✏ Conventions

↑, ↓, ←, →	*Up*, *Down*, *Left*, *Right*
↩, ⌃, ⇄	*Alt*, *Ctrl*, *Shift*
a, **b**, **c**, **<key>**	*a, b, and c* keys, etc.
[Home], **[Insert]**, **[<Key>]**	*Home*, *Insert* keys, etc.
{ 🖱 L}	**Click** *left key* of mouse
{ 🖱 R}	**Click** *right key* of mouse
✱, ✱✱	Additionl infomration in note area

This cheat sheet comes from `man tmux`.

See Tmux on Github

❓ Help

?	**List** all key bindings
`:list-keys`	
:	**Start** command prompt

Default *<prefix>*: ⌃ + **b**

Prefix key table

Following sheets show shortcuts and commands under key table of default prefix.

```
set-option -g prefix "C-b"
set-option -g prefix2 "<your prefix>"
```

🗄 Buffer

#		
`:list-buffer`	**List** the global buffers	
-		
`:delete-buffer`	**Delete** the most *recently added* buffer	
=		
`:choose-buffer`	Start **buffer mode**(as a pane) for choosing one	
]		
`:paste-buffer`	**Insert** the content of *current buffer*	

▱ Pane

x `:kill-pane`	**Kill** pane ✖
; `:last-pane`	*Focus at* **Last** one
o `:select-pane -t :.+`	*Focus at* **Next** one
↑, ↓, ←, → `:select-pane -U/-D/-L/-R`	*Focus at* **Up**, **Down**, **Left**, **Right** one
! `:break-pane`	**Move** this pane to new window
{ `:swap-pane -U`	**Swap position** with *previous pane*
} `:swap-pane -D`	**Swap position** with *next pane*
↩ + **1** `:select-layout even-horizontal`	Spread out evenly(**from left to right**) ▱
↩ + **2** `:select-layout even-vertical`	Spread out evenly(**from top to bottom**)
↩ + **3** `:select-layout main-horizontal`	1. *Large pane* is on the **top** 2. *Rest panes* are spread on **bottom**-(from left to right)
↩ + **4** `:select-layout main-vertical`	1. *Large pane* is on the **left** 2. *Rest panes* are spread on **right**-(from top to bottom)
↩ + **5** `:select-layout tiled`	Spread out as evenly as possible(**rows and columns**) ▦
q `:display-panes`	**Display** *indicators(as number)* of panes
m `:select-pane -m`	**Mark** pane
M `:select-pane -M`	**Clear mark** of pane

⊞ Pane (cont)

↩ + ↑, ↓, ←, → `:resize-pane -U/-D/-L/-R 5`	**Resize** pane by *5 lines* *Up, Down, Left, Right*
⌃ + ↑, ↓, ←, → `:resize-pane -U/-D/-L/-R`	**Resize** pane by *1 lines* *Up, Down, Left, Right*
z `:resize-pane -Z`	*Resize* pane to **Maximum/Normal** size

⊡ Window

& `:kill-window`	**Kill** window ✖
, `:rename-window`	**Rename** window(prompt)
. `:move-window`	**Move** window(prompt)
[0-9] `:select-window -t := <N>`	*Select* <N> window
' `:select-window -t :%%`	*Select* <N> window(prompt)
l `:last-window`	*Select* **last** window
p `:previous-window`	*Select* **previous** window
f `:find-window`	**Search** for the *fnmatch(3) pattern* in window names, titles, and visible content
" `:split-window`	**Split** window *vertically*
% `:split-window -h`	**Split** window *horizontally*
⌃ + o `:rotate-window`	**Rotate panes** in window

⌨ Client

d `:detach-client`	**Detach** client
r `:refresh-client`	**Refresh** current client
⌃ + z `:suspend-client`	**Suspend** client
i `:display-message`	**Display** info of current pane ⓘ
- `:show-messages`	**Show** client messages
L `:switch-client -l`	**Switch** the current session for client to *Last one*
(`:switch-client -p`	**Switch** the current session for client to *Previous one*
) `:switch-client -n`	**Switch** the current session for client to *Next one*
t ":clock-mode"	**Display** a *large clock* ⊙

◖ Session

$ `:rename-session`(prompt)	**Rename** session

▤ :choose-buffer

[Up], [Down]	Select **previous/next** buffer
[Enter]	**Choose** selected buffer
⌃ + s	**Search** by name
n	**Repeat** last search
t	**Toggle** if *buffer is tagged*
⌃ + t	Tag **all** buffers
T	Tag **no** buffers
d, D	Delete **selected/tagged** buffers

By **Mike Lue** (Mike Lue)
cheatography.com/mike-lue/
gh.mikelue.guru/

Not published yet.
Last updated 26th August, 2019.
Page 2 of 6.

