Python 3 Best Practices

Following Python Style Guidelines: Pep8

Reindert-Jan Ekker

@rjekker www.codesensei.nl

Python 3 Best Practices

Version Check

Version Check

This version was created by using:

- Python 3.10
- Pylint 2.15
- Black 22.10
- Sphinx 5.3
- Mypy 0.982

Version Check

This course is 100% applicable to:

- Python 3.10 and later
- Any version of Pylint
- Any version of Black
- Any version of Sphinx
- Any version of Mypy

Overview

PEP8: Python style guidelines

What is a PEP?

Overview of PEP8

Applying rules to your code

- Pycharm
- Visual Studio Code

Tools

- pylint
- black

Python Enhancement Proposal (PEP)

A design document providing information to the python community or describing a new feature for python or its processes or environment

PEP8: Coding Style

It’s about style, not about program correctness

Formatting

Lay-out of code and comments

Whitespace/punctuation

Indentation, quotes, operators, …

Naming

Classes, functions, variables, …

If you get it wrong

Your tools will help you

“A Foolish Consistency is the Hobgoblin of Little Minds”

Ralph Waldo Emerson

Demo

Reading PEP8

Applying the rules to your code

Lay-out

Indentation: 4 spaces per level

Max. line length 79 characters

2 lines between top-level functions and classes

1 line between methods inside a class

Use of spaces in expressions

Imports

Should be on separate lines

Three groups, separated by blank lines

- Standard library
- Third-party libraries
- Local application/library

Naming

Modules: short, lowercase names

Classes: CapitalizedNaming

Functions: lowercase_with_underscores

Constants: ALL_CAPS

Non-public names start with underscore

Documentation

Docstrings for all public modules, functions, classes and methods

Guidelines for documentation are in PEP257

Demo

Checking your code

- pylint
- flake8

Fixing your code

- black

Tools: Checkers

pycodestyle:

- check PEP8 rules

flake8

- PEP8
- code smells
- complexity

pylint

- very flexible, can check many things
- PEP8 checks are not as complete

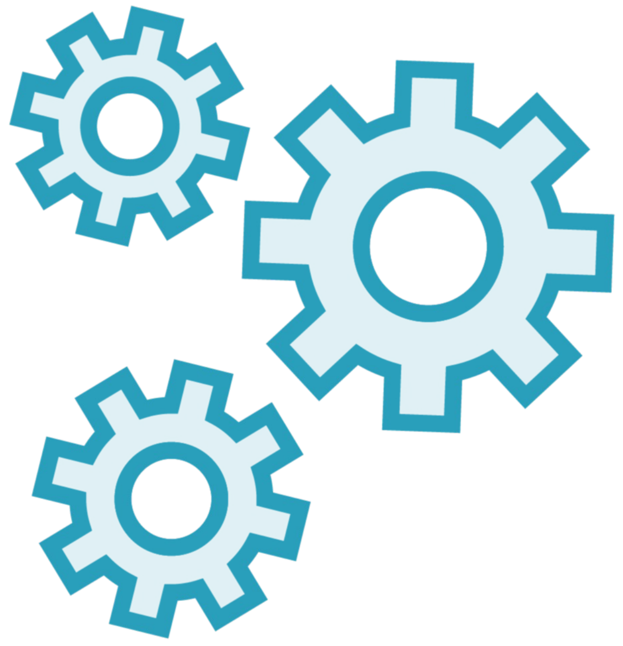
Tools: Fixing

black

- code formatter

reorder-python-imports

Automation



Most of these tools are used best in an automated workflow (CI/CD)

A very popular option is to use the pre-commit framework (https://pre-commit.com/)

Summary

PEP8: Python style guidelines

What is a PEP?

Overview of PEP8

Applying rules to your code

Tools

Up Next: Documenting Your Project