

Solution-编辑器 (Vim)

1. 完成 vimtutor。备注：它在一个 80x24 (80 列, 24 行) 终端窗口看起来最好。

```
vimtutor
```

2. 下载我们的 [vimrc](#)，然后把它保存到 ~/.vimrc。通读这个注释详细的文件（用 Vim!），然后观察 Vim 在这个新的设置下看起来和使用起来有哪些细微的区别。

3. 安装和配置一个插件：ctrlp.vim.

1. 用 `mkdir -p ~/.vim/pack/vendor/start` 创建插件文件夹

2. 下载这个插件：`cd ~/.vim/pack/vendor/start; git clone`

`https://github.com/ctrlpvim/ctrlp.vim`

下载后需要在 ~/.vimrc 中添加如下设置，参考[这里](#)

```
set runtimepath^=~/.vim/pack/vendor/start/ctrlp.vim
```

3. 请阅读这个插件的[文档](#)。尝试用 CtrlP 来在一个工程文件夹里定位一个文件，打开 Vim, 然后用 Vim 命令控制行开始 :CtrlP.

```

1  # 3-Solution-编辑器 (Vim)
1  ### 1. 完成 vimtutor。备注： 它在一个 80x24 (80 列, 24 行)
   终端窗口看起来最好。
2  ```bash
3  vimtutor
4  ```
2  5  ### 2. 下载我们的 [vimrc](https://missing-semester-cn.github.io/2020/files/
   vimrc)，然后把它保存到 `~/.vimrc`。通读这个注释详细的文件（用 Vim!），
   然后观察 Vim 在这个新的设置下看起来和使用起来有哪些细微的区别。
6
3  7  ### 3. 安装和配置一个插件： `ctrlp.vim`。
3  8  1. 用 `mkdir -p ~/.vim/pack/vendor/start` 创建插件文件夹
3  9  2. 下载这个插件： `cd ~/.vim/pack/vendor/start; git clone https://github.
   com/ctrlpvim/ctrlp.vim`
3  10 3. 请阅读这个插件的 [文档](https://github.com/ctrlpvim/ctrlp.vim/blob/
   master/readme.md)。尝试用 CtrlP 来在一个工程文件夹里定位一个文件， 打开
   Vim, 然后用 Vim 命令控制行开始 :CtrlP.
3  11 下载后需要在 ~/.vimrc 中添加如下设置，参考 [这里](http://ctrlpvim.github.
   io/ctrlp.vim/#installation)
3  12
NORMAL #3-editors-solution.md          unix | utf-8 | pandoc   3%   1:1
> missing/debug_for.sh
> missing/debug.sh
> solutions/#7-debugging-profiling-solution.md
prt  path <mru>={ files }=<buf> <-> <ngfeng/GitDownload/The-Missing-Solutions
>>> debug
0 ↑ 26d 13h 54m ██████████ 100% | 22:41 | 12 5 lingfeng lingfenaideAir2
  
```

4. 自定义 CtrlP： 添加 [configuration](#) 到你的 ~/.vimrc 来用按 Ctrl-P 打开 CtrlP

```
let g:ctrlp_map = '<c-p>'
let g:ctrlp_cmd = 'CtrlP'
let g:ctrlp_working_path_mode = 'ra' #设置默认路径为当前路径
```

```
1 # 3-Solution-编辑器 (Vim)
2 ### 1. 完成 vintutor。备注： 它在一个 80x24 (80 列, 24 行)
3 终端窗口看起来最好。
4 ```bash
5 vintutor
6 ```
7 ### 2. 下载我们的 [vimrc](https://missing-semester-cn.github.io/2020/files/
8 vimrc), 然后把它保存到 `~/.vimrc`。通读这个注释详细的文件 (用 Vim!),
9 然后观察 Vim 在这个新的设置下看起来和使用起来有哪些细微的区别。
10
11 ### 3. 安装和配置一个插件: `ctrlp.vim`.
12 1. 用 `mkdir -p ~/.vim/pack/vendor/start` 创建插件文件夹
13 2. 下载这个插件: `cd ~/.vim/pack/vendor/start; git clone https://github.com/ctrlp
14 NORMAL #3-editors-solution.md unix | utf-8 | pandoc 3% 1:1
15 > missing/debug_for.sh
16 > missing/debug.sh
17 > solutions/#6-version-control-solution.md
18 > solutions/#4-data-wrangling-solution.md
19 > solutions/#1-course-shell-solution.md
20 > solutions/#5-command-line-solution.md
21 > solutions/#2-shell-tools-solution.md
22 > solutions/#10-potpourri-solution.md
23 > solutions/#9-security-solution.md
24 > solutions/#11-qa-solution.md
25 prt path <mru>={ files }=<buf> <-> <ngfeng/GitDownload/The-Missing-Solutions
26 >>> md
27 0 ↑ 26d 14h ↑ 100% | 22:47 | 12 5 lingfeng lingfenaideAir2
```

