

## VS Code 试用截图 - 简书

VS Code 是微软出品的、难得一见的、macOS/Linux(Fedora、Ubuntu)/Windows 三操作系统通用的、开源的、轻量级的编辑器。

网上大部分用VSCode和SublimeText、Atom等相比，notepad++等正在逐步被摒弃。

今晚走马观花撸了一遍，总结几个它的优点：

1. 支持了众多编程语言，支持指的是可以：智能提示、查找和跳转定义、查找引用、竟然还可以 **Debug**..... —— 此货简直不甘心只做编辑器，是有点IDE的小野心的
2. 内置了 **git** 的支持，点鼠标和命令行都行，简直是为 **windows** 上的程序员释放了天性
3. 所有配置用 **json** 文件来做，没有传统软件的配置界面，就是为了隔离非程序员，给非程序员设置门槛。因为程序员和非程序员的需求是不一样的，想做一个全天下人都满意的产品基本不切实际，不如精准定位自己产品的目标人群。
4. 没有IDE通常的 **project** 的概念，但需要指定 **folder**，然后快捷键什么的才能快速搜索 **folder** 中的文件。**vi** 的同学很好理解，在哪个目录下打开 **vi**，在 **vi** 中 **find** 文件就是在这个 **folder** 中。
5. 专门为 **markdown** 做了定制性优化：支持实时预览、更换编译器、更好 **css** 文件等
6. 平台性做得好所以插件很多.....
7. 速度超快，内存比 **sublime** 占用少很多，打开百兆的大文件也是秒开
8. 为宽屏写代码的程序员想的很周到，分列的快捷键很好使

可以这么说：当下程序员的兵器谱已经基本分成这么几个阵营

- 冷兵器

- **vi/vim、emacs**：干将、莫邪，愈久弥坚，真粉和情怀粉都无数
- **sublime、atom、vscode**：第二阵营的三驾马车，各种操作都被命令化，被有范儿、有追求的程序员所喜爱。三者脸长的高度撞衫，操作方式也高度趋同，连快捷键恨不得都一样，大家比的是速度、性能、细节.....把下面第三梯队甩了老远
- **notepad++、editplus、ultraedit**：10年前的锐意先锋，当下的第三梯队，一切操作都还是拿鼠标点点点.....

- 热兵器: IDE 需要按需选择，不予赘述

必须要补充说明的是：若论工作效率，兵器并不是最重要的，手艺才重要，为“用什么兵器”争个你死我活的，大多是没什么实战经验的。别管那么多，抄个家伙上战场先砍死几个再说。

