

一、VS Code 的介绍

VS Code 的全称是 Visual Studio Code，是一款开源的、免费的、跨平台的、高性能的、轻量级的代码编辑器。它在性能、语言支持、开源社区方面，都做得很不错。

微软有两种软件：一种是 VS Code，一种是其他软件。

IDE 与 编辑器的对比

IDE 和编辑器是有区别的：

- **IDE** (Integrated Development Environment, 集成开发环境)：对代码有较好的智能提示和相互跳转，同时侧重于工程项目，对项目的开发、调试工作有较好的图像化界面的支持，因此比较笨重。比如 Eclipse 的定位就是 IDE。
- **编辑器**：要相对轻量许多，侧重于文本的编辑。比如 Sublime Text 的定位就是编辑器。再比如 Windows 系统自带的「记事本」就是最简单的编辑器。

需要注意的是，VS Code 的定位是**编辑器**，而非 IDE，但 VS Code 又比一般的编辑器的功能要丰富许多。可以这样理解：VS Code 的体量是介于编辑器和 IDE 之间。

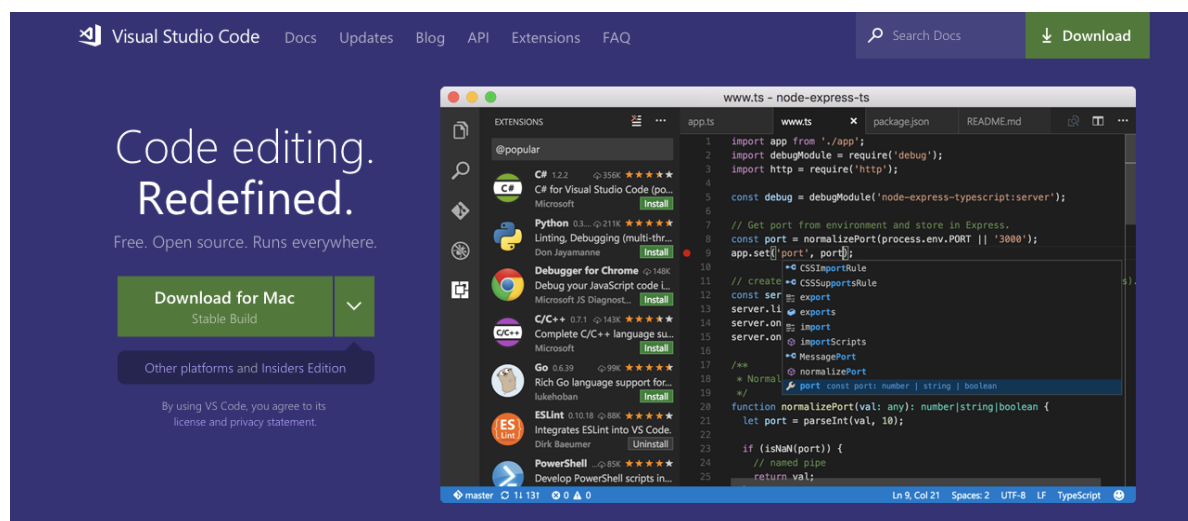
VS Code 的特点

- VS Code 的使命，是让开发者在编辑器里拥有 IDE 那样的开发体验，比如代码的智能提示、语法检查、图形化的调试工具、插件扩展、版本管理等。
- 跨平台支持 MacOS、Windows 和 Linux 等多个平台。
- VS Code 的源代码以 MIT 协议开源。
- 支持第三方插件，功能强大，生态系统完善。
- VS Code 自带了 JavaScript、TypeScript 和 Node.js 的支持。也就是说，你在书写 JS 和 TS 时，是自带智能提示的。当然，其他的语言，你可以安装相应的**扩展包**插件，也会有智能提示。

VS Code 的安装

- VS Code 官网：<https://code.visualstudio.com>

VS Code 的安装很简单，直接去官网下载安装包，然后双击安装即可。



上图中，直接点击 download，一键下载安装即可。

二、VS Code 快捷键

VS Code 用得熟不熟，首先就看你是否会用快捷键。以下列出的内容，都是常用快捷键，而加粗部分的快捷键，使用频率则非常高。

1、工作区快捷键

Mac 快捷键	Win 快捷键	作用	备注
Cmd + Shift + P	Ctrl + Shift + P	显示命令面板	
Cmd + B	Ctrl + B	显示/隐藏侧边栏	很实用
Cmd + \	Ctrl + \	创建多个编辑器	【重要】抄代码利器
Cmd + 1、2	Ctrl + 1、2	聚焦到第 1、第 2 个编辑器	同上重要
cmd +/-	Ctrl +/-	将工作区放大/缩小（包括代码字体、左侧导航栏）	在投影仪场景经常用到
Cmd + J	Ctrl + J	显示/隐藏控制台	
Cmd + Shift + N	Ctrl + Shift + N	重新开一个软件的窗口	很常用
Cmd + Shift + W	Ctrl + Shift + W	关闭软件的当前窗口	
Cmd + N	Ctrl + N	新建文件	
Cmd + W	Ctrl + W	关闭当前文件	

2、跳转操作

Mac 快捷键	Win 快捷键	作用	备注
Cmd + `	没有	在同一个软件的 多个工作区 之间切换	使用很频繁
Cmd + Option + 左右方向键	Ctrl + Pagedown/Pageup	在已经打开的 多个文件 之间进行切换	非常实用
Ctrl + Tab	Ctrl + Tab	在已经打开的多个文件之间进行跳转	不如上面的快捷键快
Cmd + Shift + O	Ctrl + shift + O	在当前文件的各种 方法 之间进行跳转	
Ctrl + G	Ctrl + G	跳转到指定行	
Cmd+Shift+\	Ctrl+Shift+\	跳转到匹配的括号	

3、移动光标

Mac 快捷键	Win 快捷键	作用	备注
方向键	方向键	在 单个字符 之间移动光标	大家都知道
option + 左右方向键	Ctrl + 左右方向键	在 单词 之间移动光标	很常用
Cmd + 左右方向键	Fn + 左右方向键	在 整行 之间移动光标	很常用
Cmd + ←	Fn + ← (或 Win + ←)	将光标定位到当前行的最左侧	很常用
Cmd + →	Fn + → (或 Win + →)	将光标定位到当前行的最右侧	很常用
Cmd + ↑	Ctrl + Home	将光标定位到文章的第一行	
Cmd + ↓	Ctrl + End	将光标定位到文章的最后一行	
Cmd + Shift + \		在 代码块 之间移动光标	

4、编辑操作

Mac 快捷键	Win 快捷键	作用	备注
Cmd + Enter	Ctrl + Enter	在当前行的下方新增一行，然后跳至该行	即使光标不在行尾，也能快速向下插入一行
Cmd+Shift+Enter	Ctrl+Shift+Enter	在当前行的上方新增一行，然后跳至该行	即使光标不在行尾，也能快速向上插入一行
Option + ↑	Alt + ↑	将代码向上移动	很常用
Option + ↓	Alt + ↓	将代码向下移动	很常用
Option + Shift + ↑	Alt + Shift + ↑	将代码向上复制	
Option + Shift + ↓	Alt + Shift + ↓	将代码向下复制	写重复代码的利器

5、多光标编辑

Mac 快捷键	Win 快捷键	作用	备注
Cmd + Option + 上下键	Ctrl + Alt + 上下键	在连续的多列上，同时出现光标	
Option + 鼠标点击任意位置	Alt + 鼠标点击任意位置	在任意位置，同时出现光标	
Option + Shift + 鼠标拖动	Alt + Shift + 鼠标拖动	在选中区域的每一行末尾，出现光标	
Cmd + Shift + L	Ctrl + Shift + L	在选中文本的所有相同内容处，出现光标	

其他的多光标编辑操作：

- 选中某个文本，然后反复按住快捷键「**Cmd + D**」键（windows 用户是按住「**Ctrl + D**」键），即可将全文中相同的词逐一加入选择。
- 选中一堆文本后，按住「**Option + Shift + i**」键（windows 用户是按住「**Alt + Shift + I**」键），既可在**每一行的末尾**都创建一个光标。

6、删除操作

Mac 快捷键	Win 快捷键	作用	备注
Cmd + shift + K	Ctrl + Shift + K	删除整行	「Cmd + X」的作用是剪切，但也可以删除整行
option + Backspace	Ctrl + Backspace	删除光标之前的一个单词	英文有效，很常用
option + delete	Ctrl + delete	删除光标之后的一个单词	
Cmd + Backspace		删除光标之前的整行内容	很常用
Cmd + delete		删除光标之后的整行内容	

备注：上面所讲到的移动光标、编辑操作、删除操作的快捷键，在其他编辑器里，大部分都适用。

7、编程语言相关

Mac 快捷键	Win 快捷键	作用	备注
Cmd + /	Ctrl + /	添加单行注释	很常用
Option + Shift + F	Alt + shift + F	代码格式化	很常用
F2	F2	以重构的方式进行 重命名	改代码备
Ctrl + J		将多行代码合并为一行	Win 用户可在命令面板搜索“合并行”
Cmd +			
Cmd + U	Ctrl + U	将光标的移动回退到上一个位置	撤销光标的移动和选择

8、搜索相关

Mac 快捷键	Win 快捷键	作用	备注
Cmd + Shift + F	Ctrl + Shift + F	全局搜索代码	很常用
Cmd + P	Ctrl + P	在当前的项目工程里，全局搜索文件名	
Cmd + F	Ctrl + F	在当前文件中搜索代码，光标在搜索框里	
Cmd + G	Ctrl + Fn + F3	在当前文件中搜索代码，光标仍停留在编辑器里	很巧妙

