

Flake8 linter 快速入门

Dezeming Family

2022 年 1 月 24 日

DezemingFamily 系列书和小册子因为是电子书，所以可以很方便地进行修改和重新发布。如果您获得了 DezemingFamily 的系列书，可以从我们的网站 [<https://dezeming.top/>] 找到最新版。对书的内容建议和出现的错误欢迎在网站留言。



前言

所谓 lint，一般意义上表示代码规范工具，来自于 C 语言的编译工具 lint。Flake8 是一款 Python 官方发布的辅助检测 Python 代码是否规范的工具。可能有不少人用过 Pylint 作为 python 代码规范工具，相比于 Pylint，flake8 的可扩展性和可开发性都很强。简单来说，flake8 包含了下面三种工具：

- PyFlakes：用于静态检查 Python 代码逻辑错误。
- PEP 8：用于静态检查 PEP 8（Python 的代码格式规范，比如如何缩进、注释等）编程风格。
- NedBatchelder's McCabe script：用于静态分析 Python 代码的复杂度。

现在有很多 IDE 都支持一些自动的检查，例如 pycharm、VSCode。我们也会选择使用和安装更简单（不依赖于平台）的 flake8。

目录

一 环境搭建与快速使用	1
二 违规类型	1
三 小结	1
参考文献	2

一 环境搭建与快速使用

安装命令如下，在 cmd 中执行。可以根据自己默认的 python 版本安装或者自己设置：

```
python -m pip install flake8
python<version> -m pip install flake8
```

Flake8 对版本的要求非常严格，要与 python 版本相对应。类似于 pip，Flake8 也是一个可执行文件，可以用 cmd 调用。

使用较为简单，cmd 进入某个目录，然后调用：

```
Flake8 .
```

就会分析这个目录下的所有 .py 文件。如果想只分析某个文件，就把 . 替换为文件名。

```
D:\Develop\Python\PSF Test\Black>flake8 .
.\BlackPlayground.py:1:1: F401 'seven_dwarfs.Grumpy' imported but unused
.\BlackPlayground.py:1:1: F401 'seven_dwarfs.Happy' imported but unused
.\BlackPlayground.py:1:1: F401 'seven_dwarfs.Sneezy' imported but unused
.\BlackPlayground.py:1:1: F401 'seven_dwarfs.Dopey' imported but unused
.\BlackPlayground.py:8:5: F821 undefined name 'very_long_variable_name'
.\BlackPlayground.py:9:9: F821 undefined name 'very_long_variable_name'
.\BlackPlayground.py:30:25: E203 whitespace before ':'
.\BlackPlayground.py:34:38: F821 undefined name 'os'
.\BlackPlayground.py:37:29: F841 local variable 'f' is assigned to but never used
```

可以看到，不同的错误或警告类型有着不同的标志，例如 F401、F821 和 E203。我们下一节介绍不同的违规类型的描述。

二 违规类型

Flake8 有一个我们使用的违规类 (violation classes) 的默认列表。这份名单是 C90 [5]。

- 所有 E 类违规都是 pycodestyle 报告的“错误”，也就是代码规范性问题（可以不用处理）。
- 所有 F 类违规都是 pyflakes 报告的。在 [4] 中有各种 F 类违规的错误类型。
- 所有 W 类违规都是 pycodestyle 报告的“警告”（可以不用处理）。

我们用户可以自定义来屏蔽哪些警告或错误，具体方法可以自行查阅 [5]。

三 小结

关于 Flake8 的基本介绍就介绍到这里，我们只需要了解这些，就可以应付不少需求了。本文开始写于在学习 miniTorch 时介绍的基础代码工具，因此暂时不再进行深究。

有趣的是，我用 Black 规范化后的代码，在使用 Flake8 检查时会检查出不少 E 类错误。实际使用时我们需要手动屏蔽掉一些不需要的报错类型，否则看起来会很乱。

参考文献

- [1] <https://pypi.org/project/flake8/>
- [2] <https://pypi.org/project/flake8/#description>
- [3] <https://flake8.pycqa.org/en/latest/index.html#quickstart>
- [4] <https://flake8.pycqa.org/en/latest/user/error-codes.html>
- [5] <https://flake8.pycqa.org/en/latest/user/violations.html>