言归正传：

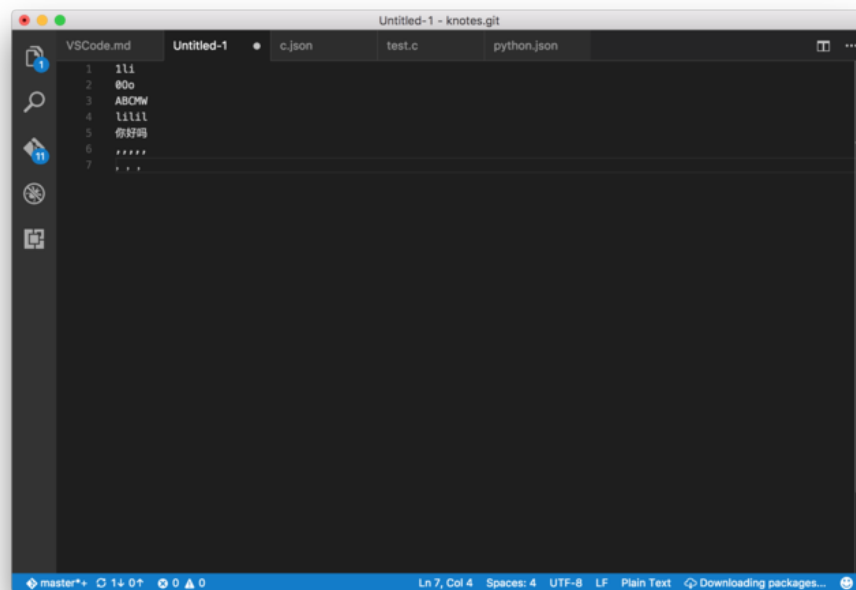
官网：<https://code.visualstudio.com/>

Extensions 的下载地址：<https://marketplace.visualstudio.com/VSCode>

整体操作方法和习惯和 **sublime**、**atom** 一致：

- 一个快捷键进入一个模式
- 模式下直接输入字符，进行快速、模糊的查询
- 执行查询出的命令、文件、等

开篇第一个截图，先看看字符对齐，一个程序员的编辑器，必须是等宽字体的

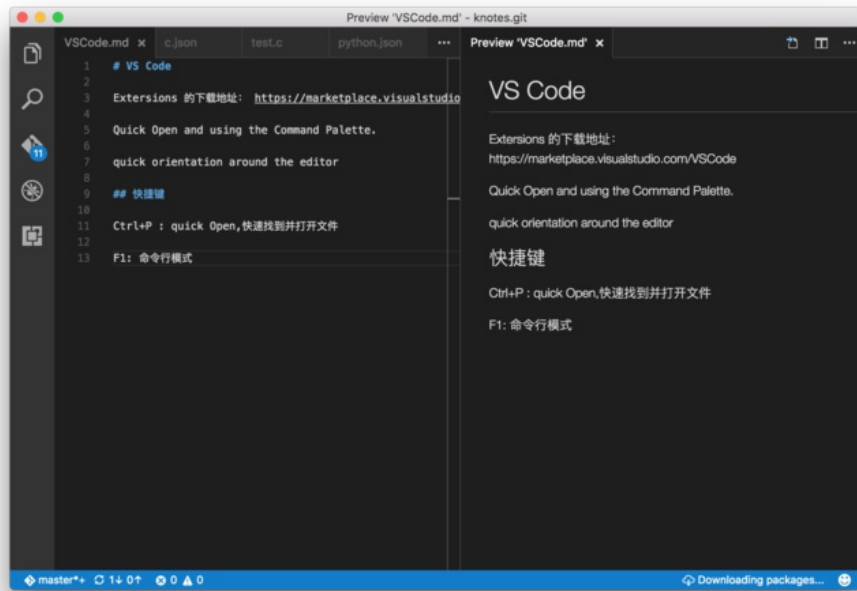


Screen Shot 2016-11-16 at 00.27.19.png

- li、00o都还分得清
- MW和li都能等宽
- 可惜中文字符和中文标点不能严格按照2个英文字符来对齐，这样中英文夹杂的程序员的代码就又不能看了——不过我这是macOS版的，不知道windows版的如何。

