

PYTHON- PYLINT

Table of Contents

1	Introduction.....	3
1.1	Static Analysis:	3
1.2	Code Quality:	3
1.3	Error Detection.....	3
1.4	Style Consistency.....	3
1.5	Customization	3
1.6	Integration:	3
1.7	Reporting.....	4
1.8	Extensibility:	4
2	Python- Pylint Standard	4
2.1	Fatal.....	4
2.2	Error	5
2.3	Warning.....	9
2.4	Convention.....	15
2.5	Refactor.....	17
2.6	Information	19
3	Conclusion	20

1 Introduction

Python Pylint is a widely used open-source static analysis tool for the Python programming language. It is designed to analyze Python code, check for adherence to coding standards, and identify potential errors, code quality issues, and style violations. Pylint provides a detailed and customizable set of rules that help developers improve their code's quality, maintainability, and readability.

Purpose of Python Pylint include:

1.1 Static Analysis:

Pylint performs static code analysis, which means it examines the code without actually running it. This analysis helps identify issues without the need to execute the code.

1.2 Code Quality:

Pylint assesses code quality by enforcing coding standards and best practices, such as PEP 8, the Python Enhancement Proposal for code style. It checks for issues like indentation, naming conventions, and style consistency.

1.3 Error Detection

Pylint detects potential errors, such as undefined variables, unused imports, or exceptions that are never raised. It helps you catch and fix these issues before they become runtime errors.

1.4 Style Consistency

Pylint checks that your code adheres to a consistent and readable style, ensuring that the code is easily understandable by developers and maintainers.

1.5 Customization

Pylint allows you to customize its behavior by configuring rules and adjusting settings to match your coding standards and project-specific requirements.

1.6 Integration:

Pylint can be integrated into various development environments and continuous integration (CI) pipelines. It's commonly used with IDEs like Visual Studio Code, PyCharm, and others.

1.7 Reporting

Pylint provides detailed reports and feedback about issues found in the code, making it easier for developers to locate and address the problems.

1.8 Extensibility:

Pylint is extensible, allowing you to write custom rules and checks to enforce specific coding guidelines or requirements for your project.

2 Python- Pylint Standard

2.1 Fatal

All messages in the fatal category:

- astroid-error / F0002
- config-parse-error / F0011
- fatal / F0001
- method-check-failed / F0202
- parse-error / F0010

All renamed messages in the fatal category:

- old-import-error / F0401

2.2 Error

All messages in the error category:

- abstract-class-instantiated / E0110
- access-member-before-definition / E0203
- assigning-non-slot / E0237
- assignment-from-no-return / E1111
- assignment-from-none / E1128
- await-outside-async / E1142
- bad-configuration-section / E0014
- bad-except-order / E0701
- bad-exception-cause / E0705
- bad-format-character / E1300
- bad-plugin-value / E0013
- bad-reversed-sequence / E0111
- bad-str-strip-call / E1310
- bad-string-format-type / E1307
- bad-super-call / E1003
- bidirectional-unicode / E2502
- broken-collections-callable / E6005
- broken-noreturn / E6004
- catching-non-exception / E0712
- class-variable-slots-conflict / E0242
- continue-in-finally / E0116
- dict-iter-missing-items / E1141
- duplicate-argument-name / E0108
- duplicate-bases / E0241
- format-needs-mapping / E1303
- function-redefined / E0102
- import-error / E0401
- inconsistent-mro / E0240

