



文本编辑器之神 vim



the editor

讲师：王晓春

本章内容



- ◆ 文本编辑工具介绍
- ◆ vim的三种模式
- ◆ 命令模式
- ◆ 扩展命令模式
- ◆ 其它模式
- ◆ 定制vim特性

vim简介



- ◆ vi: Visual editor , 文本编辑器
- ◆ 文本 : ASCII, Unicode
- ◆ 文本编辑种类 :
 - 行编辑器 : sed
 - 全屏编辑器 : nano, vi, vim
- ◆ vim
 - Visual editor iMproved
 - 官网 : www.vim.org
- ◆ 其他编辑器 :
 - gedit 一个简单的图形编辑器
 - gvim 一个Vim编辑器的图形版本

vim 小抄



IT 人的高薪职业学院

Esc Normal Revision 2.0 Sept. 11, 2011 Vim 7.3+ :version	Vim Cheat Sheet for Programmers Copyright © 2011 May be freely distributed! Sharing is Caring. http://michael.People@ottawacollege.com/vim/
HOW-To make Vim not suck out of the Box: :help statusline :set nocompatible ruler laststatus=2 showcmd showmode number Best tips: http://vim.wikia.com/ Best scripts: http://www.vim.org/scripts/index.php	Search: :set incsearch ignorecase smartcase hlsrch Remove useless splash screen: :set shortmess+=!
Best tips: http://vim.wikia.com/ Best scripts: http://www.vim.org/scripts/index.php	Best tips: http://vim.wikia.com/ Best scripts: http://www.vim.org/scripts/index.php
Ctrl-` toggle case Ctrl-1 ! extern filter Ctrl-@ @* play macro Ctrl-3 # prev identifier Ctrl-4 \$ → % goto match Ctrl-^ ^ soft → & repeat :s Ctrl-7 * next identifier Ctrl-9 (begin sentence Ctrl-0) end sentence Ctrl-_- _ cur line Ctrl-= + ↓ auto-format	Ctrl-1 ! extern filter Ctrl-2 1 Ctrl-3 2 Ctrl-4 3 Ctrl-5 4 Ctrl-6 5 Ctrl-7 6 Ctrl-8 7 Ctrl-9 8 Ctrl-0 9 Ctrl-_- 0 Ctrl-= = auto-format
Tab Ctrl-Q ex mode Ctrl-W WORD ↘ Ctrl-E end WORD ↘ Ctrl-R Replace Ctrl-T ← until char Ctrl-Y copy line Ctrl-U undo line Ctrl-I insert → Ctrl-O open ↑ Ctrl-P paste ↑ Ctrl-{ { paragraph Ctrl>} } paragraph Ctrl-q record macro Ctrl-w word ↘ Ctrl-e end word ↘ Ctrl-r replace char Ctrl-t until char → Ctrl-y copy Ctrl-u undo Ctrl-i insert Ctrl-o open ↓ Ctrl-p paste ↓ Ctrl[[misc. Ctrl]] misc.	Ctrl-Q ex mode Ctrl-W WORD ↘ Ctrl-E end WORD ↘ Ctrl-R Replace Ctrl-T ← until char Ctrl-Y copy line Ctrl-U undo line Ctrl-I insert → Ctrl-O open ↑ Ctrl-P paste ↑ Ctrl-{ { paragraph Ctrl>} } paragraph Ctrl-q record macro Ctrl-w word ↘ Ctrl-e end word ↘ Ctrl-r replace char Ctrl-t until char → Ctrl-y copy Ctrl-u undo Ctrl-i insert Ctrl-o open ↓ Ctrl-p paste ↓ Ctrl[[misc. Ctrl]] misc.
Ctrl-7 incr. # Ctrl-A append → Ctrl-S subst line Ctrl-D del → Ctrl-F ← find char Ctrl-G goto eof / goto line# Ctrl-H Top screen Ctrl-J Join lines Ctrl-K man page identifier Ctrl-L Bottom screen Ctrl-: cmd line Ctrl-" register Ctrl-g goto mark Ctrl-\' ' register Ctrl-\ \ goto col#	Ctrl-7 incr. # Ctrl-A append → Ctrl-S subst line Ctrl-D del → Ctrl-F ← find char Ctrl-G goto eof / goto line# Ctrl-H Top screen Ctrl-J Join lines Ctrl-K man page identifier Ctrl-L Bottom screen Ctrl-: cmd line Ctrl-" register Ctrl-g goto mark Ctrl-\' ' register Ctrl-\ \ goto col#
Ctrl-^ :suspend Ctrl-Z quit Ctrl-X ← del char Ctrl-C change → Ctrl-V select lines Ctrl-B \ WORD Ctrl-N "prev" find Ctrl-M Middle screen Ctrl-← undent Ctrl-→ indent Ctrl-? find ↘ Ctrl-/ find ↘	Ctrl-^ :suspend Ctrl-Z quit Ctrl-X ← del char Ctrl-C change → Ctrl-V select lines Ctrl-B \ WORD Ctrl-N "prev" find Ctrl-M Middle screen Ctrl-← undent Ctrl-→ indent Ctrl-? find ↘ Ctrl-/ find ↘
