Code Quality Tools

Why and How?

ZEN OF PYTHON

- Beautiful is better than ugly.
- Explicit is better than implicit.
- Simple is better than complex.
- Complex is better than complicated.
- Flat is better than nested.
- Sparse is better than dense.
- Readability counts.
- Special cases aren't special enough to break the rules.
- Although practicality beats purity.
- Errors should never pass silently.
- Unless explicitly silenced.

ZEN OF PYTHON (cont’d)

- In the face of ambiguity, refuse the temptation to guess.
- There should be one-- and preferably only one --obvious way to do it.
- Although that way may not be obvious at first unless you're Dutch.
- Now is better than never.
- Although never is often better than *right* now.
- If the implementation is hard to explain, it's a bad idea.
- If the implementation is easy to explain, it may be a good idea.
- Namespaces are one honking great idea -- let's do more of those!

Why?

- Consistency
- Catch bugs
- Enable automation
- Confident refactor
- Reduce Review time
- Readability, Maintainability,Extensibility
- Catch easy to miss problems
- Check complexity

PEP8

- Indentation
- Tabs or Spaces?
- Maximum Line Length
- Imports
- Line Breaks
- Whitespace

TOOLS

flake8

- Most pep8 rules
- Bare minimum check

Pylint

- Static code analysis
- Descriptive names
- Docstrings
- Configuration needed
- Helps with refactoring
- Use score to increase code quality

isort

- Import sorting

Others

- Pylama
- Pyflakes
- Radon
- Mcabe
-

Integrations with IDEs

- Pycharm
- Sublime
- VSCode
- Vim
- Emacs??

Google Python StyleGuide

- Good conventions

Code Reviews

- Check each other’s code
- Improve quality
- Should not include automate-able steps
- Find small mistakes