
welcome to

SETTING UP A REPOSITORY TEMPLATE FOR YOUR TEAM

with Consistency, Reproducibility, and Compliance in Mind

791

Views

32

Claps



Python HOW: Using Poetry, Make, and pre-commit-hooks to Setup a Repo Template for your Team

Bring consistency, rigour, and best practices to your messy data science team

 Medium / Aug 5

SENIOR DATA SCIENTIST & TECH LEAD

introduce **ABOUT ME**

A husband, a dad for 5yo and 1yo girls, a teacher, an archer and mountain biker

PhD from King's College London in Medical Imaging Sciences in 2015

MLOps at NielsenIQ BrandBank

medium.com/@DrGabrielHarris

github.com/DrGabrielHarris



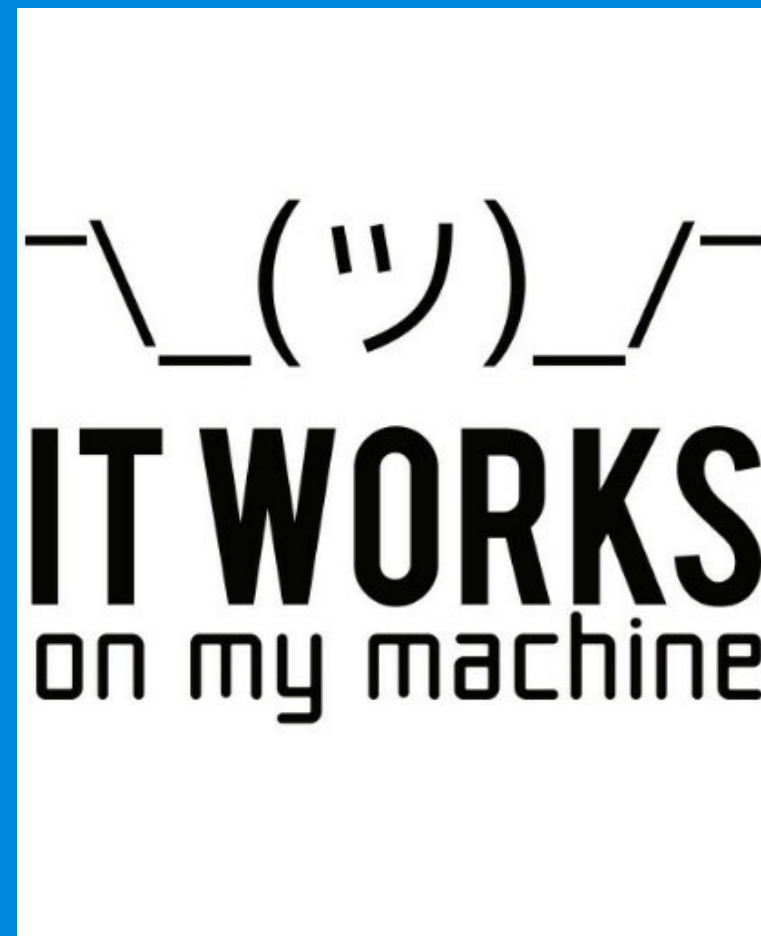
**GABRIEL
HARRIS**

why?



