

目录

前言	1.1
VSCode简介	1.2
界面布局	1.3
常用快捷键	1.4
代码编辑器常用功能	1.5
代码格式化	1.5.1
语法高亮	1.5.2
文件编码	1.5.3
搜索	1.5.4
普通搜索	1.5.4.1
正则搜索	1.5.4.2
VSCode的智能好用之处	1.6
Git代码管理	1.7
集成终端	1.8
插件	1.9
插件市场	1.9.1
常用插件	1.9.2
Markdown Preview Enhanced	1.9.2.1
indent-rainbow	1.9.2.2
Bracket Pair Colorizer	1.9.2.3
Paste Image	1.9.2.4
文件图标主题	1.9.2.5
调试代码	1.10
调试Python	1.10.1
性能	1.11
附录	1.12
参考资料	1.12.1

史上最好用的编辑器：VSCode

- 最新版本：`v0.5`
- 更新时间：`20190316`

简介

介绍史上用过的最好用的编辑器 `vscode`，以及各种好用的功能，包括但不限于常用快捷键，调试Python，常用插件，搜索功能，包括正则搜索，文件编码，git仓库代码管理，各种主题字体和插件，强大好用的插件市场，代码格式化等等各种好用的功能，且性能优越。是目前最值得推荐的编辑器，没有之一。

源码+浏览+下载

本书的各种源码、在线浏览地址、多种格式文件下载如下：

Gitbook源码

- [crifan/best_editor_vscode 史上最好用的编辑器：VSCode](#)

如何使用此Gitbook源码去生成发布为电子书

详见：[crifan/gitbook_template: demo how to use crifan gitbook template and demo](#)

在线浏览

- [史上最好用的编辑器：VSCode book.crifan.com](#)
- [史上最好用的编辑器：VSCode crifan.github.io](#)

离线下载阅读

- [史上最好用的编辑器：VSCode PDF](#)
- [史上最好用的编辑器：VSCode ePUB](#)
- [史上最好用的编辑器：VSCode MOBI](#)

crifan.com, 使用[知识署名-相同方式共享4.0协议](#)发布 all right reserved, powered by Gitbook该文件修订时间：2019-03-16 15:15:36

VSCode简介

目前常见的 编辑器(Editor) , IDE 很多, 之前也有总结: [编辑器和IDE总结](#)。

对于代码编辑器来说, 之前用过很多, 比如:

- Win平台
 - Notepad2
 - Notepad++
 - = npp
 - 等等
- Mac平台
 - Atom
 - Sublime
 - 等等

除了Win平台中[Notepad++](#)还算不错外, 其他都不够好用

vsCode = Visual Studio Code , 是目前用过的最好用的、跨平台的、免费的、编辑器。

VSCode的优点

- 轻量级
- 界面美观
 - 代码高亮效果好
- 功能强大且易用
 - 除了 文本编辑器 该有的功能
 - 支持 插件 : 可以无限扩展额外功能
 - 加上很多插件, 几乎可以实现很多复杂的, 高级的IDE才能实现的功能
 - 比如 调试Python代码 等等
 - 本身内置支持 git
 - 便于代码版本管理
- 跨平台
 - 支持多种平台:
 - Windows
 - Mac
 - Linux
 - 等等

TODO: 增加更多介绍

crifan.com, 使用[知识署名-相同方式共享4.0协议](#)发布 all right reserved, powered by Gitbook该文件修订时间: 2019-03-16 09:19:58

界面布局

下面整理 `VSCode` 的界面和布局。

左右布局显示方便文件对比

在已经打开了一个文件：想要去和另外一个文件对比时，`Command + 鼠标点击`



即可把第二个文件放在右边，左右布局，便于对比2个文件的内容：



还是很方便的。

crifan.com, 使用[知识署名-相同方式共享4.0协议](#)发布 all right reserved, powered by Gitbook该文件修订时间：2019-03-16 14:03:28

常用快捷键

此处整理出 `vsCode` 中最常用的一些快捷键，方便查询和使用。

常用的快捷键

光标位置移动

- 鼠标/光标返回之前的位置 = 上一个位置 = 前一个位置：
 - Mac : `Control + -`
- 鼠标/光标返回之后的位置
 - Mac : `Shift + Control + -`

后来也看到了工具栏中有显示快捷键：



大小写转换

默认是没有的，可以自己去设置。比如：

- `Ctrl + Shift + l` : 转为小写
 - `l` : 表示 lowercase
- `Ctrl + Shift + u` : 转为大写
 - `u` : 表示 uppercase

详见： [【已解决】VSCode中对选中内容切换大小写的快捷键](#)

如何找到某个功能的快捷键是什么

另外，可以去搜到自己要的快捷键是什么：



然后就可以通过搜短横线 - 就可以找到对应的快捷键的功能是 上一步：



crifan.com, 使用[知识署名-相同方式共享4.0协议](#)发布 all right reserved, powered by Gitbook该文件修订时间： 2019-03-16 10:36:35

代码编辑器常用功能

支持不同主题和字体

[【记录】VSCode试试其他主题和字体和插件](#)

显示特殊的不可见的控制字符

设置显示控制字符：

Code -> 首选项 -> Settings -> 搜 editor.renderControlCharacters -> 勾选： Editor: Render Control Characters 中的 Controls whether the editor should render control characters



效果：



详见：

[【已解决】VSCode中显示特殊的不可见的控制字符](#)

其他方便好用的功能

快速在终端中打开文件夹

右键某文件夹 -» 在终端中打开



则可以直接快速调用内置终端打开对应文件夹，然后就可以继续在终端中做事情了，比如此处去用ffmpeg分割mp3。

crifan.com, 使用[知识署名-相同方式共享4.0协议](#)发布 all right reserved, powered by Gitbook该文件修订时间： 2019-03-16 15:06:35