Shift-↑ Z* quit Shift-↓ X* del char Shift-← X del char → Shift-→ c change Shift-V select chars Shift-B \ word Shift-N "next" find Shift-M set mark Shift-← * f/F/U/T Shift-→ * f/F/U/T Shift-? find ↘ Shift-/ find ↘	Shift-↑ Z* quit Shift-↓ X* del char Shift-← X del char → Shift-→ c change Shift-V select chars Shift-B \ word Shift-N "next" find Shift-M set mark Shift-← * f/F/U/T Shift-→ * f/F/U/T Shift-? find ↘ Shift-/ find ↘
Legend: Macro (M) Op (O) Cmd (C) Ins (I) Move (M) Find (F) Tag (T) Code (C) Extra (E) Modes (M) n Normal i Insert v Visual o Op pending c Command Line word Foo (src), b dst, b len ; WORD Foo (src), b dst, b len ; Startup vim <filename> +123 vim <file> ... -t Foo vim <file> ... -c "Foo" vim -g or gvim Linux :set guifont=ProggyTinyT12 OSX :set guifont=ProggyTinyT12h diff gvimdiff <file1> <file2> <file3> Broken Keys Ctrl-I = Tab, Ctrl-[] = ESC Vim is still unable to map certain keys for your own use... Caps, Ctrl-I, Ctrl-Shift-I, Ctrl-I, Ctrl-I, etc. See: src/ops.c -c "/valid_yank_reg" for "reg. names" See: src/norm.c -c "/nv_cmds" for "g-x extra cmds" See: src/edit.c -c "/ctrl_x_msgs" for "x* insert cmds"	Legend: Macro (M) Op (O) Cmd (C) Ins (I) Move (M) Find (F) Tag (T) Code (C) Extra (E) Modes (M) n Normal i Insert v Visual o Op pending c Command Line word Foo (src), b dst, b len ; WORD Foo (src), b dst, b len ; Startup vim <filename> +123 vim <file> ... -t Foo vim <file> ... -c "Foo" vim -g or gvim Linux :set guifont=ProggyTinyT12 OSX :set guifont=ProggyTinyT12h diff gvimdiff <file1> <file2> <file3> Broken Keys Ctrl-I = Tab, Ctrl-[] = ESC Vim is still unable to map certain keys for your own use... Caps, Ctrl-I, Ctrl-Shift-I, Ctrl-I, Ctrl-I, etc. See: src/ops.c -c "/valid_yank_reg" for "reg. names" See: src/norm.c -c "/nv_cmds" for "g-x extra cmds" See: src/edit.c -c "/ctrl_x_msgs" for "x* insert cmds"
Legend: Macro (M) Op (O) Cmd (C) Ins (I) Move (M) Find (F) Tag (T) Code (C) Extra (E) Modes (M) n Normal i Insert v Visual o Op pending c Command Line word Foo (src), b dst, b len ; WORD Foo (src), b dst, b len ; Startup vim <filename> +123 vim <file> ... -t Foo vim <file> ... -c "Foo" vim -g or gvim Linux :set guifont=ProggyTinyT12 OSX :set guifont=ProggyTinyT12h diff gvimdiff <file1> <file2> <file3> Broken Keys Ctrl-I = Tab, Ctrl-[] = ESC Vim is still unable to map certain keys for your own use... Caps, Ctrl-I, Ctrl-Shift-I, Ctrl-I, Ctrl-I, etc. See: src/ops.c -c "/valid_yank_reg" for "reg. names" See: src/norm.c -c "/nv_cmds" for "g-x extra cmds" See: src/edit.c -c "/ctrl_x_msgs" for "x* insert cmds"	Legend: Macro (M) Op (O) Cmd (C) Ins (I) Move (M) Find (F) Tag (T) Code (C) Extra (E) Modes (M) n Normal i Insert v Visual o Op pending c Command Line word Foo (src), b dst, b len ; WORD Foo (src), b dst, b len ; Startup vim <filename> +123 vim <file> ... -t Foo vim <file> ... -c "Foo" vim -g or gvim Linux :set guifont=ProggyTinyT12 OSX :set guifont=ProggyTinyT12h diff gvimdiff <file1> <file2> <file3> Broken Keys Ctrl-I = Tab, Ctrl-[] = ESC Vim is still unable to map certain keys for your own use... Caps, Ctrl-I, Ctrl-Shift-I, Ctrl-I, Ctrl-I, etc. See: src/ops.c -c "/valid_yank_reg" for "reg. names" See: src/norm.c -c "/nv_cmds" for "g-x extra cmds" See: src/edit.c -c "/ctrl_x_msgs" for "x* insert cmds"

◆ vim [OPTION]... FILE...

