Flake8 Python linter

START HERE

`flake8`
Lint the current directory

`flake8 --extend-ignore=<code>`
Skip a rule

`flake8 <file.py>`
Lint one file

`flake8 --max-line-length=100`
Allow longer lines

`flake8 --show-source`
Show the offending line

`flake8 --statistics -qq`
Count by code

`flake8 --select=E,F`
Only errors and pyflakes

`flake8 --version`
Version and plugins

Python files found by path → `parse` → **Plugins** E W F C90 + extras → `filter` → **select / ignore** config + CLI + noqa → `report` → **Output** exit 1 if any

CONCEPTS

Flake8 wraps pycodestyle (E, W), pyflakes (F) and mccabe (C90). It only reports; it never rewrites your code.

Codes are prefixes: E501 is one rule, E5 a group, E all style errors. Options take comma-separated prefixes.

Config is read from setup.cfg, tox.ini or .flake8 in the current directory; pyproject.toml is not supported without a plugin.

RUN

`flake8`
Check all Python files here

`flake8 <path> ...`
Check files or directories

`flake8 -`
Read source from stdin

`flake8 --stdin-display-name <name> -`
Name to show for stdin input

`flake8 --version`
Show versions of flake8 and plugins

`flake8 --count`
Print total number of violations

`flake8 -q`
Print only file names with errors

`flake8 -qq`
Print nothing, just set the exit code

`flake8 --exit-zero`
Always exit 0 despite violations

`flake8 -j <n>`
Number of parallel jobs

`flake8 --color never`
Disable coloured output

OUTPUT

`flake8 --show-source`
Print the offending source line

`flake8 --statistics`
Count violations per code

`flake8 --format=pylint`
Pylint-style output

OUTPUT (cont.)

`flake8 --format='%(path)s:%(row)d %(code)s'`
Custom format string
Keys: path, row, col, code, text.

`flake8 --output-file <file>`
Write report to a file

`flake8 --tee --output-file <file>`
Write to file and stdout

`flake8 --benchmark`
Print timing information

`flake8 --bug-report`
Dump versions and config as JSON

SELECT AND IGNORE

`flake8 --select=<codes>`
Report only these codes or prefixes
Replaces the default set; plugin codes you leave out are silenced.

`flake8 --extend-select=<codes>`
Add codes to the default selection

`flake8 --ignore=<codes>`
Replace the default ignore list
Dropping the defaults (E121,E123,E126,E226,E24,E704,W503,W504) turns those checks on.

`flake8 --extend-ignore=<codes>`
Ignore more codes, keep the defaults

`flake8 --per-file-ignores='<glob>:<codes>'`
Ignore codes for matching files

`flake8 --disable-noqa`
Report even lines with # noqa

`flake8 --max-complexity=<n>`
Enable C901 above this complexity
Off by default; 10 is a common starting point.

`flake8 --max-line-length=<n>`
Line length limit for E501

`flake8 --max-doc-length=<n>`
Limit comments and docstrings (W505)

`flake8 --indent-size=<n>`
Indentation width, default 4

SELECT AND IGNORE (cont.)

`flake8 --hang-closing`
Expect hanging closing brackets

`flake8 --doctests`
Also check syntax in doctests

`flake8 --builtins=<names>`
Extra names F821 should accept

FILES

`flake8 --exclude=<patterns>`
Replace the default exclude list

`flake8 --extend-exclude=<patterns>`
Skip more paths, keep defaults

`flake8 --filename=<patterns>`
Which file names to check (default *.py)

`git diff --name-only --'*.py' | xargs flake8`
Lint only changed files

INLINE CONTROL

`# noqa`
Silence all codes on this line

`# noqa: <code>[, <code>]`
Silence only these codes on this line
Always name the code; a bare noqa hides real problems.

`# flake8: noqa`
Skip the whole file (own line)
With codes, as '# flake8: noqa: E123', it silences those codes file-wide.

CONFIG

`flake8 --config <file>`
Use only this config file

`flake8 --append-config <file>`
Read an extra config file last

`flake8 --isolated`
Ignore all config files

`flake8 --require-plugins=<names>`
Fail if a plugin is missing

`flake8 --enable-extensions=<codes>`
Turn on off-by-default plugins

CONFIG (cont.)

`flake8 -v`
Show which config and plugins load

A [flake8] section in setup.cfg, tox.ini or .flake8; option names use dashes or underscores, e.g. max-line-length.

COMMON CODES

Frequent codes

Code	Meaning
`E501`	line too long
`E302`	expected 2 blank lines
`E231`	missing whitespace after ','
`E203`	whitespace before ':'
`W291`	trailing whitespace
`W605`	invalid escape sequence
`F401`	imported but unused
`F841`	local variable assigned, never used
`F821`	undefined name
`E722`	bare except
`E741`	ambiguous variable name
`C901`	function too complex

Code prefixes

Prefix	Source
`E1-E7`	pycodestyle errors
`W`	pycodestyle warnings
`F`	pyflakes
`C90`	mccabe

RECIPES

setup.cfg / .flake8

```
[flake8]
max-line-length = 100
extend-ignore = E203
per-file-ignores =
    __init__.py: F401
    tests/*: S101
exclude = .git,build,.venv
```

With Black

```
# Black-compatible settings
[flake8]
max-line-length = 88
extend-ignore = E203,E701
```

CI: hard errors then advice

```
flake8 --count --
select=E9,F63,F7,F82 --show-
source --statistics
flake8 --count --exit-zero --
max-complexity=10 --statistics
```

Pre-commit hook

```
# .pre-commit-config.yaml
repos:
  - repo: https://github.com/
PyCQA/flake8
    rev: 7.4.1
    hooks:
      - id: flake8
```