


# VSCode Python 开发中的 Formatting 与 Linting

## 一、什么是 formatter 和 linter?

编写规范而美观易读的代码是提高代码质量的第一步，我们可以使用 formatter 和 linter 来帮助我们做到这一点。

Formatter 通过应用特定的规则和约定，例如行距、缩进和运算符周围的间距，使代码在视觉上变得更加有条理和易于理解，保持一致的风格。Formatter 并不会改变代码本身的功能。



```
1 import numpy as np
2 import math
3
4 s= int( input('Enter the size: '))
5 img = np.random.random((s,s,3))
6 b=img/ math.pi
7 print(f"Max value: {np.max(img )}")
```

```
1 import math
2
3 import numpy as np
4
5 s = int(input("Enter the size: "))
6 img = np.random.random((s, s, 3))
7 b = img / math.pi
8 print(f"Max value: {np.max(img )}")
```

Linter 通过分析代码，会高亮出代码中的常见语法、风格和功能错误以及非常规的编程实践，以此来减少代码编写出错的可能性。例如，linter 可以检测未定义变量的使用、对未定义函数的调用、缺少括号。尽管 linting 和 formatting 之间有一点重叠，但这两种功能是互补的。

导入包中的全部内容（不推荐这样做，这样其他人在阅读代码时不清楚 `sin` 是从哪里来的）

```
1 from math import *
2
3 sin(pi / 2)
```

问题 16 输出 终端 端口 JUPYTER 调试控制台 筛选器(例如 text、\*\*/\*.ts、!\*\*/\*)

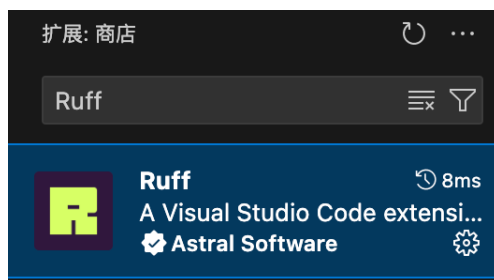
Python基础知识教程 (加长版).ipynb 10

- ⚠ Redefinition of unused ``f`` from cell 85, line 2 Ruff(F811) [行 1, 列 5]
- ⚠ Redefinition of unused ``cos`` from cell 111, line 2 Ruff(F811) [行 6, 列 23]
- 💡 ``from math import *`` used; unable to detect undefined names Ruff(F403) [行 1, 列 1]
- ⚠ ``pi`` may be undefined, or defined from star imports Ruff(F405) [行 3, 列 5]
- ⚠ Redefinition of unused ``B`` from cell 115, line 1 Ruff(F811) [行 9, 列 7]
- ⚠ Comparison to ``True`` should be ``cond is True`` or ``if cond:`` Ruff(E712) [行 1, 列 9]

## 二、VSCode 中 formatter 和 linter 的安装与配置

在本篇教程中，我们使用 Ruff 作为 python 的 formatter 与 linter (Ruff 包含了这两者的功能)。当然，你也可以参考文末的两个链接，自行选择喜欢的 formatter 与 linter。

Ruff 的安装很简单，只用在 VSCode 的扩展商店搜索“Ruff”并安装对应的扩展即可：



安装完后，需要稍作配置。点击 VSCode 左下角的齿轮按钮，在弹出的菜单中点击“设置”。



然后点击右上角的“打开设置 (JSON)”按钮。



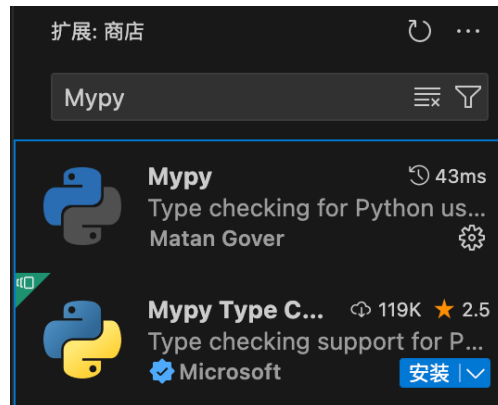
在打开的编辑器中增加下面这一段内容（如果已经存在一部分内容则进行合并）：

```
"[python]": {
  "editor.tabSize": 4,
  "editor.defaultFormatter": "charliermarsh.ruff",
  "editor.formatOnSave": true,
  "editor.codeActionsOnSave": {
    "source.organizeImports": "explicit",
  },
},
```

保存该配置文件，大功告成！

### 三、(可选) 安装 Type Checker

因为 Ruff 本身不包含类型检查功能, 所以可以再额外安装互补的 type checker, 例如 Mypy。应用市场的两个扩展都可以选择, 只不过非官方版需要事先额外安装全局的 mypy 包。



附: VSCode 中关于 formatting 和 linting 的官方文档

- [Formatting Python in Visual Studio Code](#)
- [Linting Python in Visual Studio Code](#)