Doom Emacs

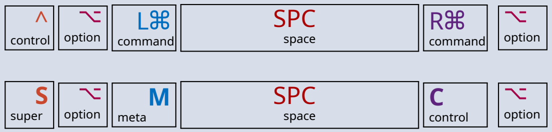
Unofficial Reference Sheet October 25, 2023

A. Setup

A.1. Key Assignments: MacOS

The following code was added to `config.el` and intended for MacOS using a Swedish keyboard:

```
(setq mac-option-modifier nil
mac-command-modifier 'meta
mac-right-command-modifier 'control
mac-control-modifier 'super
mac-function-modifier 'hyper)
```



SPC	Space
:	enter command mode
L⌘	left command = Meta = **M**
R⌘	right command = Control = **C**
⌥	left and right option keys
⇧	shift keys
⏎	enter
^	control = Super = **S**

B. Help

SPC h c	given keychord → get command
SPC h f	describe functions
SPC h k	given keychord → get command docs
SPC h w	given command → get keychord
SPC h v	variable full docs
SPC h V	custom variable full docs

C. Configuration Files

C.1. Configuration Files Location

Your configuration files: config.el , init.el , packages.el are located in the directory: *~/.config/doom/*

◎ SPC f P	browse private config files
◎ SPC f p	find file in private config files

C.2. Changing Your Configuration Files

After making changes to any of your configuration file,

1. :! *doom sync*	run doom sync
2. SPC h r r	reload doom emacs

C.3. Reloading Configuration Files

SPC h r r	reload doom
SPC h r f	reload fonts
SPC h r t	reload theme
SPC h r p	reload packages
SPC h r e	reload environment

C.4. Add Doom to your Path

Add the following to your .zshrc or .zprofile in your home direcotry: `~/`

```
# Add doom emacs to the Path
export PATH=$HOME/.config/emacs/bin:$PATH
```

After editing your .zshrc or .zprofile file, you need to reload it (source it) from your home directory in the terminal.

```
$ cd SPC ENTER
$ source .zshrc
```

D. Commands

D.1. Run Emacs Commands

M–x	run emacs commands -> the meta x
SPC :	run emacs commands -> the meta x

D.2. Run Commands

:! *cmd-name [o-opt] <input>*	run shell command
: *yourVimCommand*	Vim command-line mode

D.3. Command Chords

OPERATOR COUNT MOTION

Example:

◎ d 4 j	delete 4 lines down

E. Quit, Restart & Save

E.1. Quit

SPC q q	quit emacs
SPC q Q	quit emacs without saving
:wq	write and quit
:x	write and quit
:q	quit -> throws error if unsaved changes
:q!	discard unsaved changes and quit
:wq	write (save) and quit
Z Z	write and quit
Z Q	discard unsaved changes and quit

E.2. Restart

SPC q r	restart emacs and restore
SPC q R	restart emacs

E.3. Save

SPC f s	save buffer
SPC b s	save buffer
SPC b S	save all open buffers
:w	write (save)

F. Cursor Motions

Prefix cursor commands with a COUNT number, #, to repeat the command #-times.

F.1. Basic Motions

h or ◀	move cursor left 1 character
l or ▶	move cursor right 1 character
j or ▼	move cursor down 1 character
k or ▲	move cursor up 1 character

F.2. Word Motions

b	jump backwards start word
B	jump backwards (punctuated words) start word
e	jump forwards end word
g e	jump backwards end previous word
g E	jump backwards (punctuated words) previous word
E	jump forwards (punctuated words) end word
w	jump forwards start next word
W	jump forwards (punctuated words) start next word
⇧_	jump to first non-blank character of line

F.3. Buffer Motions

G	jump to start of buffer
g g	jump to end of buffer

F.4. Find Motions

f ?	jump to next occurance of character ?
t ?	jump to before next occurance of character ?
F ?	jump to previous occurance of character ?
T ?	jump to after previous occurance of character ?
;	repeat f, t, F, T movement forwards
,	repeat f, t, F, T movement backwards
⇧5	jump to matching delimiter in pair (), {}, []

F.5. Paragraph Motions

⇧⌥8	jump to previous paragraph, function or block
⇧⌥9	jump to next paragraph, function or block

F.6. Screen Motions

H	move top of screen
L	move bottom of screen
M	move middle of screen
C–b	move screen up one page
C–e	move screen down one line
C–f	move screen down one page
C–d	move cursor & screen down 1/2 page
C–u	move cursor & screen up 1/2 page
C–y	move screen up one line
z z	center cursor on screen
z t	position cursor at top of screen
z b	position cursor at bottom of screen

