PyCQA Meta Documentation

Release 1.0

PyCQA Developers

November 30, 2016

Contents

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

CHAPTER 1

Contributor Covenant Code of Conduct

1.1 Our Pledge

In the interest of fostering an open and welcoming environment, we as contributors and maintainers pledge to making participation in our project and our community a harassment-free experience for everyone, regardless of age, body size, disability, ethnicity, gender identity and expression, level of experience, nationality, personal appearance, race, religion, or sexual identity and orientation.

1.2 Our Standards

Examples of behavior that contributes to creating a positive environment include:

- Using welcoming and inclusive language
- Being respectful of differing viewpoints and experiences
- Gracefully accepting constructive criticism
- Focusing on what is best for the community
- Showing empathy towards other community members

Examples of unacceptable behavior by participants include:

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

1.3 Our Responsibilities

Project maintainers are responsible for clarifying the standards of acceptable behavior and are expected to take appropriate and fair corrective action in response to any instances of unacceptable behavior.

Project maintainers 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, or to ban temporarily or permanently any contributor for other behaviors that they deem inappropriate, threatening, offensive, or harmful.

1.4 Scope

This Code of Conduct applies both within project spaces and in public spaces when an individual is representing the project or its community. Examples of representing a project or community include using an official project e-mail address, posting via an official social media account, or acting as an appointed representative at an online or offline event. Representation of a project may be further defined and clarified by project maintainers.

1.5 Enforcement

Instances of abusive, harassing, or otherwise unacceptable behavior may be reported by contacting the project team at [INSERT EMAIL ADDRESS]. All complaints will be reviewed and investigated and will result in a response that is deemed necessary and appropriate to the circumstances. The project team is obligated to maintain confidentiality with regard to the reporter of an incident. Further details of specific enforcement policies may be posted separately.

Project maintainers who do not follow or enforce the Code of Conduct in good faith may face temporary or permanent repercussions as determined by other members of the project's leadership.

1.6 Attribution

This Code of Conduct is adapted from the Contributor Covenant, version 1.4, available at http://contributor-covenant.org/version/1/4/

CHAPTER 2

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
  - ceridwen
  - Claudiu Popa
  - Dmitry Pribysh
  - Florian Bruhin
  - jcristau
  - Sylvain Thénault
- 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
- Pylint
  - ceridwen
  - Claudiu Popa
  - Dmitry Pribysh
  - Florian Bruhin
  - jcristau
  - 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.

CHAPTER 3

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.

CHAPTER 4

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

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 organzation. 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.

CHAPTER 5

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 OS X and Linux)

5.1.2 OS X

- 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