



Motivation

Code quality is a critical factor for software's long-term maintenance cost. It is therefore crucial to teach programming students not only how to write functionally correct code but also how to write code that does not suffer from code quality defects.

Teaching this skill through manual code reviews scales poorly. They require a lot of time, which is costly and sometimes entirely unfeasible.

Automating code quality feedback can have a tremendous positive impact, especially if students receive the relevant feedback often, quickly, and consistently. However, the output from currently available automated tools is often unsuitable for novice programmers.

Example

Code with a defect

```
a = a // 2
```

Feedback

Use augmented assignment: `a /= 2`

Existing automated tools for Python

There are several industry-grade linters for Python (Pylint; Flake8 and its plugins; Ruff). They focus on formatting or advanced constructs (e.g. docstrings or exceptions). On the other hand, they do not detect some of the simplest (and most common) code quality defects in novice code (like using a while loop where a for loop would suit better).

There are two teaching-specific linters built on top of the industry-grade linters. PyTA has only a few custom detectors for code quality; it is primarily focusing on finding bugs. Hyperstyle was developed for one specific learning environment, so it is difficult to reuse it anywhere else. It provides no custom detectors.

Both tools also report many false positives and code defects deemed irrelevant by current state-of-the-art research.

Implemented solution: EduLint

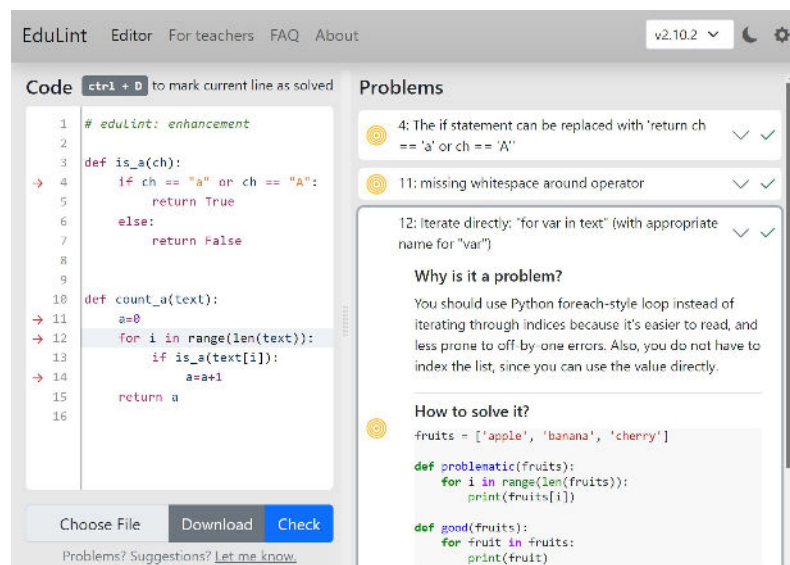
EduLint is a Python code linter customized for novice programmers. It uses, filters, and enhances Pylint and Flake8.

By default it detects 185 code quality defects, 17 of which are not detected by any other tool. Additional 16 detectors were reimplemented to improve precision or recall of existing ones.

It can be used at edulint.com or as a Python pip package. It is fully open-source, and can be easily deployed using Docker.

Main features

1. Reports defects relevant to novice programmers. Their relevance is supported by several recent papers and the teaching experience of several educators at FI MU.
2. Avoids false positives. Each detector is carefully vetted before EduLint starts reporting it.
3. Can be configured through special comments in the linted file. This accommodates educators' different opinions and spares students from handling configuration files.
4. Provides descriptions and code examples for defects. These explain to the students why the defect should be addressed and how to fix it.
5. Can be used easily at edulint.com, without the need to install anything.



EduLint's web interface at edulint.com

Most frequently reported defect by tool

EduLint was evaluated on 100 000 submissions from novice programmers from different sources (a university course, a contest for high school students, and publicly available datasets). The thesis includes a comprehensive evaluation.

Tool	Most frequent defect	Files with the defect
Pylint	missing docstring	99%
PyTA	missing docstring	99%
Hyperstyle	possibly misspelled name (frequently a false positive)	52%
EduLint	use augmented assign	8%

Many educators and researchers consider missing docstrings irrelevant to novice programmers.

In the default configuration, EduLint intentionally reports fewer defects so students can focus on the relevant ones. If students want more feedback, they can enable some or all of the defects Pylint and Flake8 can report.

Feature comparison

	Pylint	PyTA	Hyperstyle	EduLint
Relevance	~ 1)	~ 1)	~ 1)	✓
Precision	~ 2)	~ 2)	~ 2)	✓
Configurability	✓	✗ 3)	~ 4)	✓
Explanations	~ 5)	✓	✗ 6)	✓
Ease of use	✗ 7)	~ 8)	✗ 7)	✓

- ✓ fully implements the feature
 - = partially implements the feature
 - ~ = does not implement the feature
 - ✗ = does not implement the feature
- 1) Frequently reports defects deemed irrelevant by current research.
 - 2) Reports defects with a high ratio of false positives.
 - 3) Does not allow enabling detection of individual defects.
 - 4) Allows only enabling of underlying linters.
 - 5) Provides explanations only in its documentation.
 - 6) Does not provide any explanations.
 - 7) Needs a pip package or a docker image installed.
 - 8) Needs a pip package installed, but generates HTML files.

Current state and future work

The project itself and the accompanying research are ongoing. We have one paper in review, and are preparing several others.

EduLint is used in both introductory programming courses at FI MU and several of its seminars for high-school students. It is also fully available to the general public.