

CHEAT-SHEET

➔ ** = pylint

Commands:

Options which are actually commands. Options in this group are mutually exclusive.

- **--rcfile RCFILE**

Specify a configuration file to load.

- **--output OUTPUT**

Specify an output file.

- **--help-msg HELP_MSG [HELP_MSG ...]**

Display a help message for the given message id and exit. The value may be a comma-separated list of message ids.

- **--list-msgs**

Display a list of all pylint's messages divided by whether they are emittable with the given interpreter.

- **--list-msgs-enabled**

Display a list of what messages are enabled, disabled, and non-emittable with the given configuration.

- **--list-groups**

List pylint's message groups.

- **--list-conf-levels**

Generate pylint's confidence levels.

- **--list-extensions**

List available extensions.

- **--full-documentation**

Generate pylint's full documentation.

- **--generate-rcfile**

Generate a sample configuration file according to the current configuration. You can put other options before this one to get them in the generated configuration.

- **--generate-toml-config**

Generate a sample configuration file according to the current configuration. You can put other options before this one to get them in the generated configuration. The config is in the .toml format.

- **--long-help**

Show more verbose help.

Main:

- **--init-hook INIT_HOOK**

Python code to execute, usually for sys.path manipulation such as pygtk.require().

- **--errors-only, -E**

In error mode, messages with a category besides ERROR or FATAL are suppressed, and no reports are done by default. Error mode is compatible with disabling specific errors.

- **--verbose, -v**

In verbose mode, extra non-checker-related info will be displayed.

- **--enable-all-extensions**

Load and enable all available extensions. Use --list-extensions to see a list of all available extensions.

- **--ignore** <file>[,<file>...]**

Files or directories to be skipped. They should be base names, not paths. (default: ('CVS',))

- **--ignore-patterns** <pattern>[,<pattern>...]**

Files or directories matching the regular expression patterns are skipped. The regex matches against base names, not paths. The default value ignores Emacs file locks (default: (re.compile('^\\.#'),))

- **--ignore-paths** <pattern>[,<pattern>...]**

Add files or directories matching the regular expressions patterns to the ignore list. The regex matches against paths and can be in Posix or Windows format. Because '\\' represents the directory delimiter on Windows systems, it can't be used as an escape character. (default: [])

- **--persistent** <y or n>**

Pickle collected data for later comparisons. (default: True)

- **--load-plugins** <modules>**

List of plugins (as comma separated values of python module names) to load, usually to register additional checkers. (default: ())

- **--fail-under** <score>**

Specify a score threshold under which the program will exit with error. (default: 10)

- **--fail-on** <msg ids>**

Return non-zero exit code if any of these messages/categories are detected, even if the score is above the --fail-under value. Syntax same as enable. Messages specified are enabled, while categories only check already-enabled messages. (default:)

- **--jobs** <n-processes>, -j <n-processes>**

Use multiple processes to speed up Pylint. Specifying 0 will auto-detect the number of processors available to use and will cap the count on Windows to avoid hangs. (default: 1)

- **--limit-inference-results <number-of-results>**

Control the amount of potential inferred values when inferring a single object. This can help the performance when dealing with large functions or complex, nested conditions. (default: 100)

- **--extension-pkg-allow-list <pkg[,pkg]>**

A comma-separated list of package or module names from where C extensions may be loaded. Extensions are loading into the active Python interpreter and may run arbitrary code. (default: [])

- **--extension-pkg-whitelist <pkg[,pkg]>**

A comma-separated list of package or module names from where C extensions may be loaded. Extensions are loading into the active Python interpreter and may run arbitrary code. (This is an alternative name to extension-pkg-allow-list for backward compatibility.) (default: [])

- **--suggestion-mode <y or n>**

When enabled, pylint would attempt to guess common misconfiguration and emit user-friendly hints instead of false-positive error messages. (default: True)

- **--exit-zero**

Always return a 0 (non-error) status code, even if lint errors are found. This is primarily useful in continuous integration scripts. (default: False)

- **--from-stdin**

Interpret the stdin as a python script, whose filename needs to be passed as the module_or_package argument. (default: False)

- **--source-roots <path>[,<path>...]**

Add paths to the list of the source roots. Supports globbing patterns. The source root is an absolute path or a path relative to the current working directory used to determine a package namespace for modules located under the source root. (default: ())

- **--recursive <yn>**

Discover python modules and packages in the file system subtree. (default: False)

- **--py-version <py_version>**

Minimum Python version to use for version dependent checks. Will default to the version used to run pylint. (default: (3, 12))

- **--ignored-modules <module names>**

List of module names for which member attributes should not be checked (useful for modules/projects where namespaces are manipulated during runtime and thus existing member attributes cannot be deduced by static analysis). It supports qualified module names, as well as Unix pattern matching. (default: ())

- **--analyse-fallback-blocks <y or n>**

Analyse import fallback blocks. This can be used to support both Python 2 and 3 compatible code, which means that the block might have code that exists only in one or another interpreter, leading to false positives when analysed. (default: False)

- **--clear-cache-post-run <y or n>**

Clear in-memory caches upon conclusion of linting. Useful if running pylint in a server-like mode. (default: False)

Reports:

Options related to output formatting and reporting.

- **--output-format <format>, -f <format>**

Set the output format. Available formats are: text, parseable, colored, json2 (improved json format), json (old json format) and msvs (visual studio). You can also give a reporter class, e.g. mypackage.mymodule.MyReporterClass.

- **--reports** <y or n>, -r <y or n>

Tells whether to display a full report or only the messages. (default: False)

- **--evaluation** <python_expression>

Python expression which should return a score less than or equal to 10. You have access to the variables 'fatal', 'error', 'warning', 'refactor', 'convention', and 'info' which contain the number of messages in each category, as well as 'statement' which is the total number of statements analyzed. This score is used by the global evaluation report (RP0004). (default: $\max(0, 10.0 - ((5 * \text{error} + \text{warning} + \text{refactor} + \text{convention}) / \text{statement}))$)

- **--score** <y or n>, -s <y or n>

Activate the evaluation score. (default: True)

- **--msg-template** <template>

Template used to display messages. This is a python new-style format string used to format the message information. See doc for all details. (default:)

Messages Control:

Options controlling analysis messages.

- **--confidence** <levels>

Only show warnings with the listed confidence levels. Leave empty to show all. Valid levels: HIGH, CONTROL_FLOW, INFERENCE, INFERENCE_FAILURE, UNDEFINED. (default: ['HIGH', 'CONTROL_FLOW', 'INFERENCE', 'INFERENCE_FAILURE', 'UNDEFINED'])

- **--enable** <msg ids>, -e <msg ids>

Enable the message, report, category, or checker with the given id(s). You can either give multiple identifiers separated by comma (,) or put this option multiple times (only on the command line, not in the configuration file where it should appear only once). See also the "--disable" option for examples.

- **--disable <msg ids>, -d <msg ids>**

Disable the message, report, category, or checker with the given id(s). You can either give multiple identifiers separated by comma (,) or put this option multiple times (only on the command line, not in the configuration file where it should appear only once). You can also use "--disable=all" to disable everything first and then re-enable specific checks. For example, if you want to run only the similarities checker,