# Vim学习笔记：

Vi是一个简单的文档编辑器，vim是vi的高级版，可以使用颜色高亮显示程序源文件，可以认为是程序编辑器。

vi m.txt 打开m.txt文档，如果当前目录下有这个文件则打开该文件，如果没有这个文件就会新建一个文件。

观察窗口最后一行，这个是状态栏，不属于文档的编辑区，显示内容表示m.txt是一个新建的文件。此时，vim编辑区处于一般模式，按i进入编辑模式，左下角显示，表示进入编辑模式。编辑文本后，按Esc退回到一般模式，输入：，即Ctrl+； 进入命令行模式，然后键入wq保存文档并退出(write quit)。在一般模式下按ＺＺ也可保存文件退出。

可以看到，vim一共有3种模式，其中打开文档时，处于一般模式，此时不能编辑文档，但可以在文档中移动光标，并对文档进行删除字符（x）、删除整行（dd），复制和粘贴操作。按下i后进入编辑模式，此时可以对文档进行任何编辑、删除、复制粘贴操作。命令行模式就是执行一些保存和退出指令的命令。这3种模式中命令行模式和编辑模式之间无法相互切换，只能如下切换。

命令行模式

编辑模式

一般模式

再次vi m.txt,表面此时文档有4行，31个字符。

尽量使用h j k l这几个字母来移动光标。

nx：n是数字，即删除光标后的n个字符

yy:复制光标所在行，y0复制光标到所在行行首的所有字符，y$复制到行尾。

u:撤销上一次操作。

ctrl+r：重复上一次操作。

ctrl+u/d：翻半页 f/b全页

i:i插入到光标前，I插入到本行第一个非空字符前。

a：插入到光标后，A插入到本行最后一个非空字符后。

o：在光标所在行后另起一行插入，O前一行。

/patten: 查找文本中的patten字符，n/N为将光标移到下一处或者上一处。

本行移动光标：

0：本行第一个字符

^:第一个非空字符处

$:本行最后一个字符处

g\_：本行最后一个非空字符处

f m：快速定位到本行的m字符位置，f是命令，m是参数。类似F t T。

以变量为参考移动：

e：移动到光标所在变量的最后一个字符。E变量以空白字符为间隔划分。

w：移动到下一个变量的起始字符处，W以空白字符为间隔划分变量。

变量匹配移动：

%：匹配括号移动，包括 (, {, [. （陈皓注：你需要把光标先移到括号上）#/\*：移动到当前光标所在单词的.

\* 和 #:  匹配光标当前所在的单词，移动光标到下一个（或上一个）匹配单词（\*是下一个，#是上一个）

以行为单位移动光标：

N enter：向下移动N行

NG：移动到第N行

gg:移动到第一行

G:移动到最后一行

常用命令：以下N表示任意数字

x：删除当前字符，X删除前一个字符

y：复制，不能单独使用，配合yy复制本行，y0,复制光标所在处至本行第一个字符，y$复制当前位置至最后一个字符。

p:从光标后开始粘贴，P从光标前开始粘贴。

dd：删除一行，d0 d$,同y

u: undo 撤销操作

v:选择操作，进入选择模式。V选中当前行。N v 选中N行。

. :重复上一次操作。

J：将光标所在行与下一行数据合并成一行。

h j k l ：4个方向移动光标，Nj，向上移动N行

使用的命令行均可以N command，执行Ｎ此操作。

有些操作需要最后按　Esc，如连续10次输入lucky：10ilucky Esc

全局替换命令：:[range]s/pattern/string/[c,e,g,i]

g 大概都是要加的，否則只會替換每一行的第一個符合字串。可以合起來用，如 cgi，表示不分大小寫，整行替換，替換前要詢問是否替換。

vi/vim 中可以使用 :s 命令来替换字符串。以前只会使用一种格式来全文替换，今天发现该命令有很多种写法（vi 真是强大啊，还有很多需要学习），记录几种在此，方便以后查询。