9、自定义快捷键

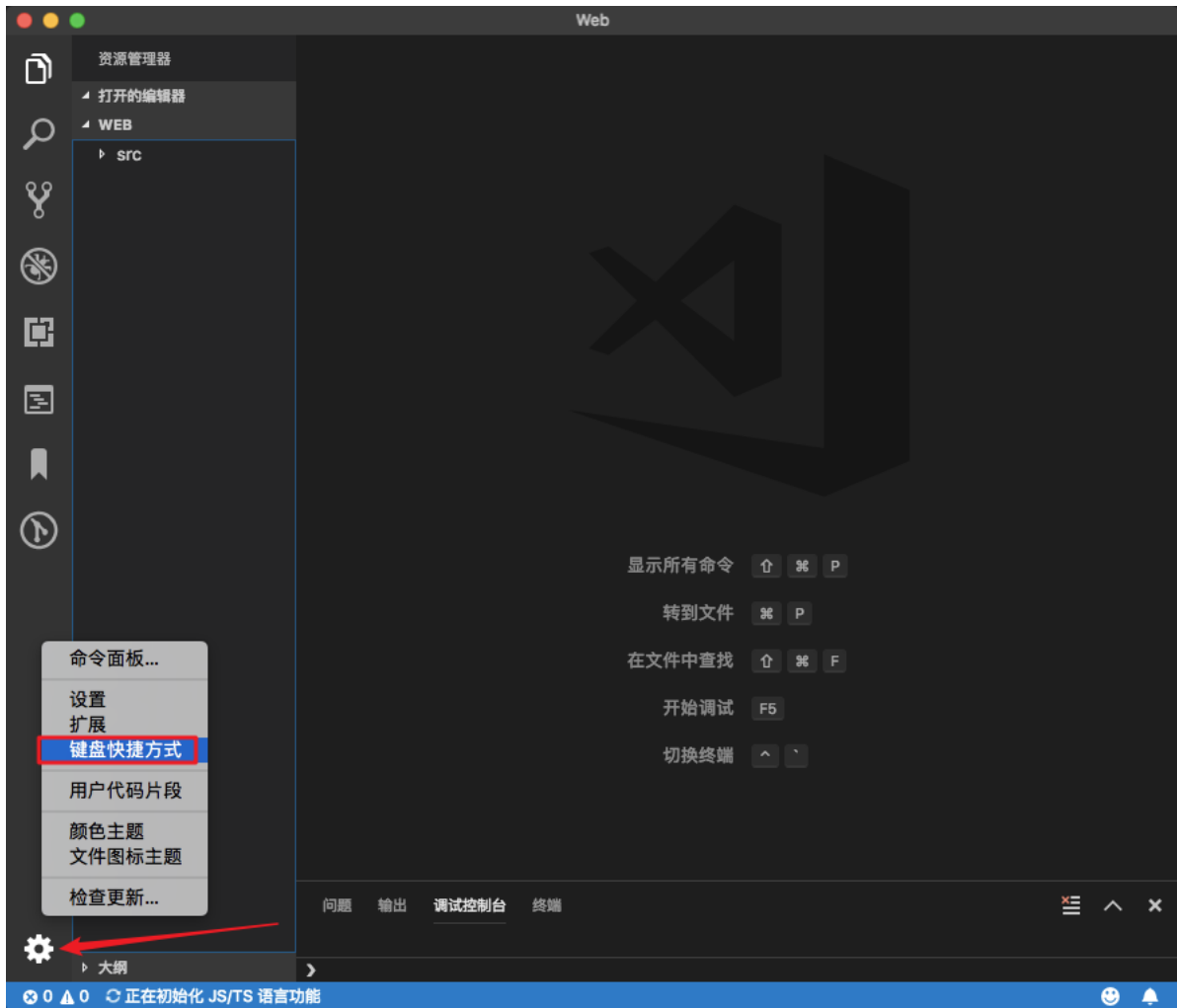
按住快捷键「Cmd + Shift + P」，弹出命令面板，在命令面板中输入“快捷键”，可以进入快捷键的设置。

