Using Linters to Achieve Python Code Quality

Linter	Purpose
pylint	Performs comprehensive static analysis. Checks for errors, style issues, unused variables, naming conventions, complexity and coding standards. Provides a score.
pyflakes	Detects logical errors and unused imports/variables. Faster and lighter than pylint, but with fewer checks. Ignores style.
pycodestyle	Checks compliance with PEP 8 (Python style guide). Focuses only on formatting (indentation, spacing, line length, etc.).
pydocstyle	Checks compliance of docstrings with PEP 257. Focuses on presence, format and structure of docstrings.

pylint

Strengths:

- Covers logic, structure, naming, and formatting
- Finds unreachable code and shadowing
- Gives a quality score

Weaknesses:

- Too many warnings by default
- Slower than others

pyflakes

Strengths:

- Fast and low-overhead
- Finds real bugs like unused or undefined names
- Few false positives

Weaknesses:

- No checks for style or docs
- Misses structural issues

pycodestyle

Strengths:

- Enforces clean layout
- Flags bad indentation, spacing, and line length

Weaknesses:

- Style-only focus
- Ignores logic and bugs

pydocstyle

Strengths:

- Enforces docstring rules
- Checks structure and punctuation

Weaknesses:

- Does not assess clarity or accuracy
- No impact on code logic

Conclusions

No single linter ensures complete code quality. Each tool has a defined scope:

- pylint: broad coverage
- pyflakes: fast correctness check
- pycodestyle: visual and formatting consistency
- pydocstyle: documentation standards

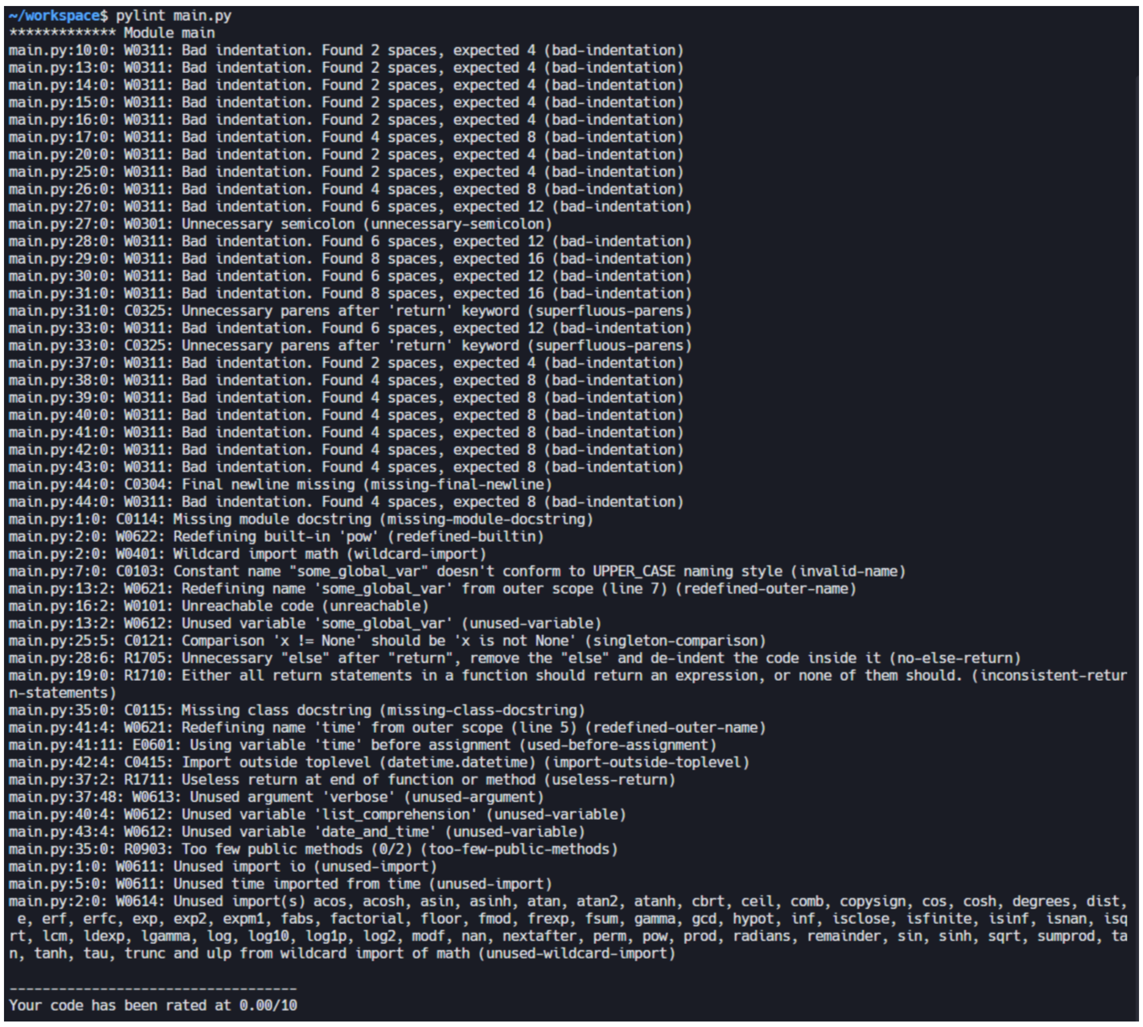
Best approach, use all four.

