GEEKFLARE

Advertiser disclosure

10 Python Linter Platforms to Clean Up Your Code

Last updated: September 5, 2023

Geekflare articles are written by humans for humans. Learn more →

Anesu Kafesu
Contributor

Bugs are any developer's worst nightmare. At best, we catch bugs during development and testing. But even then, they can take hours of frustrating debugging before being resolved. At worst, they sneak into the production code base, where they cause wreak havoc, causing the software to malfunction.

This could result in minor inconveniences to end-users or large, life-threatening disasters. Regardless, developers should strive to catch and eliminate bugs as early as possible.

In this article, we will explore a method of achieving this in Python called linting and explore the best Python linter tools and platforms.

programmatic errors. The name linting comes from the Unix utility Lint used to examine source code written in the C programming language.

Over the years, linting has grown beyond the C programming language and is now performed for source code in the most popular languages as well, including Python. Python Linters are the tools used to perform linting for different Python programs.

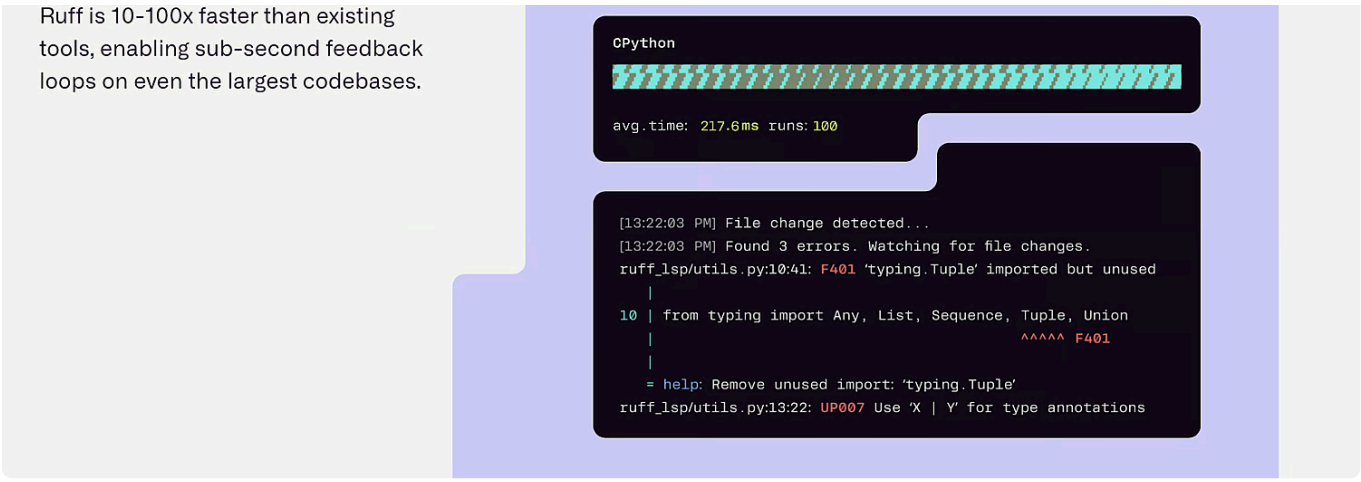
Why Are Linters Important in Programming?

- Linting reduces the number of bugs in production code by checking your code for you. It functions as an extra pair of eyes to help you catch bugs early and with less effort. As a result, it improves your productivity and your code quality.
- It also helps optimize code for efficiency and performance by flagging unused constructs such as variables and unreachable code. This will help reduce the source code size and the resulting distributed program.
- It also helps standardize code by replacing tabs with spaces or the other way around so that the codebase is written consistently.
- Linting makes it easier to review code because it ensures the reviewer that certain standards are already met. This means the code reviewer does not have to check if all variables use snake cases, for example.

There are many tools you can use to lint your Python code. To choose the best tool, you have to consider the features offered, such as the size of the ruleset, flexibility, cost, and ability to share rules with other team members.

Considering those and many other aspects, here is a list of the best tools.

Ruff



Ruff is a Python linter made by Astral. It is free and open source. Ruff is written in Rust and is, therefore, incredibly fast compared to other linters. It is easily installed via pip and enforces over 500 rules on your code base.

In addition, Ruff integrates well with editors such as Visual Studio Code, Neovim, Sublime Text, and others. It has auto-fix support, so you can easily fix errors without needing to rewrite code yourself.