当然，你也可以选择菜单栏「偏好设置 --> 键盘快捷方式」，进入快捷键的设置：

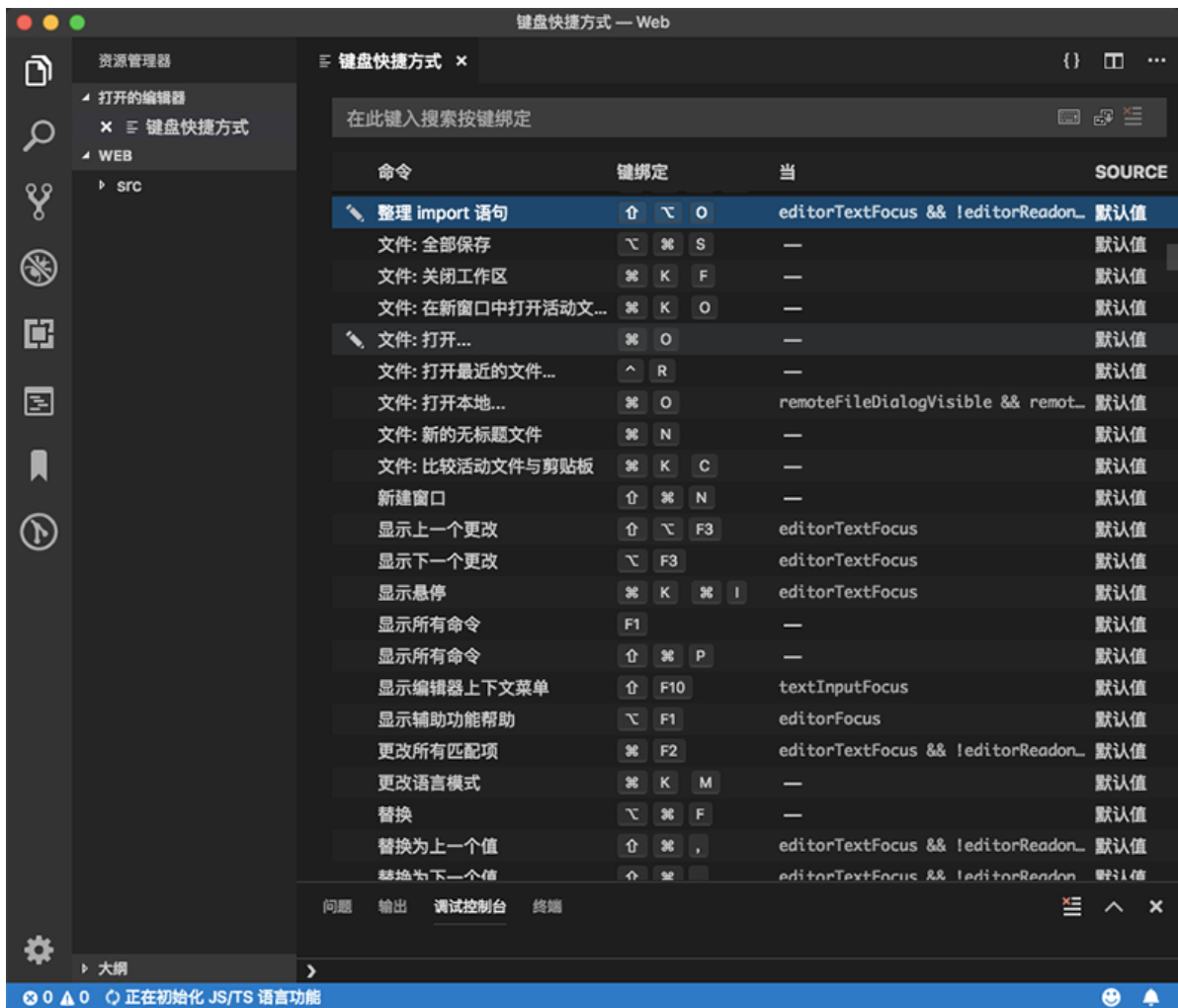


10、快捷键列表

你可以点击 VS Code 左下角的齿轮按钮，效果如下：



上图中，在展开的菜单中选择「键盘快捷方式」，就可以查看和修改所有的快捷键列表了：



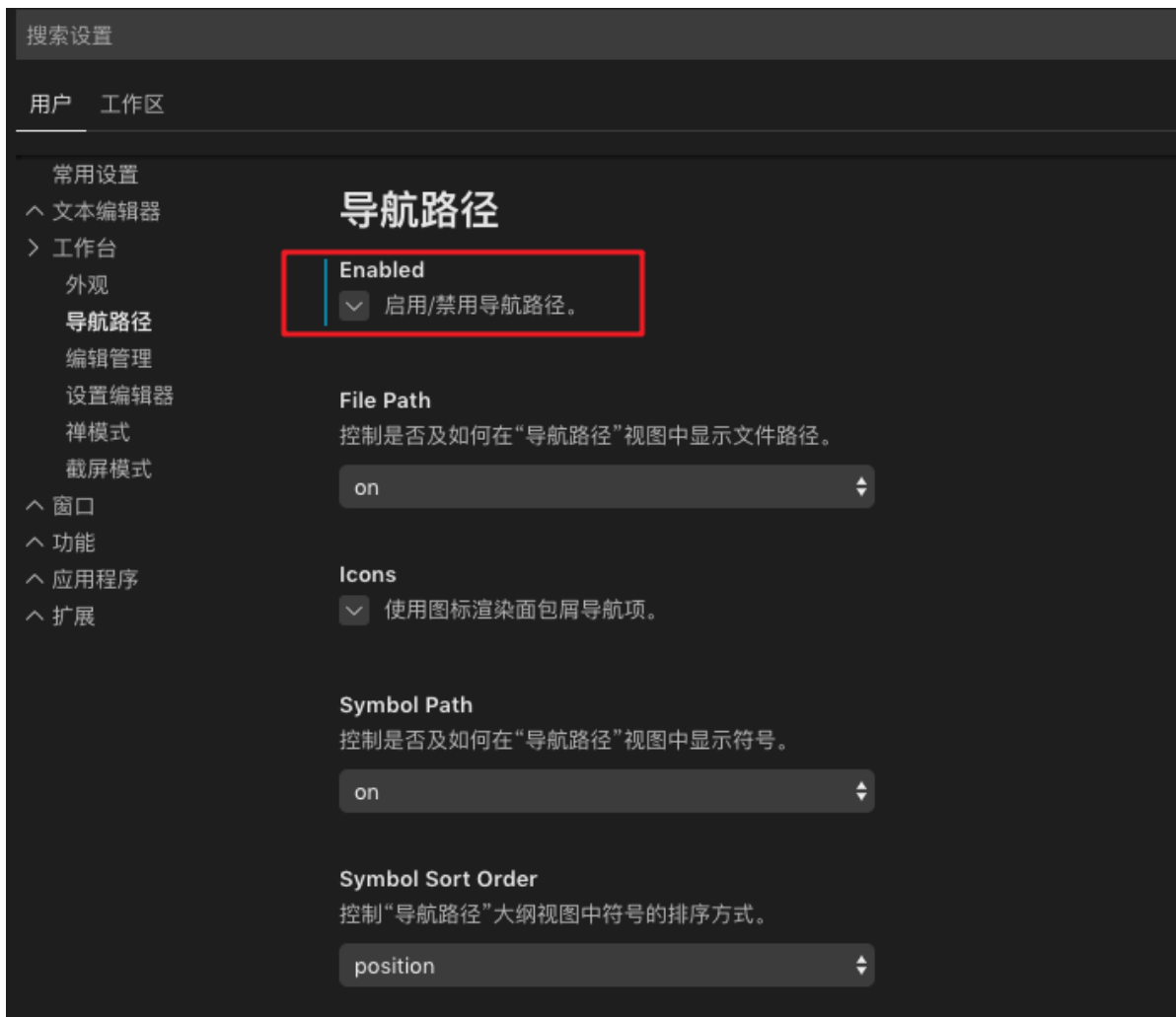
快捷键参考链接

- 快捷键速查表[官方]: <https://code.visualstudio.com/shortcuts/keyboard-shortcuts-windows.pdf>

三、私人订制：VS Code 的常见配置

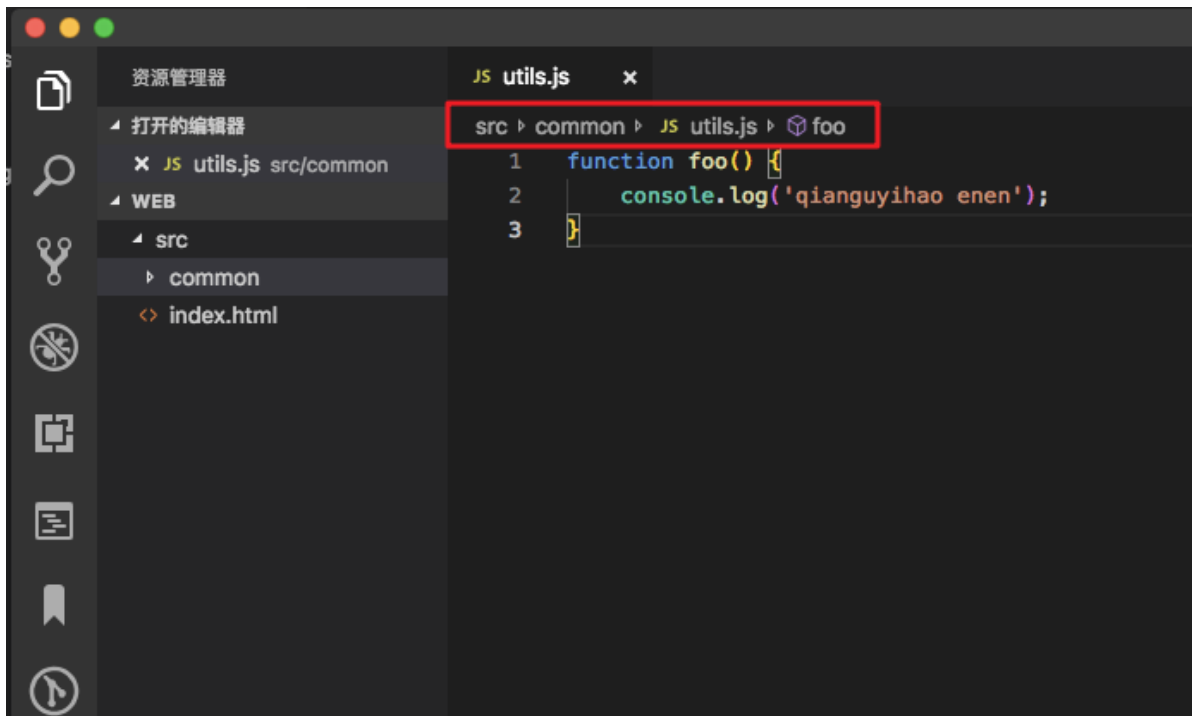
1、面包屑（Breadcrumb）

打开 VS Code 的设置项，选择「用户设置 -> 工作台 -> 导航路径」，如下图所示：



上图中，将红框部分打钩即可。

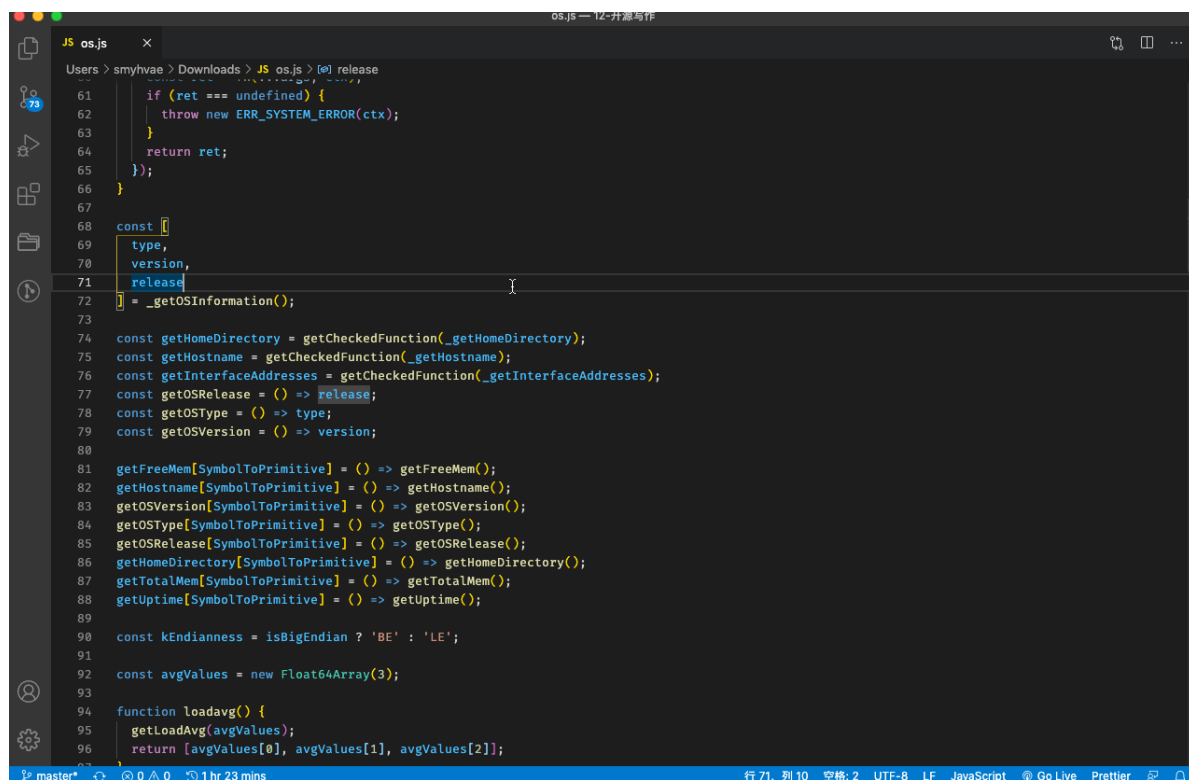
设置成功后，我们就可以查看到当前文件的「层级结构」，非常方便。如下图所示：



有了这个面包屑导航，我们可以在任意目录、任意文件之间随意跳转。

2、左右显示多个编辑器窗口（抄代码利器）

Mac 用户按住快捷键 `Cmd + \`，Windows 用户按住快捷键 `ctrl + \`，即可同时打开多个编辑器窗口，效果如下：



按快捷键「`Cmd + 1`」切换到左边的窗口，按快捷键「`Cmd + 2`」切换到右边的窗口。随时随地，想切就切。

学会了这一招，以后抄代码的时候，leader 再也不用担心我抄得慢了，一天工资到手。

3、改完代码后立即自动保存

方式一：

改完代码后，默认不会自动保存。你可以在设置项里搜索 `files.autoSave`，修改配置项如下：



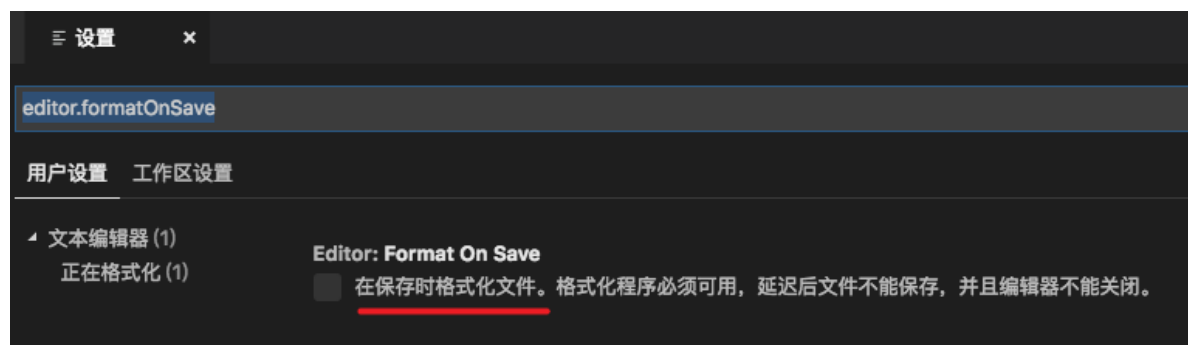
上图中，我们将配置项修改为 `onFocusChange` 之后，那么，当光标离开该文件后，这个文件就会自动保存了。非常方便。