Test Code

Code from unit 10, Software Engineering Project Management

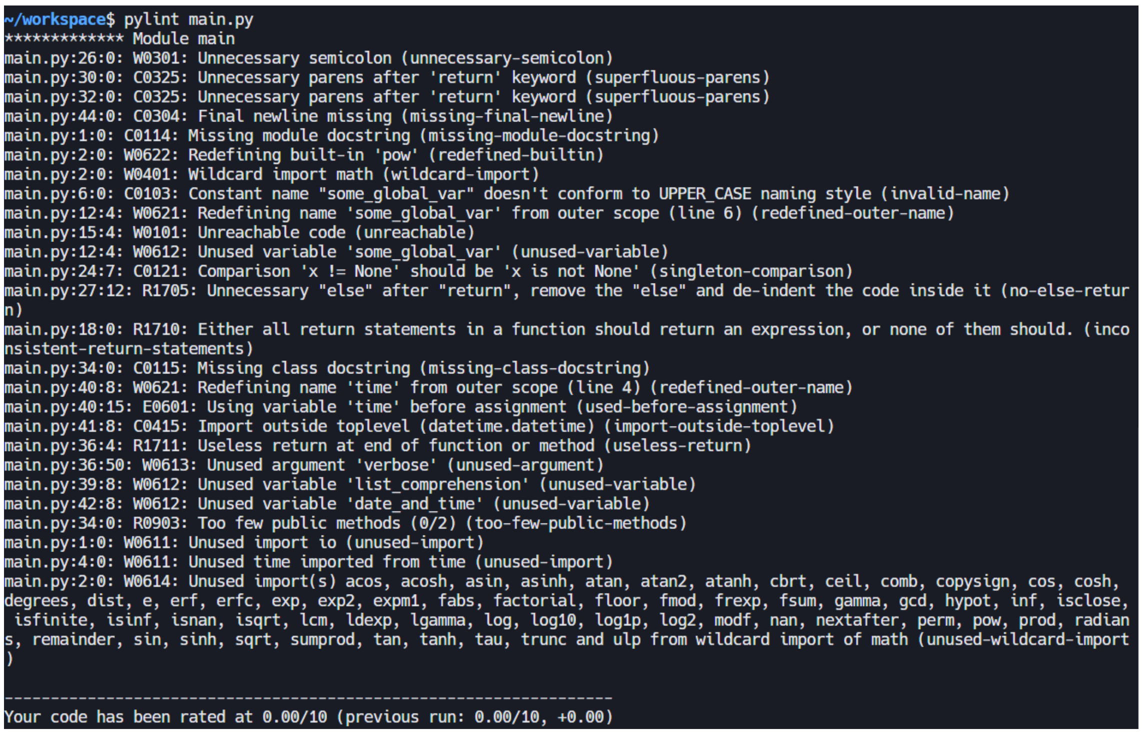
```
import io
from math import *


from time import time

some_global_var = 'GLOBAL VAR NAMES SHOULD BE IN ALL_CAPS_WITH_UNDERSCOES'

def multiply(x, y):
    """
    This returns the result of a multiplation of the inputs
    """
    some_global_var = 'this is actually a local variable...'
    result = x* y
    return result
    if result == 777:
        print("jackpot!")

def is_sum_lucky(x, y):
    """This returns a string describing whether or not the sum of input is lucky
    This function first makes sure the inputs are valid and then calculates the
    sum. Then, it will determine a message to return based on whether or not
    that sum should be considered "lucky"
    """
    if x != None:
        if y is not None:
            result = x+y;
            if result == 7:
                return 'a lucky number!'
            else:
                return( 'an unlucky number!')

            return ('just a normal number')

class SomeClass:

    def __init__(self, some_arg,  some_other_arg, verbose = False):
        self.some_other_arg  =  some_other_arg
        self.some_arg        =  some_arg
        list_comprehension = [((100/value)*pi) for value in some_arg if value != 0]
        time = time()
        from datetime import datetime
        date_and_time = datetime.now()
        return
```