代码格式化

crifan.com, 使用[知识署名-相同方式共享4.0协议](#)发布 all right reserved, powered by Gitbook该文件修订时间: 2019-03-07 20:15:33

语法高亮

crifan.com, 使用[知识署名-相同方式共享4.0协议](#)发布 all right reserved, powered by Gitbook该文件修订时间: 2019-03-07 20:15:52

文件编码

VSCode的文件编码应用案例

比如：

[【已解决】后台导入csv文件后解析成功但导入失败](#)

中就用到了 VSCode 中识别并转换文件的编码，效果很不错。因为可以自动根据文件的内容去推断最可能的实际的编码。

crifan.com, 使用[知识署名-相同方式共享4.0协议](#)发布 all right reserved, powered by Gitbook 该文件修订时间：2019-03-16 10:50:49

搜索

crifan.com, 使用[知识署名-相同方式共享4.0协议](#)发布 all right reserved, powered by Gitbook该文件修订时间: 2019-03-07 20:16:56

普通搜索

注意事项

有些东西搜不到是因为被默认排除了 `node_modules`

如果在VSCode中无法搜索到，确定存在的内容，则很可能是和我此处遇到的现象一样：

很多项目通过npm安装很多库到`node_modules`中，通过bower安装库到`bower_components`中，

而VSCode此处默认是通过：

```
"search.exclude": {
  "**/node_modules": true,
  "**/bower_components": true
}
```

排除了这两个文件夹到，从而导致搜索不到。

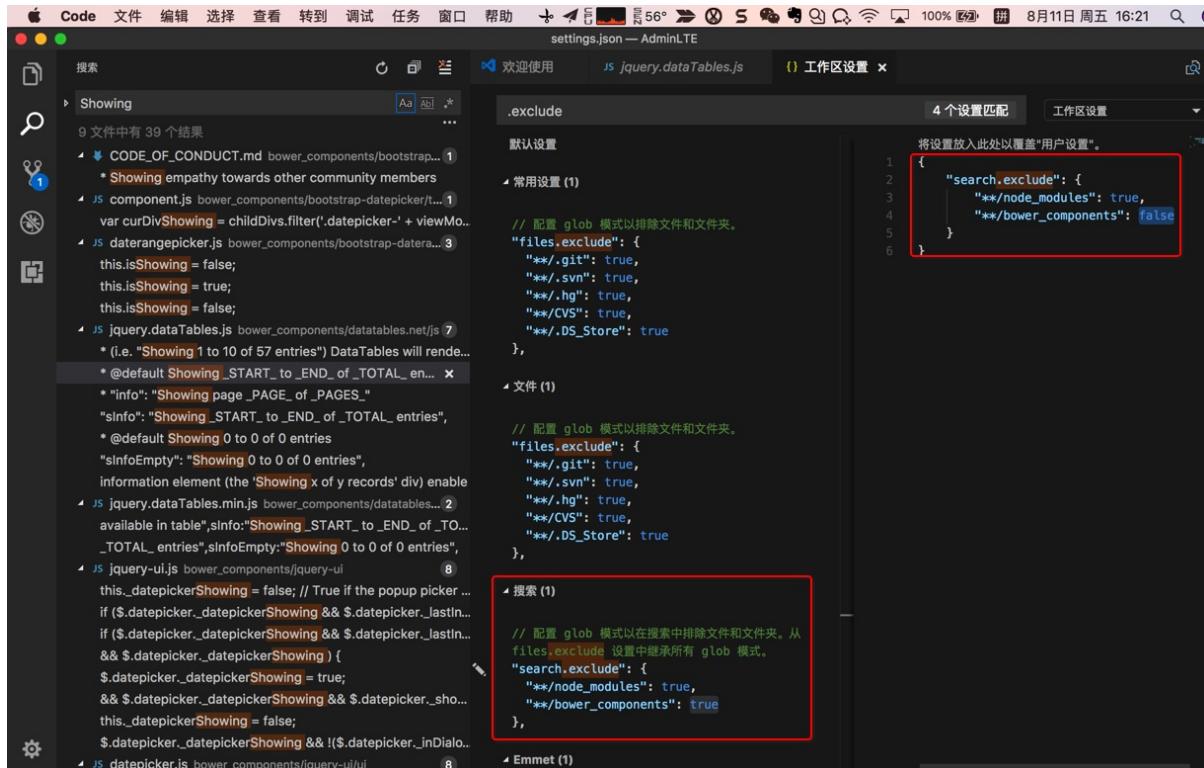
解决办法：

对于此处，取消掉`bower_components`的排除设置，改为`false`，不排除。

Code->首选项->设置->搜索：

拷贝上述的配置，放到右边到用户设置（或者是工作区设置）

把`bower_components`改为`false`即可：



详见： [【已解决】VSCode中搜索不到实际上是存在的字符串](#)

