
PyCQA Meta Documentation

Release 1.0

PyCQA Developers

Mar 09, 2020

Contents

| | |
|--|-----------|
| 1 Contributor Covenant Code of Conduct | 3 |
| 1.1 Our Pledge | 3 |
| 1.2 Our Standards | 3 |
| 1.3 Enforcement Responsibilities | 4 |
| 1.4 Scope | 4 |
| 1.5 Enforcement | 4 |
| 1.6 Enforcement Guidelines | 4 |
| 1.7 Attribution | 5 |
| 2 An Introduction to the PyCQA | 7 |
| 2.1 What is the PyCQA? | 7 |
| 2.2 Who is part of the PyCQA? | 7 |
| 2.3 Where does the PyCQA live? | 9 |
| 2.4 How can I add my project to the PyCQA? | 9 |
| 3 The History of the PyCQA | 11 |
| 3.1 Creation of the PyCQA | 11 |
| 3.2 pep8 joins the organization on GitHub | 11 |
| 3.3 Pylint and Astroid move to GitHub and the PyCQA | 11 |
| 3.4 mccabe and pep8-naming move to the PyCQA organization | 12 |
| 3.5 pydocstyle (formerly pep257) moves to the PyCQA organization | 12 |
| 3.6 Baron and RedBaron join the PyCQA | 12 |
| 3.7 Bandit moves to the PyCQA organization | 12 |
| 4 The Management of the PyCQA | 13 |
| 4.1 Project Management | 14 |
| 4.2 Team Structure | 14 |
| 5 Join the Community! | 15 |
| 5.1 Using IRC | 15 |
| 6 Indices and tables | 17 |

So you're wondering what this "Pie Cee Que Aye" is and you've come here for answers? You're in luck!

This website is here to provide you with:

- a little bit of the history of the PyCQA
- an idea of who is part of the PyCQA
- how to become a part of the PyCQA
- and other fun things (okay maybe they're not *actually* fun)

Below you'll find the map to this content and anything else we add as we go along.

Cheers! Ian

Contributor Covenant Code of Conduct

1.1 Our Pledge

We as members, contributors, and leaders pledge to make participation in our community a harassment-free experience for everyone, regardless of age, body size, visible or invisible disability, ethnicity, sex characteristics, gender identity and expression, level of experience, education, socio-economic status, nationality, personal appearance, race, religion, or sexual identity and orientation.

We pledge to act and interact in ways that contribute to an open, welcoming, diverse, inclusive, and healthy community.

1.2 Our Standards

Examples of behavior that contributes to a positive environment for our community include:

- Demonstrating empathy and kindness toward other people
- Being respectful of differing opinions, viewpoints, and experiences
- Giving and gracefully accepting constructive feedback
- Accepting responsibility and apologizing to those affected by our mistakes, and learning from the experience
- Focusing on what is best not just for us as individuals, but for the overall community

Examples of unacceptable behavior include:

- The use of sexualized language or imagery, and sexual attention or advances of any kind
- Trolling, insulting or derogatory comments, and personal or political attacks
- Public or private harassment
- Publishing others' private information, such as a physical or email address, without their explicit permission
- Other conduct which could reasonably be considered inappropriate in a professional setting

1.3 Enforcement Responsibilities

Community leaders are responsible for clarifying and enforcing our standards of acceptable behavior and will take appropriate and fair corrective action in response to any behavior that they deem inappropriate, threatening, offensive, or harmful.

Community leaders have the right and responsibility to remove, edit, or reject comments, commits, code, wiki edits, issues, and other contributions that are not aligned to this Code of Conduct, and will communicate reasons for moderation decisions when appropriate.

1.4 Scope

This Code of Conduct applies within all community spaces, and also applies when an individual is officially representing the community in public spaces. Examples of representing our community include using an official e-mail address, posting via an official social media account, or acting as an appointed representative at an online or offline event.

1.5 Enforcement

Instances of abusive, harassing, or otherwise unacceptable behavior may be reported to the community leaders responsible for enforcement at Ian Stapleton Cordasco <graffatcolmingov@gmail.com>, Ian Lee <IanLee1521@gmail.com> or Florian Bruhin <me@the-compiler.org>. All complaints will be reviewed and investigated promptly and fairly.

All community leaders are obligated to respect the privacy and security of the reporter of any incident.

1.6 Enforcement Guidelines

Community leaders will follow these Community Impact Guidelines in determining the consequences for any action they deem in violation of this Code of Conduct:

1. Correction

Community Impact: Use of inappropriate language or other behavior deemed unprofessional or unwelcome in the community.

Consequence: A private, written warning from community leaders, providing clarity around the nature of the violation and an explanation of why the behavior was inappropriate. A public apology may be requested.

2. Warning

Community Impact: A violation through a single incident or series of actions.