pylint



```
~/workspace$ pylint main.py
************* Module main
main.py:10:0: W0311: Bad indentation. Found 2 spaces, expected 4 (bad-indentation)
main.py:13:0: W0311: Bad indentation. Found 2 spaces, expected 4 (bad-indentation)
main.py:14:0: W0311: Bad indentation. Found 2 spaces, expected 4 (bad-indentation)
main.py:15:0: W0311: Bad indentation. Found 2 spaces, expected 4 (bad-indentation)
main.py:16:0: W0311: Bad indentation. Found 2 spaces, expected 4 (bad-indentation)
main.py:17:0: W0311: Bad indentation. Found 4 spaces, expected 8 (bad-indentation)
main.py:20:0: W0311: Bad indentation. Found 2 spaces, expected 4 (bad-indentation)
main.py:25:0: W0311: Bad indentation. Found 2 spaces, expected 4 (bad-indentation)
main.py:26:0: W0311: Bad indentation. Found 4 spaces, expected 8 (bad-indentation)
main.py:27:0: W0311: Bad indentation. Found 6 spaces, expected 12 (bad-indentation)
main.py:27:0: W0301: Unnecessary semicolon (unnecessary-semicolon)
main.py:28:0: W0311: Bad indentation. Found 6 spaces, expected 12 (bad-indentation)
main.py:29:0: W0311: Bad indentation. Found 8 spaces, expected 16 (bad-indentation)
main.py:30:0: W0311: Bad indentation. Found 6 spaces, expected 12 (bad-indentation)
main.py:31:0: W0311: Bad indentation. Found 8 spaces, expected 16 (bad-indentation)
main.py:31:0: C0325: Unnecessary parens after 'return' keyword (superfluous-parens)
main.py:33:0: W0311: Bad indentation. Found 6 spaces, expected 12 (bad-indentation)
main.py:33:0: C0325: Unnecessary parens after 'return' keyword (superfluous-parens)
main.py:37:0: W0311: Bad indentation. Found 2 spaces, expected 4 (bad-indentation)
main.py:38:0: W0311: Bad indentation. Found 4 spaces, expected 8 (bad-indentation)
main.py:39:0: W0311: Bad indentation. Found 4 spaces, expected 8 (bad-indentation)
main.py:40:0: W0311: Bad indentation. Found 4 spaces, expected 8 (bad-indentation)
main.py:41:0: W0311: Bad indentation. Found 4 spaces, expected 8 (bad-indentation)
main.py:42:0: W0311: Bad indentation. Found 4 spaces, expected 8 (bad-indentation)
main.py:43:0: W0311: Bad indentation. Found 4 spaces, expected 8 (bad-indentation)
main.py:44:0: C0304: Final newline missing (missing-final-newline)
main.py:44:0: W0311: Bad indentation. Found 4 spaces, expected 8 (bad-indentation)
main.py:1:0: C0114: Missing module docstring (missing-module-docstring)
main.py:2:0: W0622: Redefining built-in 'pow' (redefined-builtin)
main.py:2:0: W0401: Wildcard import math (wildcard-import)
main.py:7:0: C0103: Constant name "some_global_var" doesn't conform to UPPER_CASE naming style (invalid-name)
main.py:13:2: W0621: Redefining name 'some_global_var' from outer scope (line 7) (redefined-outer-name)
main.py:16:2: W0101: Unreachable code (unreachable)
main.py:13:2: W0612: Unused variable 'some_global_var' (unused-variable)
main.py:25:5: C0121: Comparison 'x != None' should be 'x is not None' (singleton-comparison)
main.py:28:6: R1705: Unnecessary "else" after "return", remove the "else" and de-indent the code inside it (no-else-return)
main.py:19:0: R1710: Either all return statements in a function should return an expression, or none of them should. (inconsistent-retur
n-statements)
main.py:35:0: C0115: Missing class docstring (missing-class-docstring)
main.py:41:4: W0621: Redefining name 'time' from outer scope (line 5) (redefined-outer-name)
main.py:41:11: E0601: Using variable 'time' before assignment (used-before-assignment)
main.py:42:4: C0415: Import outside toplevel (datetime.datetime) (import-outside-toplevel)
main.py:37:2: R1711: Useless return at end of function or method (useless-return)
main.py:37:48: W0613: Unused argument 'verbose' (unused-argument)
main.py:40:4: W0612: Unused variable 'list_comprehension' (unused-variable)
main.py:43:4: W0612: Unused variable 'date_and_time' (unused-variable)
main.py:35:0: R0903: Too few public methods (0/2) (too-few-public-methods)
main.py:1:0: W0611: Unused import io (unused-import)
main.py:5:0: W0611: Unused time imported from time (unused-import)
main.py:2:0: W0614: Unused import(s) acos, acosh, asin, asinh, atan, atan2, atanh, cbrt, ceil, comb, copysign, cos, cosh, degrees, dist,
 e, erf, erfc, exp, exp2, expm1, fabs, factorial, floor, fmod, frexp, fsum, gamma, gcd, hypot, inf, isclose, isfinite, isinf, isnan, isq
rt, lcm, ldexp, lgamma, log, log10, log1p, log2, modf, nan, nextafter, perm, pow, prod, radians, remainder, sin, sinh, sqrt, sumprod, ta
n, tanh, tau, trunc and ulp from wildcard import of math (unused-wildcard-import)

-----------------------------------
Your code has been rated at 0.00/10
```