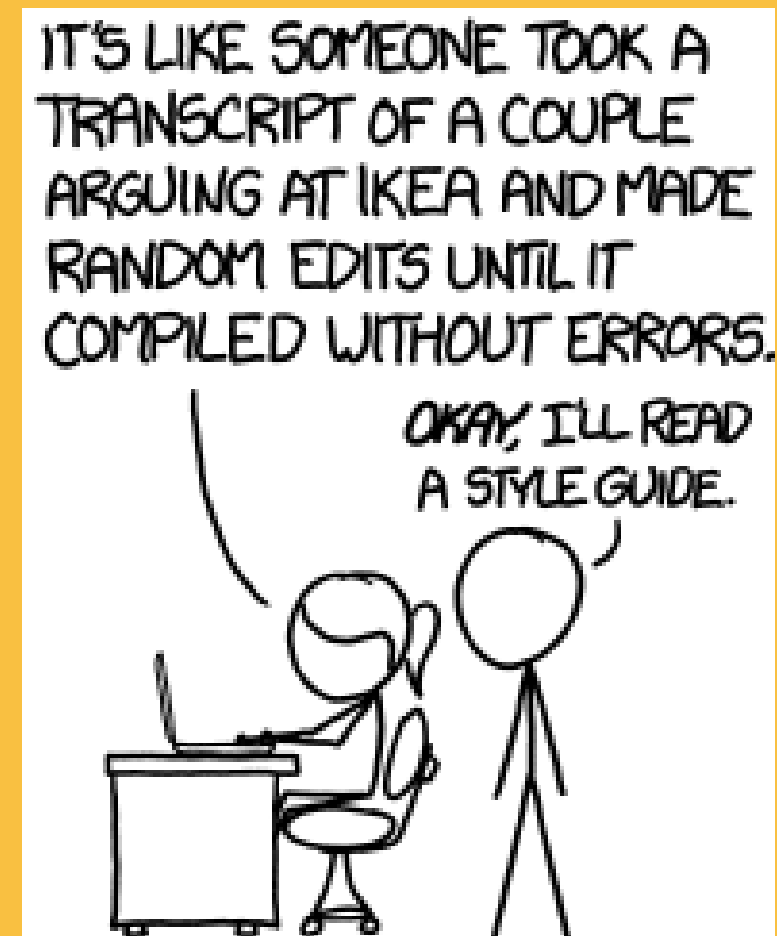
CONSISTENCY

why?



REPRODUCIBILITY

why?



COMPLIANCE

how?

GNU MAKE



CONSISTENCY

 Makefile

how?

POETRY



REPRODUCIBILITY

 pyproject.toml

how?

PRE-COMMIT-HOOKS



COMPLIANCE

 .pre-commit-config.yaml

MAKEFILE



GNU MAKE

Rule

target: prerequisites
recipes

Commands

make target

```
Makefile
1  .EXPORT_ALL_VARIABLES:
2  .PHONY: venv install pre-commit clean
3
4  GLOBAL_PYTHON = $(shell py -3.9 -c 'import sys; print(sys.executable)')
5  LOCAL_PYTHON = .\\.venv\\Scripts\\python.exe
6  LOCAL_PRE_COMMIT = .\\.venv\\Lib\\site-packages\\pre_commit
7
8  setup: venv install pre-commit
9
10 venv: $(GLOBAL_PYTHON)
11     @echo "Creating .venv..."
12     poetry env use $(GLOBAL_PYTHON)
13
14 install: ${LOCAL_PYTHON}
15     @echo "Installing dependencies..."
16     poetry install --no-root --remove-untracked
17
18 pre-commit: ${LOCAL_PYTHON} ${LOCAL_PRE_COMMIT}
19     @echo "Setting up pre-commit..."
20     .\\.venv\\Scripts\\pre-commit install
21     .\\.venv\\Scripts\\pre-commit autoupdate
22
23 clean:
24     if exist .\\.git\\hooks ( rmdir .\\.git\\hooks /q /s )
25     if exist .\\.venv\\ ( rmdir .\\.venv /q /s )
26     if exist poetry.lock ( del poetry.lock /q /s )
27
```

