vim

光标移动

光标移动

字符移动：

- `w`：移动到后一个单词开头
- `b`：向前移动一个单词开头
- `e`：移动到后一个单词末尾

e和a搭配使用迅速在单词末尾进行插入

行内移动：

- `0`：移动到行首
- `$`：移动到行尾

快速跳转：

- 文件内跳转
  - `gg`：移动到文件开头
  - `G`：移动到文件末尾
  - `数字+G`：移动到指定行
  - `g;`：跳转到上一个修改点
  - `g,`：跳转到下一个修改点
  - `{`：移动到上一段开头
  - `}`：移动到下一段开头
  - `Crtl+f`：向下翻一页
  - `Crtl+b`：向上翻一页
  - `Crtl+o`: 返回到上一次跳转处
- 屏幕内跳转
  - `H`：移动到屏幕顶端
  - `M`：移动到屏幕中间
  - `L`：移动到屏幕底端
  - `zz`：将光标置于中心

插入模式

- `i`：在当前光标前插入
- `a`：在当前光标后插入
- `A`：在当前行末尾插入
- `o`：在当前行下方新建一行并进入插入模式
- `O`：在当前行上方新建一行并进入插入模式

A和o都很好用，注意灵活使用

文本编辑

删除

- `x`：删除当前光标下的字符
- `dd`：删当前行
- `dw`：删除一个单词
- `D`：删除从光标到行尾的内容

复制粘贴

- `yy`：复制当前行
- `yw`：复制一个单词
- `p`：在光标后粘贴

更精细的删除，复制可以配合可视模式
vim中所有的删除，包括x与d等，都会将删除的内容放在寄存器中

撤销

- `u`：撤销上一步操作
- `Ctrl+r`：重做上一次撤销的操作

搜索与替换

搜索

- `/text`：向下搜索text
- `?text`：向上搜索text
- `n`：跳转到下一个匹配项
- `N`：跳转到上一个匹配项
- `#`：向上搜索光标所在单词
- `*`：向下搜索光标所在单词

替换 `:[range]s/pattern/replacement/[flags]`

- `s`：subsitute的缩写，表示进行替换操作
- `range`：指定替换的范围
- `flags`：控制替换的标志

`range`：

- `空`：只对当行进行操作
- `%`：对整个文件进行操作
- `n1,n2`：从n1行到n2行进行操作
- `g`：从光标到文件末尾

`flags`：

- `g`：全局替换（global）

- `c`：替换前确认（confirm）
- `i`：不区分大小写
- `:%s/old/new/g`：将整个文件中的old替换为new
- `:%s/old/new/gc`：在整个文件中替换，每次替换前询问（c是confirm的缩写）

多文件操作

缓冲区

- `e 文件名`：创建新缓冲区并打开一个新文件
- `:ls`：列出所有缓冲区
- `:bn`：切换至下一个缓冲区（buffer next）
- `:bp`：切换至上一个缓冲区（buffer previous）
- `Ctrl+^`：切换到上一个活跃的缓冲区，在频繁在两个文件之间切换时非常有用
- `:bN`：切换到编号为N的缓冲区
- `:b文件名`：切换到该文件的缓冲区
- `:bd`：关闭当前缓冲区

视图

- `:sp 文件名`：水平分割窗口并打开文件
- `:vsp 文件名`：垂直分割窗口并打开文件
- `Crtl+w+w`：在分割的窗口之间切换，切换至下一个
  - `Crtl+w+h`：切换至左边一个
  - `Crtl+w+j`：切换至下边一个
  - `Crtl+w+k`：切换至上边一个
  - `Crtl+w+l`：切换至右边一个
- `:q`：关闭当前窗口
- `:only`：保留当前窗口，关闭其他所有窗口

标签页

- `:tabnew 文件名`：在新标签页打开文件
- `:tabs`：列出所有标页
- `gt`：切换到下一个标签页
- `gT`：切换到上一个标签页
- `Ngt`：切换到指定编号的缓冲区
- `tabc`：关闭当前标签页

---

vim-plug

专为vim设计的轻量、快速的插件管理器，解决了插件难安装，难管理的问题（没有的话需要手动下载文件、解压，然后放在正确的位置，非常麻烦，且难以管理）

在.vimrc文件中加上 `plug#begin()`和 `plug#end()`两个函数，然后将想安装的插件放在这两个函数之间

- `:PlugInstall`：读取安装声明，自动下载插件
- `:PlugUpdate`：检查下载并进行更新
- `:PlugClean`：移除声明外的插件
- `:PlugStatus`：显示插件的安装状态以及是否懒加载等

懒加载：
如果想让vim启动的更快，可以在Plug声明之后加上 `{ 'on': '...'}`，来指定插件在什么命令下被加载

---

fzf.vim

fzf以其强大的模糊搜索闻名，拥有fzf和fzf.vim之力，仅用零碎几个字符即可定位到想查找的信息

- `:Files`：输入文件名的一部分，会立即过滤出包含该字符的文件
  - `Enter`：在当前窗口打开文件
  - `Ctrl+v`：垂直分割窗口并打开文件
  - `Ctrl+x`：水平分割窗口并打开文件
- `:Lines`：输入任何函数名，变量名或者文本片段，可以快速定位到所在行，按enter进行跳转，比传统的`/`好用一点

此外还有 `:Buffers`和 `:History`以及git相关搜索命令，不常用

---

vim-commentary

快速方便的注释和取消注释代码

使用 `gc`自动识别文件类型，注释或取消注释该行代码

---

coc.nvim

coc.nvim是一个专为vim和neovim设计的语言服务器协议（LSP）客户端，实现将vim连接到各种编程语言的语言服务器，获得智能代码补全、语法检查、代码跳转、重构等一系列功能

- `gd`：跳转到光标所在函数或变量的定义处（go definition）
- `rn`：对光标所在函数或变量进行重命名