方式二：

当然，你也可以直接在菜单栏选择「文件-自动保存」。勾选后，当你写完代码后，文件会立即实时保存。

4、保存代码后，是否立即格式化

保存代码后，默认不会立即进行代码的格式化。你可以在设置项里搜索 `editor.formatOnSave` 查看该配置项：



我觉得这个配置项保持默认就好，不用打钩。

5、在当前文件中搜索

在上面的快捷键列表中，我们已经知道如下快捷键：

- `Cmd + F` (Win 用户是 `Ctrl + F`)：在当前文件中搜索，光标在搜索框里
- `Cmd + G` (Win 用户是 `F3`)：在当前文件中搜索，光标仍停留在编辑器里

另外，你可能会注意到，搜索框里有很多按钮，每个按钮都对应着不同的功能，如下图所示：

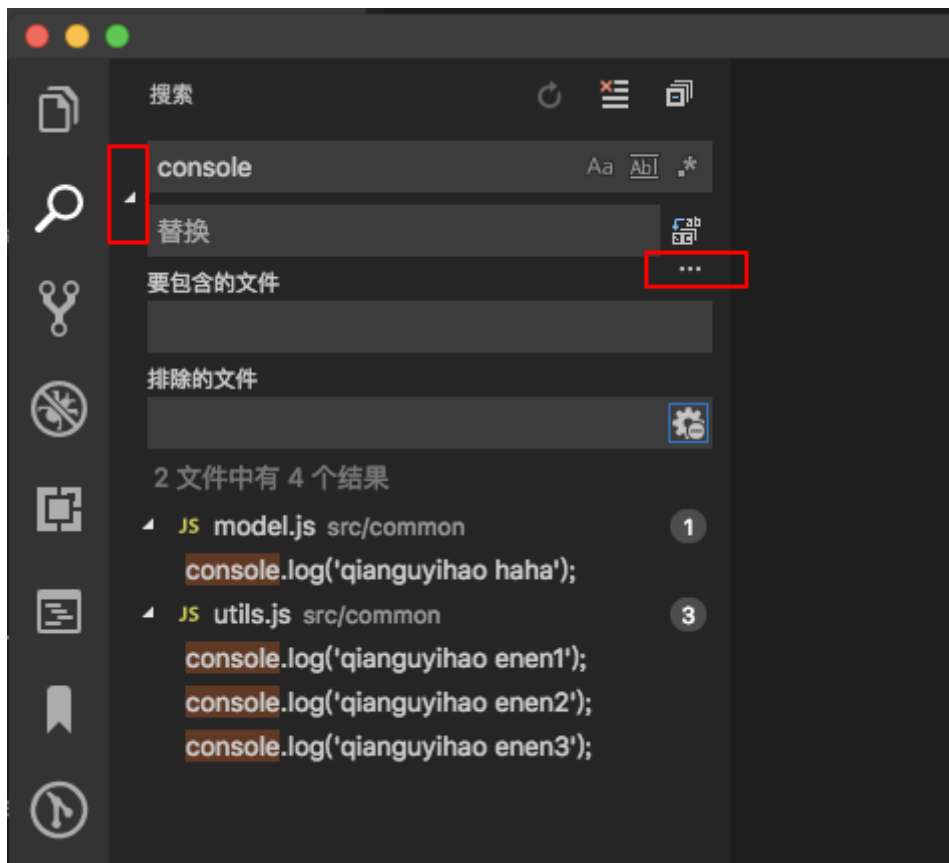


上图中，你可以通过「Tab」键和「Shift + Tab」键在输入框和替换框之间进行切换。

6、全局搜索

在上面的快捷键列表中，我们已经知道如下快捷键：

- `Cmd + Shift + F` (Win 用户是 `Ctrl + Shift + F`)：在全局的文件夹中进行搜索。效果如下：



上图中，你可以点击红框部分，展开更多的配置项。

7、将工作区放大/缩小

我们在上面的设置项里修改字体大小后，仅仅只是修改了代码的字体大小。

如果你想要缩放整个工作区（包括代码的字体、左侧导航栏的字体等），可以按下快捷键「**cmd +/-**」。windows 用户是按下「**ctrl +/-**」。

当我们在投影仪上给别人演示代码的时候，这一招十分管用。

如果你想恢复默认的工作区大小，可以在命令面板输入 **重置缩放**（英文是 **reset zoom**）

8、Emmet in VS Code

Emmet 可以极大的提高 html 和 css 的编写效率，它提供了一种非常简练的语法规则。

举个例子，我们在编辑器中输入缩写代码：**ul>li*6**，然后按下 Tab 键，即可得到如下代码片段：

```
<ul>
  <li></li>
  <li></li>
  <li></li>
  <li></li>
  <li></li>
  <li></li>
</ul>
```