+ # 打开文件后，让光标处于第#行的行首，+默认行尾

+ /PATTERN 让光标处于第一个被PATTERN匹配到的行的行首

-b file 二进制方式打开文件

-d file1 file2... 比较多个文件

-m file 只读打开文件

-e file 直接进入ex模式，相当于执行ex file

◆ 说明：

如果该文件存在，文件被打开并显示内容

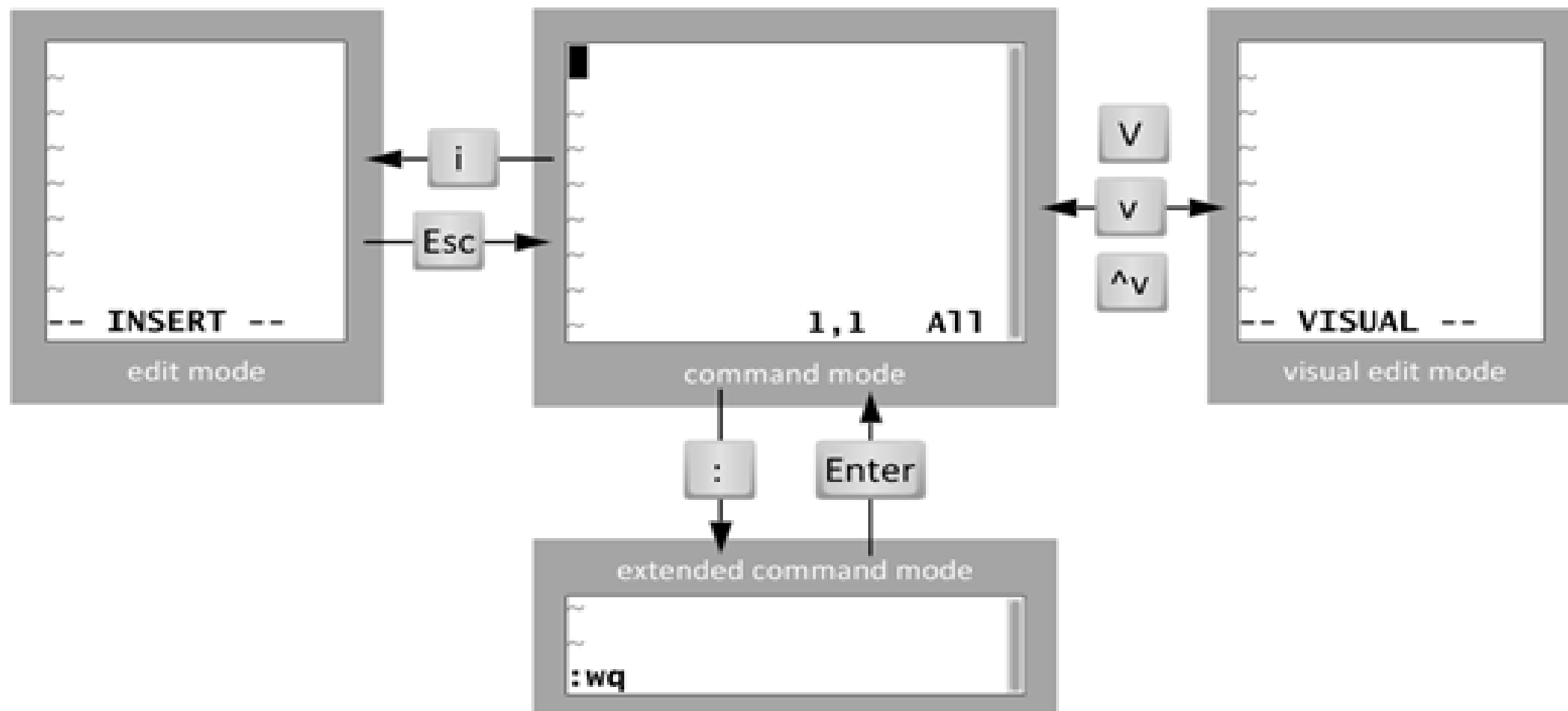
如果该文件不存在，当编辑后第一次存盘时创建它

vim：一个模式编辑器



- ◆ 击键行为是依赖于 vim 的 “模式”
- ◆ 三种主要模式：
 - 命令或普通(Normal)模式：默认模式，移动光标，剪切/粘贴文本
 - 插入(Insert)或编辑模式：修改文本
 - 扩展命令(extended command)或命令行模式：保存，退出等
- ◆ Esc键 退出当前模式
- ◆ Esc键 Esc键 总是返回到命令模式

vim模式



◆ 命令模式 --> 插入模式

- i insert, 在光标所在处输入
- I 在当前光标所在行的行首输入
- a append, 在光标所在处后面输入
- A 在当前光标所在行的行尾输入
- o 在当前光标所在行的下方打开一个新行
- O 在当前光标所在行的上方打开一个新行