:s/vivian/sky/ 替换当前行第一个 vivian 为 sky

:s/vivian/sky/g 替换当前行所有 vivian 为 sky

:n,$s/vivian/sky/ 替换第 n 行开始到最后一行中每一行的第一个 vivian 为 sky

:n,$s/vivian/sky/g 替换第 n 行开始到最后一行中每一行所有 vivian 为 sky

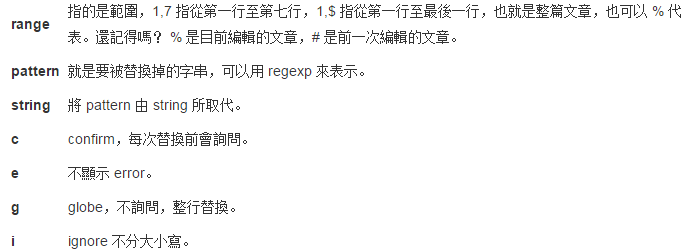
n 为数字，若 n 为 .，表示从当前行开始到最后一行

:%s/vivian/sky/（等同于 :g/vivian/s//sky/） 替换每一行的第一个 vivian 为 sky

:%s/vivian/sky/g（等同于 :g/vivian/s//sky/g） 替换每一行中所有 vivian 为 sky

可以使用 # 作为分隔符，此时中间出现的 / 不会作为分隔符

:s#vivian/#sky/# 替换当前行第一个 vivian/ 为 sky/



很多命令都可以和这些移动光标的命令连动。很多命令都可以如下来干：

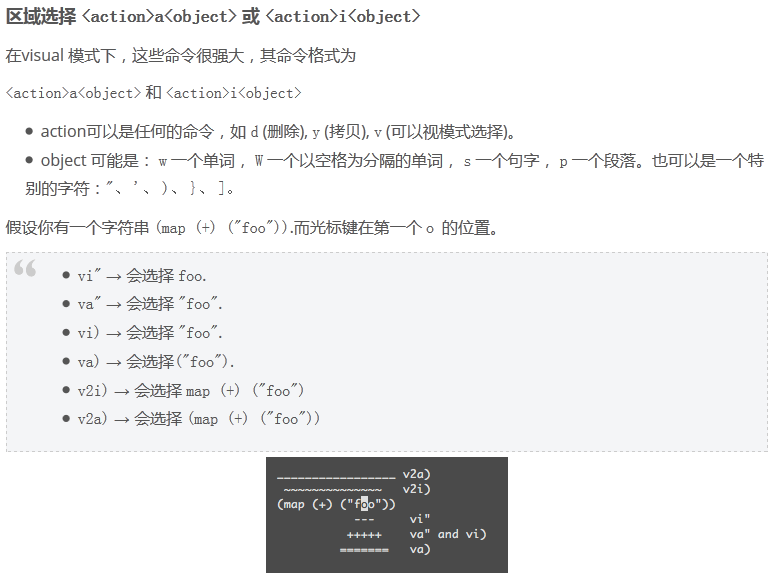
<start position><command><end position>

例如 0y$ 命令意味着：

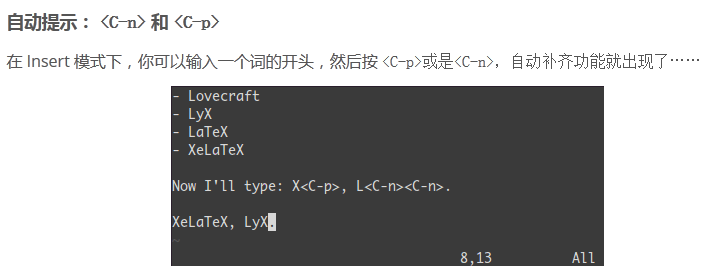
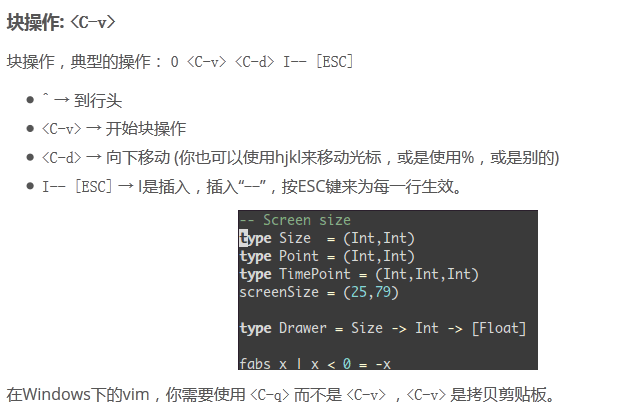