第1个要说的快捷键是：Ctrl+1/2/3 —— 并列分几个栏。比如下面分了2栏，默认最多3栏，可以在setting中修改。

左边是markdown的书写栏，右边是markdown的实时预览栏。当然，两栏也可以完全没有任何联系，完全独立显示两个文件。

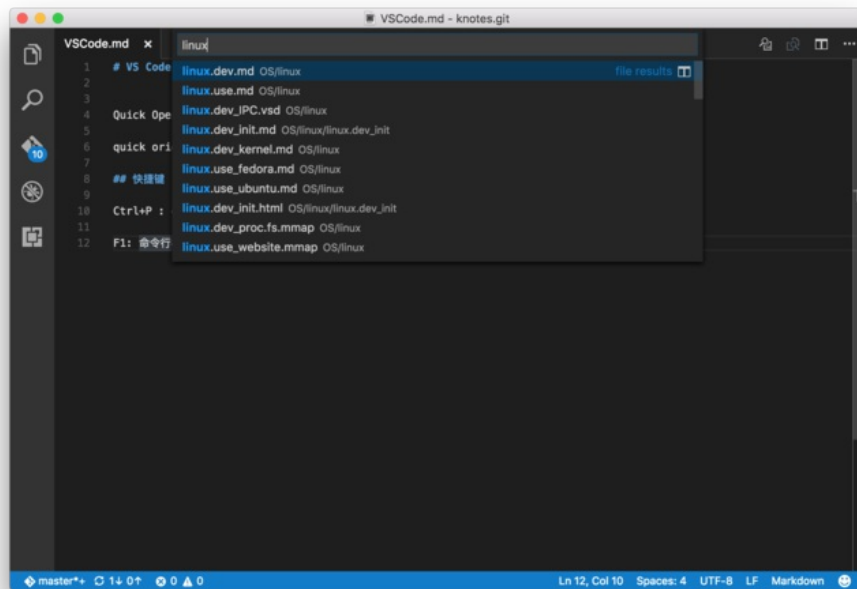


Screen Shot 2016-11-16 at 00.04.25.png

同时引出 “用VS Code 打造 markdown 环境” 的话题，看下面文章：

1. <http://code.visualstudio.com/Docs/languages/markdown>
2. <http://thisdavej.com/build-an-amazing-markdown-editor-using-visual-studio-code-and-pandoc/>

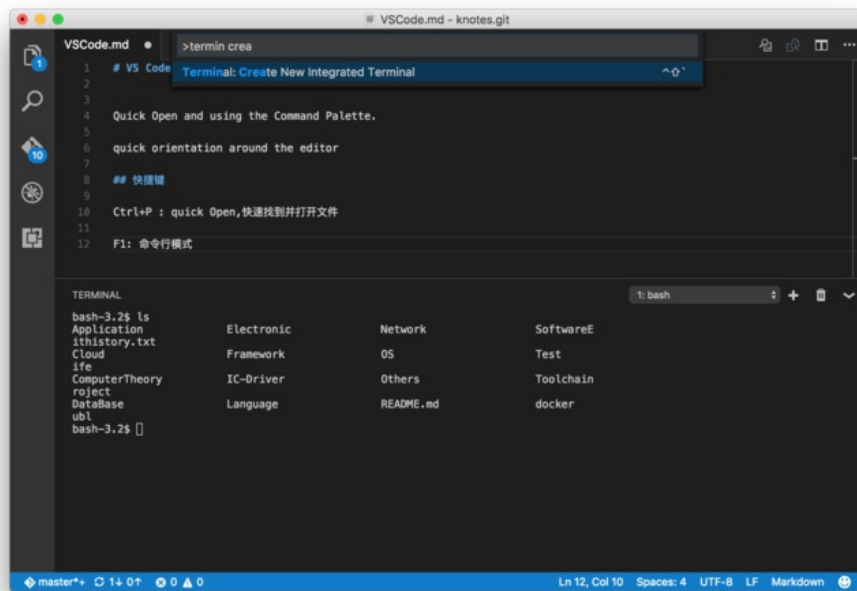
第2个要说的快捷键是：Ctrl+P —— quick Open, 快速找到并打开文件



