

Vim笔记

- [Vim笔记](#)
 - [Vim模式](#)
 - [Vim打开和切换文件](#)
 - [Vim退出](#)
 - [常用快捷键](#)
 - [Vim分屏](#)
 - [复制粘贴](#)
 - [查找和替换](#)
 - [查找](#)
 - [替换](#)
 - [代码折叠](#)
 - [拼写检查](#)

Vim作为Linux系统自带的文本编辑器，掌握如何使用是很有必要的。

Vim模式

- **Normal** 模式：进入Vim后的一般模式。
- **Insert** 模式：按下 `i` 键后进入插入模式，可以修改文档。
- **Visual** 模式：按下 `v` 键后进入选择模式，可以选择文档内容

Vim打开和切换文件

1. 终端 `vim file1 file2 ...` 可以打开多个文件。
2. `:ls` 显示打开的文件，可以使用 `:bn` 在文件间切换(`n`也可以换成 `:ls` 里给出的文件序号)。
3. 打开vim后，可以用 `:e fileName` 来打开文件或者新建文件。
4. 在终端 `vim -o file1 file2 ...` 可以打开多个文件(横向分隔屏幕)。
5. 终端 `vim -O file1 file2 ...` 可以打开多个文件(纵向分隔屏幕)。
6. `Ctrl + w + w` 在窗口间切换光标，第二个 `w` 也可以用 `h、j、k、l` 来光标代表移动方向。

Vim退出

1. `:q`：退出。

2. `:q!` : 强制退出, 放弃所有修改。
3. `:wq` : 保存修改并退出。

常用快捷键

1. 方向键`←↓↑→`也可以用 `h`、`j`、`k`、`l` 。
2. `0` 到行首, `$` 到行尾。
3. `gg` 到文档首行, `G` 到文档结尾。
4. `Ctrl + f` 下一页, `Ctrl + b` 上一页。
5. `Ctrl + u` 往上半页, `Ctrl + d` 往下半页。
6. `w` 或 `e` 光标往后跳一个单词, `b` 光标往前跳一个单词。
7. `:98` 跳转到第98行。
8. `q:` 显示命令行历史记录窗口。
9. `!bash Command` 不退出vim暂时返回终端界面执行该命令。
10. `H` 将光标移动到屏幕首行, `M` 将光标移动到屏幕中间行, `L` 将光标移动到屏幕最后一行

Vim分屏

1. `:sp` 或者 `:split` 横向分隔屏幕, 后面可以加文件名。
2. `:vs` 或者 `:vsplit` 纵向分隔屏幕, 后面可以加文件名。
3. `:only` 只保留光标所在分屏, 关闭其他分屏。
4. 在`nerdtree`插件中, 选中文件后按 `s` 纵向分隔屏幕, 按 `i` 水平分隔屏幕。

复制粘贴

1. 在**Visual**模式下选择文档内容后按 `y` 键, 复制被选择内容。
2. 按 `p` 键粘贴, 注意粘贴从紧跟光标后的那个字符之后才开始。

`| abc | d "COPIED_TEXT" efghk... (| 是光标)`

3. 选择内容后按 `d` 删除或者剪贴。
4. `yy` 复制当前行, `dd` 删除(剪贴)当前行。

查找和替换

查找

1. 在**Normal**模式下, 按 `/` 进入查找模式, 输入 `/word` 后回车, 高亮显示所有文档 `word`, 按 `n` 跳到下一个 `word`, 按 `N` 跳到上一个。

2. 若输入 `/word\c` 代表大小写不敏感查找, `\c` 代表大小写敏感。
3. 输入 `:noh` 取消高亮。
4. 按下 `*` 高亮查找光标位置处的单词, 但若查找 `word`, `helloworld` 中的 `word` 不会被高亮。
5. 按下 `g*` 高亮查找光标位置处的单词, 若查找 `word`, `helloworld` 中的 `word` 也会被高亮。
6. 在**Normal**模式下按 `q + /` 显示查找历史记录窗口。

替换

1. `:s/word/excel` : 替换当前行所有 `word` 为 `excel` 。
2. `:s/word/excel/g` : 替换当前行第一个 `word` 为 `excel` , `/g` 代表只替换每行第一个
3. `:%s/word/excel` : 替换全文所有 `word` 为 `excel` 。
4. `:%s/word/excel/gc` : 其中 `/c` 代表需要确认, 并提示: `replace with excel (y/n/a/q/l/^E/^Y)?` , 其中 `a` 表示替换所有, `q` 表示退出查找模式, `l` 表示替换当前位置并退出, `^E` 与 `^Y` 是光标移动快捷键。
5. `:2,11s/word/excel` : 替换第2到11行的 `word` 为 `excel` 。
6. `:$s/word/excel` : 替换当前行到最后一行的 `word` 为 `excel` , `.` 代表当前行, `$` 代表最后一行。
7. `:+2s/word/excel` : 替换当前行与接下来2行的 `word` 为 `excel` 。
8. 在**Visual**模式下选择后按 `:` , Vim自动补全为 `:'<,'>` , 然后手动补全 `:'<,'>s/word/excel` , 将选择区域的 `word` 替换为 `excel` 。
9. `:s/word/excel/i` : `/i` 代表大小不敏感查找, 等同于 `:s/word\c/excel` , 而 `/I` 代表大小写敏感查找。

代码折叠

代码折叠是**Vim**的高级功能, 意思就是即便没掌握也不影响**Vim**的正常使用。

1. `set foldenable` : 打开代码折叠功能, 可以写在根目录下的**.vimrc**文件里。
2. `set foldmethod=syntax` : 设置折叠方式(`foldmethod` 可缩写为 `fdm`), 常用的有 `syntax` : 按C\C++语法折叠; `indent` : 按缩进折叠, 适用于Python。
3. `set foldlevelstart=99` : 打开文件时不自动折叠代码。
4. `zc` & `zo` : 在**Normal**模式下将光标移动到代码的可折叠位置, 按 `zc` 折叠代码(close), `zo` 打开折叠(open)。

小技巧: 将 `nnoremap <space> @=((foldclosed(line('.')) < 0) ? 'zc' : 'zo')<CR>` 写入**.vimrc**文件, 可将 `zc` 和 `zo` 映射为空格键。

拼写检查

对于英文单词的拼写, **Vim** 可以自动检查拼写 (**Spell Checking**)。

1. `set spell` : 打开拼写检查, 可以写在根目录下的 **.vimrc** 文件里, 不过更推荐在需要使用时在 **Vim**

中手动打开 `:set spell`。

2. 在 `.vimrc` 文件里写入 `inoremap <C-l> <c-g>u<Esc>[slz=`]a<c-g>u` 来将 `Ctrl+l` 映射为快捷键，其中 `[s` 代表光标跳到上一个拼写错误处，`lz=` 选择第一个推荐的正确拼写，``]a` 光标跳回原来位置。设置完就可以通过 `Ctrl+l` 快速改正单词。