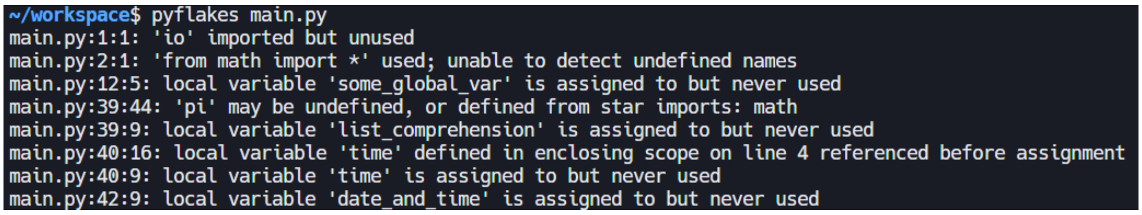
Due to the Replit IDE using tabs instead of spaces the first thing highlighted by pylint was the incorrect indentation.



```
~/workspace$ pylint main.py
************* Module main
main.py:26:0: W0301: Unnecessary semicolon (unnecessary-semicolon)
main.py:30:0: C0325: Unnecessary parens after 'return' keyword (superfluous-parens)
main.py:32:0: C0325: Unnecessary parens after 'return' keyword (superfluous-parens)
main.py:44:0: C0304: Final newline missing (missing-final-newline)
main.py:1:0: C0114: Missing module docstring (missing-module-docstring)
main.py:2:0: W0622: Redefining built-in 'pow' (redefined-builtin)
main.py:2:0: W0401: Wildcard import math (wildcard-import)
main.py:6:0: C0103: Constant name "some_global_var" doesn't conform to UPPER_CASE naming style (invalid-name)
main.py:12:4: W0621: Redefining name 'some_global_var' from outer scope (line 6) (redefined-outer-name)
main.py:15:4: W0101: Unreachable code (unreachable)
main.py:12:4: W0612: Unused variable 'some_global_var' (unused-variable)
main.py:24:7: C0121: Comparison 'x != None' should be 'x is not None' (singleton-comparison)
main.py:27:12: R1705: Unnecessary "else" after "return", remove the "else" and de-indent the code inside it (no-else-retur
n)
main.py:18:0: R1710: Either all return statements in a function should return an expression, or none of them should. (inco
nsistent-return-statements)
main.py:34:0: C0115: Missing class docstring (missing-class-docstring)
main.py:40:8: W0621: Redefining name 'time' from outer scope (line 4) (redefined-outer-name)
main.py:40:15: E0601: Using variable 'time' before assignment (used-before-assignment)
main.py:41:8: C0415: Import outside toplevel (datetime.datetime) (import-outside-toplevel)
main.py:36:4: R1711: Useless return at end of function or method (useless-return)
main.py:36:50: W0613: Unused argument 'verbose' (unused-argument)
main.py:39:8: W0612: Unused variable 'list_comprehension' (unused-variable)
main.py:42:8: W0612: Unused variable 'date_and_time' (unused-variable)
main.py:34:0: R0903: Too few public methods (0/2) (too-few-public-methods)
main.py:1:0: W0611: Unused import io (unused-import)
main.py:4:0: W0611: Unused time imported from time (unused-import)
main.py:2:0: W0614: Unused import(s) acos, acosh, asin, asinh, atan, atan2, atanh, cbrt, ceil, comb, copysign, cos, cosh,
degrees, dist, e, erf, erfc, exp, exp2, expm1, fabs, factorial, floor, fmod, frexp, fsum, gamma, gcd, hypot, inf, isclose,
 isfinite, isinf, isnan, isqrt, lcm, ldexp, lgamma, log, log10, log1p, log2, modf, nan, nextafter, perm, pow, prod, radian
s, remainder, sin, sinh, sqrt, sumprod, tan, tanh, tau, trunc and ulp from wildcard import of math (unused-wildcard-import
)

------------------------------------------------------------------
Your code has been rated at 0.00/10 (previous run: 0.00/10, +0.00)
```

