

Flake8

Christian Külker

2023-04-28

Contents

1	Installation	2
2	Versions	2
3	Sources	2
4	Usage	3
4.1	Single File Usage	3
4.2	Multiple File Usage	3
4.3	Output	3
5	Useful Command Line Options	3
6	Ignoring Errors (Violations)	4
6.1	Ignoring Files Or Directory	4
6.2	Ignoring Error (Violation) Types	4
6.3	Ignoring Lines	4
6.4	Flake8 Configuration	4
7	Discussion	5
8	Alternatives	5
9	History	5
10	Disclaimer of Warranty	5
11	Limitation of Liability	5

Flake8 is a popular Python linting tool that combines the functionality of several linting tools ([PyFlakes](#), [pycodestyle](#)). It helps identify errors, PEP 8 style guide violations, and

code complexity issues, making Python code more readable, maintainable, and efficient. This document will guide you through the installation and use of Flake8 with Python on a Debian-based Linux distribution.

With **flake8** it is possible to enforce a style guide on hand written Python code. By default, **flake8** includes checks provided by **PyFlakes**, PEP-0008 inspired style checks provided by **PyCodeStyle**, and **McCabe** complexity checking.

1 Installation

```
aptitude install flake8
```

This assumes that you are using the system version of Python. “It is very important to install Flake8 on the correct version of Python for your needs. If you want Flake8 to properly parse new language features in Python 3.5 (for example), you need it to be installed on 3.5 for Flake8 to understand those features. In many ways, Flake8 is tied to the version of Python on which it runs.” (Note from <https://flake8.pycqa.org/en/latest/>)

2 Versions

Debian Stretch

```
flake8 --version
3.2.1 (mccabe: 0.5.3, pyflakes: 1.3.0, pycodestyle: 2.2.0) CPython 3.5.3
↳ on Linux
```

Debian Buster

```
flake8 --version
3.6.0 (mccabe: 0.6.1, pycodestyle: 2.4.0, pyflakes: 2.0.0) CPython 3.7.3
↳ on Linux
```

Debian Bullseye

```
flake8 --version
3.8.4 (mccabe: 0.6.1, pycodestyle: 2.6.0, pyflakes: 2.2.0) CPython 3.9.2
↳ on Linux
```

3 Sources

- [Flake8](#)

- [Pyflakes](#)
- [PyCodeStyle](#)

4 Usage

There are basically three ways to use and influence flake8:

1. Command line (file or directory)
2. Code Comments
3. Configuration file

The command line tool `flake8` can be used on a single file or on multiple files, while comments control flake8 for special cases and the configuration file can control flake8 for one or all files aka the project.

4.1 Single File Usage

```
python3 flake8 /PATH/TO/file.py
```

4.2 Multiple File Usage

```
python3 flake8 /PATH/TO/FILES/*.py
```

4.3 Output

The `flake8` command prints a list of problems found. While most of them are style-related, it also prints if it found a variable that was assigned but not used. (Perl does this automatically via the `-w` option).

5 Useful Command Line Options

While `flake8` can be used without options, it can be customized via a configuration file or various options. This section shows only a small subset of useful command line options.

- `-q --quite` : Just print the name of file in which issues are found
- `--config=CONFIG` : specify the configuration file (for custom options)
- `--max-line-length=n` : set different line length (default 79)

6 Ignoring Errors (Violations)

6.1 Ignoring Files Or Directory

To ignore whole files or directories

```
flake8 --exclude path/to/file
```

6.2 Ignoring Error (Violation) Types

To ignore certain violation types, it is possible to specify the ID on the command line. (It is also possible to specify it in a configuration file)

New exclude list:

```
flake8 --ignore=E1,E23,W503 path/to/files/
```

Extending the exclude list:

```
flake8 --extend-ignore=E1,E23 path/to/files/
```

6.3 Ignoring Lines

If the line is short enough, it is possible to ignore a line with a special comment `# noqa`.

```
example = lambda: 'example' # noqa: E731
```

6.4 Flake8 Configuration

Create a `.flake8` file in the project directory.

```
touch .flake8
vim .flake8
[flake8]
ignore = E226,E302
max-line-length = 80
exclude = migrations,venv,tests
```

Here we're ignoring [E226](#) (missing whitespace around arithmetic operators) and [E302](#) (expected two blank lines, found 0). We've also set the maximum line length to 80 characters and excluded directories named "migrations", "venv" and "tests" from scanning. The call to `flake8` is as usual.

7 Discussion

Having **flake8** in Python is nice. Compared to other solutions, such as **perltidy**, it seems that this tool has a long way to go before it is as easy to use, useful, and time-saving. A short example: When using an editor (like vim), it is possible to define a hotkey to activate **perltidy** to format the displayed code. So as a programmer, you do not have to write all the indentation and other fancy spacing or alignment. Just press the key and the code looks nice. With **flake8**, the programmer just gets a report and has to fix everything manually. **perltidy** can batch convert files, check files embedded in other markup (such as HTML), and much more. Since Python, unlike Perl, has the “there is only one way” syntax philosophy, it is of course no wonder that **flake8** does not offer much in the way of syntax customization, but a real advantage of **flake8** is that it prints IDs referencing the rules, which can be used to look up and find more information about specific rules.

8 Alternatives

- [Pylint](#)

9 History

Version	Date	Notes
0.1.2	2023-04-28	Improve writing, add configuration file .flake8
0.1.1	2022-07-06	History, shell->bash, Debian 11 Bullseye
0.1.0	2020-05-08	Initial release

10 Disclaimer of Warranty

THERE IS NO WARRANTY FOR THIS INFORMATION, DOCUMENTS AND PROGRAMS, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE INFORMATION, DOCUMENT OR THE PROGRAM “AS IS” WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE INFORMATION, DOCUMENTS AND PROGRAMS IS WITH YOU. SHOULD THE INFORMATION, DOCUMENTS OR PROGRAMS PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

11 Limitation of Liability

IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MODIFIES AND/OR CONVEYS THE INFORMATION, DOCUMENTS OR PROGRAMS AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE INFORMATION, DOCUMENTS OR PROGRAMS (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED

INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE INFORMATION, DOCUMENTS OR PROGRAMS TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.