03-16 15:13:00

正则搜索

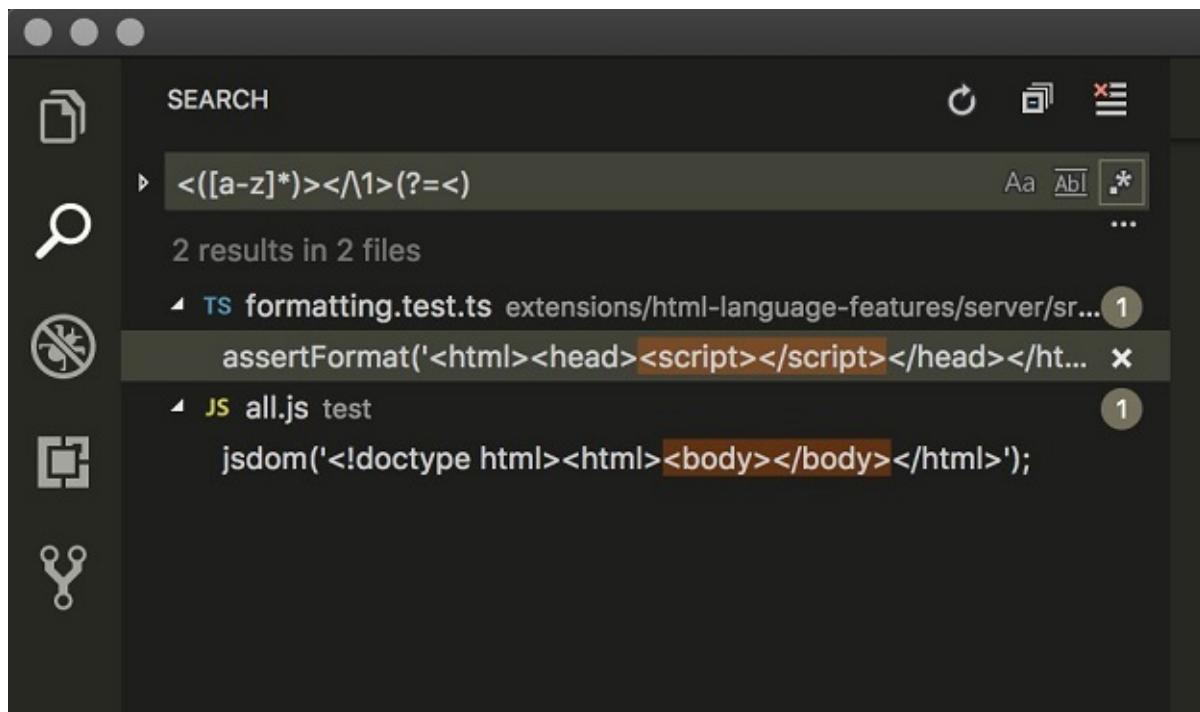
VSCode的搜索中支持正则的高级语法，比如：

- 【已解决】VSCode中如何使用正则表达式去替换且被替换中使用分组group
- 正则去替换掉特殊不可见字符：
 - 【已解决】替换掉VSCode中显示出的特殊字符：NAK

向后引用

后来新增了高级的正则搜索中的，后向引用和前向引用：[Visual Studio Code October 2018 -> Backreferences and lookahead in search](#)

效果：



以及[Multiline search](#)

所以去更新了帖子 [Visual Studio Code 的正则匹配好用吗？ - 知乎](#)的回答：

试了试其想要的效果：

```
127
126
127
```

用正则替换：

```
(\d+)\n$1,
```

The screenshot shows a code editor window titled "Untitled-1". In the search bar at the top, the regular expression `(\d+)\n` is entered. Below the search bar, the results show three matches: line 1 contains "127", line 2 contains "126", and line 3 contains "127". To the right of the search bar, there are buttons for "Aa" (case sensitive), "Ab" (regular expression), and a highlighted "Match case" button. The status bar indicates "1 / 2" matches found. The bottom part of the editor shows the replacement text "\$1,".

即可替换为：

127,126,127

This screenshot shows the same code editor window after the replacement has been applied. The search bar still displays the regular expression `(\d+)\n`. The results pane now shows the replaced text: "127,126,127". The status bar now shows "无结果" (No results). The bottom part of the editor shows the replacement text "\$1,".

crifan.com, 使用[知识署名-相同方式共享4.0协议](#)发布 all right reserved, powered by Gitbook该文件修订时间: 2019-03-16 15:07:43

VSCode的智能好用之处

自动更新import语句

当从已有的一个文件，复制并改名后，VSCode可以自动检测出来，提示是否需要更新import语句



即可自动更新import语句：



不过，此处更新后的内容不是我要的，竟然把本来正常的都改错误了，所以还要自己去改回来。以后慎用这个自动修改import的功能。

快速跳转文件

Command + P 后，输入（部分）文件名（支持模糊搜索）：



选中回车即可跳转文件：



log日志中点击文件路径可以跳转到该文件

比如调试期间出错了，点击对应log错误日志中的文件路径：

```
File "/usr/local/lib/python3.6/site-packages/rest_framework/views.py", line 483, in dispatch
    response = self.handle_exception(exc)
File "/usr/local/lib/python3.6/site-packages/rest_framework/views.py", line 443, in handle_exception
    self.raise_uncaught_exception(exc)
File "/usr/local/lib/python3.6/site-packages/rest_framework/views.py", line 480, in dispatch
    response = handler(request, *args, **kwargs)
File "/Users/crifan/dev/dev_root/company/naturling/projects/xxx/server/xxx/apps/script/views.py", line 136, in create
    if i['type'] == '0':
TypeError: string indices must be integers
```



可以跳转到对应的文件：



crifan.com, 使用[知识署名-相同方式共享4.0协议](#)发布 all right reserved, powered by Gitbook该文件修订时间： 2019-03-16 14:44:43

Git代码管理

VSCODE中用git管理代码仓库



然后尝试用VSCODE中的git界面去提交代码：



会有提示：



不过去 `git add` 后：

```
→ NaturlingCmsWeb git:(master) ✘ git add *
The following paths are ignored by one of your .gitignore files:
node_modules
package-lock.json
Use -f if you really want to add them.
→ NaturlingCmsWeb git:(master) ✘ git status
On branch master
Your branch is up to date with 'origin/master'.

Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

    modified: README.md
    modified: package.json
```

git就显示出对应的已经添加了的改动的文件了：



暂存的更改。而 更改 中就是空了。

-》说明：暂存的更改，就是add进去的，本地临时保存的改动

```
→ NaturlingCmsWeb git:(master) ✘ git commit -m "1. update readme"
[master 31efica] 1. update readme
 2 files changed, 51 insertions(+), 10 deletions(-)
```