* 0 → 先到行头
* y → 从这里开始拷贝
* $ → 拷贝到本行最后一个字符

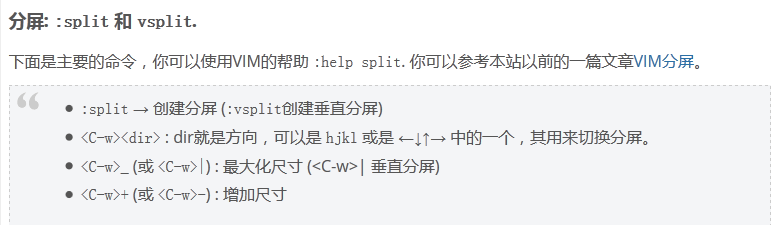
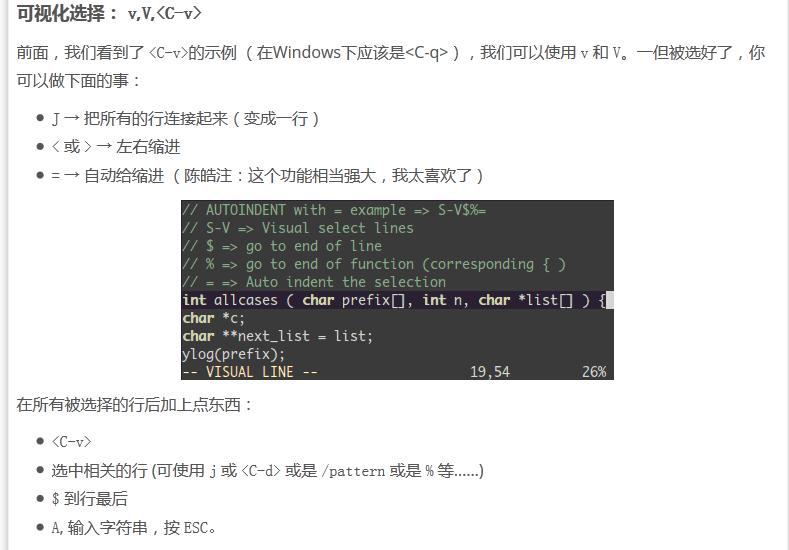
你可可以输入 ye，从当前位置拷贝到本单词的最后一个字符。

你也可以输入 y2/foo 来拷贝2个 “foo” 之间的字符串。



上面的v可以换位y、d等。好像object 无法使用单词，只能是（ “这些符号。





常用设定：

设定行号：set nu;

设定配色：colorscheme zellner

设定无菜单栏：set go=

语法高亮：syntac on

"设置缩进有三个取值cindent(c风格)、smartindent(智能模式，其实不觉得有什么智能)、autoindent(简单的与上一行保持一致)   
set cindent

"用空格键替换制表符   
:set expandtab   
"制表符占4个空格   
set tabstop=4   
"默认缩进4个空格大小   
set shiftwidth=4

"增量式搜索   
set incsearch   
"高亮搜索   
set hlsearch   
"有时中文会显示乱码，用一下几条命令解决   
let &termencoding=&encoding   
set fileencodings=utf-8,gbk