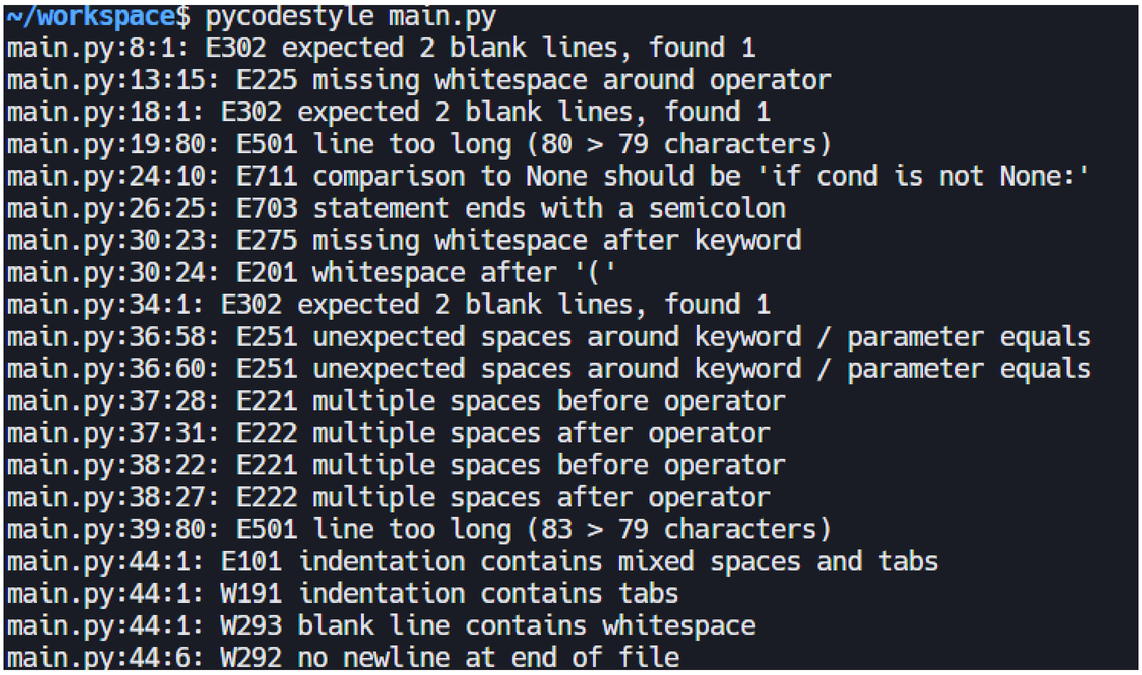
Fixed the indentation but still a lot of errors.