然后再去push：

```
→ NaturlingCmsWeb git:(master) git push
Counting objects: 4, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (4/4), done.
Writing objects: 100% (4/4), 1.31 KiB | 1.31 MiB/s, done.
Total 4 (delta 2), reused 0 (delta 0)
remote: Powered by Gitee.com
To https://gitee.com/naturling_tech/NaturlingCmsWeb.git
 20b96a1..31efica  master -> master
```



查看文件历史版本和改动差异

如果需要对比之前某次的文件代码和当前最新代码有何改动，可以通过 `File History` 文件历史，得到我们要的效果：

比如想要对此这段代码和之前的写法有何不同：



则可以：文件中右键选择 `Git: View File History`



或Tab页中右键选择：



然后就可以看到各个历史版本了：



点击其中一个版本可以查看提交详情：



点击某次提交中的某个文件：



选择：Compare against workspace file



即可去该版本和当前最新文件，去对比内容差异：



其中左边是最新内容，右边是之前该版本的内容。

另外，如果需要还可以在查看历史版本时，根据不同条件筛选，比如根据作者：



即可只查看某人的提交的代码：



crifan.com, 使用[知识署名-相同方式共享4.0协议](#)发布 all right reserved, powered by Gitbook该文件修订时间： 2019-03-16 15:05:12

集成终端

VSCode内置了终端：

查看 -> 终端

即可打开自带集成的终端：



不用再去切换到外部的终端命令行：



直接可以方便的在一个工具界面下，同时处理不同的事情了。

另外一个例子：



在更新了git中源码后，再去上传，直接在集成终端中进行，很方便。

且支持多个终端

可以点击 加号去新增额外的终端，用于不同的用途。

实际使用场景举例：

比如此处在当前项目所在文件夹下，用git去管理代码：



然后需要切换到另外一个目录，去导入mysql，所以点击+去新建终端，处理mysql：



集成的终端中对于git的支持很好

在VSCode的集成的终端中，除了git的颜色看起来很清晰明确，且git的commit后的log中，都可以识别commit哪些文件了：



且Command+鼠标点击可以直接跳转打开对文件：



-》正好省去了我本身要去：找找刚才已经git的commit并push了哪些文件，直接在VSCode的git的log中找到并打开相应文件了，很方便。

crifan.com, 使用[知识署名-相同方式共享4.0协议](#)发布 all right reserved, powered by Gitbook该文件修订时间： 2019-03-16 14:56:59

插件

crifan.com, 使用[知识署名-相同方式共享4.0协议](#)发布 all right reserved, powered by Gitbook该文件修订时间: 2019-03-07 20:18:46

插件市场

crifan.com, 使用[知识署名-相同方式共享4.0协议](#)发布 all right reserved, powered by Gitbook该文件修订时间: 2019-03-07 20:19:20

常用插件

crifan.com, 使用[知识署名-相同方式共享4.0协议](#)发布 all right reserved, powered by Gitbook该文件修订时间: 2019-03-07 20:19:50

Markdown Preview Enhanced

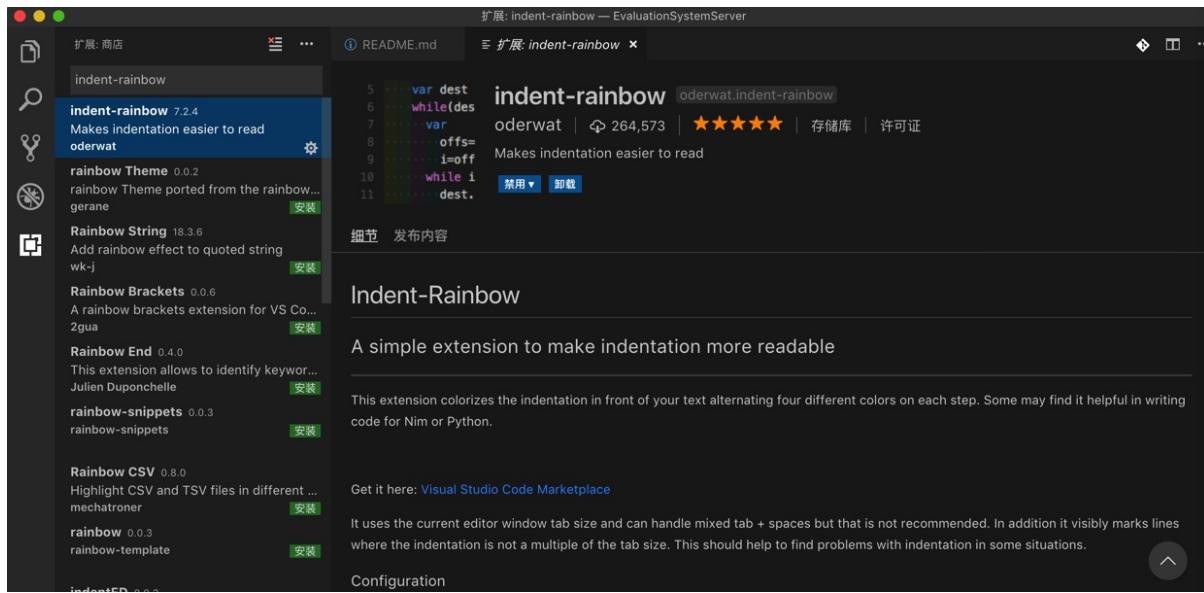
crifan.com, 使用[知识署名-相同方式共享4.0协议](#)发布 all right reserved, powered by Gitbook该文件修订时间: 2019-03-07 20:20:09

indent-rainbow

插件地址：[indent-rainbow - Visual Studio Marketplace](#)

作用：高亮（每行代码之前的）缩进

安装 indent-rainbow：



效果：

- 对于 Python 代码效果：更方便清晰的看到代码的缩进

◦

- 对于 HTML 效果：方便看清是哪个元素

- - 对于缩进层次较多的：效果尤其明显

Bracket Pair Colorizer

crifan.com, 使用[知识署名-相同方式共享4.0协议](#)发布 all right reserved, powered by Gitbook该文件修订时间: 2019-03-07 20:20:46

