

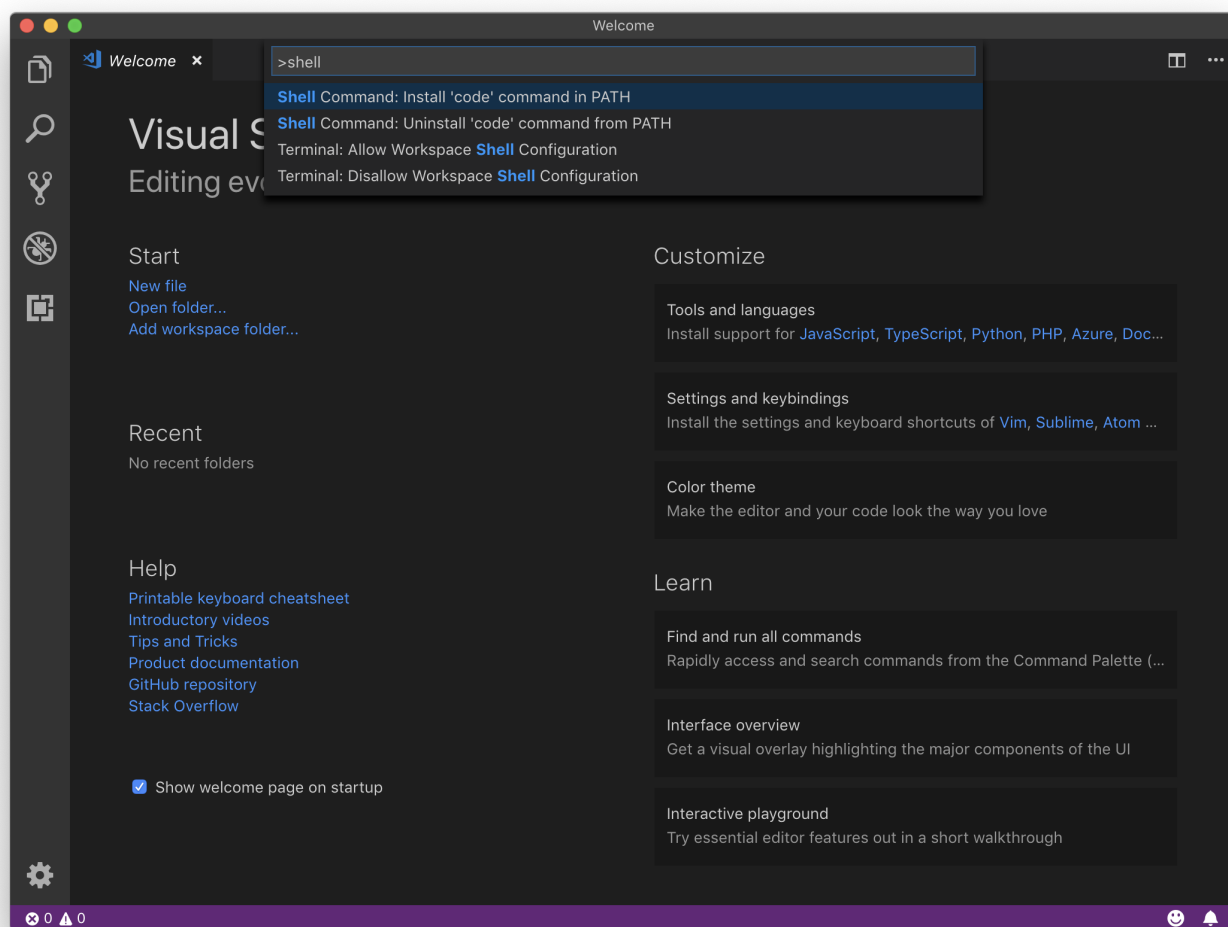
# Visual Studio Code 的安装与配置

官方文档请查询：

<https://code.visualstudio.com/docs>

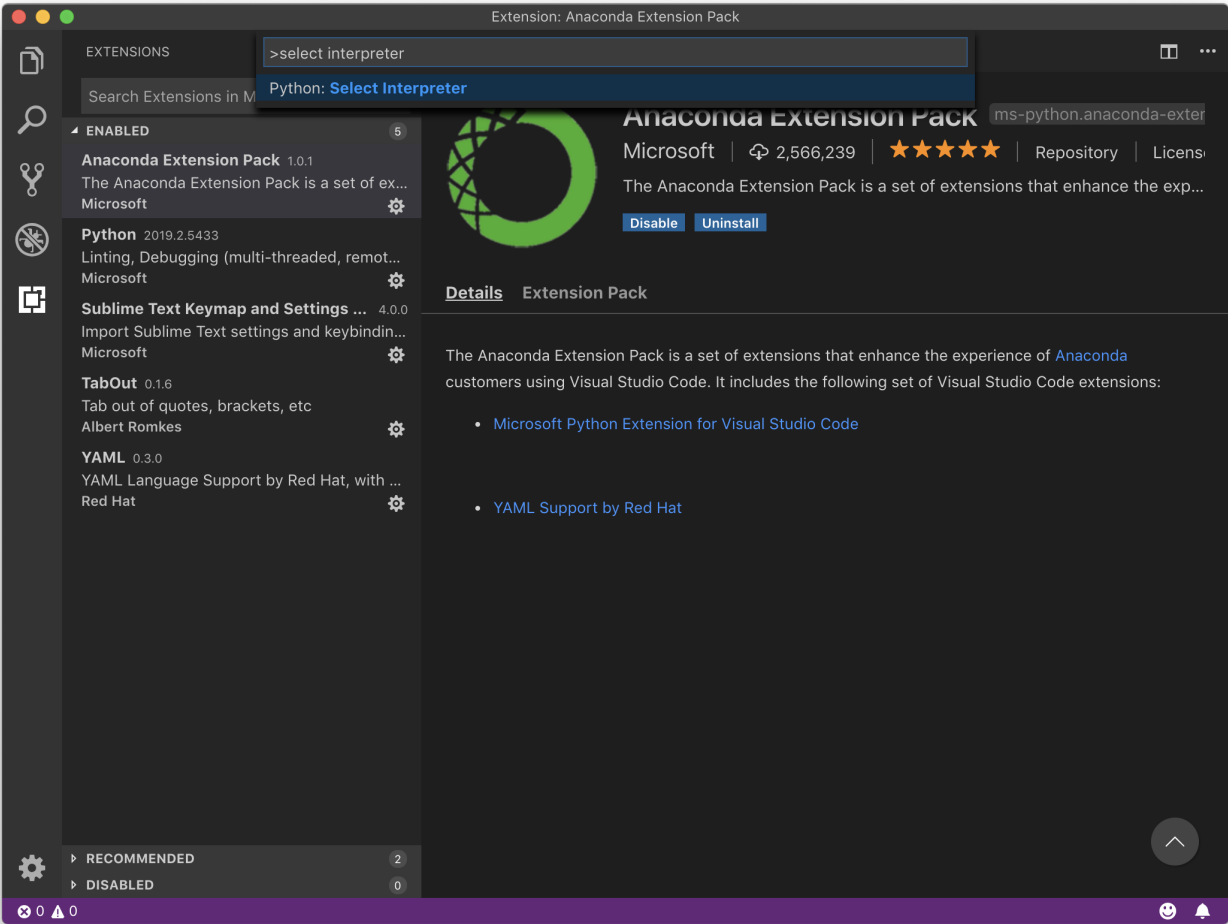
## 允许命令行启动 VS Code

使用快捷键 `⇧⌘P` 呼出 Command Palette，在其中输入 `shell command`，而后选中 `Install 'code' command in PATH`。此后，就可以在 Terminal 命令行中使用 `code` 命令了。(Windows 系统安装 VS Code 时会自动配置好，并不需要此步骤)

