

Terminal Programs

iterm2

Download, install and use iterm2 instead of the default MacOS terminal

double click + CMD-v

double click text to copy and then CMD-v to paste

zsh

install and use z shell instead of bash

oh my zsh

install oh my zsh to easily configure zsh

~/.zshrc

include common aliases and other shell configuration here

plugins=(git node npm kubectl)

add these shell completion plugins to .zshrc

ZSH_THEME="robbyrussell"

shell theme set in .zshrc <https://github.com/ohmyzsh/ohmyzsh/wiki/Themes>

alias abc=cmd

run cmd when you type abc

Shell Shortcuts

CTRL- c Stop the current command

CTRL- a Go to start of line

CTRL- e Go to end of line

CTRL- r Search command history

CTRL- l Clear the screen

!! Run last command

!abc Run last command that starts with abc

Shell Variables

env Show all environment variables

echo \$ABC Show a specific environment variables

ABC=123 Create a local variable

export ABC=123 Create a variable accessible by child processes

\$NODE_ENV Used node.js code; expects value to be production or anything else

\$HOME Home directory

IO Redirection

cmd < file

cmd receives contents of file as input (stdin)

cmd > file

cmd output (stdout) is written to file

cmd > /dev/null

discard cmd output

cmd >> file

append cmd output to end of file

cmd 2> file

cmd error output (stderr) is written to file

cmd 2>&1

send cmd error output (stderr) to the standard output (stdout)

cmd 2>&1 > file

cmd output (stdout) and error output (stderr) is sent to file

cmd 2>&1 | grep abc

search cmd output and error output for abc

cmd | cmd2

cmd2 receives output from cmd as input

Process Management

cmd1 ; cmd2

run cmd1 and then cmd2, regardless of cmd1 exit status

cmd1 && cmd2

run cmd1; run cmd2 if cmd1 is successful

cmd1 || cmd2

run cmd1; run cmd2 if cmd1 is not successful

cmd &

run cmd in the background

CTRL-z

cause currently running command to sleep

bg

continue running sleeping command in background

fg

move command running in the background, to run in the foreground

jobs

list all currently running processes

fg 2

move the second command listed by jobs, to run in the foreground

Directory Commands

mkdir abc Create directory named abc

mkdir -p abc Create directory if it does not exist

mkdir -p abc/xyz/123 Create all directories and subdirectories if they do not exist

pwd Show the current directory

cd abc Change to directory abc

cd .. Change to the parent directory

cd ../xyz Change to the sibling directory xyz



Directory Commands (cont)

cd -	Change back to previous directory
ls	List files in directory
ls -la	List files, including dot-files, with extra info
ls -lhSr	List files in reverse size order; w/friendly size

File Operations

touch file	update file last modified date; create file if not exists
cat file	output contents of file
less file	paged output contents of file
cp file1 file2	copy file1 to file2
mv file1 file2	mv file1 to file2
rm file	delete file
head -5 file	show the first 5 lines of file
tail -5 file	show the last 5 lines of file
tail -f file	show the last few lines of the file and follow output

Command help

tldr	https://tldr.sh/
man cmd	get command documentation
cmd -h OR cmd -help	get help from the command

Finding Files

find .	list all files and directories under the current directory and subdirectories
find . -type f	list all files under the current directory and subdirectories
find . -type f -iname 'abc'	list all files that have abc (case insensitive) in the file name
find . -type f -maxdepth 3	list all files up to 3 directories in depth
find . -type f -mmin -5	list files that were modified in the last 5 minutes
find . -type f -iname "abc" -exec cmd {} \;	run cmd on each file that contains abc; use {} in place of the filename; end the command with backslash+semi-colon
find . -type f -iname 'abc' -exec chown root {} \;	make root the owner of all abc files
grep -Er 'this that' src/	find files under the src/ dir that contain "this" or "that"
grep -Er -l 'test' src/	list files only that contain 'test'
grep -Er -h 'test' src/	list lines in files only that contain 'test'

Text Manipulation

cut -d, -f1	split input by comma and output the first field
cut -d' ' -f2-5	split input by space and return the fields 2, 3, 4, 5

Text Manipulation (cont)

cut -c1-10	return the first 10 characters of each line of input
sort	sort each line of input alphabetically
sort -n	sort each line of input numerically
sort -nr	sort each line of input numerically, descending (reverse)
uniq	return only one line for each duplicate adjacent line of input
uniq -c	same as above, but includes a count of how many times the line appears
sed -E -e 's/this/that/' file	replace the first instance of "this" with "that" on each line in file
sed -E -e 's/this/that/g' file	replace all instances of "this" with "that" on each line in file
sed -i.bak -E -e 's/this/that/g' file	same as above, but modifies file in place, and creates a backup
awk '{ print \$2 }'	print out the second field in each line

Simple Regular Expressions

.	match any character once
*	match the preceding character 0 or more times
+	match the preceding character 1 or more times
?	match the preceding character 0 or 1 time
()	group patterns



Simple Regular Expressions (cont)

| match the pattern on the left OR right

^ match the start of the line

\$ match the end of the line

Common Command Combos

tmux

vim

emacs

git

C

By **Edward** (edward.z)

cheatography.com/edward-z/

Published 23rd February, 2022.

Last updated 24th February, 2022.

Page 3 of 3.

Sponsored by **Readable.com**

Measure your website readability!

<https://readable.com>