



VIM快速入门

jingmi@baidu.com



intro

- ~ vim is incredibly powerful
- ~ but you have to speak its language
- ~ and it will do everything you can imagine
- ~ languages in general are messy, ambiguous, and hard
- ~ yet fantastically brilliant, for they allow us to communicate

Need help?

- ~ learn how to get help effectively
- ~ **:help** is your friend
- ~ use **CTRL-V** before a CTRL sequence command
- ~ use **i_** and **v_** prefixes to get help for CTRL sequences in Insert and Visual modes
- ~ use **CTRL-]** (jump to tag) and **CTRL-T** (go back) in help window



-
- | 条目 | 前缀 | 例子 |
|-------------------|-------|------------------|
| Normal模式命令 | 无 | :h k |
| CTRL控制字符 | CTRL- | :h CTRL-r |
| Visual视图模式 | v | :h v_u |
| Insert插入模式命令 | i | :h i_CTRL-o |
| ex-模式命令 | : | :h :set |
| Command-line命令行编辑 | c | :h c_<BS> |
| 命令参数 | - | :h -m |
| 选项 | , | :h 'winminwidth' |
- 还有一些关键词是用尖括号<>括起来的,如
:h <Left>
:h <CR>
:h <Up>
:helpg[rep] {pattern} 搜索所有的帮助文本并给出一个匹配 {pattern} 行的列表。

另外，在输入关键词的时候，可以按Tab键补全，ctrl-d列出与光标前面的关键词所匹配的名字，ctrl-a列出全部模式

VIM modes (Cool!)

- Vim has six BASIC modes(:h vim-modes)

Normal mode

Visual mode

Select mode

Insert mode

Command-line mode

Ex mode

Where am i?

How do you tell where you are?

~ simple - CTRL-G

~ detailed - gCTRL-G

~ do yourself a favor and set ruler

~ shows line, column, and percentage in status line

~ or configure it however you want with 'rulerformat'

Where am i?

- Example (Insert to ~/.vimrc)

```
set laststatus=2
```

```
set
```

```
statusline=%<%f\%h%m%r%=%(l=%l,c=%c%V,t=%L%)  
%=%10(%y\ %P%)
```

Move, move, move(Jack Bauer said)

- ~ do you use **h/j/k/l** for moving? or are you stuck in GULARrowy world?
- ~ start/end of buffer? **gg** and **G**
- ~ line n? **nG** or **ngg**
- ~ n% into the file? **n%**
- ~ the first non-blank character in the line? **^**
- ~ first non-blank character on next line? **<CR>**
- ~ first non-blank character on previous line? **—**
- ~ first character on this line? **0**
- ~ last character on the line? **\$**

marks

- ~ we can bookmark locations in the buffer
- ~ **m<letter>** sets mark named <letter> at current location
- ~ **`<letter>** jumps precisely to that mark
- ~ **'<letter>** jumps to the line with the mark
- ~ lowercase letter: mark is local to the buffer
- ~ uppercase letter: mark is global, your buffer will be switched to the file with the mark

Tips 1

~ tip: **CTRL-O** and **CTRL-I** move between positions in the full jump history

~ use **:jump** to see jump list

Insert mode

- Useful commands:

i I: Insert

a A: Append

o O: Begin a new line below the cursor and insert text

s S: Delete [count] characters [into register x] and start insert (s stands for Substitute).

- **c C:** Delete {motion} text [into register x] and start insert.