- inherit-non-class / E0239
- init-is-generator / E0100
- invalid-all-format / E0605
- invalid-all-object / E0604
- invalid-bool-returned / E0304
- invalid-bytes-returned / E0308
- invalid-character-backspace / E2510
- invalid-character-carriage-return / E2511
- invalid-character-esc / E2513
- invalid-character-nul / E2514
- invalid-character-sub / E2512
- invalid-character-zero-width-space / E2515
- invalid-class-object / E0243
- invalid-enum-extension / E0244
- invalid-envvar-value / E1507
- invalid-field-call / E3701
- invalid-format-returned / E0311
- invalid-getnewargs-ex-returned / E0313
- invalid-getnewargs-returned / E0312
- invalid-hash-returned / E0309
- invalid-index-returned / E0305
- invalid-length-hint-returned / E0310
- invalid-length-returned / E0303
- invalid-metaclass / E1139
- invalid-repr-returned / E0306
- invalid-sequence-index / E1126
- invalid-slice-index / E1127
- invalid-slice-step / E1144
- invalid-slots / E0238
- invalid-slots-object / E0236
- invalid-star-assignment-target / E0113

- invalid-str-returned / E0307
- invalid-unary-operand-type / E1130
- invalid-unicode-codec / E2501
- logging-format-truncated / E1201
- logging-too-few-args / E1206
- logging-too-many-args / E1205
- logging-unsupported-format / E1200
- method-hidden / E0202
- misplaced-bare-raise / E0704
- misplaced-format-function / E0119
- missing-format-string-key / E1304
- missing-kwoa / E1125
- mixed-format-string / E1302
- modified-iterating-dict / E4702
- modified-iterating-set / E4703
- no-member / E1101
- no-method-argument / E0211
- no-name-in-module / E0611
- no-self-argument / E0213
- no-value-for-parameter / E1120
- non-iterator-returned / E0301
- nonexistent-operator / E0107
- nonlocal-and-global / E0115
- nonlocal-without-binding / E0117
- not-a-mapping / E1134
- not-an-iterable / E1133
- not-async-context-manager / E1701
- not-callable / E1102
- not-context-manager / E1129
- not-in-loop / E0103
- notimplemented-raised / E0711

- positional-only-arguments-expected / E3102
- potential-index-error / E0643
- raising-bad-type / E0702
- raising-non-exception / E0710
- redundant-keyword-arg / E1124
- relative-beyond-top-level / E0402
- repeated-keyword / E1132
- return-arg-in-generator / E0106
- return-in-init / E0101
- return-outside-function / E0104
- singledispatch-method / E1519
- singledispatchmethod-function / E1520
- star-needs-assignment-target / E0114
- syntax-error / E0001
- too-few-format-args / E1306
- too-many-format-args / E1305
- too-many-function-args / E1121
- too-many-star-expressions / E0112
- truncated-format-string / E1301
- undefined-all-variable / E0603
- undefined-variable / E0602
- unexpected-keyword-arg / E1123
- unexpected-special-method-signature / E0302
- unhashable-member / E1143
- unpacking-non-sequence / E0633
- unrecognized-inline-option / E0011
- unrecognized-option / E0015
- unsubscriptable-object / E1136
- unsupported-assignment-operation / E1137
- unsupported-binary-operation / E1131
- unsupported-delete-operation / E1138

- unsupported-membership-test / E1135
- used-before-assignment / E0601
- used-prior-global-declaration / E0118
- yield-inside-async-function / E1700
- yield-outside-function / E0105

All renamed messages in the error category:

- bad-context-manager / E0235
- bad-exception-context / E0703
- bad-option-value / E0012
- maybe-no-member / E1103
- old-non-iterator-returned-2 / E0234
- old-unbalanced-tuple-unpacking / E0632
- unhashable-dict-key / E1140

2.3 Warning

All messages in the warning category:

- abstract-method / W0223
- anomalous-backslash-in-string / W1401
- anomalous-unicode-escape-in-string / W1402
- arguments-differ / W0221
- arguments-out-of-order / W1114
- arguments-renamed / W0237
- assert-on-string-literal / W0129
- assert-on-tuple / W0199
- attribute-defined-outside-init / W0201
- bad-builtin / W0141
- bad-chained-comparison / W3601
- bad-dunder-name / W3201
- bad-format-string / W1302
- bad-format-string-key / W1300
- bad-indentation / W0311