PYPROJECT.TOML



Commands

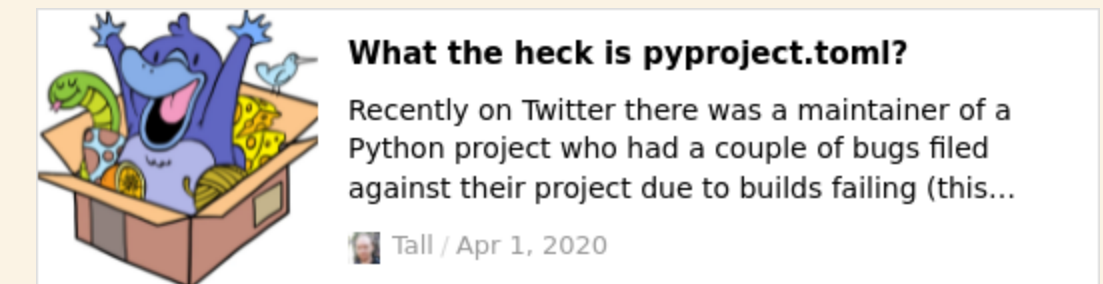
```
poetry add scikit-learn
poetry remove scikit-learn

poetry add pytest --group test

poetry install --without test,docs
poetry install --only dev
poetry install --sync

poetry update
```

```
[T] pyproject.toml M X
[T] pyproject.toml
1  [tool.poetry]
2  name = "Project Name"
3  version = "0.1.0"
4  description = "Project Description"
5  authors = ["Authors Names"]
6
7  [tool.poetry.dependencies]
8  python = "3.9.*"
9  pandas = "^1"
10
11 [tool.poetry.group.dev.dependencies]
12 black = "*"
13 flake8 = "*"
14 isort = "*"
15 nbstripout = "*"
16 notebook = "*"
17 rich = "*"
18 pre-commit = "*"
19
20 [tool.poetry.group.docs.dependencies]
21 pydocstyle = {extras = ["toml"], version = "*"}
22
23 [tool.poetry.group.test.dependencies]
24 pytest = "*"
25
26 [build-system]
27 requires = ["poetry-core>=1.0.0"]
28 build-backend = "poetry.core.masonry.api"
29
30 [tool.black]
31 line-length = 88
32
33 [tool.isort]
34 profile = 'black'
35
36 [tool.pydocstyle]
37 convention = "google"
38 add-ignore = "D100,D101,D102,D103,D104,D105,D106,D107"
```



*"...the purpose of **PEP 518** was to come up with a way for projects to specify what build tools they required ... all projects should (eventually) have ... development tools realized they ... could put their own configuration"*
– Bret Connan

.PRE-COMMIT-CONFIG.YAML



Hook

```
- repo: <link to a git package with a .pre-commit-hooks.yaml>
  rev: <package version>
  hooks:
  - id: hook id found in .pre-commit-hooks.yaml
  - arg: list of parameters to pass to the hook (default [])
```

```
.pre-commit-config.yaml X
.pre-commit-config.yaml > YAML > [ ] repos
1 repos:
2 - repo: https://github.com/psf/black
3   rev: '22.10.0'
4   hooks:
5   - id: black
6   - id: black-jupyter
7 - repo: https://github.com/PyCQA/flake8
8   rev: '5.0.4'
9   hooks:
10  - id: flake8
11    args: ['--max-line-length=88', '--extend-ignore=E203,E501,W503', '--exclude=.venv']
12 - repo: https://github.com/PyCQA/isort
13   rev: '5.10.1'
14   hooks:
15   - id: isort
16     args: ['--profile=black']
17 - repo: https://github.com/kynan/nbstripout
18   rev: '0.6.1'
19   hooks:
20   - id: nbstripout
21 - repo: https://github.com/pycqa/pydocstyle
22   rev: '6.1.1'
23   hooks:
24   - id: pydocstyle
25     args: ['--convention=google', '--add-ignore=D100,D101,D102,D103,D104,D105,D106,D107']
26 - repo: https://github.com/pre-commit/pre-commit-hooks
27   rev: 'v4.3.0'
28   hooks:
29   - id: check-ast
30   - id: end-of-file-fixer
31   - id: trailing-whitespace
32
```


BLACK 30.2K Stars

Black is able to read project-specific default values for its command line options from a `pyproject.toml` file in the `[tool.black]` section

```
[T] pyproject.toml x
[T] pyproject.toml
28 build-backend = "poetry.core.masonry.api"
29
30 [tool.black]
31 line-length = 88
32
33 [tool.isort]
```

