

# Advanced Vim

## Hacker's Café Brown Bag

Aaron Young

ORNL

June 14, 2022

ORNL is managed by UT-Battelle, LLC for the US Department of Energy



# Overview

Mastering Vim

Vim Plugins

Advanced Vim Tips

# Current Topic

Mastering Vim

Vim Plugins

Advanced Vim Tips

# Mastering Vim

## Getting Started with Vim:

- `vimtutor` — Start here if you are new to vim.
- Vim Help Files (aka `:h`)
- ExCL Cheat Sheet → Vim Cheat Sheet

## Mastering Vim:

- Practical Vim by Drew Neil
- More Vim Help Files (aka `:h`)

The  
Pragmatic  
Programmers

for  
Vim 8

## Practical Vim

Second Edition

Edit Text at the  
Speed of Thought



Drew Neil

Foreword by Tim Pope



# Vim Documentation

The Vim documentation consists of two parts:

1. The User Manual:<sup>1</sup>

Task oriented explanations, from simple to complex. Reads from start to end like a book.

2. The Reference Manual:<sup>2</sup>

Precise description of how everything in Vim works.

There is also a Quick Reference<sup>3</sup> and a Tutorial.<sup>4</sup>

---

<sup>1</sup>:h user-manual

<sup>2</sup>:h reference\_toc

<sup>3</sup>:h quickref

<sup>4</sup>:h tutor

# Vim Cheat Sheet

version 1.1  
April 1st, 06

vi / vim graphical cheat sheet															
<b>Esc</b> normal mode															
~ toggle case	! external filter	@ play macro	# prev ident	\$ eol	% goto match	^ "soft" bol	& repeat is	* next ident	( begin sentence	) end sentence	"soft" bol down	- prev line	+ next line	= auto-format	
~ goto mark	1 2 3 4 5 6 7 8 9 0	" hard" bol	_ prev line	= auto-format											
Q ex mode	W next WORD	E end WORD	R replace mode	T back 'till	Y yank line	U undo line	I insert at bol	O open above	P paste before	{ begin parag.					
Q record macro	W next word	E end word	R replace char	T 'till	Y yank	U undo	I insert mode	O open below	P paste after	{ end parag.					
A append at col	S subst line	D delete to col	F "back" find ch	G eof/ find ln	H screen top	J join lines	K help	L screen bottom	: ex cmd line	": reg. ^	": spec	": goto mk. bol	": not used!		
a append	s subst char	d delete	f find char	g extra cmd	h ←	j ↓	k ↑	l →	:	repeat	t/T/f/F				
Z quit	X back-space	C change to col	V visual lines	B prev WORD	N prev (find)	M screen mid'l	< un- <sup>3</sup> indent	> indent <sup>3</sup>	?* find (rev.)						
Z extra cmd	X delete char	C change	V visual mode	b prev word	n next (find)	m set mark	,	, t/T/f/F	repeat cmd	/ * find					

**motion** moves the cursor, or defines the range for an operator

**command** direct action command, if red, it enters insert mode

**operator** requires a motion afterwards, operates between cursor & destination

**extra** special functions, requires extra input

**Q** commands with a dot need a char argument afterwards  
bol = beginning of line, eol = end of line, mk = mark, yank = copy

words: `guux(foo, bar, baz)`  
WORDS: `guux(foo, bar, baz)`

**Main command line commands ('ex')**:

:w (save), :q (quit), :q! (quit w/o saving)  
:e f (open file f),  
:%s/x/y/g (replace 'x' by 'y' filewide),  
:h (help in vim), :new (new file in vim),

**Other important commands:**

CTRL-R: redo (vim),  
CTRL-F/-B: page up/down,  
CTRL-E/-Y: scroll line up/down,  
CTRL-V: block-visual mode (vim only)

**Visual mode:**  
Move around and type operator to act on selected region (vim only)

**Notes:**

(1) use "x before a yank/paste/del command to use that register ('clipboard') (x=a..z,")  
(e.g.: "ay\$ to copy rest of line to reg 'a')

(2) type in a number before any action to repeat it that number of times  
(e.g.: 2p, d2w, 5i, d4j)