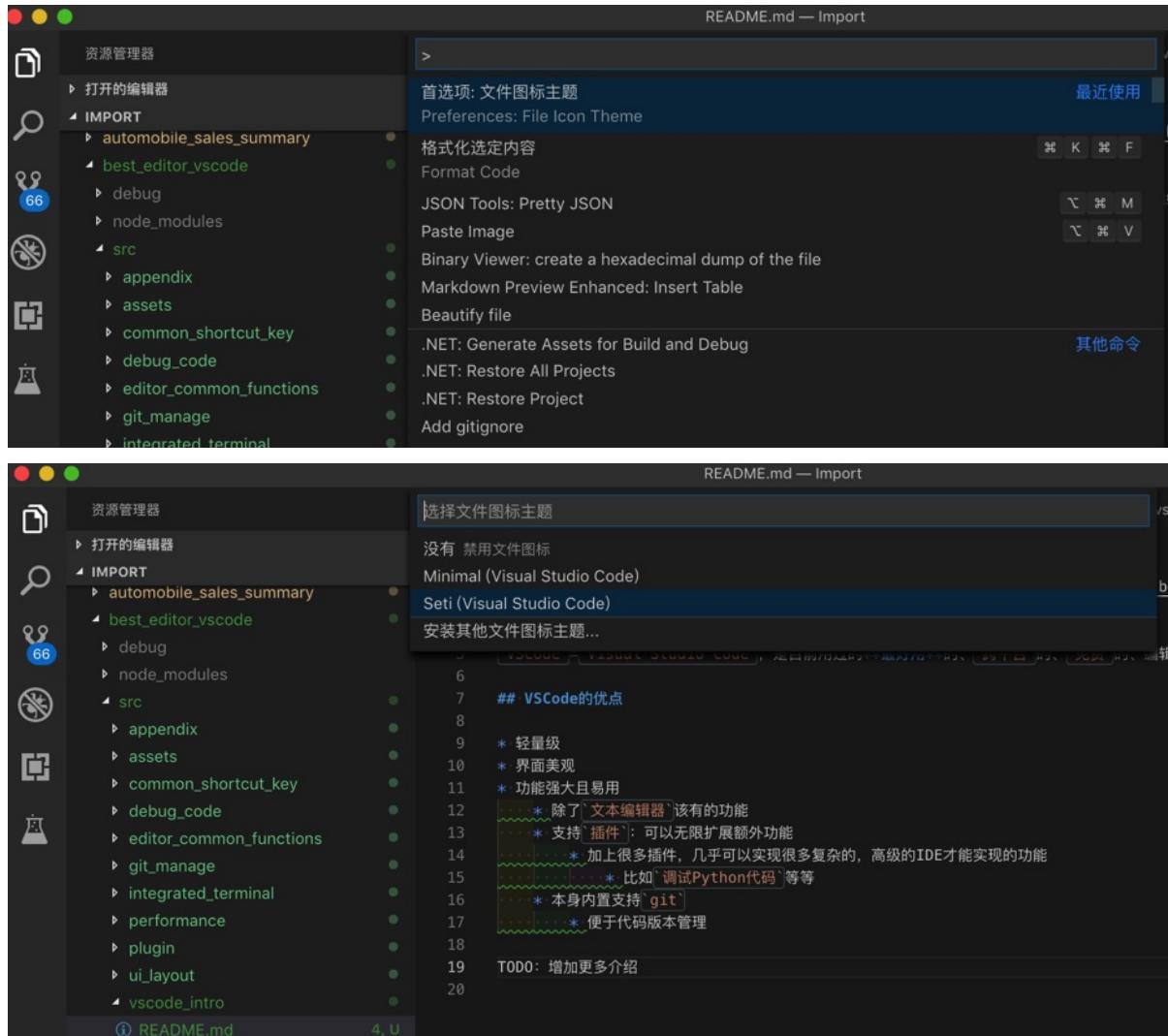
Paste Image

crifan.com, 使用[知识署名-相同方式共享4.0协议](#)发布 all right reserved, powered by Gitbook该文件修订时间: 2019-03-07 20:21:07