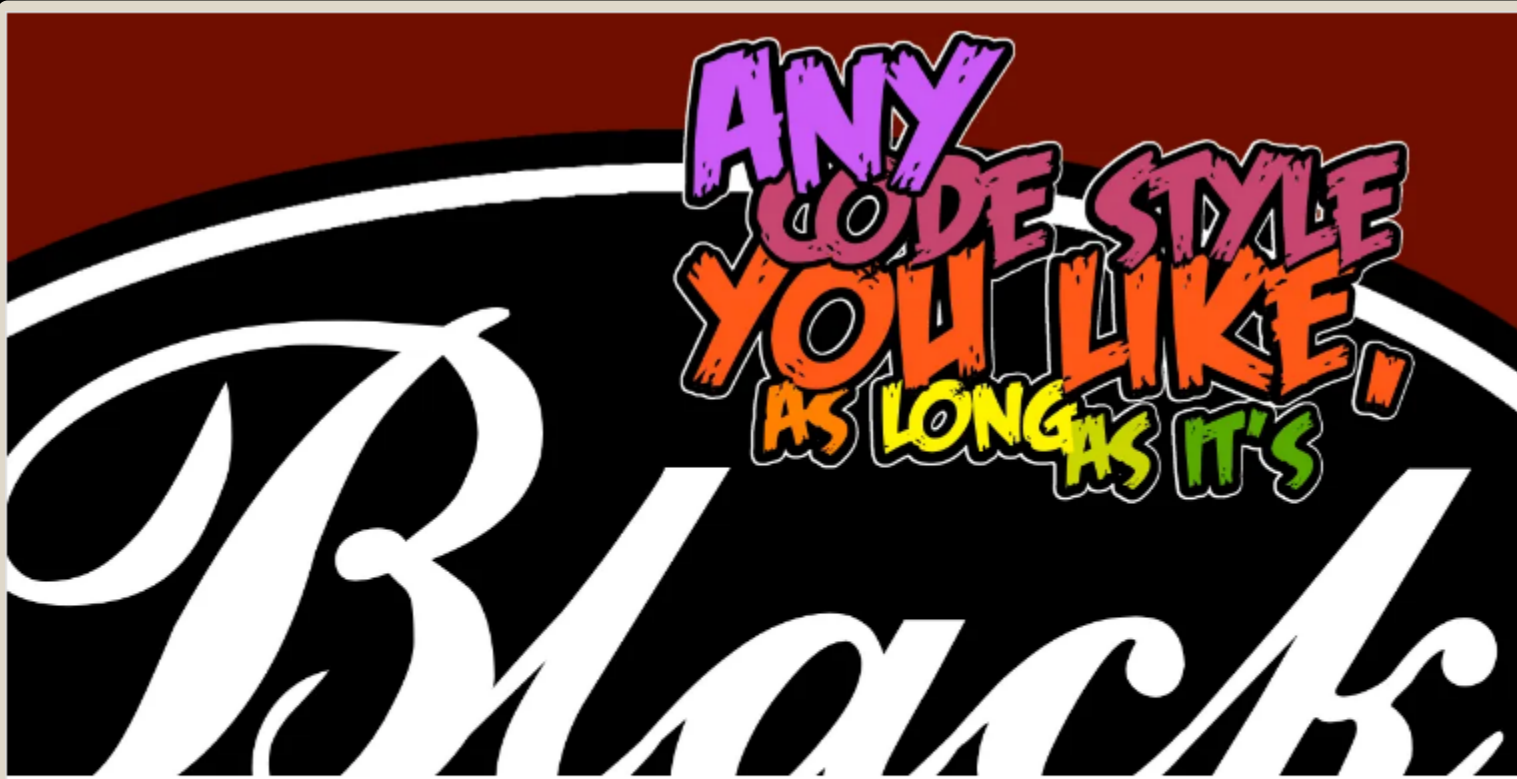
Configs

```
{ } settings.json M x
{ } settings.json > ...
1 {
2   "python.formatting.provider": "black",
3   "[python]": {
4     "editor.formatOnSave": true
5   }
6 }
```

VS Code


```
pre-commit-config.yaml x
pre-commit-config.yaml > YAML > [ ] repos
1 repos:
2 - repo: https://github.com/psf/black
3   rev: '22.10.0'
4   hooks:
5     - id: black
6     - id: black-jupyter
```

Hook



psf/black: The uncompromising Python code formatter

The uncompromising Python code formatter. Contribute to psf/black development by creating an account on GitHub.

