⌥ GIT CHEAT SHEET created by si9ma

Basic

Shell

中文测试 (Meta) key	`M`
Shift key	`S`
Command (Super) key	`CMD`
Function (Fn) key	`fn`
Home/end	`fn-left / fn-right`

Basic

Execute command by name	`M-x`
Cancel command	`C-g`
Exit Emacs	`C-x C-c`
Help: describe a key combination	`C-h k`
Help: describe a function	`C-h f`

Navigation

Previous/next character	`C-b / C-f`
Previous/next line	`C-p / C-n`
Previous/next word	`M-left / M-right`
Previous/next current symbol	`M-a / M-e`
Previous/next s-expression	`C-M-b / C-M-f`
Previous/next defun	`C-M-a / C-M-e`
Previous/next page	`C-v / M-v`
Beginning/end of line	`C-a / C-e`
Beginning/end of file	`M-S-, / M-S-.`

Mark

Select region while navigating	Hold Shift
Select next s-expression	`M-S-Space`

Files

Find (open) file	`C-x C-f`
Save file	`C-x C-s`
Don't suggest names in minibuffer	`C-f`
Find recent file	`M-m`
Find file in project	`M-i`

Search

Search forward, next match	`C-s`
Search backward, prev match	`C-r`
Occur (show all matches)	`M-o`
Grep	`M-x rgrep`
Interactive grep in project (Helm)	`M-h`
Helm-grep: preview match at point	`C-z`

Replace

Interactive replace	`M-%`
Replace current occurrence	`y`
Skip current occurrence	`n`
Replace all occurrences	`!`

Buffers

Previous/next buffer	`C-, / C-.`
Jump to buffer	`C-x b / M-m`
List buffers	`C-x C-l`
Kill buffer	`C-x k`

Editing

Kill current symbol	`C-d`
Kill line	`C-k`
Cut to clipboard	`C-w / CMD-x`
Copy to clipboard	`M-w / CMD-c`
Paste from clipboard	`C-y / CMD-v`
Cycle through clipboard	`M-y`
Undo	`C-z / CMD-z`
Redo	`M-z / CMD-y`
Indent line	`TAB`
Complete filename in buffer	`M-/`
Comment or uncomment region	`M-;`

Windows

Leave one window	`C-x 1`
Split window horizontally	`C-x 2`
Split window vertically	`C-x 3`
Jump cursor to other window	`C-x o`

Miscellaneous

Ediff	`M-x ediff`
Ediff two opened buffers	`C-c d`
Multiple cursors: mark all like this	`C-x m`

Lisp

SLIME

Connect	`M-x slime-connect`
Switch to next connection	`M-x slime-selector n`
Jump to REPL buffer	`C-c C-z`
Jump back from definition	`M-,`
Compile current defun	`C-c C-c`
Compile current pattern	`C-c C-k`
REPL: previous/next history item	`M-p / M-n`

Paredit

Wrap s-expression	`M-(`
Splice s-expression	`M-s`
Raise s-expression	`M-r`
Kill s-expression	`C-M-k`