

ranger cheatsheet

<http://ranger.nongnu.org>

git clone <https://github.com/hut/ranger>

#ranger on irc.freenode.net

ESC abort	F1 help	F2	F3 inspect file	F4 edit	F5 copy	F6 cut	F7 :mkdir	F8 :delete	F9	F10 quit	F11	F12
~ open bookmarks	! :shell	@ :shell %s	# :shell -p	\$	%	^	&	*	()		+ chmod (2)
1 (1)	2 (1)	3 (1)	4 (1)	5 (1)	6 (1)	7 (1)	8 (1)	9 (1)	0 (1)	- chmod (2)	= chmod (2)	
TAB prev. tab	Q quit	W show log	E edit	R reload this directory	T	Y	U	I :rename (insert)	O	P	{	} traverse sub directories
tab next tab	q close tab	w show back-ground tasks	e	r :open_with	t tag files	y yank (3)	u undo	i inspect file	o sort	p paste (3)	[move up in parent dir] move down in parent dir
				^R reset ranger			^U move up half page					
CAPSLOCK	A :rename (append)	S open shell	D	F	G go to top	H back in history	J move down half page	K move up half page	L forward in history	: console	' open bookmarks	
	a :rename (skip extension)	s :shell	d cut (3) (4)	f :find	g :cd ...	h go up 1 directory	j move down	k move up	i enter dir/ open file	: console	" tag files with custom tag	
			^D move down half page	^F move down 1 page		^H toggle hidden files			^L redraw			
SHIFT	ZZZ/ZQ = quit	X	C	V visual mode	B	N search previous	M change (5) linemode	<	>	? show help		
	z toggle options	x	c select files in certain order	v invert selection	b	n search next	m save bookmark	,	.	/ :search		
			^C abort task		^B move up 1 page	^N new tab						

Macros can be used in commands. They are like global variables with dynamic content:

%f The base name of the current file
 %d The path of the current directory
 %s The names of the currently selected files
 %t The names of all tagged files in this directory
 %c The paths of the currently copied files
 %any The key used in a key binding with "<any>"
 Example: `map x<any> shell -w echo %any`
 %rangerdir The path to the ranger python module
 %space Just a space, to avoid typing trailing spaces
 Example: `map yp shell echo %d/%f | xsel -i`
 They can be escaped by replacing % with %%.

Config files: run "**ranger --copy-config=all**" to copy the default config files to `~/.config/ranger/`.

rc.conf: A list of commands that are executed when ranger starts. Options, key bindings and aliases are found here.
 Pro tip: Adding "export RANGER_LOAD_DEFAULT_RC=FALSE" to your shell rc will skip loading the default rc.conf before your own.
commands.py: A python script containing custom commands
rifle.conf: Rules for rifle, the file opener. Each line looks like
`list of conditions = command`
 When ranger opens a file, it tests those conditions. The first command where all conditions are true will be executed.
scope.sh: The script that generates file previews.
 Plugins can be put in the **plugins/** subdirectory, colorschemes in **colorschemes/**. See `/usr/share/doc/ranger/examples`.

(1) numbers can be used as a quantifier in various commands, for example 5j will move the cursor down 5 by lines, 3<space> selects 3 files, 4<TAB> moves you to the 4th tab.
 (2) the keys -, + and = change the permissions of files. See "man chmod".
[+-][augo][rwxXst] (e.g. +gw means "add write permissions to the group")
[+-][rwxXst] (e.g. -x means "remove execute permissions from everybody")
<octal>= (e.g. 777= means "give full permissions to everybody")
 (3) yank, copy, paste: To copy files, select them with the cursor (or <space>, in case of multiple files) → type **dd** (to cut) or **yy** (to copy) → move to the destination → type **pp**.
 Type **da** (or **ya**) to add files to the copy buffer, allowing you to copy from multiple folders.
 (4) **d** also starts the keybindings **dc** (calculate size of the content of a directory), **du/du** (calculate directory size with the "du" program), **dd** (open the console with ":delete")
 (5) **M<key>** changes the *linemode* - the way files are drawn. **Mf** draws just the file name, **Mp** draws permissions, **Mi** draws file type information, **Mt** draws metadata, as defined with the **:meta** command. You can add custom linemodes as described in `/usr/share/doc/ranger/examples/plugin_linemode.py`

Commands can be typed in by pressing : or added to `~/.config/ranger/rc.conf` to apply then whenever ranger starts. All commands are listed in the man page. Some important ones:
:shell [<flags>] <command> calls the given <command> with the shell specified in the environment variable \$SHELL. <flags> can be "-f" to fork the process or "-p" to pipe the output to a pager. Macros like %f and %s are especially useful here. Example: `:shell -f inkscape %f` or `:shell sudo cp %c ./`
:alias <new> <old> creates the command <new> that calls <old>. The neat thing is that you can pass arguments to the next command. Example: `:alias touch shell touch` will allow you to type `:touch FILE`, which will be translated to `:shell touch FILE`.
:map <key> <command> makes the <key> run <command> when pressed. This is the typical way to define key bindings in rc.conf. There is also "pmap" to define keys in the pager and "unmap" and "punmap" to remove key bindings.