VS Code 默认支持 Emmet。更多 Emmet 语法规则，请自行查阅。

四、VS Code 配置云同步

我们可以将配置云同步，这样的话，当我们换个电脑时，即可将配置一键同步到本地，就不需要重新安装插件了，也不需要重新配置软件。

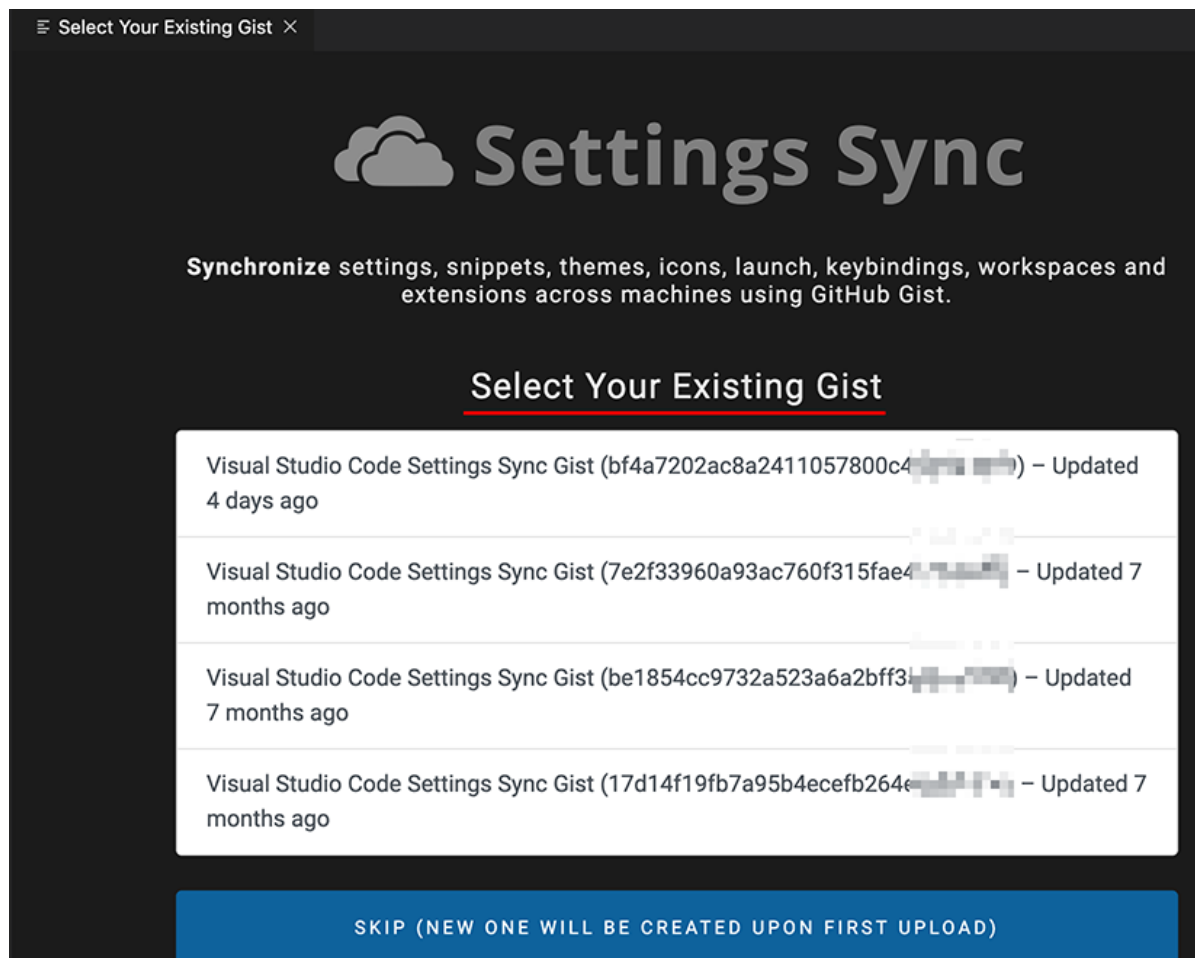
我们还可以把配置分享其他用户，也可以把其他用户的配置给自己用。

将自己本地的配置云同步到 GitHub（GitHub 是一个面向开源及私有软件项目的托管平台）：

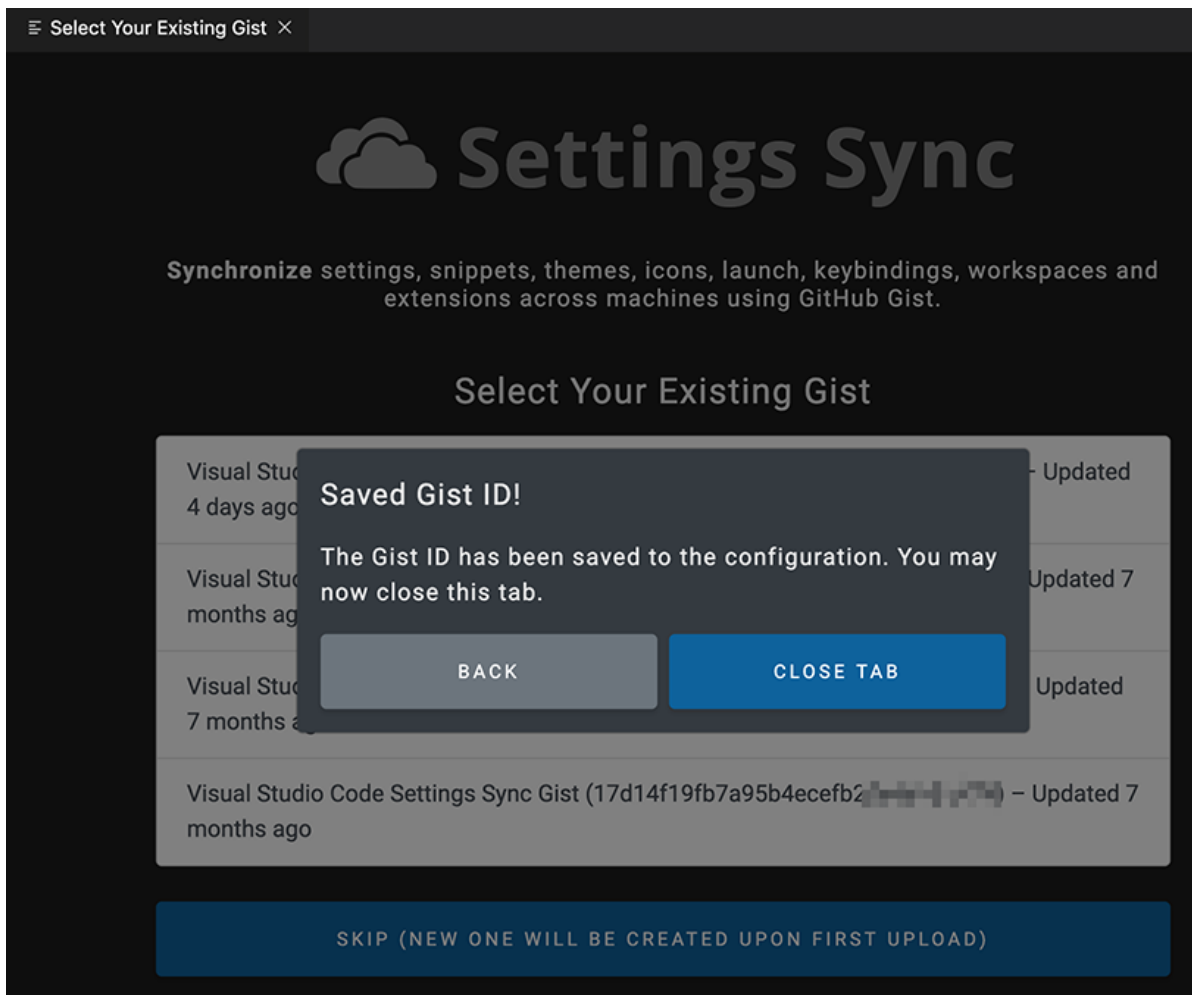
- (1) 安装插件 `settings-sync`。
- (2) 安装完插件后，在插件里使用 GitHub 账号登录。
- (3) 登录后在 vscode 的界面中，可以选择一个别人的 gist；也可以忽略掉，然后创建一个属于自己的 gist。
- (4) 使用快捷键「Command + Shift + P」，在弹出的命令框中输入 `sync`，并选择「更新/上传配置」，这样就可以把最新的配置上传到 GitHub。

换另外一个电脑时，从云端同步配置到本地：

- (1) 当我们换另外一台电脑时，可以先在 VS Code 中安装 `settings-sync` 插件。
- (2) 安装完插件后，在插件里使用 GitHub 账号登录。
- (3) 登录之后，插件的界面上，会自动出现之前的同步记录：



上图中，我们点击最新的那条记录，就可将云端的最新配置同步到本地：



如果你远程的配置没有成功同步到本地，那可能是网络的问题，此时，可以使用快捷键「Command + Shift + P」，在弹出的命令框中输入 sync，并选择「下载配置」，多试几次。

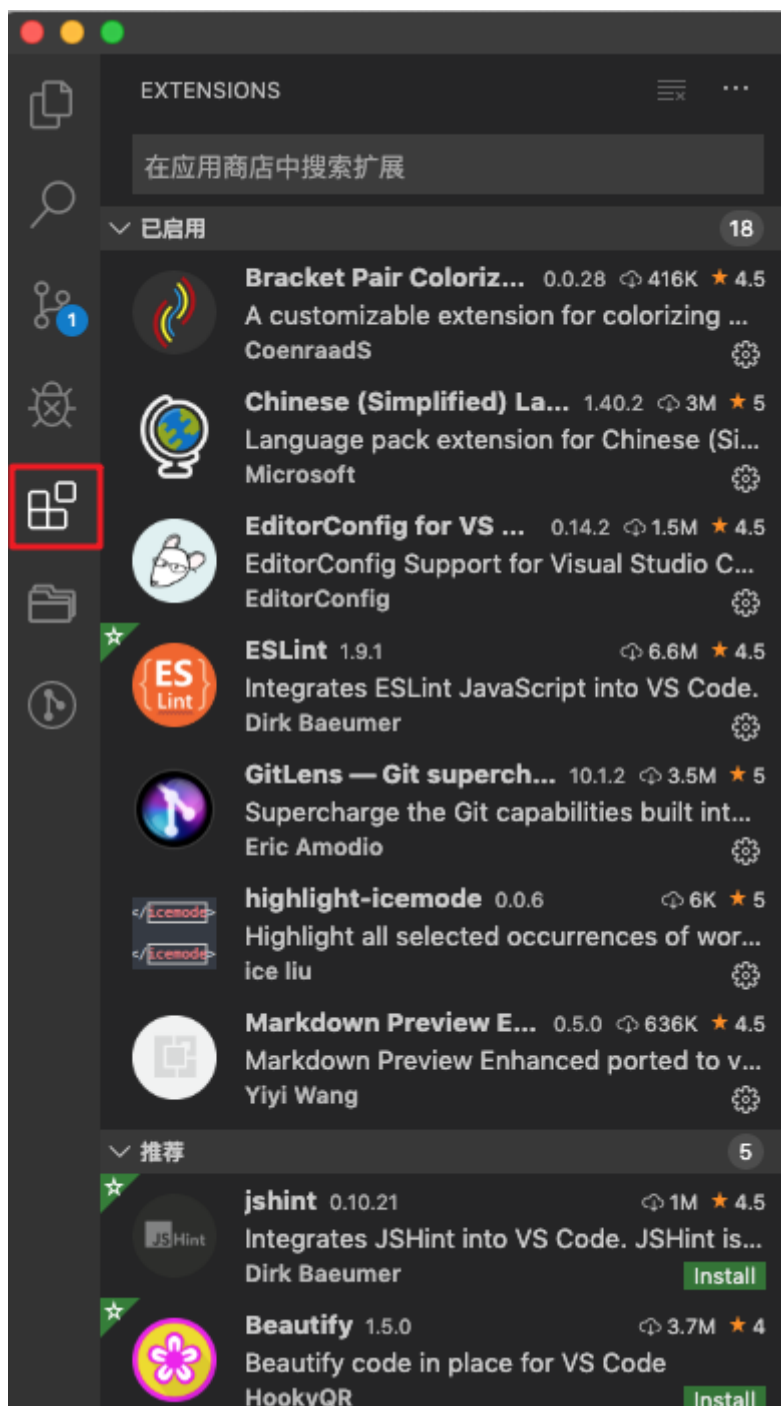
使用其他人的配置：

如果我们想使用别人的配置，首先需要对方提供给你 gist。具体步骤如下：

- (1) 安装插件 `settings-sync`。
- (2) 使用快捷键「Command + Shift + P」，在弹出的命令框中输入 sync，并选择「下载配置」
- (3) 在弹出的界面中，选择「Download Public Gist」，然后输入别人分享给你的 gist。注意，这一步不需要登录 GitHub 账号。

五、VS Code 插件推荐

VS Code 有一个很强大的功能就是支持插件扩展，让你的编辑器仿佛拥有了三头六臂。



上图中，点击红框部分，即可在输入框里，查找你想要的插件名，然后进行安装。

列举几个常见的插件，这些插件都很实用。注意：**顺序越靠前，越实用。**

1、GitLens 【荐】

我强烈建议你安装插件 `GitLens`，它是 VS Code 中我最推荐的一个插件，简直是 Git 神器，码农必备。如果你不知道，那真是 out 了。

GitLens 在 Git 管理上有很多强大的功能，比如：

- 将光标放置在代码的当前行，可以看到这样代码的提交者是谁，以及提交时间。这一点，是 GitLens 最便捷的功能。
- 查看某个 commit 的代码改动记录
- 查看不同的分支
- 可以将两个 commit 进行代码对比

- 甚至可以将两个 branch 分支进行整体的代码对比。这一点，简直是 GitLens 最强大的功能。当我们在不同分支 review 代码的时候，就可以用到这一招。

