

# Zsh Cheat Sheet

## Notation

This reference card is using the Emacs notation of keys. C- is the prefix for control. M- is the meta prefix, for example the Alt key on a PC. You can also press ESC to access meta command. For example M-w and ESC w are equivalent.

## Changing Directories

There is no need to type cd to navigate between directories if you have set the **auto\_cd** option.

There is a shortcut for changing between similarly named directories. If you type **cd old new**, then the shell will look for the first occurrence of the string **old** in the current directory, and try to replace it with **new**. For example:

```
% pwd
/src/zsh-3.0.8/Src
% cd 0.8 1.9
/src/zsh-3.1.9/Src
```

## Key Bindings

delete the line	C-u
set mark	C-SPACE
copy from point to mark	M-w
kill-from point to mark	C-w
copy the previous word	C-M-/
look in history for match of word on line	M-p
push line to be used again	C-q
extract last word from history	M-.
paste text from kill ring	C-y
display command bound to a key	M-x where-is

C-u can also be used to clear a password that you typed wrong.

If you hit M-. cycle repeatedly it will look back in history for previous last word in history.

After pressing C-y you can press several times on M-y. It will cycle through the previous cut or copy (just like emacs.)

You can see the list of all binding just by running **bindkey**.

bind string	<b>bindkey -s agi 'apt-get install'</b>
bind C-w	<b>bindkey ^w where-is</b>
bind M-a	<b>'\eA' backward-word</b>
remove key binding C-b	<b>bindkey -r ^b</b>

## History

prints the start time of the history	<b>history -d</b>
prints that as well as the date	<b>history -f</b>
prints the command's elapsed time	<b>history -D</b>

Setting the **share\_history** option will share history between sessions of zsh. The **extended\_history** option will save the the command start and duration time. You can skip duplicate entries with the **hist\_ignore\_all\_dups** option.

## Basic Glob Matching

any string	*
any character	?
any single character from class	[class]
any single character not from class	[^class]
any number between 1 and 3	foo<1-3>
alternatives	(foo bar)

## Glob Qualifier

The simplest glob qualifiers is similar to the indications displayed by **ls -F**

file	*()
directory	*/
executable	*(*)
symbolic links	*(@)

You can also pick files by modification (m) or access (a) time, either before (-), at, or after (+) a specific time, which may be measured in days (the default), months (M), weeks (w), hours (h), minutes (m) or seconds (s). These must appear in the order m or a, optional unit, optional plus or minus, number.

```
print *(m1)
```

Files that were modified one day ago i.e. less than 48 but more than 24 hours ago.

```
print *(aw-1)
```

Files accessed within the last week, i.e. less than 7 days ago.

There is also the (u) and (g) that lets you match on a given user or group.

```
print /var/*(~u:root)
```

All files in /var that are not owned by root.

## Recursive Glob

recurse directories	**
recurse and follow symlinks	***

```
ls -l **/README
```

Search for 'README' in all subdirectories.

```
rm -f **/*(mh+6)
```

Deletes recursively all files more than 6 hours old.

## Extended Glob Matching

These only work with the option **extended\_glob** set.

negative matches	~*.o
------------------	------

Matches all file except the one ending in .o.

match all file A minus pattern B	A~B
----------------------------------	-----

```
ls ~*.*()
```

Matching all files which do not have a dot in filename

zero or one occurrences of foo	(foo )
zero or more occurrences of foo	(foo)#
one or more occurrences of foo	(foo)##

## Completion

Zsh comes with a powerful completion system available when pressing the TAB key.

```
echo $P TAB
```

List all variable beginning with P.

```
acroread foo TAB
```

List all document ending in .pdf and starting with foo.

```
dpkg -l ad TAB
```

List all Debian packages starting with ad.

```
tar xjf zsh.tar.bz2 zsh-4.3.12/Doc TAB
```

Expands a directory starting with Doc within the tar file.

## Redirection

```
date > foo > bar
```

Writes the date to two files, named **foo** and **bar**.

```
echo exit 0 >> *.sh
```

Append **exit 0** a the end of each shell file.

## Options

```
setopt auto_cd
```

When you type something with no arguments which isn't a command, if it matches a directory it will move in it.

```
setopt extended_glob
```

Enables extended glob.

```
setopt correct
```

The shell will propose a command that will loosely match what you have mistyped.

```
setopt no_beep
```

Turn off beep on completion.

```
autoload -Uz compinit compinit
```

Use modern completion system, highly recommended!

```
unsetopt chase_links
```

Don't resolve symbolic links to their true values.

## Zmv

You need to run **autoload zmv** in order to use this command.

The **-n** argument does a dry run, i.e. prints what would be done and **-C** does a copy instead of a move.

```
zmv '(*).txt' '$1.tex'
```

Rename all file ending in .txt to .tex.

```
zmv -W '*.txt' '*.tex'
```

The same with automatic wildcard match.

```
zmv '(*)foo(*)' '$1bar$2'
```

Change part of files from **file-foo.name** to **file-bar.name**

## Miscellaneous

print out function	<b>which _tar</b>
edit variable	<b>vared PATH</b>
edit function in place	<b>zed -f function_name</b>