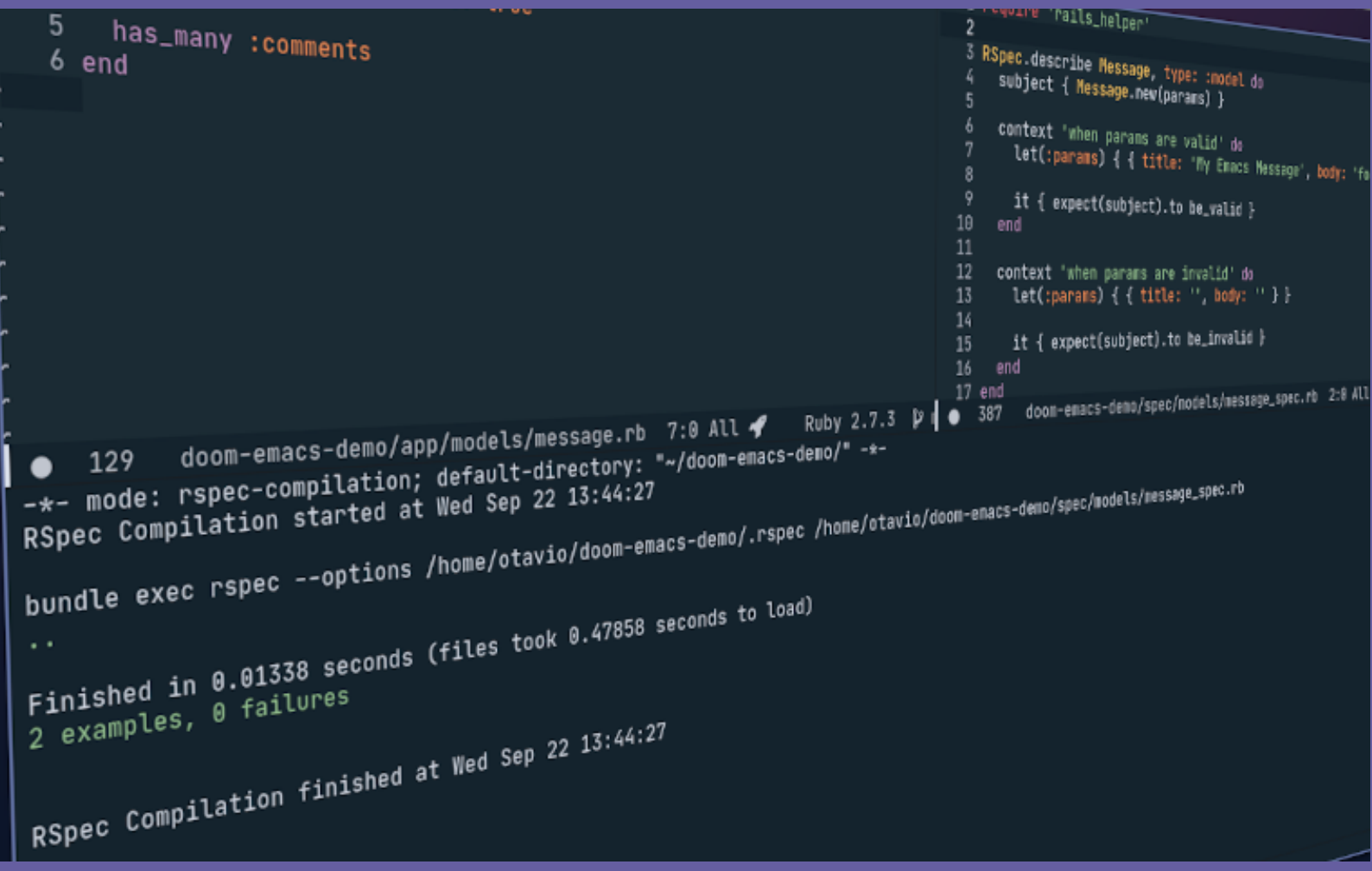
DOOM EMACS HANDBOOK

For beginners



BEST COMMANDS FOR DAILY USAGE - ROR DEVELOPMENT

Config: https://github.com/otavioschwanck/doom-emacs-on-rails

Learn vim before start

First, You have to learn the basics of Vim:

$ sudo apt install vim

$ vimtutor

UNDERSTANDING COMMANDS:

C = CONTROL. EXAMPLE: C-O = CTRL+O

M = ALT. EXAMPLE: M-O = ALT + O

S = SHIFT.