 GitHub

"Black is the uncompromising Python code formatter. By using it, you agree to cede control over minutiae of hand-formatting. In return, Black gives you speed, determinism, and freedom from pycodestyle nagging about formatting. You will save time and mental energy for more important matters."

FLAKE8 2.4K

Stars

Flake8 supports storing its configuration in your project in one of `setup.cfg`, `tox.ini`, or `.flake8`

Configs

```
[flake8]
max-line-length = 88
extend-ignore = E203,E501,W503
exclude = .venv
```

VS Code

```
{
  "python.linting.enabled": true,
  "python.linting.lintOnSave": true,
  "python.linting.flake8Enabled": true,
  "python.linting.flake8Args": [
    "--max-line-length=88",
    "--extend-ignore=E203,E501,W503",
    "--exclude=.venv"
  ]
}
```

Hook

```
repo: https://github.com/PyCQA/flake8
rev: '5.0.4'
hooks:
- id: flake8
  args: ['--max-line-length=88', '--extend-ignore=E203,E501,W503', '--exclude=.venv']
```

PyCQA/flake8



flake8 is a python tool that glues together pycodestyle, pyflakes, mccabe, and third-party plugins to check the style and quality...

146 Contributors

298k Used by

2k Stars

248 Forks



PyCQA/flake8: flake8 is a python tool that glues together pycodestyle, pyflakes, mccabe, and third-party plugins to check...

flake8 is a python tool that glues together pycodestyle, pyflakes, mccabe, and third-party plugins to check the style and quality of some python code. - GitHub - PyCQA/flake8: flake8 is a python to...

GitHub

"Flake8 is a code linter that warns you of syntax errors, possible bugs, stylistic errors, etc."

"Flake8 is a wrapper around: PyFlakes, pycodestyle (formerly called pep8), Ned Batchelder's McCabe script."

"There are a few deviations that cause incompatibilities with black."

ISORT

5.3K

Stars

isort is able to read project-specific default values for its command line options from a `pyproject.toml` file in the `[tool.isort]` section

```
[T] pyproject.toml M X
[ ] pyproject.toml
28 build-backend = "poetry.core.masonry.api"
29
30 [tool.black]
31 line-length = 88
32
33 [tool.isort]
34 profile = 'black'
35
36 [tool.pydocstyle]
```

Configs

```
() settings.json M X
() settings.json > ...
1 {
2   "python.sortImports.args": [
3     "--profile=black"
4   ],
5   "[python]": {
6     "editor.codeActionsOnSave": {
7       "source.organizeImports": true
8     }
9   }
10 }
```


VS Code

```
pre-commit-config.yaml X
pre-commit-config.yaml > YAML > [ ] repos > {} 5 > [ ] hooks > {} 2
11 - repo: https://github.com/PyCQA/isort
12   rev: '5.10.1'
13   hooks:
14     - id: isort
15       args: ['--profile=black']
16
```

Hook

PyCQA/isort

A Python utility / library to sort imports.



235

Contributors

281k

Used by

5k

Stars


495

Forks



PyCQA/isort: A Python utility / library to sort imports.

A Python utility / library to sort imports. Contribute to PyCQA/isort development by creating an account on GitHub.

 GitHub

"isort is a Python utility / library to sort imports alphabetically, and automatically separated into sections and by type."

"Black also formats imports, but in a different way from isort defaults which leads to conflicting changes."

PYDOC STYLE

966

Stars

pyDocStyle is able to read project-specific default values for its command line options from a [pyproject.toml](#) file in the `[tool.pydocstyle]` section

```
[T] pyproject.toml M x
[T] pyproject.toml
34 profile = "black"
35
36 [tool.pydocstyle]
37 convention = "google"
38 add-ignore = "D100,D101,D102,D103,D104,D105,D106,D107"
```


Configs

```
◆ .pre-commit-config.yaml x
◆ .pre-commit-config.yaml > YAML > [ ] repos > { } 3 > [ ] repo
21 - repo: https://github.com/pycqa/pydocstyle
22   rev: '6.1.1'
23   hooks:
24   - id: pydocstyle
25     args: ['--convention=google', '--add-ignore=D100,D101,D102,D103,D104,D105,D106,D107']
```

Hook

PyCQA/**pydocstyle**

docstring style checker



79

Contributors

36k


Used by

966

Stars


182

Forks



PyCQA/pydocstyle: docstring style checker

docstring style checker. Contribute to PyCQA/pydocstyle development by creating an account on GitHub.