pyflakes



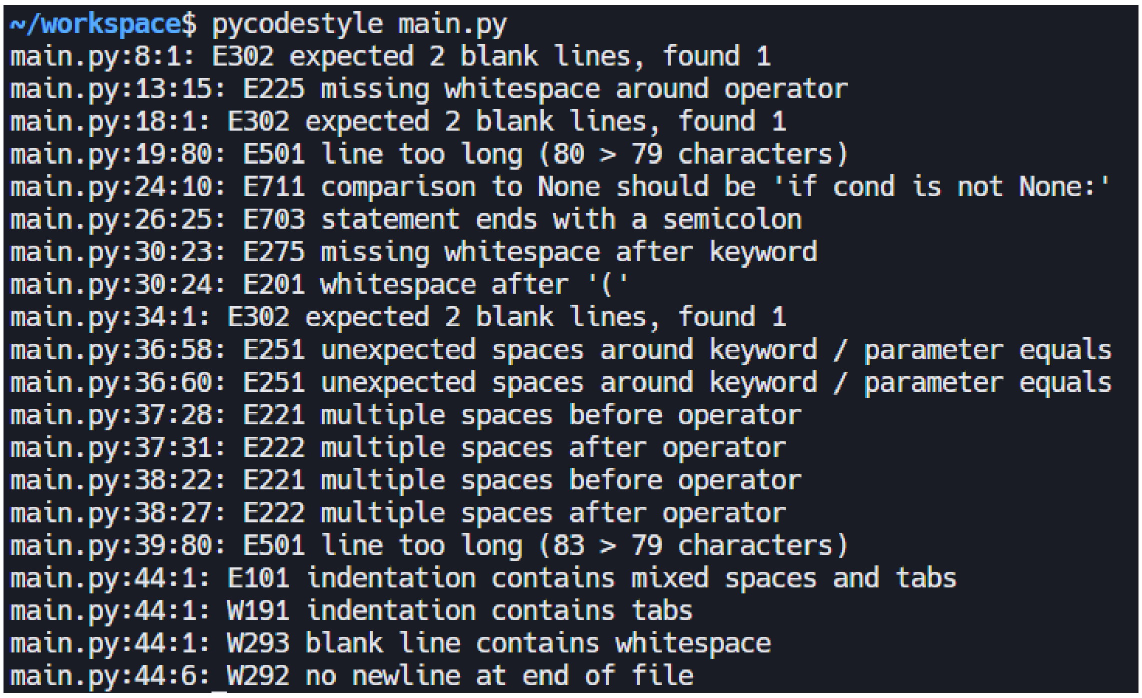
```
~/workspace$ pyflakes main.py
main.py:1:1: 'io' imported but unused
main.py:2:1: 'from math import *' used; unable to detect undefined names
main.py:12:5: local variable 'some_global_var' is assigned to but never used
main.py:39:44: 'pi' may be undefined, or defined from star imports: math
main.py:39:9: local variable 'list_comprehension' is assigned to but never used
main.py:40:16: local variable 'time' defined in enclosing scope on line 4 referenced before assignment
main.py:40:9: local variable 'time' is assigned to but never used
main.py:42:9: local variable 'date_and_time' is assigned to but never used
```