模式转换



◆ 插入模式 -----> 命令模式

ESC

◆ 命令模式 -----> 扩展命令模式

:

◆ 扩展命令模式 -----> 命令模式

ESC,enter

关闭文件



◆ 扩展命令模式：

:q 退出

:q! 强制退出，丢弃做出的修改

:wq 保存退出

:x 保存退出

◆ 命令模式

ZZ 保存退出

ZQ 不保存退出

扩展命令模式



- ◆ 按 “:” 进入Ex模式
- ◆ 创建一个命令提示符: 处于底部的屏幕左侧
- ◆ 常见命令
 - w 写 (存) 磁盘文件
 - wq 写入并退出
 - x 写入并退出
 - q 退出
 - q ! 不存盘退出, 即使更改都将丢失
 - r filename 读文件内容到当前文件中
 - w filename 将当前文件内容写入另一个文件
 - !command 执行命令
 - r!command 读入命令的输出

◆ 字符间跳转：

h: 左 l: 右 j: 下 k: 上

#COMMAND：跳转由#指定的个数的字符

◆ 单词间跳转：

w：下一个单词的词首

e：当前或下一单词的词尾

b：当前或前一个单词的词首

#COMMAND：由#指定一次跳转的单词数

◆ 当前页跳转：

H：页首 M：页中间行 L：页底

zt：将光标所在当前行移到屏幕顶端

zz：将光标所在当前行移到屏幕中间

zb：将光标所在当前行移到屏幕底端

命令模式光标跳转



◆ 行首行尾跳转：

^ 跳转至行首的第一个非空白字符

0 跳转至行首

\$ 跳转至行尾

◆ 行间移动：

#G 或者扩展命令模式下:# 跳转至由第#行

G 最后一行

1G, gg 第一行

◆ 句间移动：

) 下一句 (上一句

◆ 段落间移动：

} 下一段 { 上一段

命令模式翻页操作

- ◆ Ctrl+f 向文件尾部翻一屏
- ◆ Ctrl+b 向文件首部翻一屏
- ◆ Ctrl+d 向文件尾部翻半屏
- ◆ Ctrl+u 向文件首部翻半屏



◆ 字符编辑：

- x 删除光标处的字符
- #x 删除光标处起始的#个字符
- xp 交换光标所在处的字符及其后面字符的位置
- ~ 转换大小写
- J 删除当前行后的换行符

◆ 替换命令(r, replace)

- r 替换光标所在处的字符
- R 切换到REPLACE模式

◆ 删除命令：

d 删除命令，可结合光标跳转字符，实现范围删除

d\$ 删除到行尾

d^ 删除到非空行首

d0 删除到行首

dw

de

db

#COMMAND

◆ dd：删除光标所在的行

#dd 多行删除

◆ D：从当前光标位置一直删除到行尾，等同于d\$

命令模式操作



◆ 复制命令(y, yank) :

y 复制，行为类似于d命令

y\$

y0

y^

ye

yw

yb

#COMMAND

◆ yy : 复制行

#yy 复制多行

◆ Y : 复制整行

◆ 粘贴命令(p, paste) :

p 缓冲区存的如果为整行，则粘贴当前光标所在行的下方；否则，则粘贴至当前光标所在处的后面

P 缓冲区存的如果为整行，则粘贴当前光标所在行的上方；否则，则粘贴至当前光标所在处的前面

- ◆ 改变命令(c, change)

 - c: 修改后切换成插入模式

- ◆ 命令模式 --> 插入模式

 - c\$

 - c^

 - c0

 - cb

 - ce

 - cw

 - #COMMAND

- ◆ cc : 删除当前行并输入新内容，相当于S

 - #cc

- ◆ C : 删除当前光标到行尾，并切换成插入模式

命令模式：操作文本

	Change (replace)	Delete (cut)	Yank (copy)
Line	cc	dd	yy
Letter	cl	dl	yl
Word	cw	dw	yw
Sentence ahead	c)	d)	y)
Sentence behind	c(d(y(
Paragraph above	c{	d{	y{
Paragraph below	c}	d}	y}