 GitHub

"pydocstyle is a static analysis tool for checking compliance with Python docstring conventions."

*"There are three conventions that may be used by pydocstyle: **pep257**, **numpy** and **google**".*

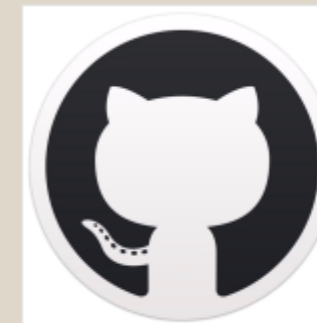
NBSTRIP OUT

957

Stars

```
.pre-commit-config.yaml X
https://github.com/kynan/nbstripout
18   rev: '0.6.1'
19   hooks:
20     - id: nbstripout
```

Hook



kynan/nbstripout: strip output from Jupyter and IPython notebooks

strip output from Jupyter and IPython notebooks. Contribute to kynan/nbstripout development by creating an account o...

GitHub

"Opens a notebook, strips its output, and writes the outputless version to the original file."

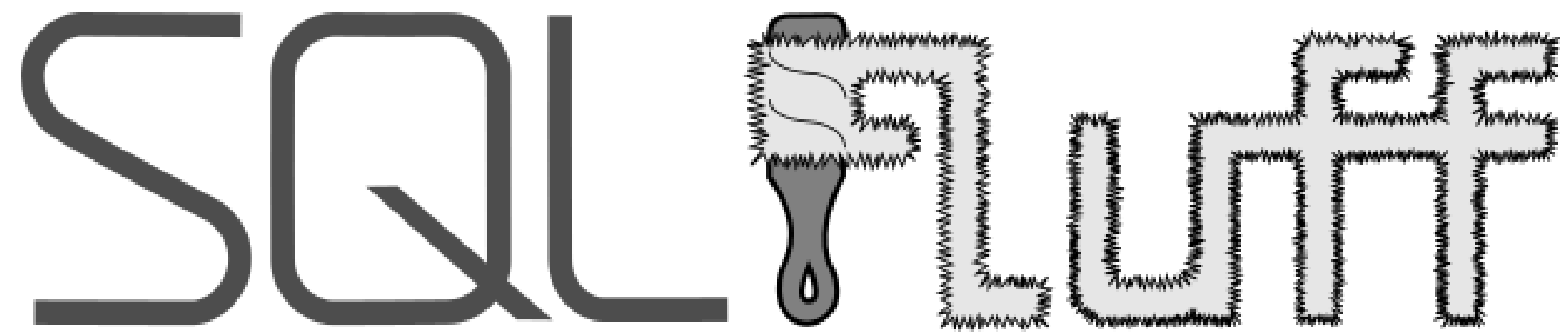
"Useful mainly as a git filter or pre-commit hook for users who don't want to track output in VCS."

"This does mostly the same thing as the Clear All Output command in the notebook UI."

SQLFLUFF

5.3K

Stars



The SQL Linter for Humans

```
[T] pyproject.toml X
Users > gabrielharris > Downloads > template > [T] pyproj
48
49 [tool.sqlfluff.core]
50 dialect = "tsql"
51 templater = "placeholder"
52 exclude_rules = "L016,L031"
53
```

Configs

```
.pre-commit-config.yaml M X
.pre-commit-config.yaml > YAML > [ ] repos
32 - repo: https://github.com/sqlfluff/sqlfluff
33   rev: '1.3.2'
34   hooks:
35   - id: sqlfluff-fix
36   - id: sqlfluff-lint
```

Hook

Demo?