FOR SOME EXERCISES, PRESS M-X AND SEARCH FOR MR-MIYAGI

FILE NAVIGATION

Generic Buffer (file) change

SPC ,	Change Buffer
C-,	Previous Buffer
C-;	Next Buffer
SPC SPC	Find Files in project
SPC .	Find File (cur dir)

Rails Navigation

SPC r m	Find Model
SPC r a	Find Locales
SPC r z	Find Serializer
SPC r v	Find View
SPC r V	Find View (current resource)
SPC r c	Find Controller
SPC r s	Find Service
SPC a	Go to test*
SPC A	Go to test and vsplit*

*upper case is always for current resource

Go-to

gd	Go to definition
gD	Go to usages (reference)
gf	Go to file
ga	Go to route Ex: (foo_path)

SEARCH / JUMP ON FILE

On Current Line

f	go to character (1)
S	go to character (2)
0	go to first character of line
-	go to end of line
w and **W**	jump 1 word (start) (W jumps higher)
e and **E**	jump 1 word (end) (E jumps higher)
b and **B**	jump -1 word (start) (B jumps higher)
t	go to before character(1)

*you can keep pressing f or t or s to repeat the search

*f, s and t can be used backwards using uppercase

On Current File

/	Search Forward
?	Search Backward
n and N	Go forward and backward (search)
SPC s s	Advanced search on current file
SPC s i	Search symbols / defs on current file
#	Search word at cursor at current file
M-s	Search and jump using two characters
gg	First line of file
G	Last line of file
20G	Go to line 20
]g and **[g**	Next / Previous git change
]] and **[[**	Next / Previous method definition
]) and **[(**	Next / Previous parenthesis
]} and **[{**	Next / Previous brackets
C-j and **C-k**	Next / Previous section

ENTERING ON INSERT MODE

On Current Line

i	Enter on insert mode
a	Enter on insert mode after point
I	Enter on insert mode at beg of line
A	Enter on insert mode at end of line
V then A	Add text at end of line at all selected
V then I	Add text at beg of line at all selected

SEARCH GLOBALLY

On Project

SPC s p	Search text on Project
SPC s d	Search text on current dir
SPC *	Search item at cursor on project
M-x find-dired	Grep search files on project

Other

SPC f P	Private doom emacs files
C-o	Return to last jump
C-i	Go forward on jump
SPC f r	Find Recent Files

FILE EDITING

Current File

SPC f D	Delete current file
SPC f Y	Copy current file path
SPC f R	Rename / Move current File
\ or ç or SPC f s	Save file

COPY DELETE AND PASTE

Intro

To use one of the operators, call the **operator** + the **motion.** **Example:** **Delete a word = dw**

The operators

d	Delete (Cut)
c	Delete (Cut) and enter on insert mode
y	Copy
p and P	Paste before | Past e after cursor
C-p	*after pasting* Navigate on copy history
M-y	Search on copy (yank) history (and paste)

Selecting

To start selecting something, press **v.** To line select, press S-**V.** To block select, press **C-v.**

After you done selecting what you want, you can:

d or **y** or **c**	Execute the operator action
S	Add delimitator to selection. Example: S) = (selection) S(= (selection) S] = [selection] S[= [selection] S" = "selection" St<emacs> = <emacs>selection</emacs>

FOR HOW TO USE SNIPPETS, CHECK THE GIF ON README

COOL STUFF WITH OPERATORS

Before read:

`lets call the operators` **`(d, y or c): OP`**

Hidden goodies

OP OP	Do the operator on current line. Ex: dd
4 OP OP	Do the operator on 4 lines. Ex: 4dd
ds	Delete the delimiter. Ex: ds" Delete ""
OP io	Operator on thing at point
OP ij	Operator on current identation
OP i)	Operator inside ()
OP i"	Operator inside ""
OP i]	Operator inside []
OP it	Operator inside tag
OP ii	Operator inside identation
UPPERCASE OP	Operator from point to end of line
DT OP	Operator from point to before char
DF OP	Operator from point to char
OP ia	Operator in argument
M-c	Toggle case (snake_case, camel, etc)