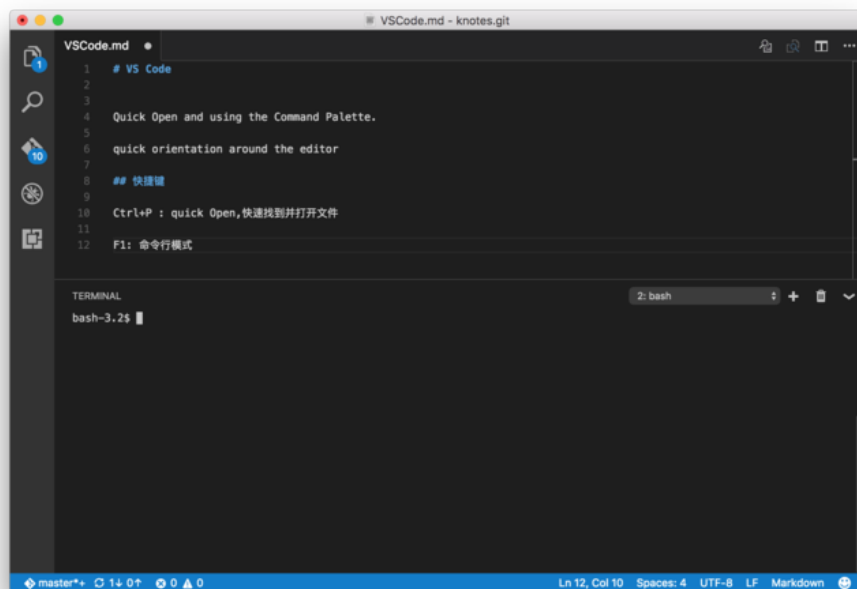
Screen Shot 2016-11-15 at 21.54.58.png

第3个要说的快捷键是：F1 —— 命令输入模式

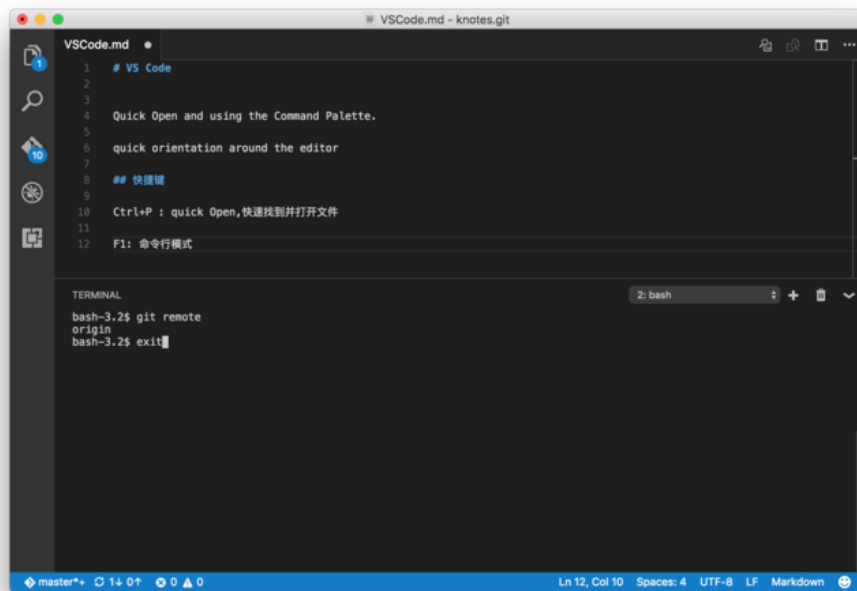
比如：terminal 命令



Screen Shot 2016-11-15 at 21.53.22.png



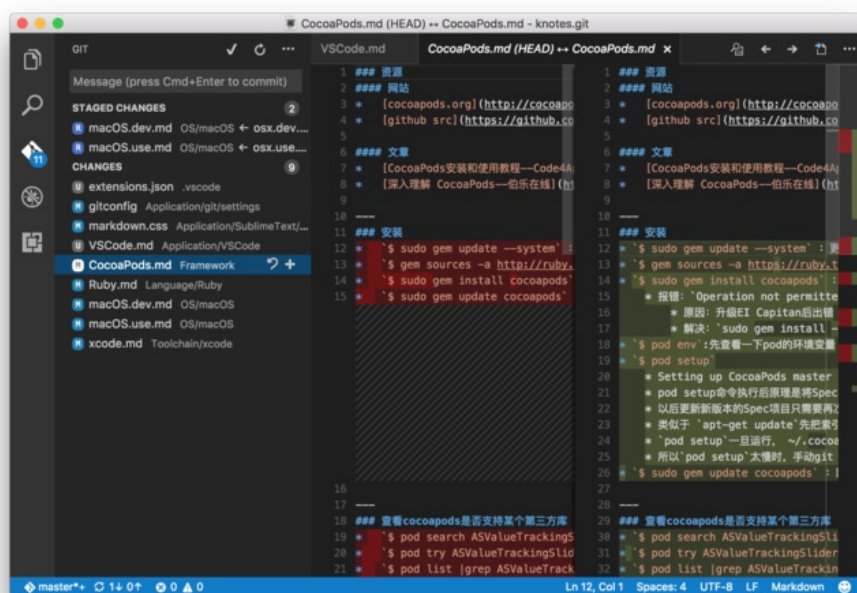
Screen Shot 2016-11-15 at 21.53.37.png



Screen Shot 2016-11-15 at 21.54.08.png

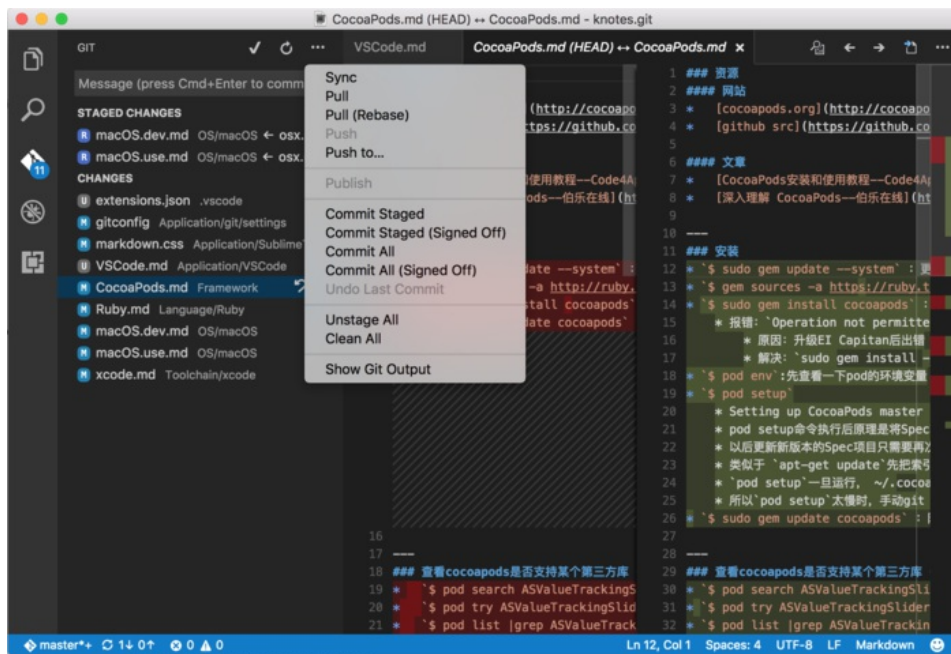
VSCode 对 git 兼容的也相当不错，一直以来 git 的 windows版本（git for windows）的效果、性能、大文件速度……都差强人意，相信VSCode自家开发的git肯定在windows上能够表现良好。