2、Git History

有些同学习惯使用编辑器中的 Git 管理工具，而不太喜欢要打开另外一个 Git UI 工具的同学，这一款插件满足你查询所有 Git 记录的需求。

3、Live Server 【荐】

在本地启动一个服务器，代码写完后可以实现「热更新」，实时地在网页中看到运行效果。就不需要每次都手动刷新页面了。

使用方式：安装插件后，开始写代码；代码写完后，右键选择「Open with Live Server」。

4、Chinese (Simplified) Language Pack for Visual Studio Code

让软件显示为简体中文语言。

5、Bracket Pair Colorizer 2: 突出显示成对的括号【荐】

Bracket Pair Colorizer 2 插件：以不同颜色显示成对的括号，并用连线标注括号范围。简称彩虹括号。

另外，还有个 Rainbow Brackets 插件，也可以突出显示成对的括号。

6、sftp: 文件传输【荐】

如果你需要将本地文件通过 ftp 的形式上传到局域网的服务器，可以安装 sftp 这个插件，很好用。在公司会经常用到。

7、open in browser

安装 open in browser 插件后，在 HTML 文件中「右键选择 --> Open in Default Browser」，即可在浏览器中预览网页。

8、highlight-icemode: 选中相同的代码时，让高亮显示更加明显【荐】

VSCode 自带的高亮显示，实在是不够显眼。用插件支持一下吧。

所用了这个插件之后，VS Code 自带的高亮就可以关掉了：

在用户设置里添加 "editor.selectionHighlight": false 即可。

参考链接：[vscode 选中后相同内容高亮插件推荐](#)

9、vscode-icons

vscode-icons 会根据文件的后缀名来显示不同的图标，让你更直观地知道每种文件是什么类型的。

10、Project Manager

工作中，我们经常会来回切换多个项目，每次都要找到对应项目的目录再打开，比较麻烦。Project Manager 插件可以解决这样的烦恼，它提供了专门的视图来展示你的项目，我们可以把常用的项目保存在这里，需要时一键切换，十分方便。

11、TODO Highlight

写代码过程中，突然发现一个 Bug，但是又不想停下来手中的活，以免打断思路，怎么办？按照代码规范，我们一般是在代码中加个 TODO 注释。比如：（注意，一定要写成大写 `TODO`，而不是小写的 `todo`）

```
//TODO:这里有个bug，我一会儿再收拾你
```

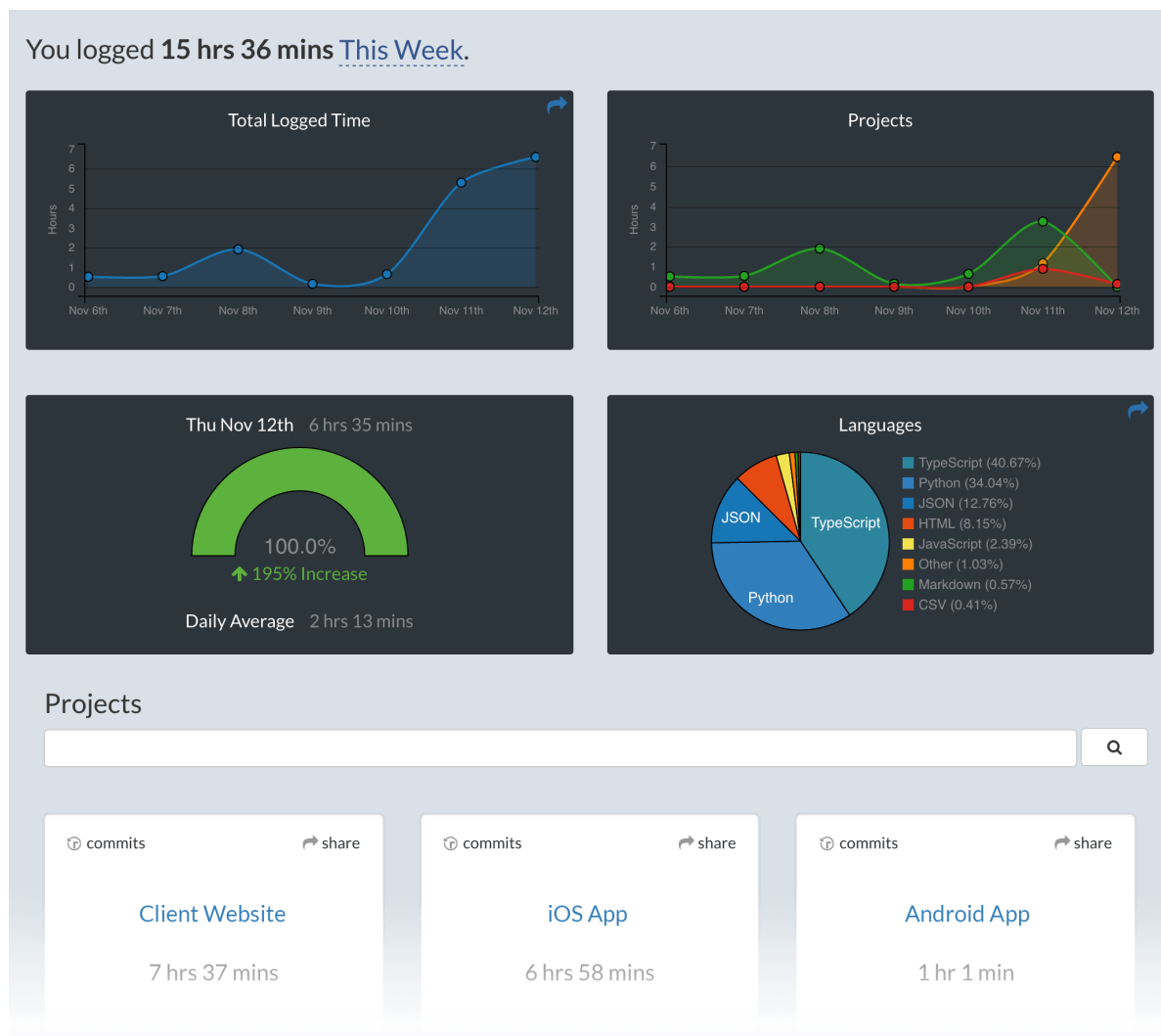
或者：

```
//FIXME:我也不知道为啥， but it works only that way.
```

安装了插件 `TODO Highlight` 之后，按住「Cmd + Shift + P」打开命令面板，输入「`Todohighlist`」，选择相关的命令，我们就可以看到一个 `todoList` 的清单。

12、WakaTime 【荐】

统计在 VS Code 里写代码的时间。统计效果如下：



13、Code Time

Code Time 插件：记录编程时间，统计代码行数。

安装该插件后，VS Code 底部的状态栏右下角可以看到时间统计。点击那个位置之后，选择「Code Time Dashboard」，即可查看统计结果。

备注：试了一下这个 code time 插件，发现统计结果不是很准。

14、Markdown Preview Github Styling 【荐】

以 GitHub 风格预览 Markdown 样式，十分简洁优雅。就像下面这样，左侧书写 Markdown 文本，右侧预览 Markdown 的渲染效果：



15、Markdown Preview Enhanced

预览 Markdown 样式。

Markdown All in One

这个插件将帮助你更高效地在 Markdown 中编写文档。

16、Settings Sync 【荐】

- 地址：<https://github.com/shanalikhan/code-settings-sync>
- 作用：多台设备之间，同步 VS Code 配置。通过登录 GitHub 账号来使用这个同步工具。

同步的详细操作已在上面讲过。

17、vscode-syncing

- 地址：<https://github.com/nonoroazoro/vscode-syncing>
- 作用：多台设备之间，同步 VS Code 配置。

18、Vetur

Vue 多功能集成插件，包括：语法高亮，智能提示，emmet，错误提示，格式化，自动补全，debugger。VS Code 官方钦定 Vue 插件，Vue 开发者必备。

19、ES7 React/Redux/GraphQL/React-Native snippets

React/Redux/react-router 的语法智能提示。

20、minapp：小程序支持

小程序开发必备插件。

21、Prettier：代码格式化

Prettier 是一个代码格式化工具，只关注格式化，但不具备校验功能。在一个多人协同开发的团队中，统一的代码编写规范非常重要。一套规范可以让我们编写的代码达到一致的风格，提高代码的可读性和统一性。自然维护性也会有所提高。