- bad-open-mode / W1501
- bad-staticmethod-argument / W0211
- bad-thread-instantiation / W1506
- bare-except / W0702
- binary-op-exception / W0711
- boolean-datetime / W1502
- broad-exception-caught / W0718
- broad-exception-raised / W0719
- cell-var-from-loop / W0640
- comparison-with-callable / W0143
- confusing-with-statement / W0124
- consider-ternary-expression / W0160
- dangerous-default-value / W0102
- deprecated-argument / W4903
- deprecated-class / W4904
- deprecated-decorator / W4905
- deprecated-method / W4902
- deprecated-module / W4901
- deprecated-typing-alias / W6001
- differing-param-doc / W9017
- differing-type-doc / W9018
- duplicate-except / W0705
- duplicate-key / W0109
- duplicate-string-formatting-argument / W1308
- duplicate-value / W0130
- eq-without-hash / W1641
- eval-used / W0123
- exec-used / W0122
- expression-not-assigned / W0106
- f-string-without-interpolation / W1309
- fixme / W0511

- forgotten-debug-statement / W1515
- format-combined-specification / W1305
- format-string-without-interpolation / W1310
- global-at-module-level / W0604
- global-statement / W0603
- global-variable-not-assigned / W0602
- global-variable-undefined / W0601
- implicit-flag-alias / W0213
- implicit-str-concat / W1404
- import-self / W0406
- inconsistent-quotes / W1405
- invalid-envvar-default / W1508
- invalid-format-index / W1307
- invalid-overridden-method / W0236
- isinstance-second-argument-not-valid-type / W1116
- keyword-arg-before-vararg / W1113
- kwarg-superseded-by-positional-arg / W1117
- logging-format-interpolation / W1202
- logging-fstring-interpolation / W1203
- logging-not-lazy / W1201
- lost-exception / W0150
- method-cache-max-size-none / W1518
- misplaced-future / W0410
- missing-any-param-doc / W9021
- missing-format-argument-key / W1303
- missing-format-attribute / W1306
- missing-param-doc / W9015
- missing-parentheses-for-call-in-test / W0126
- missing-raises-doc / W9006
- missing-return-doc / W9011
- missing-return-type-doc / W9012

- missing-timeout / W3101
- missing-type-doc / W9016
- missing-yield-doc / W9013
- missing-yield-type-doc / W9014
- modified-iterating-list / W4701
- multiple-constructor-doc / W9005
- named-expr-without-context / W0131
- nan-comparison / W0177
- nested-min-max / W3301
- non-ascii-file-name / W2402
- non-parent-init-called / W0233
- non-str-assignment-to-dunder-name / W1115
- overlapping-except / W0714
- overridden-final-method / W0239
- pointless-exception-statement / W0133
- pointless-statement / W0104
- pointless-string-statement / W0105
- possibly-unused-variable / W0641
- preferred-module / W0407
- protected-access / W0212
- raise-missing-from / W0707
- raising-format-tuple / W0715
- redeclared-assigned-name / W0128
- redefined-builtin / W0622
- redefined-loop-name / W2901
- redefined-outer-name / W0621
- redefined-slots-in-subclass / W0244
- redundant-returns-doc / W9008
- redundant-u-string-prefix / W1406
- redundant-unittest-assert / W1503
- redundant-yields-doc / W9010

- `reimported` / W0404
- `return-in-finally` / W0134
- `self-assigning-variable` / W0127
- `self-cls-assignment` / W0642
- `shadowed-import` / W0416
- `shallow-copy-environ` / W1507
- `signature-differs` / W0222
- `subclassed-final-class` / W0240
- `subprocess-popen-preexec-fn` / W1509
- `subprocess-run-check` / W1510
- `super-init-not-called` / W0231
- `super-without-brackets` / W0245
- `too-many-try-statements` / W0717
- `try-except-raise` / W0706
- `unbalanced-dict-unpacking` / W0644
- `unbalanced-tuple-unpacking` / W0632
- `undefined-loop-variable` / W0631
- `unknown-option-value` / W0012
- `unnecessary-ellipsis` / W2301
- `unnecessary-lambda` / W0108
- `unnecessary-pass` / W0107
- `unnecessary-semicolon` / W0301
- `unreachable` / W0101
- `unspecified-encoding` / W1514
- `unused-argument` / W0613
- `unused-format-string-argument` / W1304
- `unused-format-string-key` / W1301
- `unused-import` / W0611
- `unused-private-member` / W0238
- `unused-variable` / W0612
- `unused-wildcard-import` / W0614