点击一个git修改状态的文件，会直接打开对比栏显示。—— 就为这一点，很多码农从sublime转投到VSCode门下了。



Screen Shot 2016-11-15 at 21.59.55.png

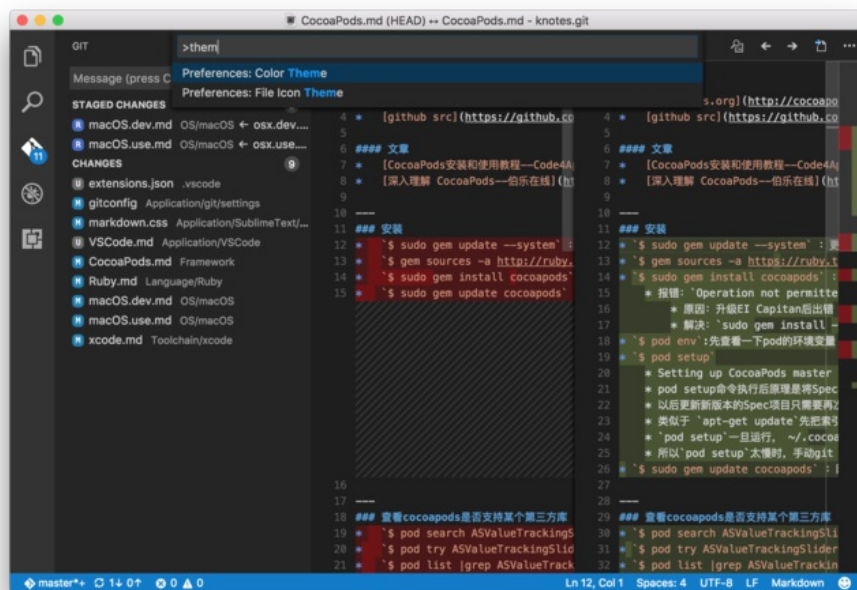
这里还有很多git的操作



Screen Shot 2016-11-15 at 22.01.39.png

来继续介绍命令。

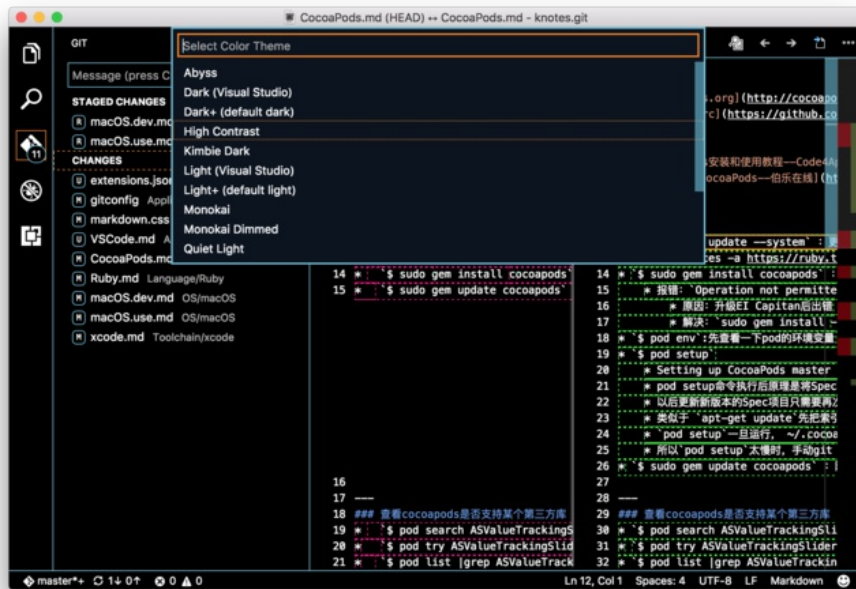
theme 命令：修改编辑器的主题。



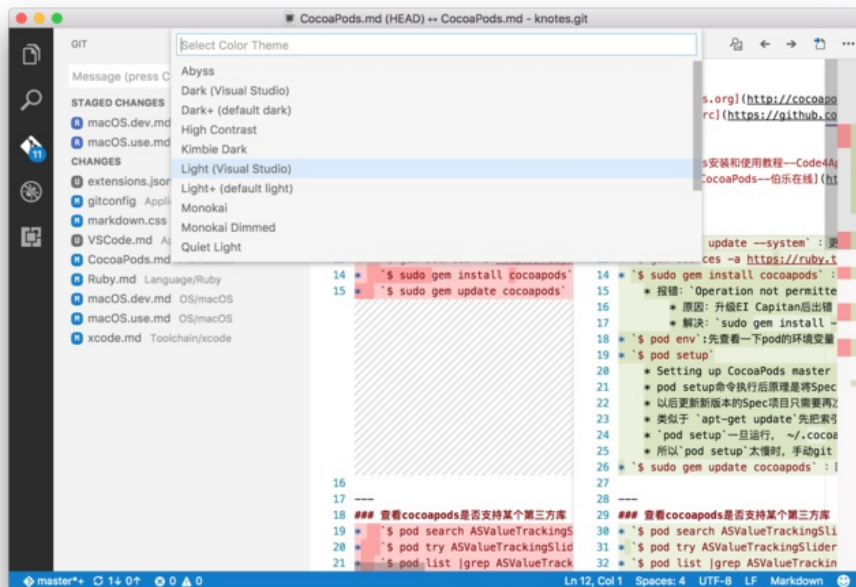
Screen Shot 2016-11-15 at 22.03.24.png