At the time of writing, Ruff is in version 0.0.267, which means it might not exactly work well, and there could be breaking changes before it hits version 1.

Sonar

Sonarlint is a free linting tool that is implemented as an IDE plug-in. It can be installed with most major IDEs, such as Visual Studio Code, PyCharm, and Eclipse.

In addition to Python, it also supports other languages such as JavaScript, Java, and C++. It runs as you write code, giving you real-time and instant feedback to fix your code.

practices.

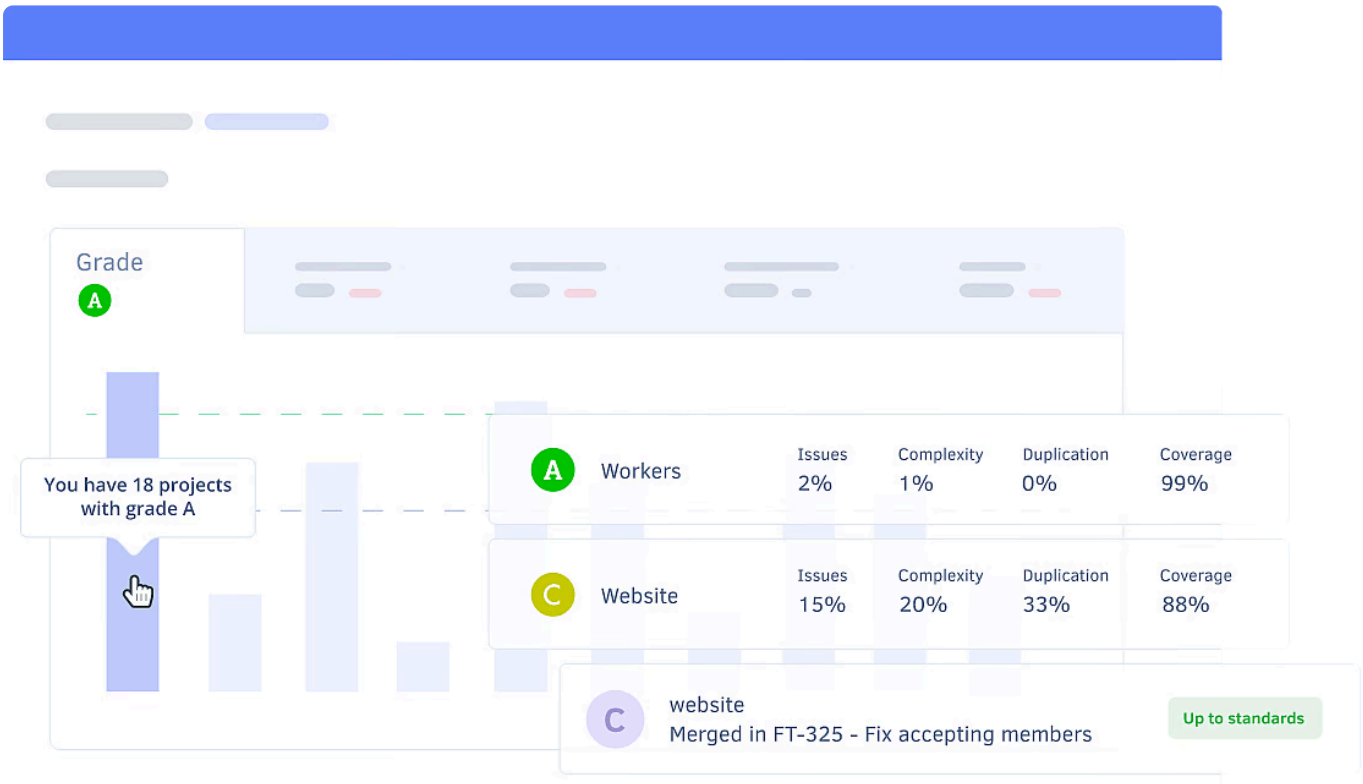
pytype

pyType is a popular linter created by and used by Google for their many Python projects. It is free to use and is open source. PyType checks your code and infers types. This means it can check your code for type-related errors without needing you to write explicit type annotations.

In addition, pyType checks code from different files to ensure correctness. It can be installed with pip and used as a command-line tool.

PyType was developed and tested on Linux. Therefore, it works best on a Linux machine. For MacOS, PyType requires OSX version 10.7 or higher and XCode version 8 or higher. Currently, it does not support Windows unless you use it in the Windows Subsystem for Linux.