22、ESLint：代码格式校验

日常开发中，建议用可以用 Prettier 做代码格式化，然后用 eslint 做校验。

23、Beautify

代码格式化工具。

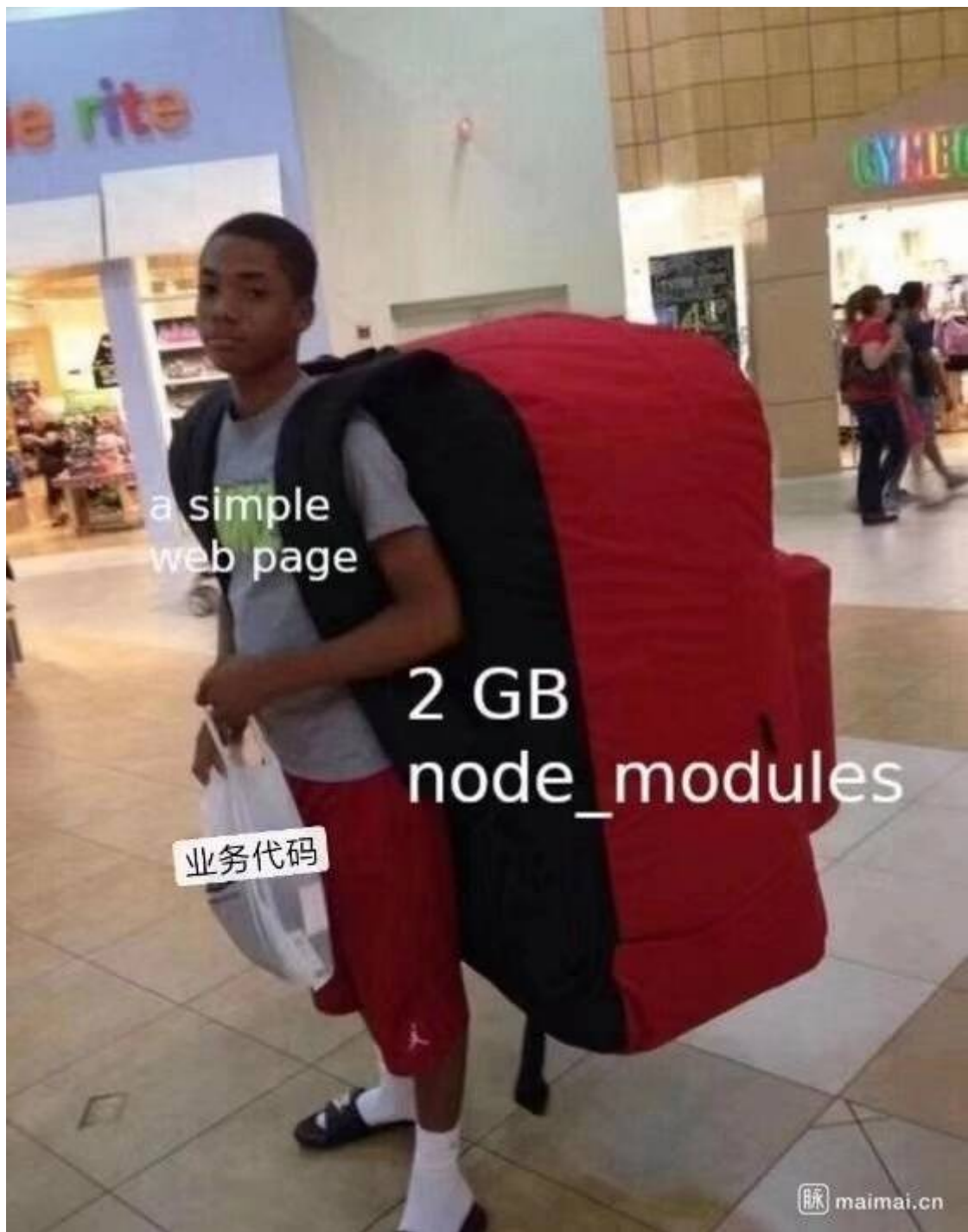
备注：相比之下，Prettier 是当前最流行的代码格式化工具，比 Beautify 用得更多。

24、JavaScript(ES6) code snippets

ES6 语法智能提示，支持快速输入。

25、Search node_modules 【荐】

node_modules 模块里面的文件夹和模块实在是太多了，根本不好找。好在安装 Search node_modules 这个插件后，输入快捷键「Cmd + Shift + P」，然后输入 node_modules，在弹出的选项中选择 Search node_modules，即可搜索 node_modules 里的模块。



26、indent-rainbow: 突出显示代码缩进

`indent-rainbow` 插件：突出显示代码缩进。

安装完成后，效果如下图所示：

```
5  ....var dest
6  ....while(des
7  ....var
8  ....offs=
9  ....i=offs
10 ....while i
11 ....dest.
```

indent-rainbow

oderwat.indent-rainbow

oderwat | 539,136 | ★★★★★ | 存储库 | 许可证

Makes indentation easier to read

安装

细节

发布内容

Indent-Rainbow

A simple extension to make indentation more readable

This extension colorizes the indentation in front of your text alternating four different colors on each step.

```
nimbench_ex2.nim src/misc
1  import nimbench
2
3  bench(str, m):
4    for n in 0..<m:
5      var dest = "apple"
6      while(dest.len<100):
7        var
8          offs=dest.len-5
9          i=offs
10         while i<offs+10:
11           dest.add(dest[i])
12           i.inc
13         doNotOptimizeAway(dest)
14
15  runBenchmarks()
16 |
```

27、javascript console utils：快速打印 log 日志【荐】

安装这个插件后，当我们按住快捷键「Cmd + Shift + L」后，即可自动出现日志 `console.log()`。简直是日志党福音。

当我们选中某个变量 `name`，然后按住快捷键「Cmd + Shift + L」，即可自动出现这个变量的日志 `console.log(name)`。

其他的同类插件还有：Turbo Console Log。

不过，生产环境的代码，还是尽量少打日志比较好，避免出现一些异常。

编程有三等境界：

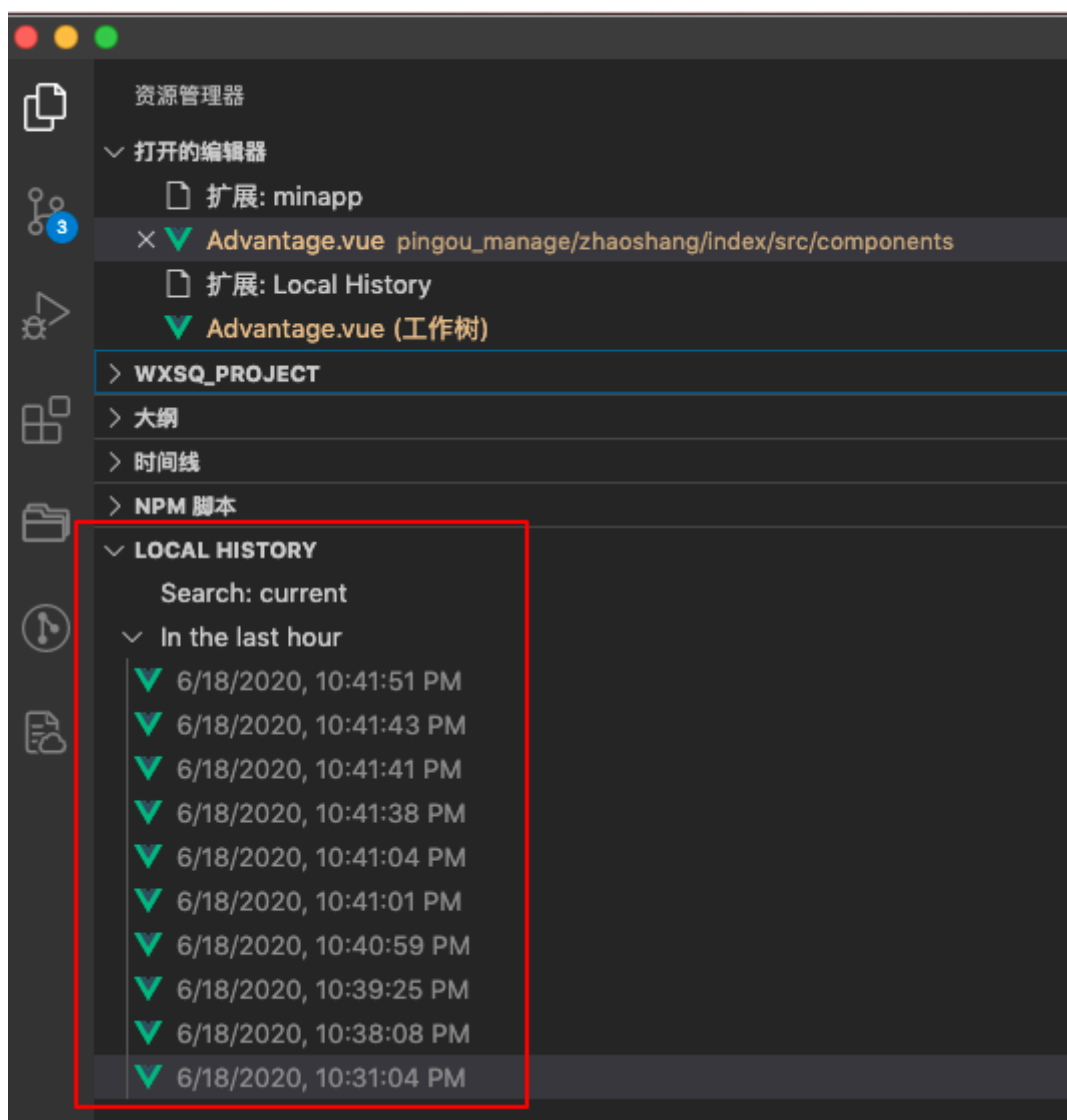
- 第三等境界是打日志，这是最简单、便捷的方式，略显低级，一般新手或资深程序员偷懒时会用。
- 第二等境界是断点调试，在前端、Java、PHP、iOS 开发时非常常用，通过断点调试可以很直观地跟踪代码执行逻辑、调用栈、变量等，是非常实用的技巧。
- 第一等境界是测试驱动开发，在写代码之前先写测试。与第二等的断点调试刚好相反，大部分人不是很习惯这种方式，但在国外开发者或者敏捷爱好者看来，这是最高效的开发方式，在保证代码质量、重构等方面非常有帮助，是现代编程开发必不可少的一部分。

28、Code Spell Checker：单词拼写错误检查

这个拼写检查程序的目标是帮助捕获常见的单词拼写错误，可以检测驼峰命名。从此告别 Chinglish.

