

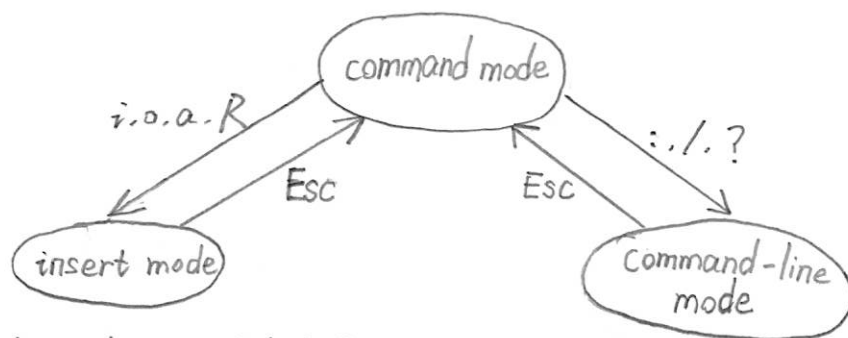
Lec 09 vim 程序编辑器

vi与vim

vim 是 vi 的进阶版本。

vi 的使用

- 一般指令模式 command mode
- 编辑模式 insert mode
- 命令行模式 command-line mode



△ command mode 的按键说明

100 ↓ : 向下移动 100 行, 并以此类推

ctrl + f : 向下 - 页

ctrl + b : 向上 - 页

0 : 光标移动到这一行的最前面

\$: 光标移动到这一行的最后面

30 G / 30 gg : 移动到这个文件的第 30 行

4 [Enter] : 向下移动 4 行

/jasmine : 向下搜索 jasmine (按 [Enter] 开始)

?fragrans : 向上搜索 fragrans (按 [Enter] 开始)

} n: next
N: 反向 next

x : 向后删除一个字符 (相当于 [Delete])

X : 向前删除一个字符

dd : 删除光标所在的一整行

3dd : 删除光标及光标之下的 2 行 (共 3 行)

d1G : 删除从开头到光标所在行 (连同光标所在行)

dG : 从光标所在行删到最后一行

d\$: 从光标删到该行的最后

do : 从该行的开头删到光标之前一个字符

yy : 复制光标所在的那一行

p : 将已复制的数据贴到光标的下一行

P : 将已复制的数据贴到光标的上一行

→ 同理, 4yy, y1G, yG, y0, y\$

u : 复原前一个操作 (撤销)

ctrl+r : 反撤销

. : 重复前一个操作

△ command mode → insert mode

i : 从当前光标处插入

A : 从光标所在行的最后一个字符插入

O : 从光标所在行的下一行插入一行

O : 从光标所在行的上一行插入一行

Esc : 退出 insert mode

△ command mode → command-line mode

:w 写数据

:w! 强行写入

:q 离开 vi

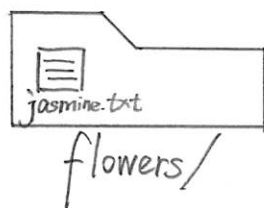
:q! 强行离开不写入

⊗⊗ 若文件没有改动, 则不保存离开; 若文件已被改动, 则保存后离开。

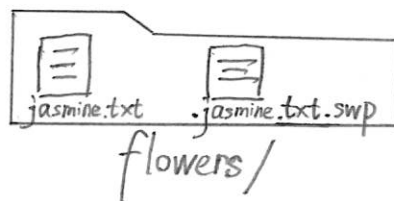
:w <filename> 将编辑的数据保存为另一个文件

:r <filename> 将 <filename> 的文件内容加到光标所在行的后面

vim 的暂存档、救援回复与开启时的警告信息



vim jasmine.txt



kill vim 进程

vim jasmine.txt

“found a swap file by the name”

[R] : recover

[D] : delete

[O] : open read-only

[E] : edit anyway

[Q] : quit

vim的额外功能

区块选择 (visual block)

v : 字符选择, 将光标经过的地方反白

V : 行选择, 将光标经过的行反白

ctrl+v : 以长方形的方式选择 (shift+alt)

y : 将反白的地方复制起来

d : 将反白的地方删掉

p : 将刚刚复制的数据贴到光标处

多文件编辑


vim <f1> <f2>

:n 下一个文件

:N 上一个文件

:files 列出这个vim开启的所有文件

多窗口功能

场景1:  需要同时看这两个地方, 却不想频繁翻页

场景2: 有两个文件要对照着看, 却不想用上一节的“多文件编辑”

:sp

:sp <filename>

ctrl+w+↑ 光标移到上面的窗口

ctrl+w+↓ 光标移到下面的窗口

ctrl+w+q 离开光标所在的窗口

vim的环境设定与记录: ~/.vimrc 与 ~/.viminfo