4. 练习使用 Vim, 在你自己的机器上重做演示。
5. 下个月用 Vim 完成_所有_的文件编辑。每当不够高效的时候, 或者你感觉 “一定有一个更好的方式”, 尝试求助搜索引擎, 很有可能有一个更好的方式。如果你遇到难题, 来我们的答疑时间或者给我们发邮件。
6. 在你的其他工具中设置 Vim 快捷键 (见上面的操作指南)。
7. 进一步自定义你的 ~/.vimrc 和安装更多插件。安装插件最简单的方法是使用 Vim 的包管理器, 即使用 vim-plug 安装插件:
 1. 安装 vim-plug

```
$ curl -fLo ~/.vim/autoload/plug.vim --create-dirs \
https://raw.githubusercontent.com/junegunn/vim-plug/master/plug.vim
```

2. 修改 ~/.vimrc

```
call plug#begin()
Plug 'preservim/NERDTree' #需要安装的插件 NERDTree
Plug 'wikitopian/hardmode' #安装 hardmode
..... # 更多插件
call plug#end()
```

3. 在 vim 命令行中执行 :PlugInstall

```

1 Installing plugins (31/54)
2 [=====
1
2 * vim-matchup: Installing ...
3 - indentline: Already installed
4 - vim-yaml: Already installed
5 - vim-colorschemes: Already installed
6 - vim-sensible: Already installed
7 - gruvbox: Already installed
8 - vim-gitgutter: Already installed
9 - vim-unimpaired: Already installed
10 - incsearch.vim: Already installed
11 - editorconfig-vim: Already installed
12 - vista.vim: Already installed
13 - vim-speeddating: Already installed
14 + vim-easymotion: Installing ...
15 - ansible-vim: Already installed
16 - securemodelines: Already installed
17 - vim-startify: Already installed
18 - python-syntax: Already installed
19 - ale: Already installed
20 - colorizer: Already installed
21 - hardmode: Already installed
22 - vim-argwrap: Already installed
< unix | utf-8 | vim-plug 5% 2:1 .vimrc 4% 19:1
:PlugInstall
0 ↑ 26d 14h 19m 100% | 23:06 | 12 5 lingfeng lingfenaideAir2

```

8. (高阶)用 Vim 宏将 XML 转换到 JSON ([例子文件](#))。尝试着先完全自己做，但是在你卡住的时候可以查看上面 [宏](#) 章节。

1. Gdd, ggdd 删除第一行和最后一行

2. 格式化最后一个元素的宏 (寄存器 e)

跳转到有 <name> 的行 qe^r"f>s": "<ESC>f<C"<ESC>q

3. 格式化一个人的宏

跳转到有 <person> 的行 qpS{<ESC>j@eA,<ESC>j@ejS},<ESC>q

4. 格式化一个人然后转到另外一个人的宏

跳转到有 <person> 的行 qq@pjq

5. 执行宏到文件尾 999@q

6. 手动移除最后的, 然后加上 [和] 分隔符