pycodestyle



```
~/workspace$ pycodestyle main.py
main.py:8:1: E302 expected 2 blank lines, found 1
main.py:13:15: E225 missing whitespace around operator
main.py:18:1: E302 expected 2 blank lines, found 1
main.py:19:80: E501 line too long (80 > 79 characters)
main.py:24:10: E711 comparison to None should be 'if cond is not None:'
main.py:26:25: E703 statement ends with a semicolon
main.py:30:23: E275 missing whitespace after keyword
main.py:30:24: E201 whitespace after '('
main.py:34:1: E302 expected 2 blank lines, found 1
main.py:36:58: E251 unexpected spaces around keyword / parameter equals
main.py:36:60: E251 unexpected spaces around keyword / parameter equals
main.py:37:28: E221 multiple spaces before operator
main.py:37:31: E222 multiple spaces after operator
main.py:38:22: E221 multiple spaces before operator
main.py:38:27: E222 multiple spaces after operator
main.py:39:80: E501 line too long (83 > 79 characters)
main.py:44:1: E101 indentation contains mixed spaces and tabs
main.py:44:1: W191 indentation contains tabs
main.py:44:1: W293 blank line contains whitespace
main.py:44:6: W292 no newline at end of file
```

pydocstyle



```
~/workspace$ pycodestyle main.py
main.py:8:1: E302 expected 2 blank lines, found 1
main.py:13:15: E225 missing whitespace around operator
main.py:18:1: E302 expected 2 blank lines, found 1
main.py:19:80: E501 line too long (80 > 79 characters)
main.py:24:10: E711 comparison to None should be 'if cond is not None:'
main.py:26:25: E703 statement ends with a semicolon
main.py:30:23: E275 missing whitespace after keyword
main.py:30:24: E201 whitespace after '('
main.py:34:1: E302 expected 2 blank lines, found 1
main.py:36:58: E251 unexpected spaces around keyword / parameter equals
main.py:36:60: E251 unexpected spaces around keyword / parameter equals
main.py:37:28: E221 multiple spaces before operator
main.py:37:31: E222 multiple spaces after operator
main.py:38:22: E221 multiple spaces before operator
main.py:38:27: E222 multiple spaces after operator
main.py:39:80: E501 line too long (83 > 79 characters)
main.py:44:1: E101 indentation contains mixed spaces and tabs
main.py:44:1: W191 indentation contains tabs
main.py:44:1: W293 blank line contains whitespace
main.py:44:6: W292 no newline at end of file
```