:choose-buffer (cont)

f	Enter a format to **filter** items
O	Change **sort order**
v	Toggle **preview**
q	**Exit** mode

:choose-client

[Up], **[Down]**	Select **previous/next** client
[Enter]	**Choose** selected client
v	Toggle **preview**
^ + s	**Search** by name
n	**Repeat** last search
t	**Toggle** if *client is tagged*
^ + t	Tag **all** clients
T	Tag **no** clients
d, **D**	Detach **selected/tagged** clients
x, **X**	*Detach and HUP* **selected/tagged** client
z, **Z**	*Suspend* **selected/tagged** clients
f	Enter a format to **filter** items
O	Change **sort order**
q	**Exit** mode

:choose-tree

[Up], **[Down]**	Select **previous/next** item
[<], **[>]**	*Scroll* list of previews **left/right**
[Enter]	**Choose** selected item
^ + s	**Search** by name
n	**Repeat** last search
t	**Toggle** if *item is tagged*
^ + t	Tag **all** items
T	Tag **no** items
:	**Run** a command for each *tagged item*
f	Enter a format to **filter** items
O	Change **sort order**
v	Toggle **preview**
q	**Exit** mode

Copy Mode

[	**Start** copy mode
[PageUp]	**Start** copy mode and *scroll one page up*
q	**Quit** copy mode
^ + c	

☑ Vi Copy Mode

Following sheets show keys in **Vi edit mode**

`set-window-option mode-keys vi`

<N> - As key **0-9** for times of repeating the shortcuts

E.g.: "5j" - Move the cursor for left by *5 times*

<ch> - Any printable character

<word> - Continuous characters of `[_0-9a-zA-Z]`

<WORD> - Continuous non-blank characters

#line - As integer value for line number. E.g.: "13"

☑ ⎘ Copy/Selection

[Space]	**Begin** selection
v, **V**	**Rectangle/Line-wise** selection
o	**Switch other end** in *selected area*
[Enter] ⌃ + j	**Copy selection** ✱
A	**Append selection** to copied buffer ✱
D	**Copy to end of line**(from current position) ✱
[Esc]	**Clear** selection
✱ - Will *quit the copy mode* after the action	

☑ Q Search

/, **?**	**Start Search** <text> *forward/backward*
n, **N**	*<N>* Search **next/reverse**

☑ ⌖ Cursor

	<N> Move cursor for:
k, **j**, **h**, **l** ↑, ↓, ←, →	**Up, Down, Left, Right**
⌃ + h [Backspace]	**Left**
	Move cursor to(current line):
^	**First non-blank char**
0	**First char**
$	**Last char**

☑ ⌖ Cursor (cont)

	<N> Move cursor to(current line):
f<ch>, **t**<ch> **F**<ch>, **T**<ch>	**Next** <ch>/T<ch> **Previous** <ch>/<ch> T
;, **,**	*<N>* **Repeat** the latest *"f/t<ch>"* in *same/opposite* direction
	<N> Move cursor to:
w, **W** **e**, **E** **b**, **B** **{**, **}**	**Next** *beginning* of <word>/<WORD> **Next** *end* of <word>/<WORD> **Previous** *beginning* of <word>/<WORD> **Previous/Next** paragraph
	<N> Move cursor to(current screen):
H **M** **L**	**Top**(Head) **Middle** **Bottom**(Low)

☑ 🖵 Screen Scroll

⌃ + b, ⌃ + f [PageUp], [PageDown]	*<N> Page* **Up**/**Down**
⌃ + d, ⌃ + u ⌃ + ↑, ⌃ + ↓	*<N> Half Page* **Up**/**Down**
⌃ + y ⌃ + ↑	*<N> One line* **Up**
⌃ + e ⌃ + ↓	*<N> One line* **Down**
: #line	**Goto** *"#line"* line ↓⁹₁
g **G**	**Top** of *history* **Bottom** of *history*

By **Mike Lue** (Mike Lue)
cheatography.com/mike-lue/
gh.mikelue.guru/

Not published yet.
Last updated 26th August, 2019.
Page 4 of 6.

