**【vim 命令-移动光标】：**

h 的鍵位于左邊，每次按下就會向左移動。

l 的鍵位于右邊，每次按下就會向右移動。

 j 鍵看起來很象一支尖端方向朝下的箭頭。

k朝上移动。

**【vim 命令-启动和退出】**：

1. 請按<ESC>鍵(這是為了確保您處在正常模式)。

2.:q! <回車>不保存退出。

3.:wq  <回車>保存并退出。

**【vim 命令-编辑删除】**：

1. 按x 鍵來刪除光標所在位置的字符。

**【vim 命令-编辑插入】**：

1. 欲進入vim編輯器(從命令行提示符)，請輸入︰vim 文件名 <回車>

2.按下 i 鍵來插入文本。

**【vim 命令-小结2】**：

1. 欲從當前光標刪除至單字/單詞末尾，請輸入︰dw

  2. 欲從當前光標刪除至當前行末尾，請輸入︰d$

  3. 欲刪除整行，請輸入︰dd

  4. 在正常模式下一個命令的格式是︰

       [number]   command   object     或者     command [number]   object

     其意是︰

       number - 代表的是命令執行的次數

       command - 代表要做的事情，比如 d 代表刪除

       object - 代表要操作的對象，比如 w 代表單字/單詞，$ 代表到行末等等。

$ (to the end of line), etc.

  5. 欲撤消以前的操作，請輸入︰u (小寫的u)

     欲撤消在一行中所做的改動，請輸入︰U (大寫的U)

     欲撤消以前的撤消命令，恢復以前的操作結果，請輸入︰CTRL-R

**【vim 命令-小结3】**：

1. 要重新置入已經刪除的文本內容，請輸入小寫字母 p。該操作可以將已刪除

     的文本內容置于光標之後。如果最後一次刪除的是一個整行，那麼該行將置

     于當前光標所在行的下一行。

  2. 要替換光標所在位置的字符，請輸入小寫的 r 和要替換掉原位置字符的新字

     符即可。

  3. 更改類命令允許您改變指定的對象，從當前光標所在位置直到對象的末尾。

     比如輸入 cw 可以替換當前光標到單詞的末尾的內容；輸入 c$ 可以替換當

     前光標到行末的內容。

  4. 更改類命令的格式是︰

[number]   c object      或者 c   [number]   object

**【vim 命令-小结4】**：

  1. Ctrl-g 用于顯示當前光標所在位置和文件狀態信息。Shift-G 用于將光標跳

     轉至文件最後一行。先敲入一個行號然後按 Shift-G 則是將光標移動至該行

     號代表的行。

  2. 輸入 / 然後緊隨一個字符串是則是在當前所編輯的文檔中向後查找該字符串。

     輸入問號 ? 然後緊隨一個字符串是則是在當前所編輯的文檔中向前查找該字符串。完成一次查找之後按 n 鍵則是重復上一次的命令，可在同一方向上查

     找下一個字符串所在；或者按 Shift-N 向相反方向查找下該字符串所在。

  3. 如果光標當前位置是括號(、)、[、]、{、}，按 % 可以將光標移動到配對的

     括號上。

  4. 在一行內替換頭一個字符串 old 為新的字符串 new，請輸入  :s/old/new

     在一行內替換所有的字符串 old 為新的字符串 new，請輸入  :s/old/new/g

     在兩行內替換所有的字符串 old 為新的字符串 new，請輸入  :#,#s/old/new/g

     在文件內替換所有的字符串 old 為新的字符串 new，請輸入  :%s/old/new/g

     進行全文替換時詢問用戶確認每個替換需添加 c 選項，請輸入 :%s/old/new/gc

**【vim 命令-小结5】**：

 1. :!command 用于執行一個外部命令 command。

     請看一些實際例子︰

:!dir -  用于顯示當前目錄的內容。

:!rm FILENAME -  用于刪除名為 FILENAME 的文件。

  2. :w FILENAME  可將當前 VIM 中正在編輯的文件保存到名為 FILENAME

     的文件中。

  3. :#,#w FILENAME 可將當前編輯文件第 # 行至第 # 行的內容保存到文件

     FILENAME 中。

  4. :r FILENAME 可提取磁盤文件 FILENAME 並將其插入到當前文件的光標位置後面

**【vim 命令-选项】**：

1. 要查找單詞 ignore 可在正常模式下輸入 /ignore 。要重復查找該詞，可以

     重復按 n 鍵。

  2. 然後設置 ic 選項(ic就是英文忽略大小寫Ignore Case的首字母縮寫詞)，即

     輸入︰

:set ic

  3. 現在可以通過鍵入 n 鍵再次查找單詞 ignore。重復查找可以重復鍵入 n 鍵。

  4. 然後設置 hlsearch 和 incsearch 這兩個選項，輸入以下內容︰

     :set hls is

  5. 現在可以再次輸入查找命令，看看會有什麼效果︰

     /ignore

**【vim 命令-小结6】**：

  1. 輸入小寫的 o 可以在光標下方打開新的一行並將光標置于新開的行首，進入

     插入模式。

     輸入大寫的 O 可以在光標上方打開新的一行並將光標置于新開的行首，進入

     插入模式。

  2. 輸入小寫的 a 可以在光標所在位置之後插入文本。

     輸入大寫的 A 可以在光標所在行的行末之後插入文本。

  3. 輸入大寫的 R 將進入替換模式，直至按 <ESC> 鍵退出替換模式而進入正常

     模式。

  4. 輸入 :set xxx 可以設置 xxx 選項。

**【vim 命令-在线帮助】**：

Vim 擁有一個細致全面的在線幫助系統。要啟動該幫助系統，請選擇如下三種方

  法之一︰

- 按下 <HELP> 鍵 (如果鍵盤上有的話)