F.7. Function Motions

g d	move to local declaration
g D	move to global declaration

G. Insert Mode

G.1. Exit

ESC	exit insert mode
C–g	exit insert mode

G.2. Cursors

a	append after the cursor
i	insert before the cursor

G.3. Macros

q	stop recording a macro
q a	record macro
@a	run macro a
@@	rerun last macro

G.4. Lines

o append new line below current line
A insert at end of line
O append new line above current line
I insert at beginning of line
C–t indent line
C–d de-indent line
C–j add line break current position

H. Buffers

H.1. Viewing Buffers

SPC b b	view workspace buffers
SPC ,	view workspace buffers
SPC b B	view all buffers
SPC b i	view all ibuffer
SPC b I	view workspace ibuffer

H.2. Navigating Buffers

SPC b n	next buffer
SPC b l	previous buffer

H.3. Creating and Deleting Buffers

SPC b c	clone current buffer
SPC b N	new empty buffer
SPC b d	kill (delete) buffer
SPC b k	kill (delete) buffer
SPC b K	kill (delete) ALL buffers
SPC b O	kill (delete) other buffers

I. Search & Replace

I.1. Search and Replace

	select current word forwards
#	select current word backwards
/pattern	search forwards for pattern
?pattern	search backwards for pattern
:noh	remove highlighting of search
n	search next same direction
N	search next opposite direction
:%s/old/new/g	replace all old with new
:%s/old/new/gc	replace all old with new confirm

J. Windows

J.1. Creating Windows

SPC w n	new window
SPC w s	split window
SPC w S	split window and navigate to window

J.2. Navigating Windows

SPC w j	navigate to window below
SPC w h	navigate to window left
SPC w k	navigate to window above
SPC w l	navigate to window right

J.3. Moving Windows

SPC w J	move window down
SPC w H	move window left
SPC w K	move window up
SPC w L	move window right

J.4. Change Window Size

SPC w +	increase window height
SPC w -	decrease window height
SPC w <	increase window width
SPC w <	decrease window width
SPC w =	balance windows

K. Normal Mode

K.1. Repeat, Undo and Replace

.	repeat last command
r	replace single character
R	replace characters until ESC
u	undo
C–r	redo

K.2. Copy and Paste

p	put (paste) clipboard after cursor
y	yank (copy)
P	paste before cursor
Y	yank to end of line
y w	yank the characters from cursor to next word
y y	yank line
x p	transpose two characters
y a w	yank word and space before or after
y i w	yank word

K.3. Delete

x	delete character
D	delete to end of line
d d	delete line
d w	delete word from cursor
d i w	delete word
:n,md	delete lines starting from n to m
:.,⌥4 d	delete from current line to end file
:.,1 d	delete from current line to start file
:n,1 d	delete from line n to start file

K.4. Change

C	change to end of line
c c	change line
c w	change to end of word
c i w	change word

K.5. Transpose Characters and Line Joins

J	join line below with line with space
g J	join line below with line with no space
x p	transpose two characters

K.6. Evil Multi-Edit

D	clear the region
C	clear to end of region and go into insert mode
A	go into insert mode at end of region
I	go into insert mode at start of region
R	highlight all matches of selection in buffer
P	replace highlighted region with the clipboard
V	select the region
$	go to end of region
0	go to start of region
gg	go to first region
G	go to last region
⏎	toggle selected region under cursor
M–d	match next selected region
M–D	match previous selected region
C–n	move to next selected region
C–N	move to previous selected region
M–**C**–D	restore last groupe multi-edit regions

K.7. Text Indentation

> >	move indent right
< <	move indent left (de-indent)
> %	indent () or {} block (cursor on bracket)
< %	de-indent () or {} block (cursor on bracket)
> i b	indent inner block
> i B	re-indent inner block
> a t	indent block with < > tags
= %	re-indent () or {} block (cursor on bracket)
g g = G	re-indent entire buffer

K.8. Text Folding

z a	toggle folds
z o	open folds
z c	close fold
z C	outline hide subtree (org)
z m	hide next fold level
z M	close all folds
z r	fold less
z x	update folds
z v	open fold this line
z i	toggle inline images
z A	toggle folds recursively
z C	close folds recursively
z M	close all
z o	open fold
z O	open fold recursively
z r	open next fold
z R	open all folds

