Cheatography

Zsh Primer(Start-Up files, Functions, Globbing) Cheat Sheet

by JohnMalaperdas (JohnMalaperdas) via cheatography.com/179057/cs/37295/

Options

EXTENDEDGLOB	Enables zsh extra features
PUSHDTOHOME	`pushd` with no arguments redirects to $HOME
NOEQUALS	prepending a command name with = replaces the command name with it's full path. (e.g. `echo =sudo` outputs `/usr/bin/sudo`) Can used with `ls -l =(command)` instead of `which command`
NOBANGHISTORY	Disable ! (exclamation mark) history substitution csh-style.
RECEXACT	TAB-completion allows exact matches to be accepted even though there could be other matches.

Globbing Qualifiers

Example	*pattern*(qualifier) e.g. *(@)
/	directories
X	executable by others files
x	executable by owner
W	world writable files
w	user writable files
R	world readable files/readable by others
r	owner readable files
U	user-owned files
@	symlinks
*(W^@)	world-writable but no symlinks
*(.)	find all plain files(e.g. no sym links)
*(s)	find setuid files(only user-set)
<->	filename containing integers

Globbing Qualifiers (cont)

<100-200>	integer range pattern
*(u1001)	search for files owned by uid 1001
if echo is used and no results are available then the pattern is echo'ed.	

Expansions

Arithmetic value Expansion	
$[...] (e.g. $[RANDOM % 5]	
Command Output File	=(command)
Explanation	
Outputs the file name holding the command's output. Can be used to edit command output on the spot. Different from `<(..)` which creates a named pipe(FIFO)	

Command Line Editing (ZLE)

Zsh supports line editing in either Emacs or vi mode	
ZLE negates the need for `fc`	
In Emacs-mode CTRL+R	fwrd-search
Link to page	
Enable hosts-name completion	hosts=(host_one.uk host_two.com)

Globbing Basics

^	(pattern) negation
<x-y>	Integer range(x and/or y are ommitable) e.g. run<>
(.c|.h|.m)	grouping(or)
**/	recursive subdirectory search

Globbing Basics (cont)

/bar/	directory search with partial path
*.c~bar.c	exclude bar.c

Functions

Simplest Func def	func_name () { ... }
One liners are allowed	func_name () echo "one liner"
aliases are parsed when function is parsed so...	
>func_name () { > ypmatch $1 passwd >} % alias other_name=func_name % other_name % func_output	
one liners are also a thing	func_name () { for i; do echo $i; done

Check argument count	if (($# ==0))
	  then echo "no args";
	fi

Declare integer	integer j=3

Important Links

Prompts and Dirs

Aliases

Define alias	
Aliases can also inlcude pipes	alias m1='cat /etc/passwd | grep mike'
Zsh defines global aliases which can be used in a number of ways.	
Username aliases	

By **JohnMalaperdas**
(JohnMalaperdas)

cheatography.com/johnmalaperdas/

Not published yet.
Last updated 14th March, 2023.

Aliases (cont)

Global aliases can contain strings in general	alias -g m2='| grep -f file'
Global alias process	% alias -g PASS='<(ypcat passwd)' % grep pfalstad PASS

Start-Up Files

ENV Var	ZDOTDIR(If not set HOME is used)
They are read in order	With some exceptions
.zshenv	
.zprofile	For ksh fans. Should not be used with .zlogin. But it's up to you.
.zshrc	Sourced in interactive shells. Contains aliases, functions, options, key bindings etc.
.zlogin	Sourced in login shells. Should contain only relevant commands. Should not be used for aliases or to change the env at all. Should only set terminal type and run series of external commands.
.zlogout	Sourced during log out. Same as above.

Operators and Vars

%	Modulus (can be used in scripts).
$RANDOM	Random Number var (returns a random num each time it is called).
$TTY	Current TTY.

Build-Ins

View function definition	`functions` func_name
View all function definition	`functions`
Autoload functions on startup	autoload func_name_1 func_name_2

Explanation

FPATH points to dir where files containing function definitions are stored. Each file must have 755 perms and be named afte the function with no extensions. Call `autoload` on .zshrc and then you can call the functions.

Bonus: Place #! on top of function file and you can call the files as scripts too! <3 This a bit slower however since a separate process is created.

`pushd`	If only two dirs exist in stack, swap their position.

History

`fc` command	Calls default editor(vi by default) to edit history and all. --It effectively "replays" the last recorded command.
`r` command	Re-does the last command replacing string with other strings
`r` command example	% echo foo % r foo=bar echo bar