每略过一个主题，不必选中，界面即可预览该主题



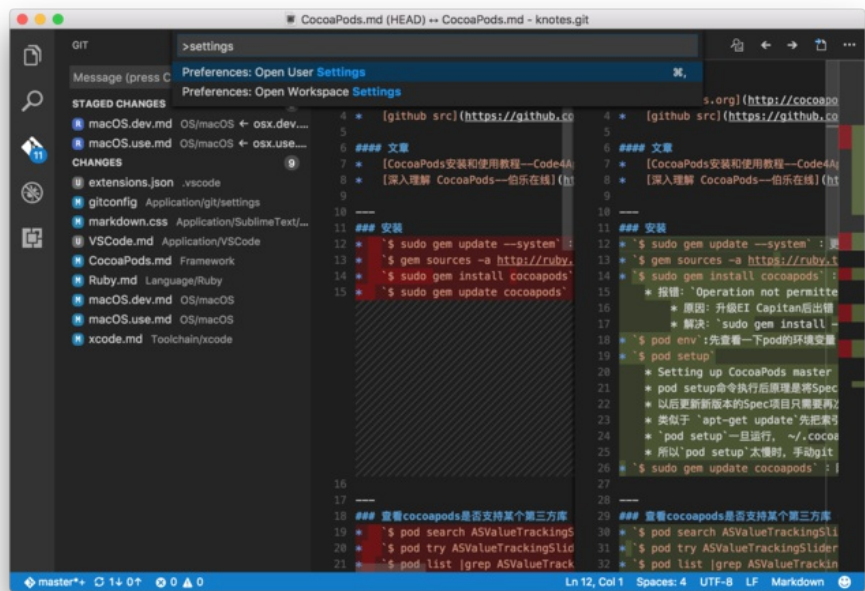


Screen Shot 2016-11-15 at 22.03.31.png



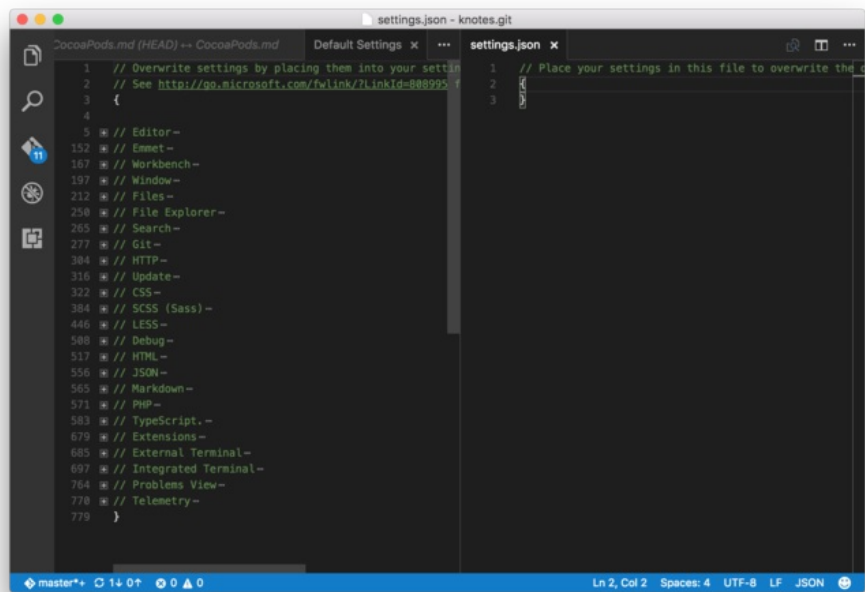
Screen Shot 2016-11-15 at 22.04.34.png

settings 命令:



Screen Shot 2016-11-15 at 22.05.01.png

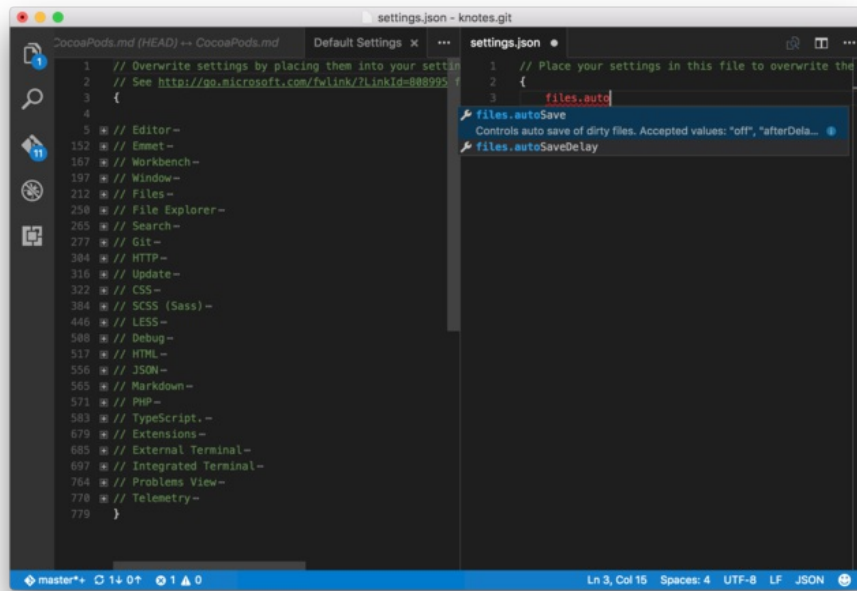
和sublime很类似，有一个系统的配置是不能修改的，只能看；然后自己一份配置可以覆盖系统配置。