文件图标主题

vsCode 支持给文件的图标设置不同的主题。

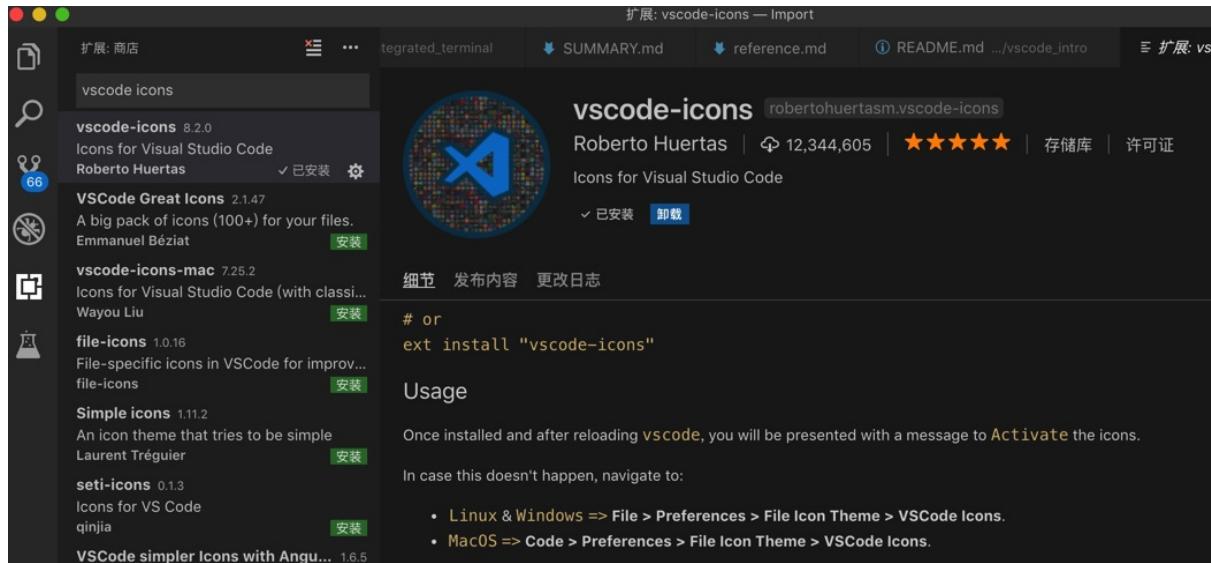
默认是 Seti 主题，效果是：



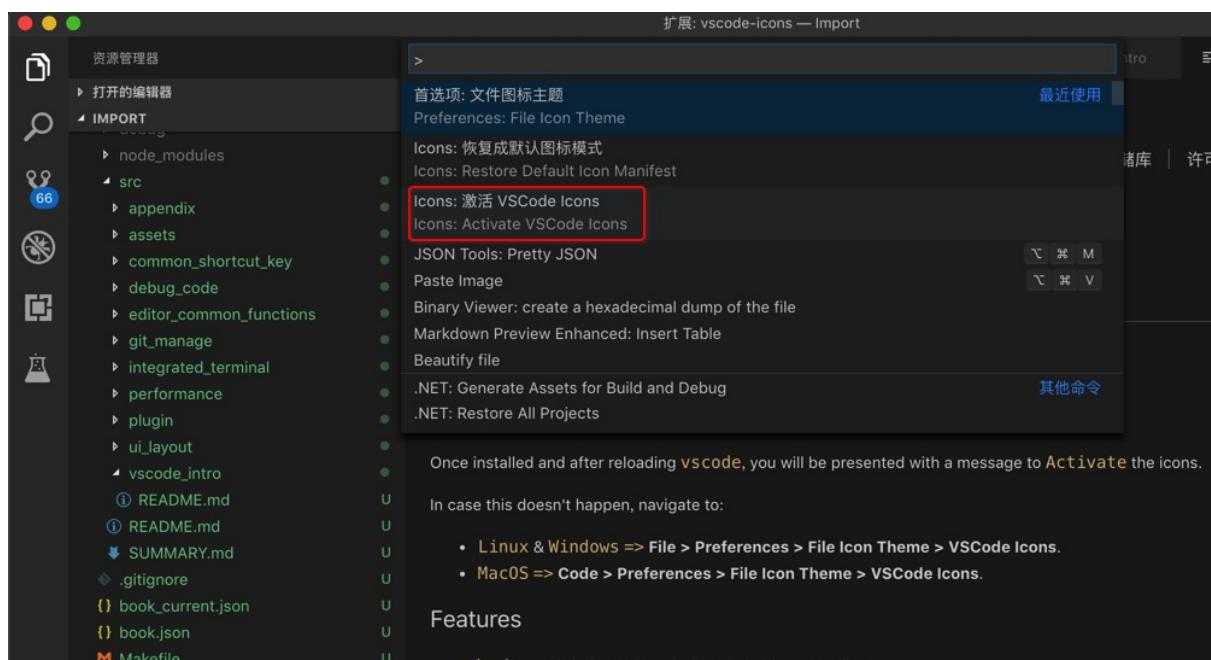
有其他新的图标的主题，比如：

[vscode-icons](#)