Codacy



Codacy is a paid linting tool with a free plan for open-source developers. It can help identify issues in your code, including security vulnerabilities and errors.

issues in your code and how to resolve them. In addition, it also provides in-line annotations.

Codacy can also be integrated into workflows with tools such as Git, Jira Software, Slack, and git providers such as GitLab and BitBucket. It supports more than 40 of the most popular programming languages.

Pylint

PyLint is a popular Python linting tool. It is free and open source. PyLint can enforce coding standards such as limiting line length and ensuring variable names follow convention. In addition, it also detects errors such as unimported modules.

You can customize PyLint using config files. It integrates well with Emacs, Vim, Eclipse, Spyder, and TextMate. PyLint can be automated using tools such as Apycot, Hudson, or Jenkins to build CI/CD pipelines. To install PyLint, you can use package managers on Linux or Pip on Windows and MacOS.

Flake8

Flake8 is a free and open-source Python linter. It verifies PEP8, pyflakes, and circular complexity. It is generally very accurate, producing a low number of false positives. This results in better code overall and a better development experience.

You can add Flake8 to your Python IDE or editor, such as PyCharm or Sublime Text. Flake8 supports both Python 2 and Python 3. Alternatively, you can run it from the command line or

options, Flake8 allows you to store them in a configuration file.

Black

Black is an uncompromising and opinionated Python linter. As a result, it is fast ad deterministic. It is deterministic because it uses its own internal standards and enforces them across different projects.

This ensures that code linted by black looks the same regardless of the project. Black is among the most popular tools used by notable open-source projects such as Pytest, Django, and SQLAlchemy.

Organizations such as Facebook, Mozilla, and Tesla use Black for their Python projects. Black is an open-source project.

autopep8

autopep8 is a popular Python linter that lints code to conform to the PEP8 style guide, which is the official style guide for Python code. Unlike some other code linters, Autopep8 focuses on fixing your code's formatting and does not modify the code's logic or structure.

Autopep8 also provides a range of configuration options to customize its behavior. You can specify formatting preferences, enable or disable specific rules, control the line length limit, and configure other aspects according to your project's requirements. Like most linters, it integrates well with existing IDEs and editors.

Pychecker

PyChecker is a Python linter that helps identify issues such as unreachable code, unused variables, and unassigned parameters. It helps developers find bugs faster and makes your code easier to maintain.

It can be used from the command line, which helps lint entire projects with a single command. This is because it can recursively traverse through all files in a directory, linting them. It can also be used within Python Code and can analyze code written in any style. It complies with PEP8 and other formatting guidelines.

Pylama is a wrapper for various individual linters. It integrates with several well-known linters, such as Pylint, PyFlakes, pycodestyle, Mypy, and others. By using multiple linters, it provides a comprehensive analysis of code quality and enforces a wide range of coding standards.

It is often integrated into continuous integration systems such as GitHub Actions. Pylama also works from the command line and can be integrated with most IDEs and code editors.

Final Words

This article discussed linting and the common tools you might use to lint your Python project. Python linting tools are a popular part of Python development tooling, and they improve your productivity.

Next, check out CSV tools to convert, format, validate, and more.