Screen Shot 2016-11-15 at 22.06.05.png

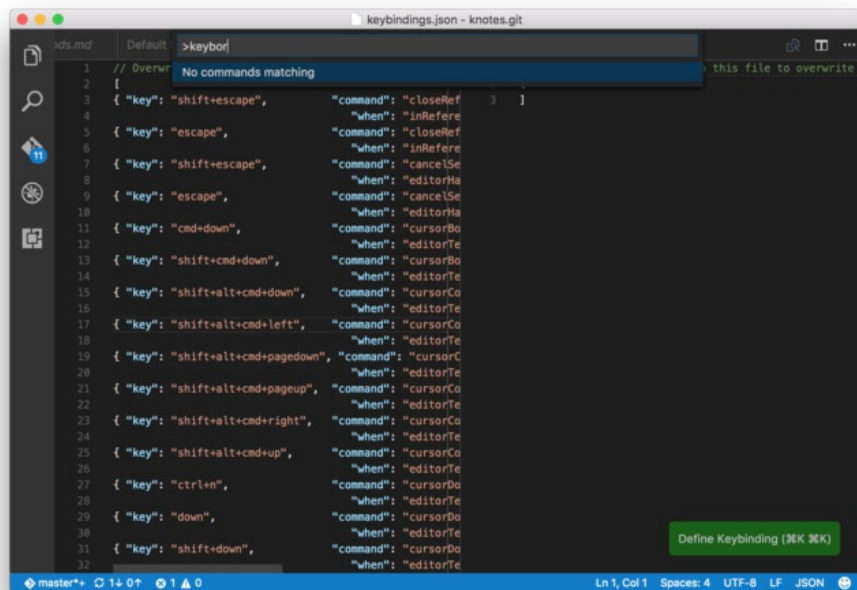
会自动弹出提示，鼠标放在 i 符号上还有详细提示。





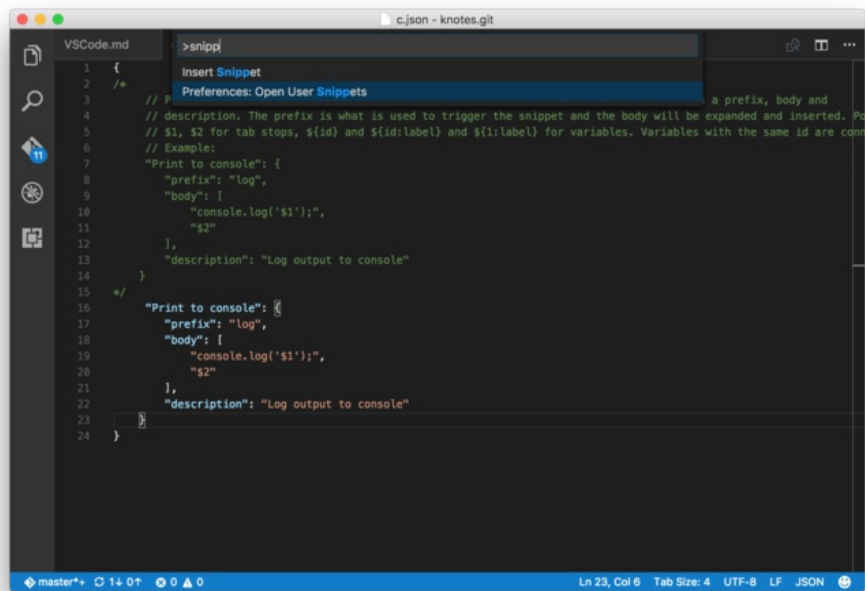
Screen Shot 2016-11-15 at 22.06.19.png

keyboard 命令：查看快捷键

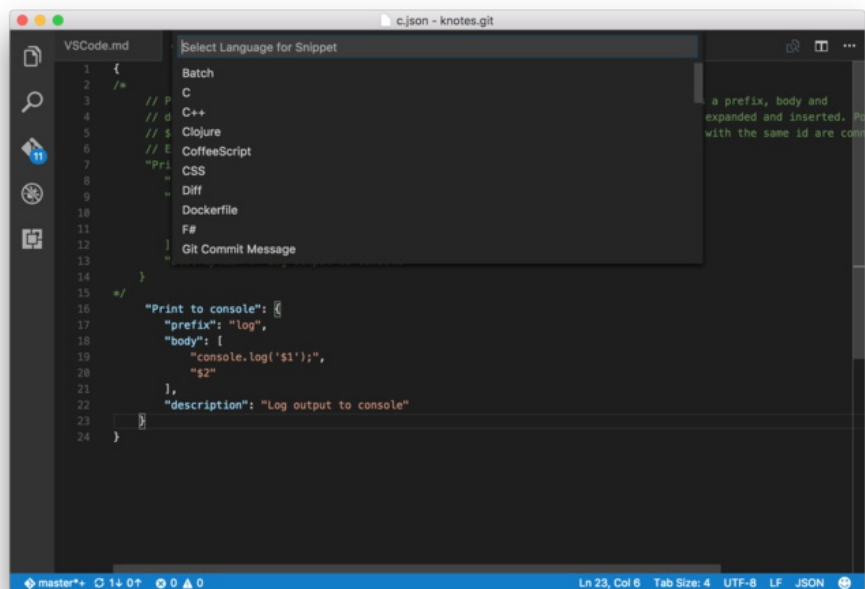


Screen Shot 2016-11-15 at 22.10.32.png

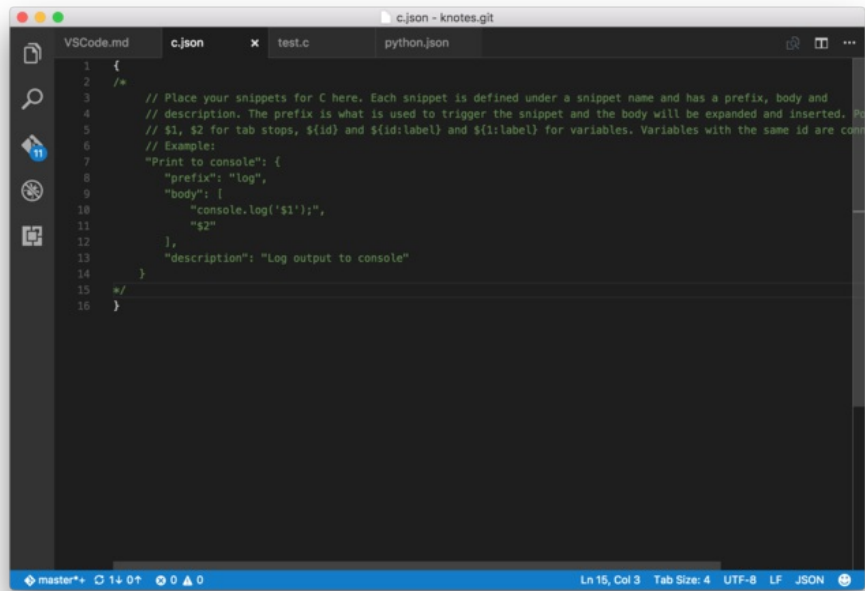
snippet命令：为某个语言添加代码片段



Screen Shot 2016-11-15 at 22.21.11.png

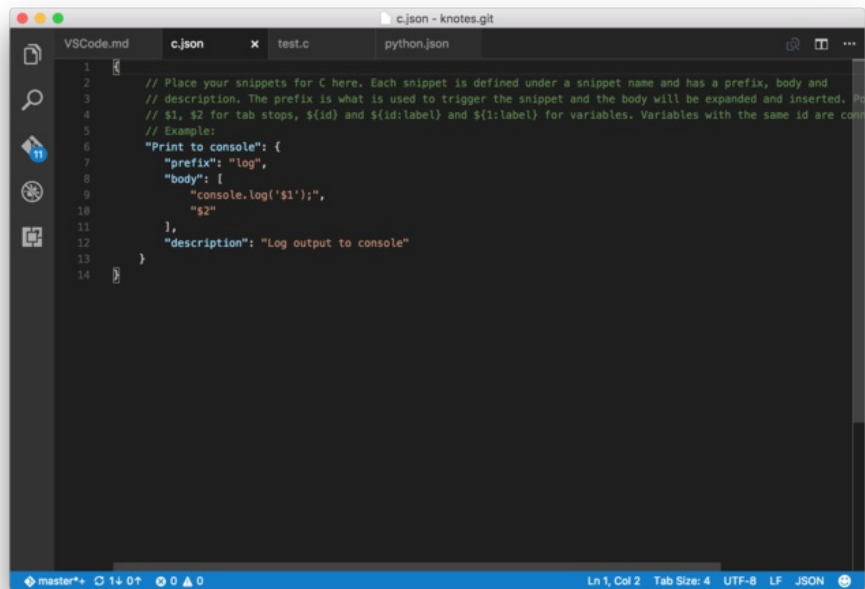


Screen Shot 2016-11-15 at 22.21.19.png



```
1 {
2   /*
3   // Place your snippets for C here. Each snippet is defined under a snippet name and has a prefix, body and
4   // description. The prefix is what is used to trigger the snippet and the body will be expanded and inserted. Pl
5   // $1, $2 for tab stops, ${id} and ${id:label} and ${1:label} for variables. Variables with the same id are con
6   // Example:
7   "Print to console": {
8     "prefix": "log",
9     "body": [
10      "console.log('$1');",
11      "$2"
12    ],
13    "description": "Log output to console"