- useless-else-on-loop / W0120
- useless-param-doc / W9019
- useless-parent-delegation / W0246
- useless-type-doc / W9020
- useless-with-lock / W2101
- using-constant-test / W0125
- using-f-string-in-unsupported-version / W2601
- using-final-decorator-in-unsupported-version / W2602
- while-used / W0149
- wildcard-import / W0401
- wrong-exception-operation / W0716

All renamed messages in the warning category:

- broad-except / W0703
- cache-max-size-none / W1517
- implicit-str-concat-in-sequence / W1403
- lru-cache-decorating-method / W1516
- old-assignment-from-none / W1111
- old-deprecated-argument / W1511
- old-deprecated-class / W1512
- old-deprecated-decorator / W1513
- old-deprecated-method / W1505
- old-deprecated-module / W0402
- old-empty-docstring / W0132
- old-missing-param-doc / W9003
- old-missing-returns-doc / W9007
- old-missing-type-doc / W9004
- old-missing-yields-doc / W9009
- old-non-iterator-returned-1 / W0234
- old-unidiomatic-typecheck / W0154
- old-unpacking-non-sequence / W0633

- useless-super-delegation / W0235

2.4 Convention

All messages in the convention category:

- bad-classmethod-argument / C0202
- bad-docstring-quotes / C0198
- bad-file-encoding / C2503
- bad-mcs-classmethod-argument / C0204
- bad-mcs-method-argument / C0203
- consider-iterating-dictionary / C0201
- consider-using-any-or-all / C0501
- consider-using-dict-items / C0206
- consider-using-enumerate / C0200
- consider-using-f-string / C0209
- dict-init-mutate / C3401
- disallowed-name / C0104
- docstring-first-line-empty / C0199
- empty-docstring / C0112
- import-outside-toplevel / C0415
- import-private-name / C2701
- invalid-characters-in-docstring / C0403
- invalid-name / C0103
- line-too-long / C0301
- misplaced-comparison-constant / C2201
- missing-class-docstring / C0115
- missing-final-newline / C0304
- missing-function-docstring / C0116
- missing-module-docstring / C0114
- mixed-line-endings / C0327
- multiple-imports / C0410
- multiple-statements / C0321
- non-ascii-module-import / C2403

- non-ascii-name / C2401
- single-string-used-for-slots / C0205
- singleton-comparison / C0121
- superfluous-parens / C0325
- too-many-lines / C0302
- trailing-newlines / C0305
- trailing-whitespace / C0303
- typevar-double-variance / C0131
- typevar-name-incorrect-variance / C0105
- typevar-name-mismatch / C0132
- unexpected-line-ending-format / C0328
- ungrouped-imports / C0412
- unidiomatic-typecheck / C0123
- unnecessary-direct-lambda-call / C3002
- unnecessary-dunder-call / C2801
- unnecessary-lambda-assignment / C3001
- unnecessary-negation / C0117
- use-implicit-booleaness-not-comparison / C1803
- use-implicit-booleaness-not-comparison-to-string / C1804
- use-implicit-booleaness-not-comparison-to-zero / C1805
- use-implicit-booleaness-not-len / C1802
- use-maxsplit-arg / C0207
- use-sequence-for-iteration / C0208
- useless-import-alias / C0414
- wrong-import-order / C0411
- wrong-import-position / C0413
- wrong-spelling-in-comment / C0401
- wrong-spelling-in-docstring / C0402

