<https://realpython.com/python-code-quality/>
[Accessed 25 January 2022]