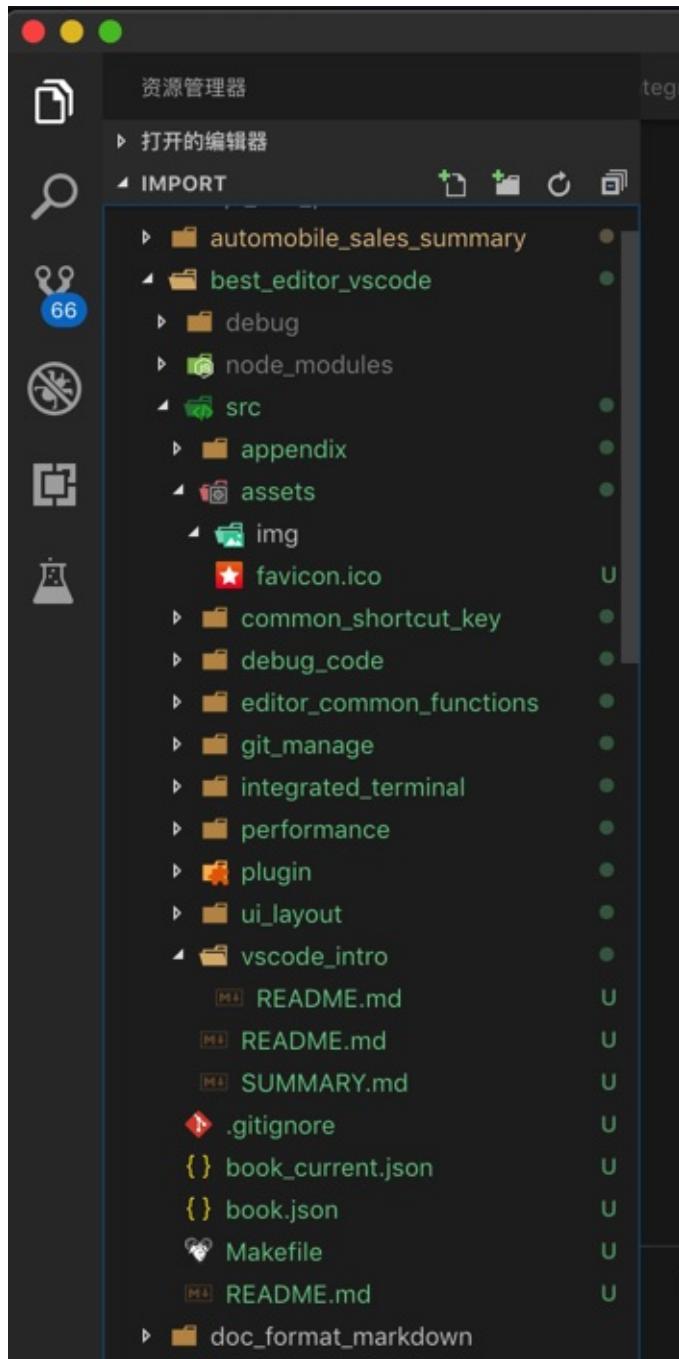
去安装：



之后再去激活：

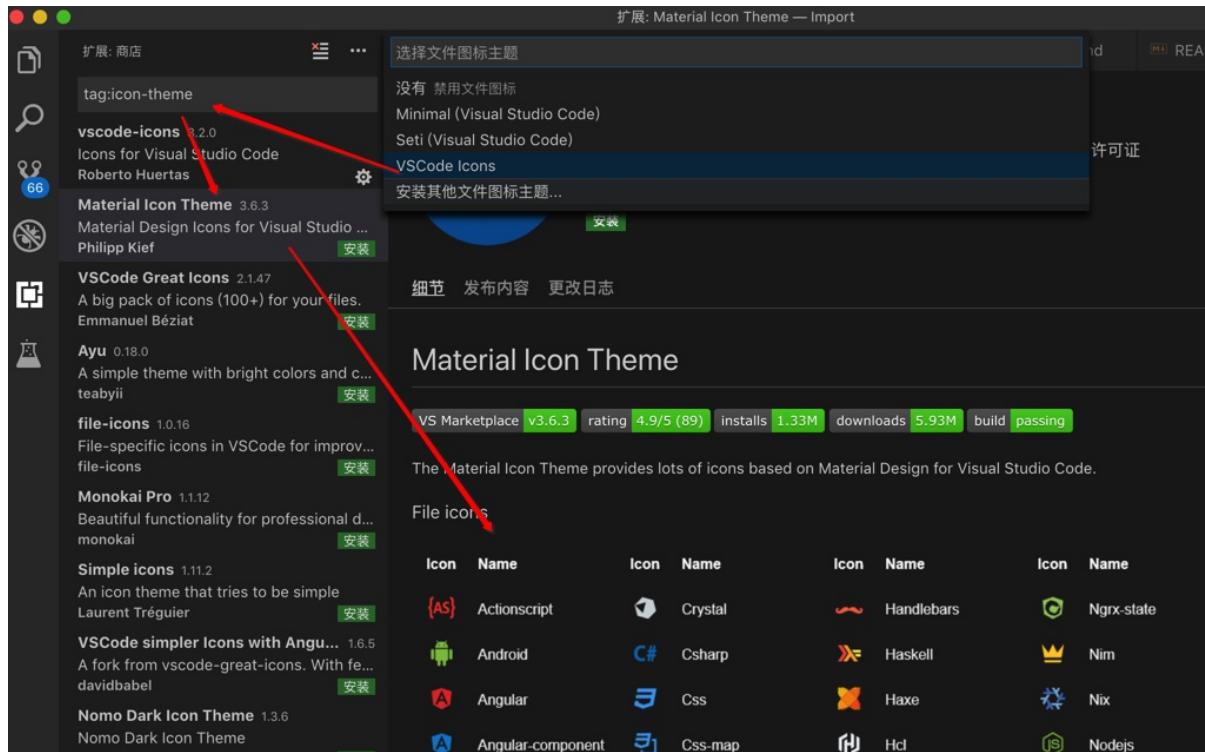


后的效果：



很不错，图标很丰富，识别度比较高。

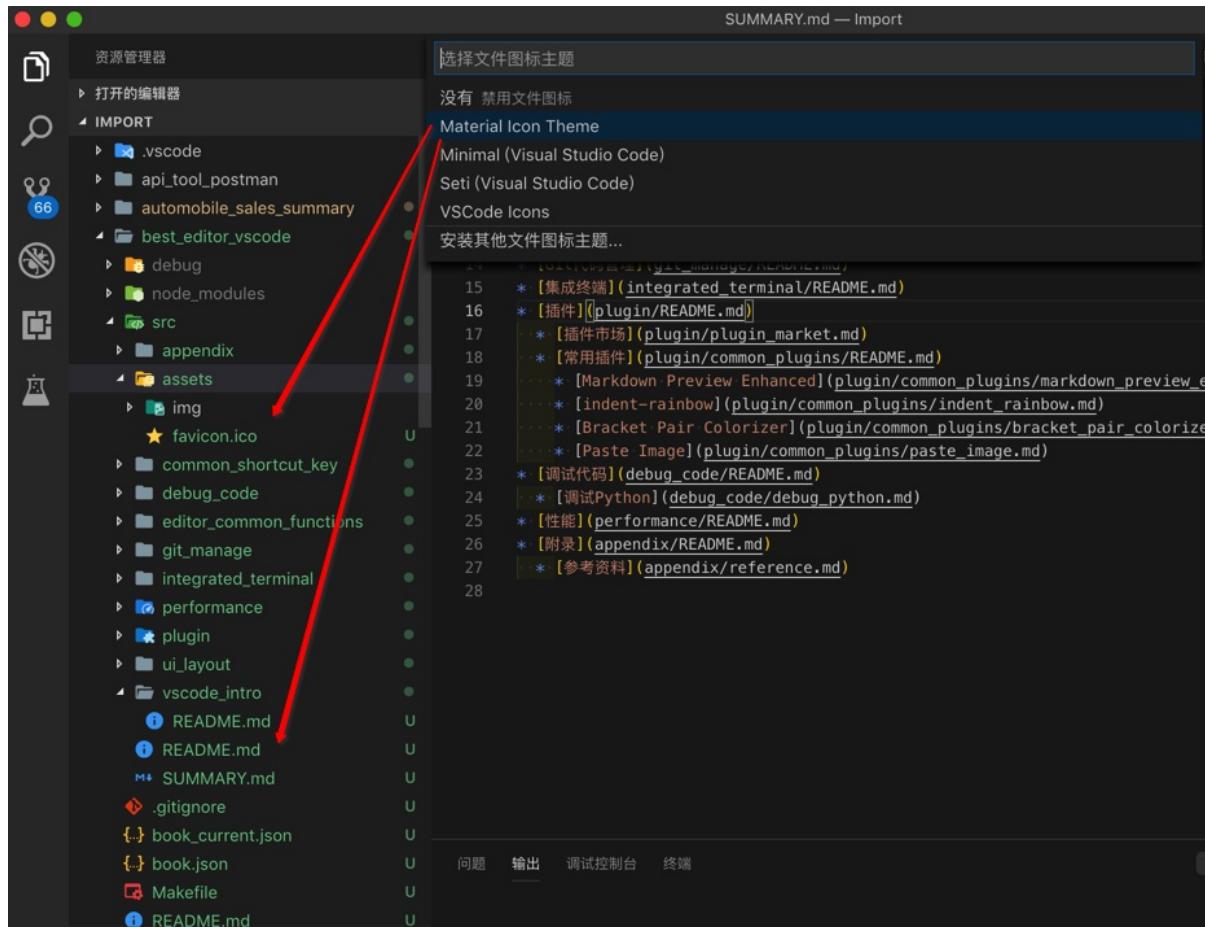
另外，可以再去根据提示去安装其他版本图标的主题：



然后去安装另外一个：

Material Icon Theme

后，去换成该主题：



感觉比前面的 `vscode-icons` 更好看，更现代化些。

crifan.com, 使用[知识署名-相同方式共享4.0协议](#)发布 all right reserved, powered by Gitbook该文件修订时间: 2019-03-16 09:35:48

调试代码

crifan.com, 使用[知识署名-相同方式共享4.0协议](#)发布 all right reserved, powered by Gitbook该文件修订时间: 2019-03-07 20:21:33

调试Python

VSCode支持调试Python代码：

之前去试了：

[【已解决】VSCode中调试Python代码](#)

[【记录】用VSCode开发和调试Python](#)

效果很好。

设置Python解析器

手动设置选择Python版本

调试之前，先要设置Python解析器。

对于python代码中：

```
#!/usr/bin/python  
# -*- coding: utf-8 -*-
```

其可以自动检测出当前的第一行的解析器，然后显示出 `Set as interpreter`，如果点击了：



则就可以从当前的解析器：



换成代码中设置的解析器了：



选择虚拟环境的Python版本

在用

```
pipenv install
```

安装了虚拟环境的python项目后，重新用VSCode打开该项目，会提示你

You have selected the macOS system install of Python, which is not recommended for use with the Python extension. Some functionality will be limited, please select a different interpreter.



然后点击 `Select Python Interpreter`，去选择刚装好的虚拟环境中的python：



选择后，左下角就可以显示出当前所选Python了：



鼠标移动上去可以查看变量值

后来又多次使用VSCode去调试Python：



鼠标移动到变量（类）的属性上，支持（直接）显示变量的属性的值：

比如datetime



的microsecond：