(3) duplicate operator to act on current line  
(dd = delete line, >> = indent line)

(4) ZZ to save & quit, ZQ to quit w/o saving

(5) zt: scroll cursor to top,  
zb: bottom, zz: center

(6) gg: top of file (vim only),  
gf: open file under cursor (vim only)

For a graphical vi/vim tutorial & more tips, go to [www.viemu.com](http://www.viemu.com) - home of ViEmu, vi/vim emulation for Microsoft Visual Studio

# Current Topic

Mastering Vim

Vim Plugins

Advanced Vim Tips

# Vim Plugins

## My Favorite Vim Plugins

<https://geekdude.github.io/tech/vim-plugins/>

# Current Topic

Mastering Vim

Vim Plugins

Advanced Vim Tips

# Vim Lists

Vim has multiple built in lists.

- Quickfix list — commands start with :c.
- Location list — commands start with :l.
- Arguments list — List of files specified on the command line.
- Change listg — g;, g,, and :changes.

See [:h quickfix](#), [:h arglist](#), [:h changelist](#), and [The quickfix and location lists in Vim](#) for more information.

# Vim Long Line Navigation

```
+-----+  
^ |  
| c-e (keep cursor) |  
| H(igh)           zt (top) |  
| | ^ |  
|       ze      |     zs |  
| M(iddle)   zh/zH <--zz--> zl/zL |  
| | | |  
| | v |  
| L(ow)          zb (bottom) |  
| c-y (keep cursor) |  
v |  
+-----+
```

<https://stackoverflow.com/questions/5989739/vertical-navigation-in-long-lines>

# Command Line Execution

See `vim -h`

```
vim -p      open in tabs  
-o      open in split  
-O      open in vsplit  
+      open end of file  
+<num> open at line number
```

`vimdiff <(command) <(command)` to diff two commands

# Recovery

Use vim -r to list all recovery files and change status.

# Mode Line

2 spaces:

```
/* vim: set shiftwidth=2 tabstop=2 softtabstop=2 expandtab: */  
# vim: shiftwidth=2 tabstop=2 softtabstop=2 expandtab
```

3 spaces:

```
/* vim: set shiftwidth=3 tabstop=3 softtabstop=3 expandtab: */  
# vim: shiftwidth=3 tabstop=3 softtabstop=3 expandtab
```

4 spaces

```
/* vim: set shiftwidth=4 tabstop=4 softtabstop=4 expandtab: */  
# vim: shiftwidth=4 tabstop=4 softtabstop=4 expandtab
```

# Netrw — Vim's built in file browser

u Previous Directory

U Subsequent Directory

qf Display information on file

d Make a directory

% Open a new file

R Rename file or directory

D Remove file or directory

x View file with default program

mf Mark files

mp Print marked files

mu Unmark all files

mt Dir under cursor becomes markfile target

mc Copy marked files to markfile target

me Placed marked files on arg list

mx Apply shell command to marked files

# Vim Digraphs

Documentation: [:h digraph](#)

Examples:

i^km\* → μ

i^kDG → °

Cheatsheet: <https://sheet.shiar.nl/digraphs>

Note: Characters can also be inserted via character code using CTRL-V.<sup>5</sup>

---

<sup>5</sup>:h i\_ctrl-v

# Filter buffer through bash

:.!bash  
:%!bash

Filter the current line through bash  
Filter all lines through bash

The line is standard input to the command and standard out is placed on the line in vim.

So you could do things like :.!xargs ls to use the current line as arguments to ls.

What I normally do is regex/column edit the commands I want and then run them all using :%!bash. This is faster than saving and executing a script.

:!.ls on an empty line then pastes the directory listing to the current buffer. Without the . the command is run and output as a popup without filtering the information from the current buffer.

%!sort sort all lines in the file.

Visual line select + : is also powerful to run command on selection.

## Bonus: VimGolf



# Advanced Vim

## Hacker's Café Brown Bag

Aaron Young

ORNL

June 14, 2022

ORNL is managed by UT-Battelle, LLC for the US Department of Energy

