Pylint Python linter

CONCEPTS

> Every finding has an id (C0116) and a symbol (missing-function-docstring); either works in enable, disable and pragmas.

Config pyproject / pylintrc
↓ load
Targets files, packages
↓ parse
Checkers enabled messages
↓ report
Output text, json2, score + exit code

> The first letter of an id is its category: C convention, R refactor, W warning, E error, F fatal. I is informational.

RUNNING

`pylint <file.py>`
Lint a single file

`pylint <package>`
Lint a package or module by name

`pylint --recursive=y <dir>`
Lint every .py file under a directory

`pylint -j 0 <path>`
Use all CPU cores

CHOOSING MESSAGES

`pylint -E <path>`
Report errors only (--errors-only)

`pylint --disable=<msg>[,<msg>]`
Turn messages off by id or symbol

`pylint --enable=<msg>`
Turn a message back on

`pylint --disable=all --enable=<msg>`
Run only the listed messages

OUTPUT AND SCORE

`pylint --fail-under=<score> <path>`
Exit non-zero if the score is below it

HELP AND DISCOVERY

`pylint --help-msg=<msg>`
Explain a message by id or symbol

CONFIGURATION

`pylint --rcfile=<file> <path>`
Use a specific config file

`pylint --generate-toml-config`
Print a documented pyproject-style config

`pylint --generate-rcfile`
Print a documented .pylintrc

`[tool.pylint."messages control"]`
disable / enable lists in pyproject.toml

INLINE CONTROL

`# pylint: disable=<msg>`
End of a line: silence it for that line

`# pylint: disable=<msg>`
On its own line: silence to end of the block

MESSAGES YOU WILL MEET

`C0114 / C0115 / C0116`
Missing module / class / function docstring

`C0103 invalid-name`
Name breaks the naming style

`C0301 line-too-long`
Line exceeds max-line-length

`W0611 unused-import`
Imported but never used

`W0612 / W0613`
Unused variable / unused argument

`E0401 import-error`
Module cannot be imported (wrong env?)

`E1101 no-member`
Attribute not found on inferred type

`E0602 undefined-variable`
Name is not defined