而这类功能，之前只有比较高级的IDE，比如Visual Studio，PyCharm等才支持。

支持异常信息的显示

当调试代码时发生异常，则可以方便快速的显示异常堆栈错误信息：



且可以点击左下角的调用堆栈，调转到对应代码位置：



注意事项

偶尔出现小问题，可参考：

调试适配器进程意外终止

[【已解决】VSCode调试Python出错：调试适配器进程意外终止Cannot read property 'style' of undefined](#)

文件内容改动后断点错位

有时候会遇到：当前面新增行后，后面的断点位置都移动了，错位了。

比如此处，前面新增了2行，导致后面的，之前设置的断点，都错位了，无效了：



只能再：去掉之前断点，重新打断点：



所以还是有点点麻烦的

-» 而PyCharm就可以很好的支持：当代码改动（不多）的时候，可以自动保持原有的断点的位置。

-» 不过后来也发现，此问题只是偶尔发生的。有时候代码改动后，断点还正常的。

性能

crifan.com, 使用[知识署名-相同方式共享4.0协议](#)发布 all right reserved, powered by Gitbook该文件修订时间: 2019-03-07 20:22:13

附录

下面列出相关参考资料。

crifan.com, 使用[知识署名-相同方式共享4.0协议](#)发布 all right reserved, powered by Gitbook该文件修订时间: 2019-03-02 22:22:58

参考资料

- 这就是我想要的 VSCode 插件! - 知乎
- [VSCode插件推荐] Bracket Pair Colorizer: 为代码中的括号添上一抹亮色 - 知乎
- 【已解决】VSCode调试Python出错：调试适配器进程意外终止Cannot read property 'style' of undefined
- 30个极大提高开发效率的VSCode插件 - 知乎
- Debugger for Chrome - Visual Studio Marketplace
- 【已解决】VSCode中对选中内容切换大小写的快捷键
- Visual Studio Code October 2018
- Backreferences and lookahead in search
- Multiline search
- Visual Studio Code 的正则匹配好用吗? - 知乎
- 【已解决】后台导入csv文件后解析成功但导入失败
- 【记录】VSCode试试其他主题和字体和插件
- 【已解决】VSCode中搜索不到实际上是存在的字符串

crifan.com, 使用[知识署名-相同方式共享4.0协议](#)发布 all right reserved, powered by Gitbook该文件修订时间: 2019-03-16 15:13:20