

# Vim

Vi IMproved, a programmers text editor

[atdog.tw@gmail.com](mailto:atdog.tw@gmail.com)

# Version 7.4.52

VIM - Vi IMproved

version 7.4.52

by Bram Moolenaar et al.

Modified by pkg-vim-maintainers@lists.alioth.debian.org

Vim is open source and freely distributable

Become a registered Vim user!

type :help register<Enter> for information

type :q<Enter> to exit

type :help<Enter> or <F1> for on-line help

type :help version7<Enter> for version info

阿 順便看一下 bash 小技巧  
Bash Tips

`^x ^e` - Use \$EDITOR to edit your command

`^a` - head of line

`^e` - end of line

`^u` - cut line

`^y` - paste line

`^|` - clear

`^w` - delete a word

開始前

Before we start.

先看一下手要怎麼擺？

Finger position at keyboard.



重點是 F 跟 J !

Small protrusions on F and J keys.



這樣才可以買60%機械式鍵盤



手擺對，就可以很快唷

那我們可以開始了！  
Let's go through tips on vim.

Vim 中雖然有 6 種模式

不過太複雜了

先記住 3 種就好

Insert Mode - Normal Mode - Visual Mode

首先看看基本操作

## Normal mode

上下左右 `kjhl`

存檔 `:w`

離開 `:q`

存檔 + 離開 `:wq`

插入模式 `i`

還原上一步 `u`

還原下一步 `^r`

上一頁 `^u` (half-page)

下一頁 `^d` (half-page)

複製 - `yy`

貼上 - `p`

刪除 - `dd`

Delete - `x`

Backspace - `X`



Insert Mode

沒什麼特別功能，就拿來打打字

## Visual Mode

選取  $v$

行選取  $V$

區塊選取  $^v$

+ 基本操作

很簡單吧？

讓我們稍微進階一點點

首先，把 ESC 換成 ^[  
:)

這目的很單純

這樣才可以買60%機械式鍵盤



其實不是這樣



如果把 ctrl 與 capslock 互換 ...

手才不用離開鍵盤太遠



然後 Insert mode 其實有

i, l, a, A, s, S, o, O

那麼多種

另外就是， 上完這堂課

就別再一直 j j k k 了！

Page up - ^u, ^f  
Page down - ^d, ^b  
line up - ^e  
line down - ^y

然後再縮短一點時間!

儲存並離開 ZZ

不儲存離開 ZQ

執行 shell 程式 `!:shell_cmd`



不過如果只有這樣就太簡單了



Normal Mode

## TabPage

\$ vim -p filea fileb

新增 Tab :tabnew

下一個 Tab :tabnext (gt)

上一個 Tab :tabprev (gT)

# SplitWindow

分割 :split (:sp)

垂直分割 :vsplit (:vs)

離開所有視窗 :qa!

儲存離開所有視窗 :xa!

切換視窗 ^w^w

切換視窗 ^w\_hjkl

視窗交換 ^w\_HJKL

視窗大小 ^w\_+-<>

關閉其他視窗 :only

關閉當前視窗 :hide

## 讀取資料

讀取檔案至游標上 :r file

讀取執行結果至游標上 :r !shell\_cmd

編輯檔案 :e

## 搜尋全文

向下搜尋 /pattern

向上搜尋 /pattern

下一個 n

上一個 N

向下搜尋游標單字 \*

向上搜尋游標單字 #

向下搜尋游標 pattern g\*

向上搜尋游標 pattern g#

## 同一行內的移動

向右找字元 `f`

向左找字元 `F`

向右找字元前的一個字 `t`

向左找字元前的一個字 `T`

下一個 `;`

字行首 `^`

行首 `0`

行尾 `$`

向右找單字 `w, e`

向左找單字 `b`

e.g., `4f, ;`

## 全文內的移動

移動至上一個更動的地方 `g;`

移動至下一個更動的地方 `g,`

尋找下一個 bracket `%`

移動到檔案開頭 `gg`

移動到檔案結尾 `G`

以游標為位址將畫面置中 `zz`

跳回上一個所在地 ````

移動至畫面上/中/下的位置 `H, M, L`

記錄並儲存 `ma`

跳至儲存點 `'a`

移動至上/下一個段落空白處 `{ }`

移動至上/下一個段落開頭 `( )`



加一 ^c\_a

減一 ^c\_x

合併行 J

向右排版 >>

向左排版 <<

取代字元 r

取代模式 R

The objects that can be used are:

aw	a word (with white space)	v_aw
iw	inner word	v_iw
aW	a WORD (with white space)	v_aW
iW	inner WORD	v_iW
as	a sentence (with white space)	v_as
is	inner sentence	v_is
ap	a paragraph (with white space)	v_ap
ip	inner paragraph	v_ip
ab	a () block (with parenthesis)	v_ab
ib	inner () block	v_ib
aB	a {} block (with braces)	v_aB
iB	inner {} block	v_iB
at	a <tag> </tag> block (with tags)	v_at
it	inner <tag> </tag> block	v_it
a<	a <> block (with <>)	v_a<
i<	inner <> block	v_i<
a[	a [] block (with [])	v_a[
i[	inner [] block	v_i[
a"	a double quoted string (with quotes)	v_aquote
i"	inner double quoted string	v_iquote
a'	a single quoted string (with quotes)	v_a'
i'	inner simple quoted string	v_i'
a`	a string in backticks (with backticks)	v_a`
i`	inner string in backticks	v_i`

重複上一個動作 .

整行變小寫 guu

整行變大寫 gUU

顯示字元的16進位 ga

開啟已游標所在文字為檔名的檔案 gf

回去上一個檔案 :bp

全文排版 G=gg

## 移動文字、區塊

移動行 :m7

移動區塊 visual + :m7

## 錄影

記錄 q1

重放 @1

重複一次重放 @@

Visual Mode

重新選取 gv  
跳至選取區塊的開始/結束處 o  
跳至選取區塊左/右側 O  
向右排版 >  
向左排版 <  
大小寫互換 ~  
大寫 U  
小寫 u  
ROT13 g?

基本操作, 進階操作結合

還有一點小東西



## Python 排版問題

```
set tabstop=4  
set expandtab
```

Real tab -  $\wedge$ v tab

```
70     asd  
71 " Set tab  
72 " Replace tab with space  
73 set expandtab  
74 " Set tab to 4 space  
75 set tabstop=4  
76 " To change the number of space characters inserted for indentation  
77 set shiftwidth=4
```

autoread

```
90  
91 " Set to auto read when a file is changed from the outside  
92 set| autoread
```

Live demo

Use vim for binary patch.

```
:%!xxd
```

```
:%!xxd -r
```

Plugins from [vim.org](http://vim.org)

套件管理

Plugin Manger

vundle

pathogen