All renamed messages in the convention category:

- blacklisted-name / C0102
- compare-to-empty-string / C1901

- compare-to-zero / C2001
- len-as-condition / C1801
- missing-docstring / C0111
- old-misplaced-comparison-constant / C0122
- old-non-ascii-name / C0144
- unneeded-not / C0113

2.5 Refactor

All messages in the refactor category:

- chained-comparison / R1716
- comparison-of-constants / R0133
- comparison-with-itself / R0124
- condition-evals-to-constant / R1727
- confusing-consecutive-elif / R5601
- consider-alternative-union-syntax / R6003
- consider-merging-isinstance / R1701
- consider-refactoring-into-while-condition / R3501
- consider-swap-variables / R1712
- consider-using-alias / R6002
- consider-using-assignment-expr / R6103
- consider-using-augmented-assign / R6104
- consider-using-dict-comprehension / R1717
- consider-using-from-import / R0402
- consider-using-generator / R1728
- consider-using-get / R1715
- consider-using-in / R1714
- consider-using-join / R1713
- consider-using-max-builtin / R1731
- consider-using-min-builtin / R1730
- consider-using-namedtuple-or-dataclass / R6101
- consider-using-set-comprehension / R1718
- consider-using-sys-exit / R1722

- consider-using-ternary / R1706
- consider-using-tuple / R6102
- consider-using-with / R1732
- cyclic-import / R0401
- duplicate-code / R0801
- else-if-used / R5501
- empty-comment / R2044
- inconsistent-return-statements / R1710
- literal-comparison / R0123
- magic-value-comparison / R2004
- no-classmethod-decorator / R0202
- no-else-break / R1723
- no-else-continue / R1724
- no-else-raise / R1720
- no-else-return / R1705
- no-self-use / R6301
- no-staticmethod-decorator / R0203
- prefer-typing-namedtuple / R6105
- property-with-parameters / R0206
- redefined-argument-from-local / R1704
- redefined-variable-type / R0204
- redundant-typehint-argument / R6006
- simplifiable-condition / R1726
- simplifiable-if-expression / R1719
- simplifiable-if-statement / R1703
- simplify-boolean-expression / R1709
- stop-iteration-return / R1708
- super-with-arguments / R1725
- too-complex / R1260
- too-few-public-methods / R0903
- too-many-ancestors / R0901

- too-many-arguments / R0913
- too-many-boolean-expressions / R0916
- too-many-branches / R0912
- too-many-instance-attributes / R0902
- too-many-locals / R0914
- too-many-nested-blocks / R1702
- too-many-public-methods / R0904
- too-many-return-statements / R0911
- too-many-statements / R0915
- trailing-comma-tuple / R1707
- unnecessary-comprehension / R1721
- unnecessary-dict-index-lookup / R1733
- unnecessary-list-index-lookup / R1736
- use-a-generator / R1729
- use-dict-literal / R1735
- use-list-literal / R1734
- use-set-for-membership / R6201
- useless-object-inheritance / R0205
- useless-option-value / R0022
- useless-return / R1711

All renamed messages in the refactor category:

- old-no-self-use / R0201
- old-simplifiable-if-statement / R0102
- old-too-many-nested-blocks / R0101

2.6 Information

All messages in the information category:

- bad-inline-option / I0010
- c-extension-no-member / I1101
- deprecated-pragma / I0022
- file-ignored / I0013
- locally-disabled / I0011

- raw-checker-failed / I0001
- suppressed-message / I0020
- use-symbolic-message-instead / I0023
- useless-suppression / I0021

All renamed messages in the information category:

- deprecated-disable-all / I0014

3 Conclusion

Overall, Pylint is a valuable tool for Python developers and development teams looking to enhance the quality, maintainability, and correctness of their Python code. By identifying and addressing issues early in the development process, Pylint helps reduce bugs, improve code readability, and maintain coding standards, contributing to more robust and maintainable software.