29、Local History 【荐】

维护文件的本地历史记录，强烈建议安装。代码意外丢失时，有时可以救命。



30、Polacode-2020：生成代码截图 【荐】

可以把代码片段保存成美观的图片，主题不同，代码的配色方案也不同，也可以自定义设置图片的边框颜色、大小、阴影。

尤其是在我们做 PPT 分享时需要用到代码片段时，或者需要在网络上优雅地分享代码片段时，这一招很有用。

生成的效果如下：

```
function getCheckedFunction(fn) {
  return hideStackFrames(function checkError(...args) {
    const ctx = {};
    const ret = fn(...args, ctx);
    if (ret === undefined) {
      throw new ERR_SYSTEM_ERROR(ctx);
    }
    return ret;
  });
}
```

其他同类插件：CodeSnap。我们也可以通过 <https://carbon.now.sh/> 这个网站生成代码图片

有人可能会说：直接用 QQ 截图不行吗？可以是可以，但不够美观、不够干净。

31、Image Preview 【荐】

图片预览。鼠标移动到图片 url 上的时候，会自动显示图片的预览和图片尺寸。

32、Auto Close Tag、Auto Rename Tag

自动闭合标签、自动对标签重命名。

33、Better Comments

为注释添加更醒目、带分类的色彩。

34、CSS Peek

增强 HTML 和 CSS 之间的关联，快速查看该元素上的 CSS 样式。

35、Vue CSS Peek

CSS Peek 对 Vue 没有支持，该插件提供了对 Vue 文件的支持。

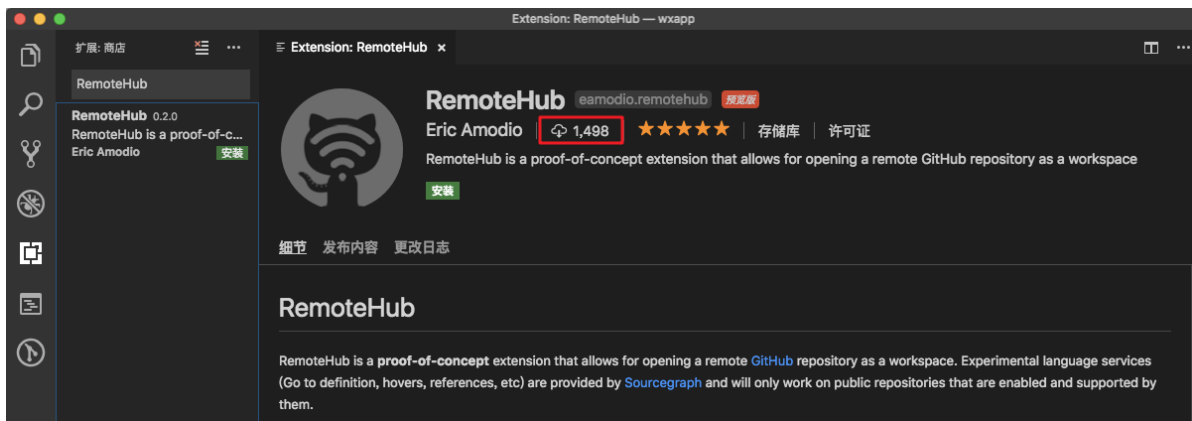
36、Color Info

这个便捷的插件，将为你提供你在 CSS 中使用颜色的相关信息。你只需在颜色上悬停光标，就可以预览色块中色彩模型的（HEX、RGB、HSL 和 CMYK）相关信息了。

37、RemoteHub

不要惊讶，RemoteHub 和 GitLens 是同一个作者开发出来的。

RemoteHub 插件的作用是：可以在本地查看 GitHub 网站上的代码，而不需要将代码下载到本地。



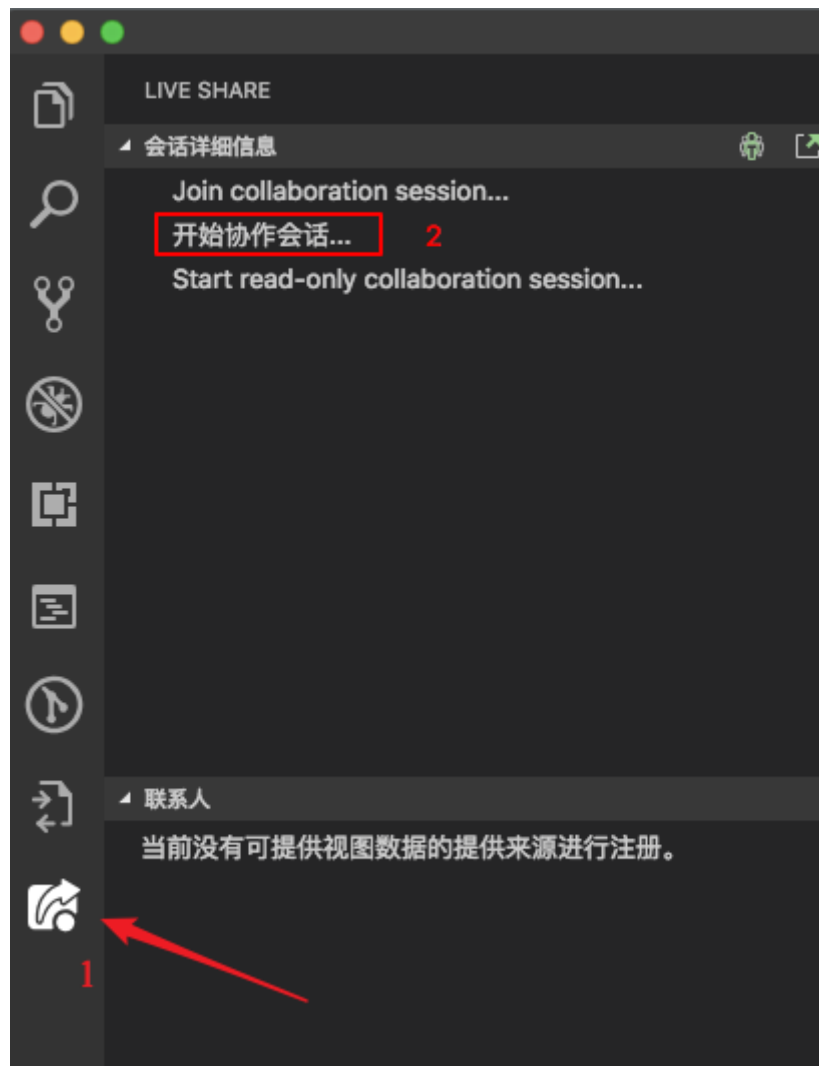
这个插件目前使用的人还不多，赶紧安装起来尝尝鲜吧。

38、Live Share：实时编码分享

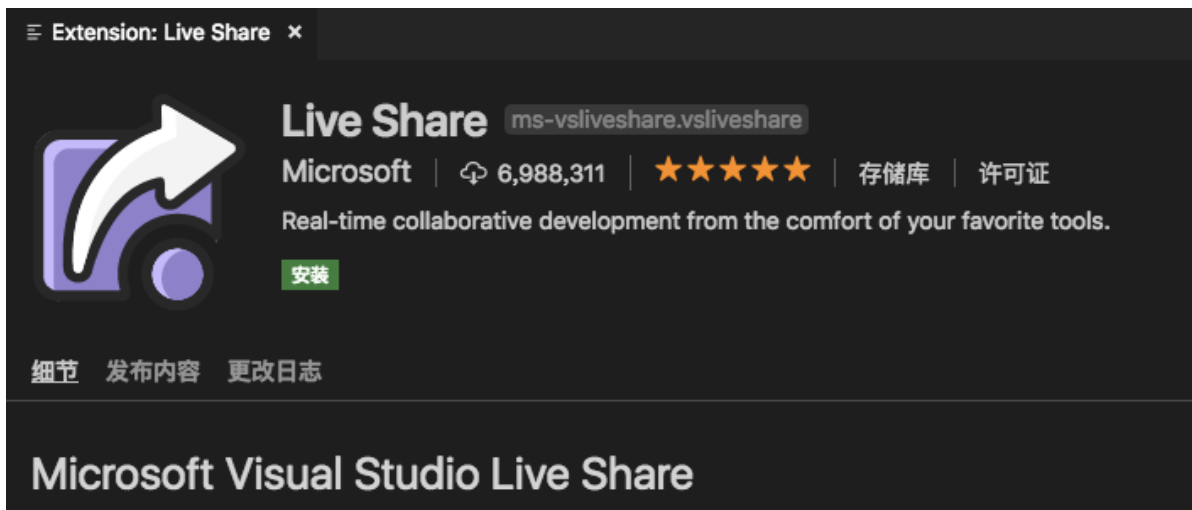
Live Share 这个神奇的插件是由微软官方出品，它的作用是：**实时编码分享**。也就是说，它可以实现你和你的同伴一起写代码。这绝对就是**结对编程**的神器啊。

安装方式：

打开插件管理，搜索“live share”，安装。安装后重启 VS Code，在左侧会多出一个按钮：



上图中，点击红框部分，登录后就可以分享你的工作空间了。



39、Import Cost

在项目开发过程中，我们会引入很多 npm 包，有时候可能只用到了某个包里的一个方法，却引入了整个包，导致代码体积增大很多。Import Cost 插件可以在代码中友好的提示我们，当前引入的包会增加多少体积，这很有助于帮我们优化代码的体积。

40、Paste JSON as Code

此插件可以将剪贴板中的 JSON 字符串转换成工作代码。支持多种语言。

六、常见主题插件

给你的 VS Code 换个皮肤吧，免费的那种。

- Dracula Theme
- Material Theme
- Nebula Theme
- [One Dark Pro](#)
- One Monokai Theme
- Monokai Pro
- Ayu
- [Snazzy Plus](#)
- [Dainty](#)
- Synthwave '84
- GitHub Plus Theme：白色主题
- Horizon Theme：红色主题

参考链接

- [VSCode 插件大全 | VSCode 高级玩家之第二篇](#)
- <http://www.supuwoerc.xyz/tools/vscode/plugins.html>
- [如何让 VS Code 更好用 10 倍？这里有一份 VS Code 新手指南](#)
- [那些你应该考虑卸载的 VSCode 扩展](#)
- [VS Code 折腾记 - \(16\) 推荐一波实用的插件集](#)
- [VSCode 前端必备插件，有可能你装了却不知道如何使用？](#)
- [能让你开发效率翻倍的 VSCode 插件配置（上）](#)
- <https://segmentfault.com/a/1190000012811886>

- [「Vscode」打造类 sublime 的高颜值编辑器](#)
- [Mac Vscode 快捷键](#)
- [使用 VSCode 的一些技巧](#)