14  }
15  */
16 }
```

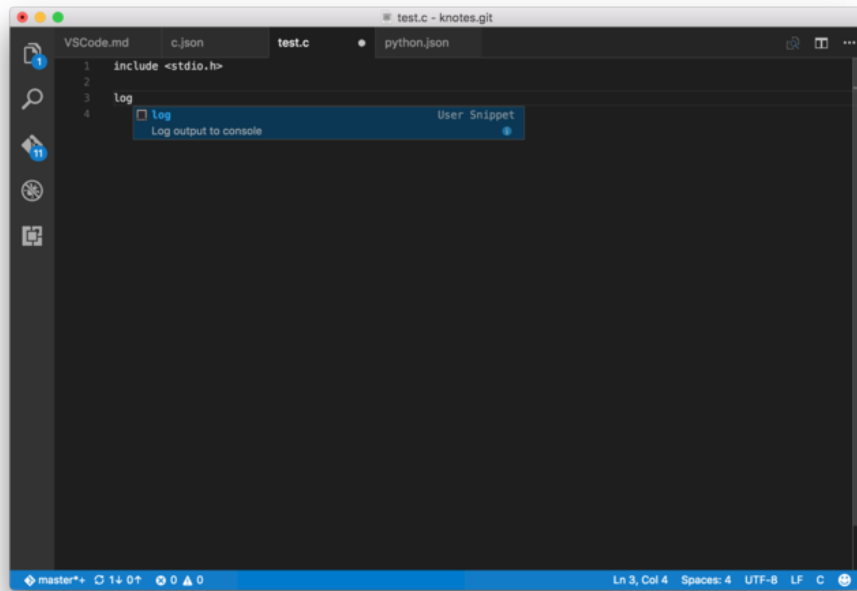
Screen Shot 2016-11-15 at 22.21.41.png



```
1 {
2   /*
3   // Place your snippets for C here. Each snippet is defined under a snippet name and has a prefix, body and
4   // description. The prefix is what is used to trigger the snippet and the body will be expanded and inserted. Pl
5   // $1, $2 for tab stops, ${id} and ${id:label} and ${1:label} for variables. Variables with the same id are con
6   // Example:
7   "Print to console": {
8     "prefix": "log",
9     "body": [
10      "console.log('$1');",
11      "$2"
12    ],
13    "description": "Log output to console"
14  }
15  */
16 }
```

Screen Shot 2016-11-15 at 22.21.47.png

输入前导字符 log ，即可弹出片段提示，选中片段后出现整段代码。



Screen Shot 2016-11-15 at 22.21.54.png

Screen Shot 2016-11-15 at 22.22.01.png

Extension:

ext 命令:

- 可以 `ext install xxx` 直接安装一个extension
- 也可以点出侧边栏，查看哪些可以安装、哪些已经安装.....

Screen Shot 2016-11-15 at 23.16.57.png

- 还可以到官网上下载extension，然后本地安装

Screen Shot 2016-11-15 at 23.14.28.png