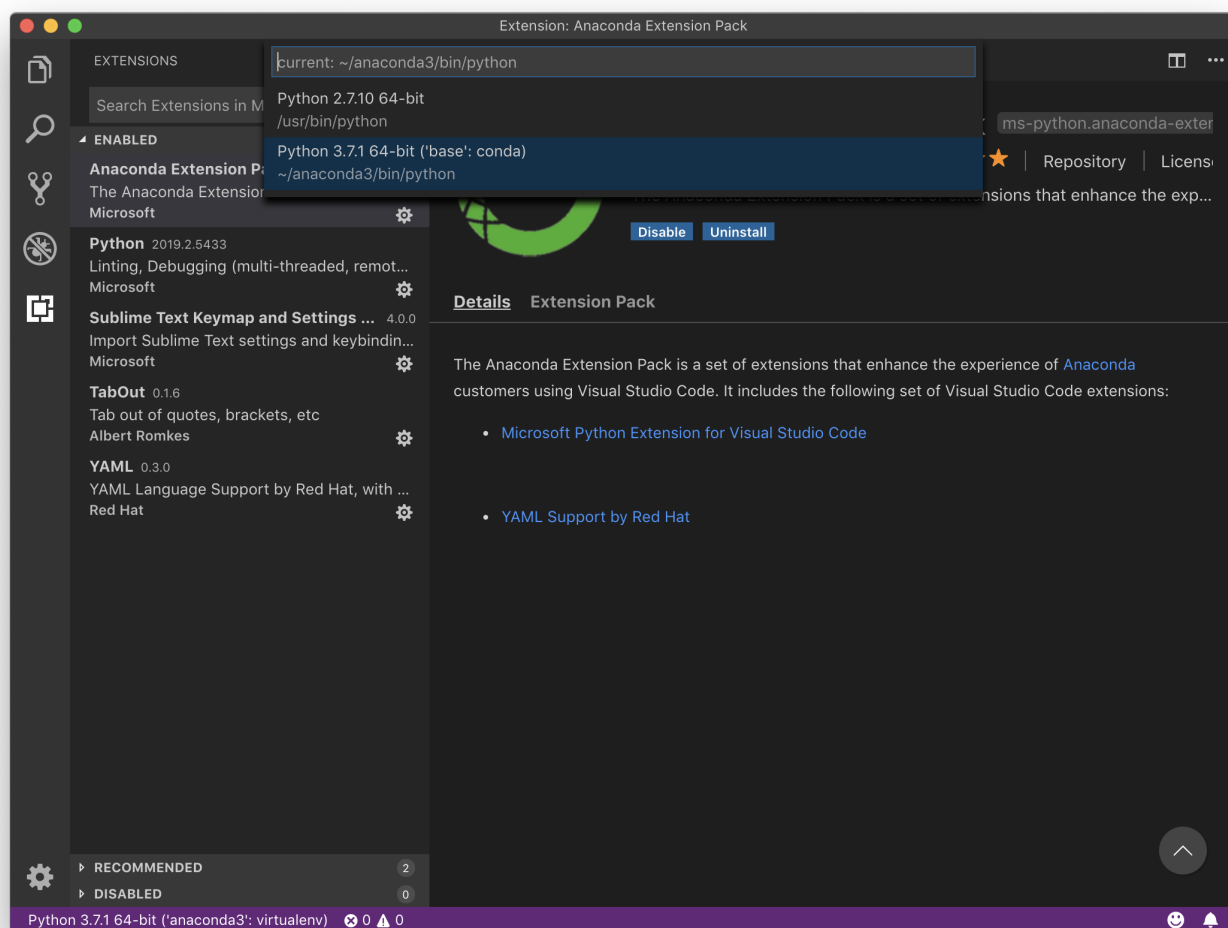


## 选择 Python 解析器版本

使用快捷键 `⇧⌘P` 呼出 Command Palette，在其中输入 `select interpreter`，而后选中 `Python: Select Interpreter`。