- ◆ 100iwang [ESC] 粘贴 “wang” 100次
 - ◆ <start position> <command> <end position>
 - ◆ 常见Command
 - y 复制、d 删除、gU 变大写、gu 变小写
- 例如：0y\$ 命令
- 0 → 先到行头
 - y → 从这里开始拷贝
 - \$ → 拷贝到本行最后一个字符

- ◆ di" 光标在" "之间，则删除" "之间的内容
- ◆ yi(光标在()之间，则复制()之间的内容
- ◆ vi[光标在[]之间，则选中[]之间的内容
- ◆ dtx 删除字符直到遇见光标之后的第一个 x 字符
- ◆ ytx 复制字符直到遇见光标之后的第一个 x 字符

◆ 查找

/PATTERN：从当前光标所在处向文件尾部查找

?PATTERN：从当前光标所在处向文件首部查找

n：与命令同方向

N：与命令反方向

命令模式：撤消更改



- ◆ u 撤销最近的更改
- ◆ #u 撤销之前多次更改
- ◆ U 撤消光标落在这行后所有此行的更改
- ◆ Ctrl - r 重做最后的“撤消”更改
- ◆ . 重复前一个操作
- ◆ #. 重复前一个操作#次

- ◆ 有26个命名寄存器和1个无命名寄存器，常存放不同的剪贴版内容，可以不同会话间共享
- ◆ 寄存器名称a, b,...,z,格式：“寄存器 放在数字和命令之间
如：3"tyy 表示复制3行到t寄存器中
"tp 表示将t寄存器内容粘贴
- ◆ 未指定，将使用无命名寄存器
- ◆ 有10个数字寄存器，用0, 1, ..., 9表示，0存放最近复制内容，1存放最近删除内容。当新的文本变更和删除时，1转存到2，2转存到3，以此类推。数字寄存器不能在不同会话间共享

标记和宏(macro)



- ◆ ma 将当前位置标记为a，26个字母均可做标记，mb、mc 等等
- ◆ 'a 跳转到a标记的位置，实用的文档内标记方法，文档中跳跃编辑时很有用
- ◆ qa 录制宏 a，a为宏的名称
- ◆ q 停止录制宏
- ◆ @a 执行宏 a
- ◆ @@ 重新执行上次执行的宏

◆ 地址定界

:start_pos,end_pos

具体第#行，例如2表示第2行

#,# 从左侧#表示起始行，到右侧#表示结尾行

#,+# 从左侧#表示的起始行，加上右侧#表示的行数

示例：2,+3 表示2到5行

. 当前行

\$ 最后一行

.,\$-1 当前行到倒数第二行

% 全文, 相当于1,\$

扩展命令模式：地址定界

- ◆ /pattern/ 匹配pattern的行

- ◆ /pat1/,/pat2/

从第一次被pat1模式匹配到的行开始，一直到第一次被pat2匹配到的行结束

#,/pat/

/pat/, \$

- ◆ 使用方式：后跟一个编辑命令

d

y

w file: 将范围内的行另存至指定文件中

r file：在指定位置插入指定文件中的所有内容

扩展命令模式：查找并替换



◆ 在扩展命令模式下完成查找替换操作

格式：s/要查找的内容/替换为的内容/修饰符

要查找的内容：可使用模式

替换为的内容：不能使用模式，但可以使用\1, \2, ...等后向引用符号；还可以使用 "&" 引用前面查找时查找到的整个内容

修饰符：

i 忽略大小写

g 全局替换，默认情况下，每一行只替换第一次出现

gc 全局替换，每次替换前询问

◆ 查找替换中的分隔符/可替换为其它字符

s@/etc@/var@g

s#/boot#/#i

- ◆ 以二进制方式打开文件

`vim -b binaryfile`

- ◆ 扩展命令模式下，利用xxd命令转换为可读的十六进制

`:%!xxd`

- ◆ 插入模式下，编辑二进制文件

- ◆ 扩展命令模式下，利用xxd命令转换回二进制

`:%!xxd -r`

- ◆ 保存退出

- ◆ 允许选择的文本块
 - ✓ 面向字符
 - ✓ 面向行
 - ctrl-v 面向块
- ◆ 可视化键可用于与移动键结合使用
 - w) } 箭头等
- ◆ 突出显示的文字可被删除，复制，变更，过滤，搜索，替换等

使用多个“窗口”



◆ 多文件分割

`vim -o|-O FILE1 FILE2 ...`

-o: 水平分割

-O: 垂直分割 (vim only)

