Effective Programming Practices for Economists

Software engineering

Style guides

Janoś Gabler and Hans-Martin von Gaudecker

Main message

- Style guides are important
- Style guides do not automatically make code good
- Style guides should be automated!

Naming conventions

- Functions and methods are `lowercase_with_underscores`
- Local variables are `lowercase_with_underscores`
- Global ~~variables~~ constants are `UPPERCASE_WITH_UNDERSCORES`
- Classes are `CamelCase`, instances are `lowercase_with_underscores`
- Private functions start with `_underscores`

Why naming conventions are important

- Help to communicate information without extra characters
- Make it easier to read code written by others
- Reduce your mental load when deciding about variable names
- Do not cost any time to implement

Examples of formatting rules

- Loop and function bodys are indented by four spaces (mandatory)
- Maximum line length of 88 characters
- Two blank lines between function definitions, one between method definitions
- Whitespaces around operators
- ...

Why formatting is important

- Many formatting rules are based on rigorous research about code readability
- Indentation helps to prevent bugs!
- Familiar formatting makes it easier to read code written by others

Why style guides should be automated

- Code style is important, so it has to be enforced
- Formatting is boring, so it should be automated
- Could never manage to be consistent enough to not get any fake changes in version control
- Manual formatting and discussions about it distract from important things

How to automate style guides?

- Linters in your editor
- Formatters in your editor
- Pre-commit hooks