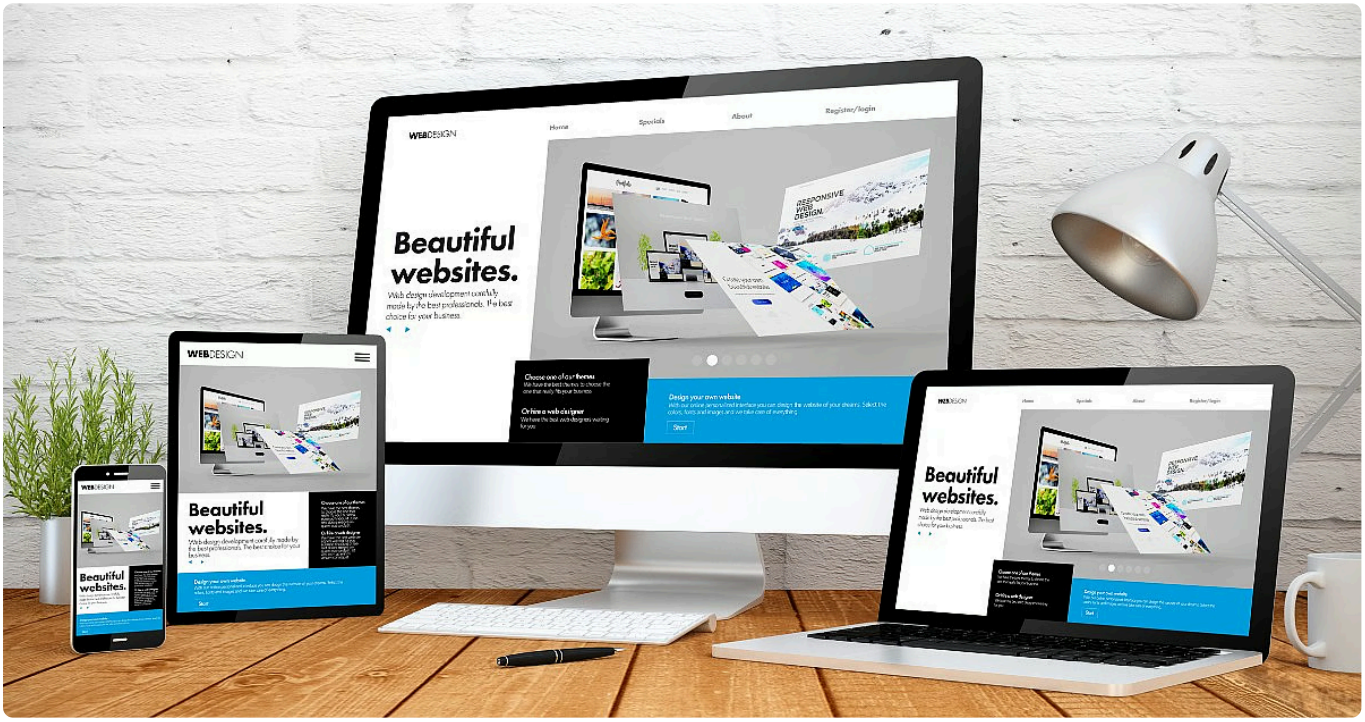
Anesu Kafesu
Contributor

I am a Software Engineer specialising in Web Development, Mobile Application Development and Artificial Intelligence. I occasionally write on the same topics.

Was this helpful?