在窗口间切换 : `Ctrl+w`, Arrow

◆ 单文件窗口分割

`Ctrl+w,s` : split, 水平分割

`Ctrl+w,v` : vertical, 垂直分割

`ctrl+w,q` : 取消相邻窗口

`ctrl+w,o` : 取消全部窗口

: `wqall` 退出

定制vim的工作特性



- ◆ 配置文件：永久有效
 - 全局：/etc/vimrc
 - 个人：~/.vimrc
- ◆ 扩展命令模式：当前vim进程有效
- ◆ (1) 行号
 - 显示：set number，简写 set nu
 - 取消显示：set nonumber, 简写 set nonu
- ◆ (2) 忽略字符的大小写
 - 启用：set ignorecase，简写 set ic
 - 不忽略：set noic
- ◆ (3) 自动缩进
 - 启用：set autoindent，简写 set ai
 - 禁用：set noai

定制vim的工作特性

- ◆ (4) 复制保留格式
 - 启用：set paste
 - 禁用：set nopaste
- ◆ (5) 显示Tab和换行符 ^I 和\$显示
 - 启用：set list
 - 禁用：set nolist
- ◆ (6) 高亮搜索
 - 启用：set hlsearch
 - 禁用：set nohlsearch
- ◆ (7) 语法高亮
 - 启用：syntax on
 - 禁用：syntax off

定制vim的工作特性

◆ (8) 文件格式

启用windows格式 : `set fileformat=dos`

启用unix格式 : `set fileformat=unix`

简写 `set ff=dos|unix`

◆ (9) 设置文本宽度

`set textwidth=65` (vim only)

`set wrapmargin=15`

◆ (10) 设置光标所在行的标识线

启用 : `set cursorline` , 简写 `set cul`

禁用 : `set nocursorline`

了解更多



马哥教育

IT 人的高薪职业学院

- ◆ set 帮助
 - :help option-list
 - :set or :set all
- ◆ vi/vim内置帮助
 - :help
 - :help *topic*
 - Use :q to exit help
- ◆ vimtutor

vi / vim 键盘图

Esc

命令
模式

~ 转换大小写	! 外部过滤器	@ 运行宏	# prev ident	\$ 行尾	% 括号匹配	^ "软"行首	& 重复:s	* next ident	(句首) 下一句首	"soft" bol down	+ 后一行行首
· 跳转到标注	1 ²	2	3	4	5	6	7	8	9	0 "硬"行首	- 前一行行首	= 自动 ³ 格式化
Q 切换至ex模式	W 下一单词	E 词尾	R 替换模式	T back 'till	Y 拷贝行	U 撤销行内命令	I 到行首插入	O 分段(前)	P 粘贴(前)	{ 段首	}	段尾
q 录制宏	w 下一单词	e 词尾	r 替换字符	t 'till	y 拷贝 ^{1,3}	u 撤销命令	i 插入模式	o 分段(后)	p 粘贴 ¹ (后)	[杂项]	杂项
A 在行尾附加	S 删除行并插入	D 删除至行尾	F 行内字符反向查找	G 文尾/行号	H 屏幕顶行	J 合并两行	K 帮助	L 屏幕底行	:	ex命令	" 寄存器 ¹ 标识	行首/列
a 附加	s 删除字符并插入	d 删除 ^{1,3}	f 行内字符查找	g 附加命令 ⁶	h ←	j ↓	k ↑	l →	;	重复 ¹ u/T/f/F	' 跳转到标注的行首	\ 未用!
Z 退出 ⁴	X 退格	C 修改至行末	V 可视行模式	B 前一单词	N 查找上一处	M 屏幕中间行	< 反缩进 ³	> 缩进 ³	?	向前搜索		
Z 附加命令 ⁵	X 删除(字符)	c 修改 ^{1,3}	V 可视模式	b 前一单词	n 查找下一处	m 设置标注	, 反向 ¹ u/T/f/F	.	重复命令	/	向后搜索	

动作

移动光标, 或者定义操作的范围

命令

直接执行的命令,
红色命令 进入编辑模式

操作

后面跟随表示操作范围的指令

extra

特殊功能,
需要额外的输入

q

后跟字符参数

w,e,b命令

小写(b): quux(foo, bar, baz);

大写(B): quux(Foo, Bar, Baz);

主要ex命令:

:w (保存), :q (退出), :q! (不保存退出)

:ef (打开文件 f),

:%s/x/y/g ('y' 全局替换 'x'),

:h (帮助 in vim), :new (新建文件 in vim),

其它重要命令:

CTRL-R: 重复 (vim),

CTRL-F/-B: 上翻/下翻,

CTRL-E/-Y: 上滚/下滚,

CTRL-V: 块可视模式 (vim only)

可视模式:

漫游后对选中的区域执行操作 (vim only)

备注:

(1) 在 拷贝/粘贴/删除 命令前使用 "x (x=a..z,*)

使用命令的寄存器('剪贴板')

(如: "ay\$ 拷贝剩余的行内容至寄存器 'a')