Consequence: A warning with consequences for continued behavior. No interaction with the people involved, including unsolicited interaction with those enforcing the Code of Conduct, for a specified period of time. This includes avoiding interactions in community spaces as well as external channels like social media. Violating these terms may lead to a temporary or permanent ban.

3. Temporary Ban

Community Impact: A serious violation of community standards, including sustained inappropriate behavior.

Consequence: A temporary ban from any sort of interaction or public communication with the community for a specified period of time. No public or private interaction with the people involved, including unsolicited interaction

with those enforcing the Code of Conduct, is allowed during this period. Violating these terms may lead to a permanent ban.

4. Permanent Ban

Community Impact: Demonstrating a pattern of violation of community standards, including sustained inappropriate behavior, harassment of an individual, or aggression toward or disparagement of classes of individuals.

Consequence: A permanent ban from any sort of public interaction within the community.

1.7 Attribution

This Code of Conduct is adapted from the [Contributor Covenant][homepage], version 2.0, available at https://www.contributor-covenant.org/version/2/0/code_of_conduct.html.

Community Impact Guidelines were inspired by [Mozilla's code of conduct enforcement ladder](<https://github.com/mozilla/diversity>).

[homepage]: <https://www.contributor-covenant.org>

For answers to common questions about this code of conduct, see the FAQ at <https://www.contributor-covenant.org/faq>. Translations are available at <https://www.contributor-covenant.org/translations>.

An Introduction to the PyCQA

“PyCQA” is an abbreviated name for “Python Code Quality Authority”.

The PyCQA is not actually an authority on anything. (See also *Creation of the PyCQA*.)

2.1 What is the PyCQA?

The PyCQA is a loose organization of people who maintain projects in roughly the same domain: automatic style and quality reporting. Almost all of these projects are widely used by the larger Python community (and by each other) to enforce style guidelines and maintain some modicum of consistency within a code base. The organization formed around these people to provide them an easy to remember namespace for their projects.

2.2 Who is part of the PyCQA?

The projects housed under the PyCQA are all presently maintained by the following people:

- Astroid
 - Ashley Whetter
 - ceridwen
 - Claudiu Popa
 - Dmitry Pribysh
 - Florian Bruhin
 - Guillaume Peillex
 - Łukasz Rogalski
 - jcristau
 - Roy Williams

- Sylvain Thénault
- Bandit
 - Eric Brown
 - Gage Hugo
 - Ian Stapleton Cordasco
 - Luke Hinds
- Baron
 - Laurent Peuch
- Flake8
 - Ian Cordasco
- flake8-docstrings
 - Ian Cordasco
- mccabe
 - Ian Cordasco
- mccabe-console-script
 - Ian Cordasco
- pycodestyle
 - Ian Lee
- pep8-naming
 - Ian Cordasco
- pydocstyle
 - Amir Rachum
 - Shachar Ohana
 - Sambhav Kothari
- Pylint
 - Ashley Whetter
 - ceridwen
 - Claudiu Popa
 - Dmitry Pribysh
 - Florian Bruhin
 - Guillaume Peillex
 - Łukasz Rogalski
 - jcristau
 - Roy Williams
 - Sylvain Thénault
- RedBaron

– Laurent Peuch

2.3 Where does the PyCQA live?

As you may have noticed, we have a [GitHub organization](#) and a [GitLab group](#). The PyCQA likes to respect all individual project decisions about where development primarily takes place. If your project lives on either of these two services, you're welcome to share our organizational structures.

You might also notice, now that you've seen both our GitHub organization and GitLab group that we mirror repositories between the sites. If your project lives within the PyCQA and you'd like a mirror made (on the other service), let us know and we'll be happy to set that up for you.

2.4 How can I add my project to the PyCQA?

If your project is related to code style or quality reporting, your project is welcome. All we ask is that you send an email to the [code-quality mailing list](#) with your request to join. In all likelihood one of the organization owners will respond to you and help you set everything up to move your project into the organization.

The History of the PyCQA

The PyCQA has a short, but interesting history.

3.1 Creation of the PyCQA

On 2014 September 12, Ian Cordasco [sent an email](#) to the code-quality mailing list with the idea to migrate Flake8 from Mercurial to Git.

The response was positive with the caveat that Free Software should be developed on similarly Free Software (GitLab). Ian was uncomfortable transferring the project to his own personal GitLab account so in jest and following the example of other groups ([PyCA](#), [PyPA](#), etc.), he created a GitLab group for the project. He also created a GitHub organization with the same name so he could mirror Flake8 to GitHub and accept contributions there. As the sole maintainer of flake8-docstrings, Ian also migrated that project from Mercurial to Git and onto GitLab (with a mirror on GitHub).

3.2 pep8 joins the organization on GitHub

After a while as the only member, Ian invited Ian Lee to join the PyCQA with pep8. Following some conversations, pep8 and its development team (mostly Ian Lee, but also its creator, Johann Rocholl, and former maintainer, Florent Xicluna) [moved to the PyCQA](#) in July of 2015.