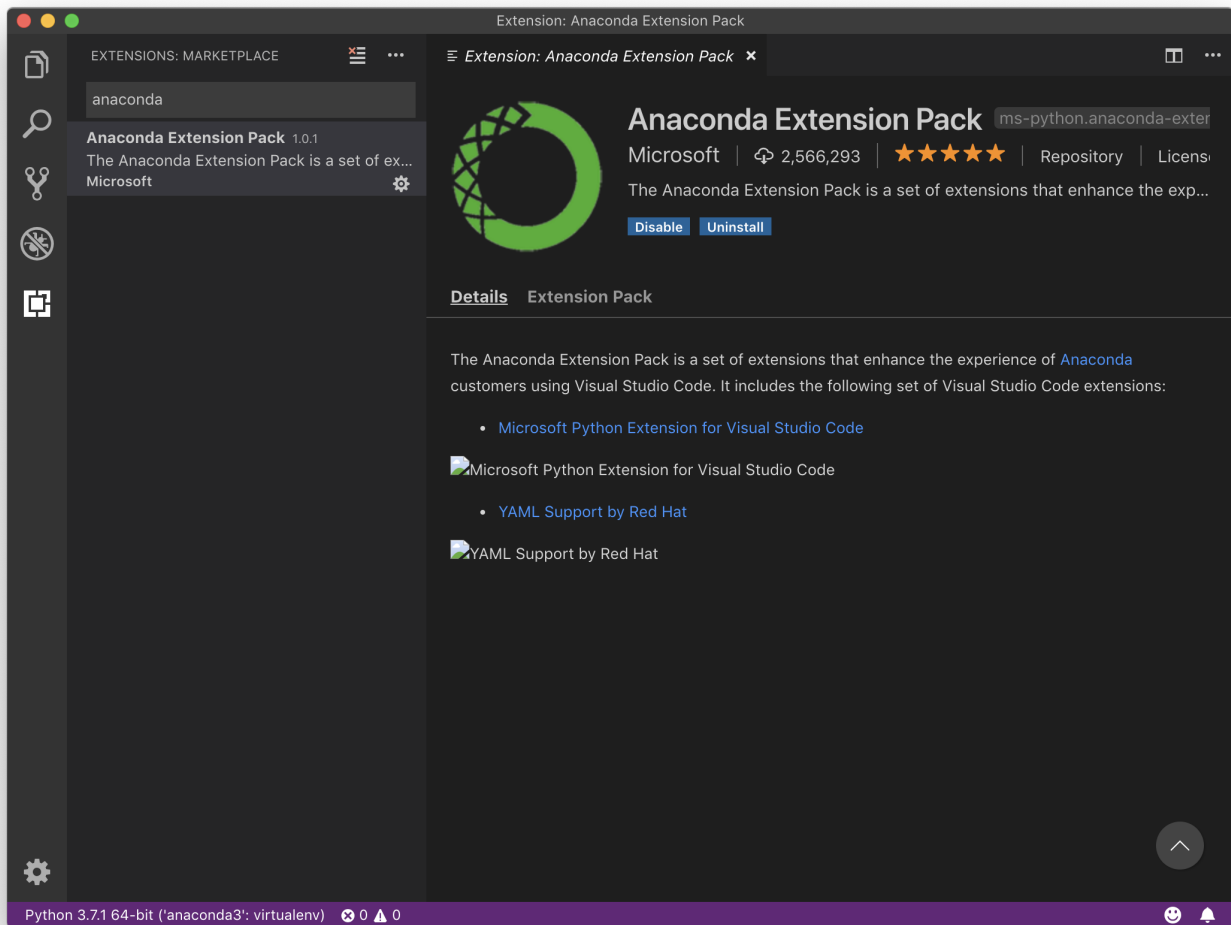


而后，在系统中已安装的若干个版本中选择你需要的那一个。MacOS 系统自带一个 Python 2.7，而我们安装的 Anaconda 为系统另外安装了一个 Python 3.7。



## 安装扩展

使用快捷键 **⇧⌘x** 呼出扩展面板。安装 **anaconda** 扩展，它会连带装上 **python** 扩展：



另外，为了输入方便，有两个扩展可选安装：

- Tabout 有它之后，可以使用 TAB 键跳出光标后的括号、引号等等；
- Sublime Text Keymap and Settings Importer 有它之后，可以在 VS Code 中使用 SublimeText 的快捷键，最重要的当属多光标编辑 `⌘⇧1`.....

## 自动补全

专业编辑器最重要的功能之一，就是能够在你输入的时候它帮你做到“自动补全”，通常使用的快捷键是 TAB 键 `→`。

TAB 键 `→` 触发的自动补全有两种：

- 当前文件中已有的字符串。比如，之前你输入过 `sum_of_word`；那么，之后，你就可以输入 `su` 或者干脆 `sow` 而后按 TAB 键 `→`，“自动补全”功能会帮你完成输入 `sum_of_word`
- 已有的 Snippets。比如，当你需要输入 `if ...: ...` 的时候，实际上当你输入 `if` 或者甚至 `i` 之后，你就可以用 TAB 键 `→`，“自动补全”功能会为你“自动完成”语句块的输入。