Sponsored by **ApolloPad.com**
Everyone has a novel in them. Finish Yours!
https://apollopad.com

⊕ Emacs Copy Mode

Following sheets show keys in **Emacs edit mode**

`set-window-option mode-keys emacs`

Cancel mode: **[Esc]**

<N> - As key "↩ + **0-9**" for times of repeating the shortcuts	
E.g.: "↩+5 ⌃+b" - Move the cursor for left by *5 times*	
<ch> - Any printable character	
#line - As integer value for line number. E.g.: "13"	

⊕ ⎙ Copy/Selection

⌃ + **[Space]**	**Begin** selection
⌃ + w ↩ + w	**Copy selection** ✻
⌃ + k	**Copy to end of line**(from current position) ✻
⌃ + g	**Clear** selection
✻ - Will *quit the copy mode* after the action	

⊕ Q Search

⌃ + r, ⌃ + s	**Start Search** `<text>` *backward/forward*
R	**Rectangle** selection
n, N	*<N>* Search **next/reverse**

⊕ ⎚ Cursor

	<N> Move cursor for:
⌃ + p, ⌃ + n ↑, ↓	**Up, Down**
⌃ + b, ⌃ + f ←, →	**Left, Right**
	Move cursor to(current line):
↩ + m	**First non-blank char**
⌃ + a **[Home]**	**First char**
⌃ + e **[End]**	**Last char**
	<N> Move cursor to(current line):
f<ch>, **t**<ch> **F**<ch>, **T**<ch>	**Next** *<ch>*/T̲*<ch>* **Previous** *<ch>*/*<ch>*T̲

⊕ ⎚ Cursor (cont)

;, ,	*<N>* **Repeat** the latest *"f/t<ch>"* in *same/opposite* direction
	<N> Move cursor to:
↩ + f	**Next end** of `<word>`
↩ + b	**Previous** `<word>`
↩ + {, ↩ + }	**Next/Previous** `paragraph`
	<N> Move cursor to(current screen):
↩ + R, ↩ + r	**Top/Middle**

⊕ 🖥 Screen Scroll

[PageUp] ↩ + v	*<N>* Page **up**
[Space] **[PageDown]** ⌃ + v	*<N>* Page **down**
↩ + ↑, ↩ + ↓	*<N> Half Page* **Up/Down**
⌃ + ↑, ⌃ + ↓	*<N> One line* **Up/Down**
g *#line*	**Goto** *"#line"* line ↓₁⁰
↩ + <	**Top** of *history*
↩ + >	**Bottom** of *history*

▥ Target pane

`<session>:<window>.<pane>`	**Full name**
Special **token**:	
! (*{last}*)	**Last** pane
+ (*{next}*)	**Next** window
- (*{previous}*)	**Previous** window
Offset	
+*<N>*, -*<N>*	**Next/Previous** offsets of pane
tokens(relative position):	
{up-of}	**Above**
{down-of}	**Below**
{left-of}	to the **Left**
{right-of}	to the **Right**
	of the *current pane*
tokens(absolute position):	
{top}	The **top**
{bottom}	The **bottom**
{left}	The **leftmost**
{right}	The **rightmost**
{top-left}	The **top-left**
{top-right}	The **top-right**
{bottom-left}	The **bottom-left**
{bottom-right}	The **bottom-right**
	pane

▭ Target window

`<session>:<window>`	**Full name**
1. As a special **token**:	
^ (*{start}*)	**Lowest-Numbered** window
$ (*{end}*)	**Highest-Numbered** window
! (*{last}*)	**Last** window
+ (*{next}*)	**Next** window
- (*{previous}*)	**Previous** window
2. A window **index**	"*session1:0*", "*session1:1*"
3. A Window **ID**(prefixed with **@**)	"*@1*", "*@2*"
4. An **exact** window name	"*session2:help*", "*session2:devs-1*"

▭ Target window (cont)

5. The **start** of a window name	"*session3:he*", "*session3:devs*"
6. As an fnmatch(3) **pattern** for matching the window name	"*session4:he**", "*session4:devs-?*"
`:list-windows (lsw)`	**List** windows
"*.:<target-window>*", "*:<target-window>*", "*<target-window>*" uses current session	

◐ Target session

1. A session **ID**(prefixed with $)	"*$0*", "*$1*", "*$2*"
2. An **exact** name	"*Work*", "*Monitor-S1*", "*Monitor-S2*"
3. The **start** of session name	"*Wo*", "*Monitor*"
4. An *fnmatch(3)* pattern for matching session name	"*Monitor**", "*Monitor-S?*"
Exact matching by prefixed with =	"*=Work*", "*=Monitor-S2*"
`:list-sessions (ls)`	**List** all sessions
`:has-session -t <target-session> (has)`	Checks if a session is **existing**

⌨ Target client

Name is normally as *pty(7)* file	"*/dev/pty0*", "*/dev/pts/1*"
`:list-clients (:lsc)`	**List** clients