"在windows版本中vim的退格键模式默认与vi兼容，与我们的使用习惯不太符合，下边这条可以改过来

set backspace=indent,eol,start

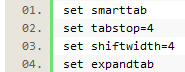
设定窗口大小

set lines=35 columns=100

设定tab：

建议在当前用户的主目录下编辑.vimrc，即~/.vimrc

或者在/etc/vimrc中添加



1. tabstop：表示一个 tab 显示出来是多少个空格，默认 8  
2. softtabstop：在编辑的时候（比如按退格或 tab 键）一个 tab 是多少个空格（有点不好理解。。。我也没完全弄明白）  
3. expandtab/noexpandtab：将 tab 扩展/不扩展成空格  
4. shiftwidth：每一级缩进是多少个空格  
5. smarttab：根据文件中其他地方的缩进空格个数来确定一个 tab 是多少个空格

复制和粘贴：

用vim这么久了，始终也不知道怎么在vim中使用系统粘贴板，通常要在网上复制一段代码都是先gedit打开文件，中键粘贴后关闭，然后再用vim打开 编辑，真的不爽；上次论坛上有人问到了怎么在vim中使用系统粘贴板，印象里回复很多，有好几页的回复却没有解决问题，今天实在受不了了又在网上找办法， 竟意外地找到了，贴出来分享一下。   
  
如果只是想使用系统粘贴板的话直接在输入模式按Shift+Inset就可以了，下面讲一下vim的粘贴板的基础知识，有兴趣的可以看看，应该会有所收获的。   
vim帮助文档里与粘贴板有关的内容如下：   
  
vim有12个粘贴板，分别是0、1、2、...、9、a、“、＋；用:reg命令可以查看各个粘贴板里的内容。在vim中简单用y只是复制到“（双引号)粘贴板里，同样用p粘贴的也是这个粘贴板里的内容；   
要将vim的内容复制到某个粘贴板，需要退出编辑模式，进入正常模式后，选择要复制的内容，然后按"Ny完成复制，其中N为粘贴板号(注意是按一下双引号然后按粘贴板号最后按y)，例如要把内容复制到粘贴板a，选中内容后按"ay就可以了，有两点需要说明一下：   
“号粘贴板（临时粘贴板）比较特殊，直接按y就复制到这个粘贴板中了，直接按p就粘贴这个粘贴板中的内容；   
+号粘贴板是系统粘贴板，用"+y将内容复制到该粘贴板后可以使用Ctrl＋V将其粘贴到其他文档（如firefox、gedit）中，同理，要把在其他地方用Ctrl＋C或右键复制的内容复制到vim中，需要在正常模式下按"+p；   
要将vim某个粘贴板里的内容粘贴进来，需要退出编辑模式，在正常模式按"Np，其中N为粘贴板号，如上所述，可以按"5p将5号粘贴板里的内容粘贴进 来，也可以按"+p将系统全局粘贴板里的内容粘贴进来。注意：在我这里，只有vim.gtk或vim.gnome才能使用系统全局粘贴板，默认的 vim.basic看不到+号寄存器。

Windows下gVim的设置：

在vim/\_vimrc文件中添加：

colorscheme zellner

set guifont=Consolas:h11:cANSI

set go=

syntax on

set hlsearch

set nu

let &termencoding=&encoding

set fileencodings=utf-8,gbk

set lines=35 columns=100

"设置缩进有三个取值cindent(c风格)、smartindent(智能模式，其实不觉得有什么智能)、autoindent(简单的与上一行保持一致)

set cindent

"用空格键替换制表符

:set expandtab

"制表符占4个空格

set tabstop=4

"默认缩进4个空格大小

set shiftwidth=4

"增量式搜索

set incsearch

"在windows版本中vim的退格键模式默认与vi兼容，与我们的使用习惯不太符合，下边这条可以改过来

set backspace=indent,eol,start