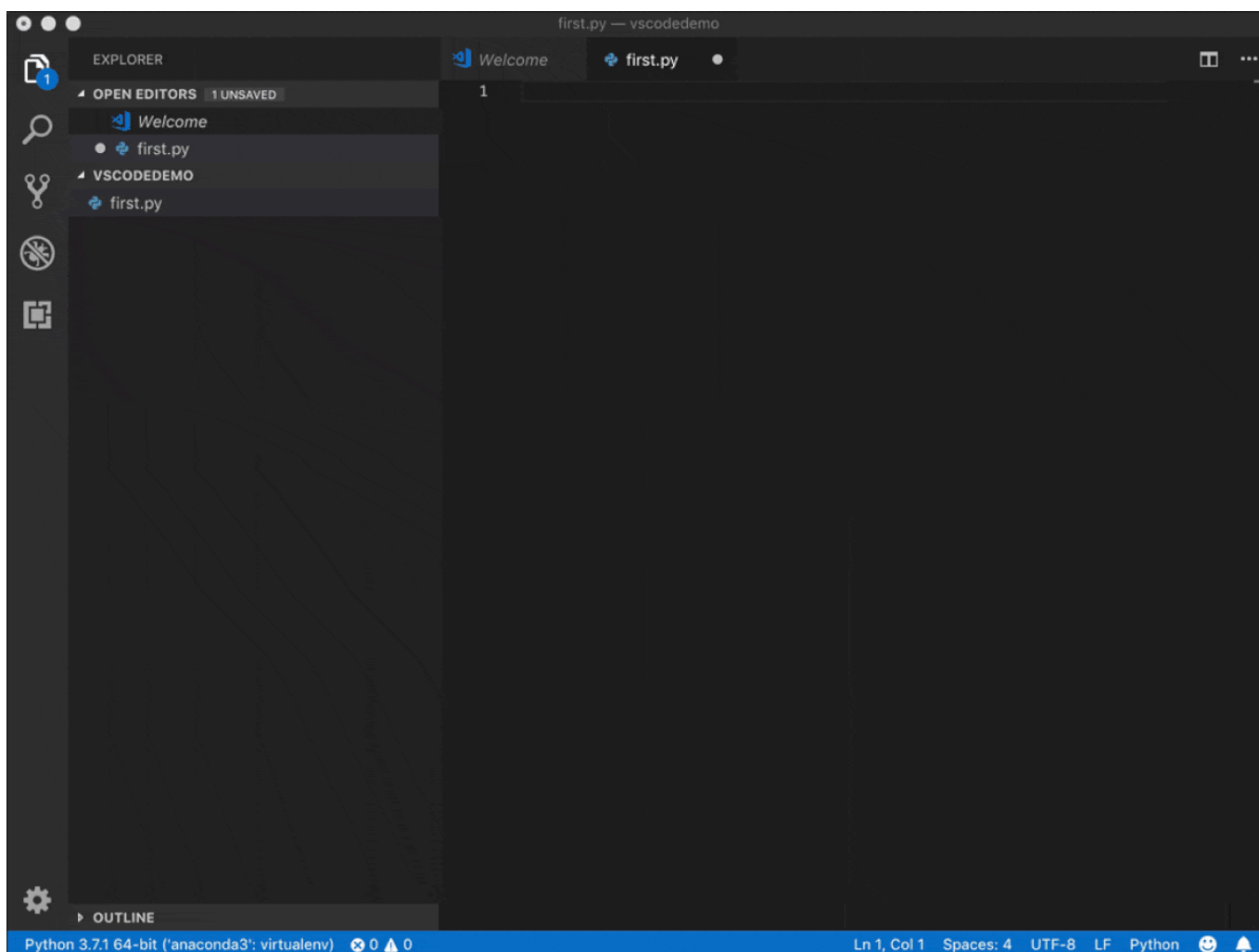
字符串自动补全，使用的是所谓的 Fuzzy Match。输入 `sum_of_word` 中所包含的任意字符的任意组合（按顺序），它都会尽量去匹配；所以，`su` 和 `sow` 都可以匹配 `sum_of_word`，再比如，`rst` 可以匹配 `result`。

在 Snippet 自动补全的过程中，常常有若干个“TAB Stop”，即，有若干个位置可以使用 TAB 键 `→`（或者，`Shift + →`）来回切换；这时，第一种字符串自动补全的功能就失效了，如果需要使用字符串自动补全，那么需要按快捷键 `ESC` 退出 Snippet 自动补全模式。

以下的 gif 文件演示的是以下代码的输入过程：

```
def sum_of_word(word):  
    sum = 0  
    for char in word:  
        sum += ord(char) - 96  
    return sum  
with open('results.txt', 'w') as results:  
    with open('words_alpha.txt', 'r') as file:  
        for word in file.readlines():  
            if sum_of_word(word.strip()) == 100:  
                results.write(word)
```

因为有这样的功能，所以你在输入程序的时候其实是非常从容的，可以很慢输入，边思考边输入..... 可实际上，完成速度却很快。



另外，SublimeText 的多光标输入是很多程序员爱不释手的功能，于是，各种编辑器里都有第三方写的 SublimeText Keymap 插件，连 Jupyterlab 都有：

[https://github.com/ryantam626/jupyterlab\\_sublime](https://github.com/ryantam626/jupyterlab_sublime)