(2) 命令前添加数字

多遍重复操作

(e.g.: 2p, d2w, 5i, d4j)

(3) 重复本字符在光标所在行执行操作

(dd = 删除本行, >> = 行首缩进)

(4) ZZ 保存退出, ZQ 不保存退出

(5) zt: 移动光标所在行至屏幕顶端,

zb: 底端, zz: 中间

(6) gg: 文首 (vim only),

gf: 打开光标处的文件名 (vim only)

Vim 命令小抄

original card by Laurent Gregoires
redesign by brohan

基本移动

h l j k	左/右 移动一个字符; 上/下 移动一行
b w	向 左/右 移动一个单词
ge e	向 左/右 移动一个单词 (光标在词尾)
{ }	移动到 前一个/后一个 段落开头
()	移动到 前一个/后一个 句子开头
0 gm	移动到 行首/行正中
^ \$	移动到行 第一个非空白字符/行尾
nG ngg	移动到第 n 行, 默认到 尾行/首行
n%	移动到文件的 n 百分比处
n 	移动到当前行的第 n 列
nH nL	移动到窗口 顶端/底部 第 n 行
%	匹配下一个项目 (括号、方括号等.....)
M	移动到窗口的中间行

编辑修改

i a	在光标 前/后 插入文本
I A	在 行首/行尾 插入文本
gl	在第一列插入文本
o O	在光标的 下方/上方 插入新行
rc	把光标下的字符换成 c
grc	类似 r , 但替换的是虚拟字符
R	进入替换模式
gR	类似 R , 但替换的是虚拟字符
cm	更改动作 m 跨越的文本
CC or S	更改当前行的内容
C	更改光标到行尾的文本
s	更改一个字符并进入插入模式
~	切换当前字符大小写, 同时光标前移
g~m	切换动作 m 跨越文本的大小写
gum gUm	小写/大写 动作 m 跨越的文本
<m >m	左移/右移 动作 m 跨越的行
n<< n>>	将 n 行 左移/右移 一个缩进位

删除

x X	删除光标 后/前 的字符
dm	删除动作 m 跨越的文本
dd D	删除 当前行/到行尾
J gJ	连接两行 删除缩进、插入空格/直接连接
rd	删除范围 r 包括的行
rdx	删除范围 r 包括的行到寄存器 x

插入模式

^Vc ^Vn	插入字符 c 的本义/十进制值 n
^A	插入最近插入的文本
^@	插入最近插入的文本, 并退出插入模式
^Rx	原样插入寄存器 x 的内容
^N ^P	补全关键字并作 正向/反向 查找
^W	删除光标前的单词
^U	删除当前行光标前所有输入的字符
^D ^T	左移/右移 一个缩进位
^KC1C2	输入 \C1,C2\ 代表的二合字母
^Oc	在临时命令模式执行命令 c
^X^E	向上滚动一行
^X^Y	向下滚动一行
<esc> ^[结束插入模式, 回到命令模式

复制

"x	指定下次删除、抽出和放置使用寄存器 x
ireg	显示所有寄存器的内容
ireg x	显示寄存器 x 的内容
ym	抽出动作 m 跨越的文本
yy or Y	抽出整行
p P	放置寄存器内容到光标 之后/之前
]p [P	类似 "p" "P" , 但调整当前行的缩进
gp gP	类似 "p" "P" , 光标停留在新文本之后

复杂改动

g?m	对动作 m 跨越的文本做 rot13 编码
n^A n^X	光标之上或之后的数值或者字母 +n/-n
gqm	格式化动作 m 跨越的所有行
:rce w	将范围 r 中的行依据宽度 w 居中对齐
:rle i	将范围 r 中的行靠左对齐, 缩进 i 列
:rri w	将范围 r 中的行依据宽度 w 靠右对齐
lmo	将动作 m 跨越的行用命令 c 过滤
n!c	将 n 行用命令 c 过滤
rlc	将范围 r 中的行用命令 c 过滤

可视模式

v V	以 字符/行 方式开始 (结束) 高亮
^V	以列块方式开始 (结束) 高亮
o	交换高亮区域的开始处和光标位置
gv	使用上一次的可视区域开始高亮
aw as ap	选择 "一个单词" / "一个句子" / "一个段落"
ab aB	选择 "一个块" () / "一个大块" { }

撤消, 重做

u U	撤销最近的改动 / 恢复最近被改动的行
. ^R	重复最近的改动 / 重做最近撤销的改动
n.	重复最近的改动 n 次

寄存器

qo qC	记录键入的字符, 存入/添加到 寄存器 c
q	停止记录
@c	执行寄存器 c 的内容
@@	重复上次的 @c 操作
:@c	将寄存器 c 的内容当作 Ex 命令来执行
:rg/p/c	在范围 r 中找到匹配内容 p 时执行 Ex 命令 c