- 按下 <F1> 鍵 (如果鍵盤上有的話)

- 輸入 :help <回車>

  輸入 :q <回車> 可以關閉幫助窗口。

**【vim 命令-创建启动脚本】**：

Vim的功能特性要比vi多得多，但大部分功能都沒有缺省激活。為了啟動更多的

  功能，您得創建一個vimrc文件。

  1. 開始編輯vimrc文件，這取決于您所使用的操作系統︰

     :edit ~/.vimrc 這是Unix系統所使用的命令

     :edit $VIM/\_vimrc 這是Windows系統所使用的命令

  2. 接著導入vimrc范例文件︰

     :read $VIMRUNTIME/vimrc\_example.vim

  3. 保存文件，命令為︰

     :write

进入vi的命令   
vi filename :打开或新建文件，并将光标置于第一行首   
vi +n filename ：打开文件，并将光标置于第n行首   
vi + filename ：打开文件，并将光标置于最后一行首   
vi +/pattern filename：打开文件，并将光标置于第一个与pattern匹配的串处   
vi -r filename ：在上次正用vi编辑时发生系统崩溃，恢复filename   
vi filename....filename ：打开多个文件，依次进行编辑   
  
移动光标类命令  
h ：光标左移一个字符   
l ：光标右移一个字符   
space：光标右移一个字符   
Backspace：光标左移一个字符   
k或Ctrl+p：光标上移一行   
j或Ctrl+n ：光标下移一行   
Enter ：光标下移一行   
w或W ：光标右移一个字至字首   
b或B ：光标左移一个字至字首   
e或E ：光标右移一个字至字尾   
) ：光标移至句尾   
( ：光标移至句首   
}：光标移至段落开头   
{：光标移至段落结尾   
nG：光标移至第n行首   
n+：光标下移n行   
n-：光标上移n行   
n$：光标移至第n行尾   
H ：光标移至屏幕顶行   
M ：光标移至屏幕中间行   
L ：光标移至屏幕最后行   
0：（注意是数字零）光标移至当前行首   
$：光标移至当前行尾   
  
屏幕翻滚类命令   
Ctrl+u：向文件首翻半屏   
Ctrl+d：向文件尾翻半屏   
Ctrl+f：向文件尾翻一屏   
Ctrl＋b；向文件首翻一屏   
nz：将第n行滚至屏幕顶部，不指定n时将当前行滚至屏幕顶部。   
  
插入文本类命令   
i ：在光标前   
I ：在当前行首   
a：光标后   
A：在当前行尾   
o：在当前行之下新开一行   
O：在当前行之上新开一行   
r：替换当前字符   
R：替换当前字符及其后的字符，直至按ESC键   
s：从当前光标位置处开始，以输入的文本替代指定数目的字符   
S：删除指定数目的行，并以所输入文本代替之   
ncw或nCW：修改指定数目的字   
nCC：修改指定数目的行   
  
删除命令   
ndw或ndW：删除光标处开始及其后的n-1个字   
do：删至行首   
d$：删至行尾   
ndd：删除当前行及其后n-1行   
x或X：删除一个字符，x删除光标后的，而X删除光标前的   
Ctrl+u：删除输入方式下所输入的文本   
  
搜索及替换命令   
/pattern：从光标开始处向文件尾搜索pattern   
?pattern：从光标开始处向文件首搜索pattern   
n：在同一方向重复上一次搜索命令   
N：在反方向上重复上一次搜索命令   
：s/p1/p2/g：将当前行中所有p1均用p2替代   
：n1,n2s/p1/p2/g：将第n1至n2行中所有p1均用p2替代   
：g/p1/s//p2/g：将文件中所有p1均用p2替换   
  
选项设置   
all：列出所有选项设置情况   
term：设置终端类型   
ignorance：在搜索中忽略大小写   
list：显示制表位(Ctrl+I)和行尾标志（$)   
number：显示行号   
report：显示由面向行的命令修改过的数目   
terse：显示简短的警告信息   
warn：在转到别的文件时若没保存当前文件则显示NO write信息   
nomagic：允许在搜索模式中，使用前面不带“\”的特殊字符   
nowrapscan：禁止vi在搜索到达文件两端时，又从另一端开始   
mesg：允许vi显示其他用户用write写到自己终端上的信息   
  
最后行方式命令   
：n1,n2 co n3：将n1行到n2行之间的内容拷贝到第n3行下   
：n1,n2 m n3：将n1行到n2行之间的内容移至到第n3行下   
：n1,n2 d ：将n1行到n2行之间的内容删除   
：w ：保存当前文件   
：e filename：打开文件filename进行编辑   
：x：保存当前文件并退出   
：q：退出vi   
：q!：不保存文件并退出vi   
：!command：执行shell命令command   
：n1,n2 w!command：将文件中n1行至n2行的内容作为command的输入并执行之，若不指定n1，n2，则表示将整个文件内容作为command的输入   
：r!command：将命令command的输出结果放到当前行   
  
寄存器操作   
"?nyy：将当前行及其下n行的内容保存到寄存器？中，其中?为一个字母，n为一个数字   
"?nyw：将当前行及其下n个字保存到寄存器？中，其中?为一个字母，n为一个数字   
"?nyl：将当前行及其下n个字符保存到寄存器？中，其中?为一个字母，n为一个数字   
"?p：取出寄存器？中的内容并将其放到光标位置处。这里？可以是一个字母，也可以是一个数字   
ndd：将当前行及其下共n行文本删除，并将所删内容放到1号删除寄存器中。