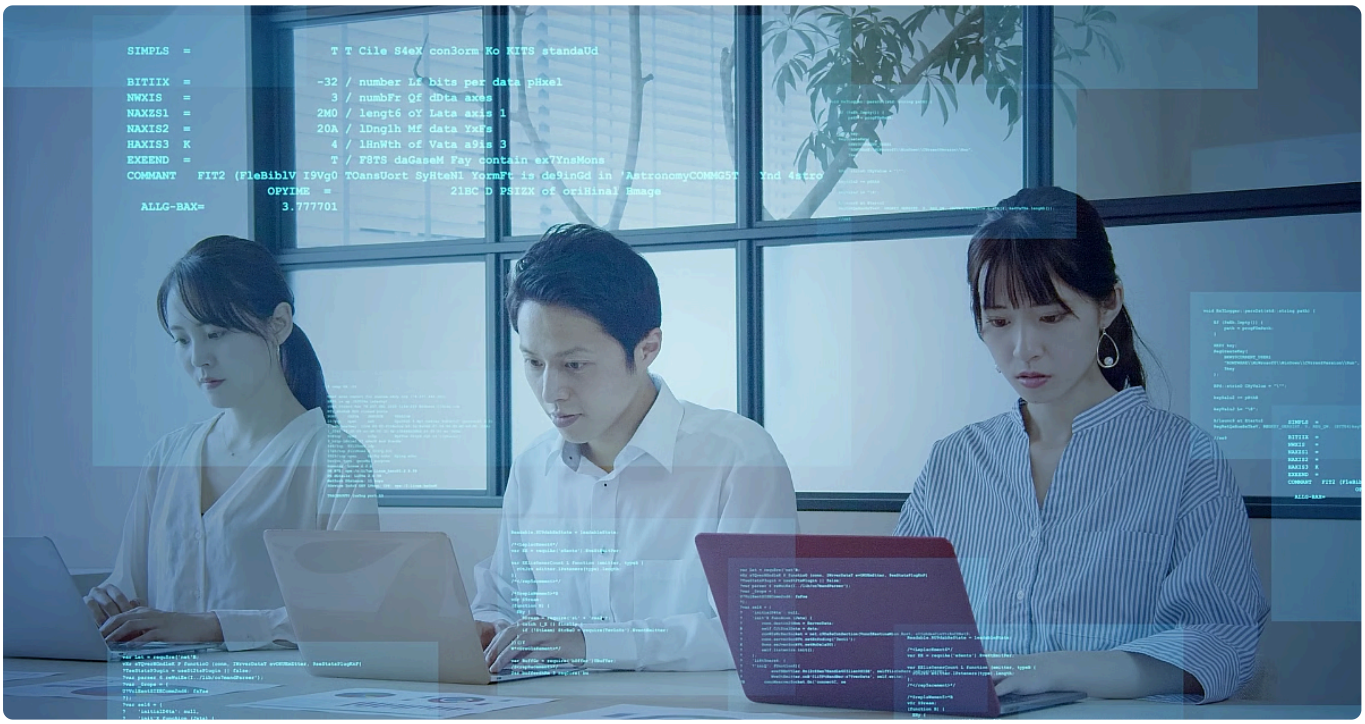
More on Development

Event-Driven Architecture – Streamlining Software Delivery

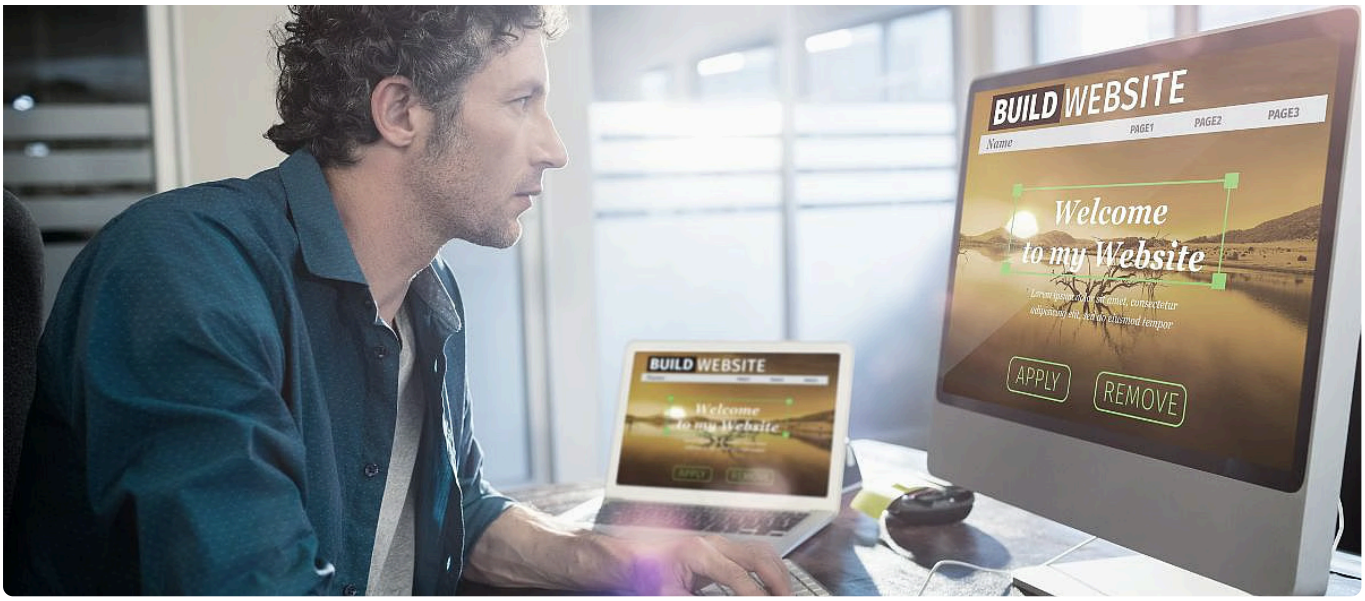


12 Best Website Builders for Non-Techies and Non-Designers

6 Best Cloud CMS Platforms for Businesses of All Sizes (2024)



Python Logging: Your Ultimate Guide to Mastering Modules and Exception Handling



How to Create CSS Sticky Headers to Enhance User Experience

12 Fast and Feature-Rich Financial Charting Libraries for Your Next App

GEEKFLARE

Privacy Policy

Cookie Policy

Disclosure

GENERAL

Editorial Policies

Sitemap

English