r R: Replace

Tips 2

- `gi`: incredibly handy
- ````: To the position before the latest jump

delete

- set your <Backspace> free
- ```
:set backspace=start,indent,eol
```

X, x, D, d, dd

# search

/, \*, #, n, N

# replace

`:%s/old/new/g`

Counting:

`:%s/string/&/gn`

# Tips 3

Insert to your ~/.vimrc

set incsearchset nobackup

set autoread

set ff=unix

set ffs=unix

Convert DOS file to UNIX file

:%s/^M//g



# undo

- simple case: u - undo, CTRL-R – redo
- :undol
- g-, g+
- :earlier Ns,m,h - go to text state as it was Nseconds, minutes, hours ago
- :later Ns,m,h - go to a later text state similarly

# windows

- **new, sp, vnew**: Create a new window
- CTRL-W h/j/k/l: 在窗口间移动(Moving between windows)
- CTRL-W +/-/</>:Resizing windows(调整窗口大小)

# Tab pages

- vim 7 supports tab pages
- ~ **:tabe <file>** to edit file in a new tab
- ~ **:tabc** to close
- ~ **:tabn, :tabp** (or gt, gT to switch)

# completion

- CTRL – n
- CTRL – p
- CTRL –x CTRL-o/l
- CTRL – y/e
- CTRL-X CTRL-F to complete filenames
- CTRL-X CTRL-N to complete identifiers

# folding

- ~ **zo** - open fold (if the cursor is on the fold line)
- ~ **zc** - close closest fold
- ~ **zR** - open all folds
- ~ **zM** - close all folds
- ~ **zj** - move to the start of the next fold
- ~ **zk** - move to the end of the previous fold
- ~ **zf**

# plugin 1(from newsmth)

- (1) [ 插件的名称 ]

BufExplorer

- (2) [ 插件的功能 ]

方便切换 buffer，切换 buffer 后列表自动关闭。

- (3) [ 插件的下载地址 ]

[http://www.vim.org/scripts/script.php?script\\_id=42](http://www.vim.org/scripts/script.php?script_id=42)

- (4) [ 备注或特别提醒 ]

装完后 Normal mode 下输入 \be 即可打开这个列表，光标放在某一个文件上按回车可以打开这个文件。

# plugin 2(from newsmth)

- (1) [ 插件的名称 ]

Taglist

- (2) [ 插件的功能 ]

实现类或者函数列表功能，支持 C/C++, Java, Perl, Python, TCL, SQL, PHP 等语言，它可以在VIM中创建一个窗口，列出编辑的源文件中的类、函数、变量定义列表，而且更方便的是这个列表也插入在了右键菜单中，分类存放，访问非常方便。

- (3) [ 插件的下载地址 ]

[http://www.vim.org/scripts/script.php?script\\_id=273](http://www.vim.org/scripts/script.php?script_id=273)

- (4) [ 备注或特别提醒 ]

依赖于 Exuberant Ctags，可以在 <http://ctags.sourceforge.net> 下载。

# plugin3 (from newsmth)

- (1) [ 插件的名称 ]

word\_complete

- (2) [ 插件的功能 ]

Insert模式下在输入的时候自动补全单词，如果补全正确可以按Tab键接受，否则可以不管它继续输入或者按Ctrl-n，Ctrl-p切换补全。注意是输入三个字符后才开始补全。

- (3) [ 插件的下载地址 ]

[http://www.vim.org/scripts/script.php?script\\_id=73](http://www.vim.org/scripts/script.php?script_id=73)

- (4) [ 备注或特别提醒 ]

在 工具 菜单中选择 Word completion 开启这个功能。



# plugin4 (from newsmth)

- (1) [ 插件的名称 ]

Mark

- (2) [ 插件的功能 ]

同时高亮显示多次搜索结果，这个特性非常有利于显示代码中某些变量或函数的调用位置。

- (3) [ 插件的下载地址 ]

[http://www.vim.org/scripts/script.php?script\\_id=1238](http://www.vim.org/scripts/script.php?script_id=1238)

- (4) [ 备注或特别提醒 ]

类似的还有：

MultipleSearch

[http://www.vim.org/scripts/script.php?script\\_id=479](http://www.vim.org/scripts/script.php?script_id=479)

MultipleSearch2

[http://www.vim.org/scripts/script.php?script\\_id=1183](http://www.vim.org/scripts/script.php?script_id=1183)

otf

[http://www.vim.org/scripts/script.php?script\\_id=634](http://www.vim.org/scripts/script.php?script_id=634)

# plugin5 (from newsmth)

- (1) [ 插件的名称 ]

SearchComplete

- (2) [ 插件的功能 ]

用 / 或者 ? 搜索时按 tab 补全要搜索的单词。

- (3) [ 插件的下载地址 ]

[http://www.vim.org/scripts/script.php?script\\_id=474](http://www.vim.org/scripts/script.php?script_id=474)

# plugin 6 (from newsmth)

- (1) [ 插件的名称 ]

vcsccommand

- (2) [ 插件的功能 ]

集成 cvs 和 svn 到 vim 中，可以在 vim 中看 log，diff，annotate，commit 等等。需要安装 cvs 和 svn 的命令行版本。

- (3) [ 插件的下载地址 ]

[http://www.vim.org/scripts/script.php?script\\_id=90](http://www.vim.org/scripts/script.php?script_id=90)

# plugin 7 (by me)

名称:NERD\_comments.vim

功能: A plugin that allows for easy commenting of code for many filetypes.

下载:

[http://www.vim.org/scripts/script.php?script\\_id=1218](http://www.vim.org/scripts/script.php?script_id=1218)

# Q&A

THANKS!