复合移动

- +	上移/下移 一行, 至第一个非空白字符
B W	向 前/后 移动一个单词
gE E	向 前/后 移动一个单词 (光标在末尾)
n_	下移 n-1 行, 至第一个非空白字符
g0	移动到屏幕行第一个字符
g^ g\$	移动到屏幕行 第一个非空白字符/最后一个字符
gk gj	上移/下移 一个屏幕行
fc Fc	移动到 后一个/前一个 字符 c 处
tc Tc	移动到 后一个/前一个 字符 c 之前
;;	正向/反向 重复上次 "f" "F" "t" 或 "T" 命令
[[]]	向后/向前 一小节, 置于小节的开始
[]]]	向后/向前 一小节, 置于小节的末尾
[()]	向后/向前 至未闭合的 '(' '/'
[{ }]	向后/向前 至未闭合的 '{' '/'
[m]m	向后/向前 至 java method 的开始
[#]#	向后/向前 至未闭合的 #if, #else, #endif
[*]*	向后/向前 至注释的 开始/结束 /* */

查找替换

/s ?s	向前/向后 查找 s
/s/o ?s/o	向前/向后 查找 s , 光标偏移量 o
n or /	向前重复上次查找
N or ?	向后重复上次查找
# *	向后/向前 查找光标下的标识符
g# g*	同 # / * , 但也查找部分匹配
gd gD	跳转到光标下标识符的 局部/全局 声明
:rs/f/t/x	将范围 r 中的 f 替换成 t
:rs x	选项 x : g -所有匹配项, c -确认替换 在新的范围 r 中以选项 x 重复上一替换

vim

复制/粘贴/删除

- yy复制当前行
- 5yy复制从当前行开始的5行
- p粘贴到下一行
- P粘贴到上一行
- D当前光标开始删除到行尾
- dd/.删除当前行
- ndd从当前行开始向后删除n行
- d1G删除第1行到当前行的数据
- dnG删除第n行到当前行的数据
- dG删除当前行到最后一行的数据
- x向后删除1个字符
- nx向后删除n个字符
- X向前删除1个字符

搜索

- /str 从光标位置向下搜索str
- ?str 从光标位置向上搜索str
- n 找下一个
- N 找上一个

替换

- :%s/abc/123/g 把abc全部替换成123
- :%s/abc/123/gc 同上，但要用户一个个确认是否替换

撤销

- u退回前一个动作
- ctrl+r重复前一个动作
- :e! 文档还原到最原始状态

块操作

- v字符选择。光标经过的字符被选中
- V行选择。光标经过的行被选中
- ctrl+v块选择。矩阵方式选中
- y复制被选中的数据
- d删除被选中的数据

多文件编辑

- vim file1 file2 同时编辑两个文件file1和file2
- :files 列出vim当前打开的文件列表
- :n 编辑下一个文件
- :N 编辑上一个文件

多窗口操作

- :sp file2 在打开一个文件后，用:sp在新窗口中打开另一个文件file2
- ctrl+w---->j 窗口向下切换。先按ctrl+w，放开，再按j
- ctrl+w---->k 窗口向上切换。先按ctrl+w，放开，再按k

其他操作

- set nu 显示行号
- set nonu 取消行号
- :r filename 把filename文件里的内容插入到当前光标所在位子下
- :!command 暂时离开vim，执行command命令。如:ldf -h

插入

- i在光标前插入
- I在行首插入
- a在光标后插入
- A在行末插入
- o在下一行插入
- O在上一行插入
- r输入替换光标位置字符
- R进入输入替换模式
- 大小写转换
- ESC退出编辑模式

保存

- :w 保存
- :w! 强制保存
- :w fileName 文件另存为fileName
- n1,n2 w fileName 将n1行到n2行的数据另存为fileName

退出

- :q 退出
- :q! 强制不保存退出
- :qw! 强制保存退出
- :x 保存并退出

光标移动

- 左下上右键
- h j k
- \$光标到行尾
- 0光标到行首
- G光标到文档最后一行
- gg光标到文档第一行
- nG/:n<enter>光标到文档第n行
- n<enter>向下移动n行
- n<space>向后移动n字符
- 向上翻页 pageup
- ctrl+b
- 向下翻页 pagedown
- ctrl+f

关于马哥教育



马哥教育

IT 人的高薪职业学院

- ◆ 博客 : <http://mageedu.blog.51cto.com>
- ◆ 主页 : <http://www.magedu.com>
- ◆ QQ : 1661815153, 113228115
- ◆ QQ群 : 203585050, 279599283

祝大家学业有成

谢 谢

咨询热线 400-080-6560