L. Visual Mode

L.1. Visual Mode

Entering Visual Mode

v	start visual mode, mark region
V	start visual-line mode
Cv	start visual-block mode
A	insert mode at end of region
I	insert mode at start of region
o	toggle cursor to other end of marked region
ESC	exit visual mode

L.2. Mark Regions

a w	mark a word
a b	mark a () block
a (	mark a () block
a B	mark a {} block
a {	mark a {} block
a t	mark a <> tags block
i b	mark the inner of () block
i B	mark the inner of {} block
i p	mark the inner of paragraph
i t	mark the inner of <> tags block

L.3. Surround Mode

S	surround a region of text
S <tagName ⏎	surround region with given tag name
S (⏎	surround region ()

L.4. Actions on Regions

d	delete marked text
y	yank (copy) marked text
u	change to lowercase
U	change to uppercase
⌥..	switch case (key beside return key)

M. Python

M.1. Running Python

M–x run-python	start a python REPL
C–c **M**–o	clear python REPL

N. Customizations

N.1. Themes

SPC h t	choose (consult) theme
SPC h r t	reload theme

O. Dired - File Management

O.1. Starting Dired

SPC f d	open directory in dired (prompts you)
SPC o -	open current directory in dired
SPC p .	open project root in dired
q	kill all dired buffers

O.2. Directory Navigation

⏎	Visit the file/directory new buffer
-	go up one directory
(	toggle file info detail
)	dired-git-info-mode
=	diff files
a	visit file/directory same buffer
g r	refresh in the dired directory
g o	visit the file in the other directory

O.3. Marking Files & Directories

d	mark file(s) for deletion
m	mark files
q	kill all dired buffers
u	unmark a file/directory
U	unmark all marked file/directory
% m	mark files by regexp
!	run shell command on marked files

O.4. Creating & Delete

SPC .	create new file
SPC f f	create new file
+	create directory
D	delete the file at point
x	delete marked files
Z	zip (compress) a file/directory

O.5. Copy & Rename

C	copy the file at point
q	kill all dired buffers
R	rename the file at point
=	diff files

O.6. Bulk Renaming files/directories

Steps renaming files/directories in bulk

Step 1: **C**–c, **C**–e	open writable grep buffer
Step 2:	rename files & directories in the buffer.
Step 3: **C**–c, **C**–c	commit changes
C–c, **C**–k	abort changes at any time

P. Org-Mode

P.1. Headings

⇧–⏎	create headline of same type below
⇧–**C**–⏎	create headline of same type above
M–h	move heading left (promote)
M–l	move heading right (demote)
M–k	move heading up
M–j	move heading down
⇧–**M**–h	move heading and children left
⇧–**M**–l	move heading and children right
⇧–**M**–k	move heading and children up
⇧–**M**–j	move heading and children down

P.2. Links

SPC m l c	clipboard link
SPC m l l	create link
SPC m l d	delete link
SPC m l s	store link
SPC m l S	insert last stored link
SPC m l t	toggle link display

P.3. Babel Source Code Blocks

src **TAB**	create a babel source block
C–c **C**–c	run babel block
SPC m k	remove babel output
SPC m K	remove all babel output

```
#+begin_src web
;; your html code goes here
#+end_src
```

P.4. Org-Agenda

SPC m d	agenda deadline
SPC m s	agenda schedule

P.5. Org-Agenda Clock

SPC m c c	clock cancel
SPC m c g	goto clock
SPC m c i	clock in
SPC m c o	clock out
SPC m c r	clock report mode
SPC m c s	show clocking issues

P.6. Calendar Minibuffer

C–h	backward 1 day
C–l	forward 1 day
C–k	backward 1 week
C–j	forward 1 week
⇧–**C**–h	backward 1 month
⇧–**C**–l	forward 1 month
⇧–**C**–k	backward 1 year
⇧–**C**–j	forward 1 year

P.7. Org Tree/Subtree

SPC m s b	tree to indirect buffer
SPC m s c	clone subtree with time shift
SPC m s d	cut subtree
SPC m s h	promote subtree
SPC m s j	move subtree down
SPC m s k	move subtree up
SPC m s l	demote subtree
SPC m s n	narrow to subtree
SPC m s N	widen
SPC m s r	org refile
SPC m s s	sparse tree
SPC m s S	org sort
SPC m s A	archive subtree