3.3 Pylint and Astroid move to GitHub and the PyCQA

In October of 2015, Florian Bruhin [sent an announcement](#) to the code-quality mailing list that Pylint and Astroid would be moving to Git and GitHub from Mercurial and BitBucket. Over the course of the ensuing discussion, it was suggested that they also join the PyCQA and in December of 2015 the migration was complete.

3.4 mccabe and pep8-naming move to the PyCQA organization

Around the same time, Ian Cordasco moved more projects that he maintained into the PyCQA namespace. This time he moved both `mccabe` and `pep8-naming` which were previously maintained in the `Flintwork` organization.

3.5 pydocstyle (formerly pep257) moves to the PyCQA organization

In early January of 2016, Guido van Rossum asked that `pep257` change its name. The usage of “PEP” was apparently creating confusion among users causing bugs to be reported to the PEP repository. Vladimir Keleshev, the creator of `pep257`, and Amir Rachum, the active maintainer, decided to move the project to the PyCQA. On 29 January 2016, the move and rename was complete and Amir joined the PyCQA with `pydocstyle`.

3.6 Baron and RedBaron join the PyCQA

On 2016 March 28, the author of `Baron` and `RedBaron` sent a message to the [code-quality mailing list](#) about a new release and the need for more contributors and maintainers. At that point, the organization extended an invitation for Laurent Peuch to join with both `Baron` and `RedBaron`. The transfer was completed on 2016 March 31.

3.7 Bandit moves to the PyCQA organization

On 2018 February 26, the authors of `Bandit` sent a request to the [code-quality mailing list](#) to accept `Bandit` from the OpenStack namespace to the PyCQA. The transfer was completed on 2018 May 2.

The Management of the PyCQA

- Creator:
 - Ian Cordasco
- Organization owners:
 - Amir Rachum
 - Claudiu Popa
 - Florian Bruhin
 - Ian Cordasco
 - Ian Lee
- Pycodestyle (formerly pep8) Administrator(s):
 - Ian Lee
- PyLint Administrator(s):
 - Claudiu Popa
 - Florian Bruhin
 - Sylvain Thénault
- Pydocstyle (formerly pep257) Administrator(s):
 - Amir Rachum
- Baron and RedBaron Administrator(s):
 - Laurent Peuch
- Flake8-import-order Administrators:
 - Alex Stapleton
 - Phil Jones
- Flake8-bugbear Administrator(s):

- Łukasz Langa
- Bandit Administrator(s):
 - Eric Brown
 - Gage Hugo
 - Ian Stapleton Cordasco
 - Luke Hinds

All projects in the PyCQA should have a Code of Conduct that they are willing to enforce.

4.1 Project Management

Each project is managed by its own members. There is no over-arching management over all projects in the PyCQA. While many people *could* merge pull requests and close issues on *all* repositories, they are trusted to not do so unless they are specifically a member of the team maintaining that project. At least one member of each project's administration will be an Owner of the organization. This is to allow them to manage their teams and membership of others in teams.

4.2 Team Structure

Projects are encouraged to have as many teams as they deem necessary to run their project. Most projects operate with the following:

- Team of administrators - this team has “Admin” permissions in GitHub for the project's repository/repositories.
- Team of developers (or cores, or whatever you want to call them) - this team has “Write” permissions in GitHub for the project's repository/repositories.

This set-up allows projects to add more people who can merge pull requests and close issues without being able to change settings on the repository itself, add or remove services, etc.

Join the Community!

There are quite a few ways to get involved with the PyCQA.

1. You can contribute directly to projects in a variety of ways
2. You can start discussions by opening issues on our [meta](#) repository
3. You can chat with us on IRC on the [Freenode](#) server in the `##python-code-quality` channel (See also *“Using IRC”*)
4. You can join our [mailing list](#) (this list does not receive a large amount of messages so it should not cause an information overload if you join)

5.1 Using IRC

IRC is an old and durable protocol that allows people to chat in real time. There are many ways to begin using IRC but we recommend using Freenode’s [webchat](#) as it is both secure (served over TLS) and should work in any browser. [IRCHelp.org](#) has a great [tutorial](#) to learn how to use IRC.

If you outgrow Freenode’s webchat, there are a number of other options for connecting to IRC. Wikipedia has an overwhelmingly long [list of IRC clients](#).

The following is a list of clients that the current members of the PyCQA use:

5.1.1 Windows

- [Hexchat](#) (Graphical, Also works on macOS and Linux)

5.1.2 macOS

- [Textual](#) (Graphical, WebKit based)
- [weechat](#) (Curses, Terminal program)

5.1.3 Linux

- `weechat` (Curses, Terminal program)
- `irssi` (Curses, Terminal program)

CHAPTER 6

Indices and tables

- `genindex`
- `modindex`
- `search`