*every command with **i** can be used with **a** (but the action includes the delimiters

Text: my_var = (i love doom emacs)
di) => my_var = ()
dai => my_var =

RUNNING STUFF

Basic Stuff

SPC v	Toggle Quick Terminal
SPC o T	Open a new terminal
SPC e	Open tree (Prefer **SPC .** over tree)
SPC m k k	Run a rake task
SPC m b i	Run bundle install
SPC =	Run rubocop on current file
SPC -	Indent current file
SPC m P	Run rubocop on project
SPC r r	Rails Console
SPC r R	Rails Server
C-c t	Google Translator

Tests

SPC t v	Run tests of current file
SPC t a	Run all tests
SPC t r	Rerun last test
SPC t l	Run only failures of last test
SPC t s	Run tests on cursor

AWESOME TIPS

Multiple Cursors

M-d	Start multiple cursors (keep pressing)
M-S-d	Find item of multiple cursor on top
C	Edit in all cursors

Search and Replace in entire project

SPC s p	Search On Entire Project
C-c C-e	Edit search result

After this, you can edit all search results like a single file. To confirm, press **C-c C-C.** To cancel, **C-c C-k**

***You can use multiple cursors or :%s/old_text/new_text/gr to change**

Rename multiple files on project

M-x find dired	Search with grep the files. To match all **foo rb** files by example, run with with -name "foo*.rb".
C-c C-e	Edit search resuls

After this, you can edit all search results like a single file. To confirm, press **C-c C-C.** To cancel, **C-c C-k**

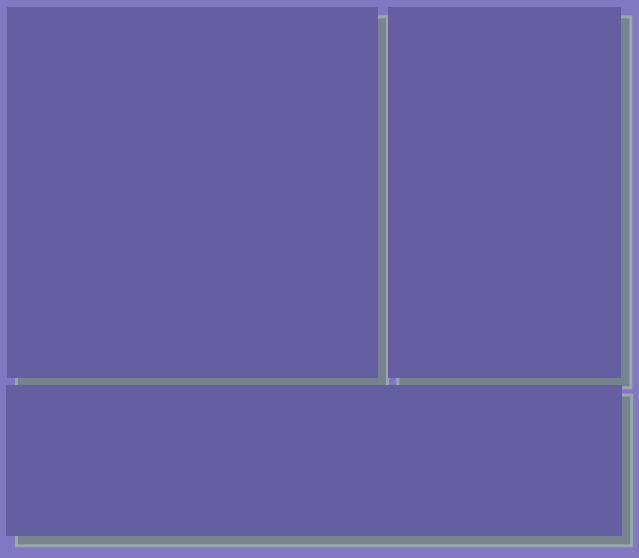
WINDOW MANAGEMENT

Window Actions

C-w v	Divide verically
C-w s	Divide horizontally
C-w C-o	Maximize Window
C-w C-u	Undo Windows Alteration
C-w C-r	Redo Windows Alteration
C-w =	Balance Windows
C-w T	Detach Window
C-w q	Close Window

Windows Moviment

M-o	Go to next window
M-h	Go to window on left
M-j	Go to window on bottom
M-k	Go to window on top
M-l	Go to window on right



RUBY REFACTORING

Refactoring like a pro

SPC m i	Toggle if unless (single / multi line)
SPC m m	Move selected text to a new method
SPC m n	Create a method from text in cursor
SPC m [	Toggle do end => { } and vice-versa
SMC m v	Move selected text to a new variable
SPC m V	Move selected text to a constant
C-c s	Add http code at point (humanized)

Using Snippets + Auto Complete

C-n or C-j	Next item on autocomplete
C-p or C-k	Previous item on autocomplete
RET	Select item on autocomplete
M-RET	Newline while autocomplete open
TAB	Start snippet / go forward on snippet Example: def<TAB> = def my_method(args) end
S-TAB	Go back on snippet selection
C-d	Delete selected text (while on snippet)
C-RET	Complete text from all buffers

Using Emmet (create methods and tags)

M-e Toggle emmet

```
On ruby: init@name;call@values<C-e> =
def initialize(name)
  @name = name
end

def call(values)
  | cursor here
end

On web-mode: .name>li.item*3<C-e>=
<div class="name">
  <li class="item"></li>
 <li class="item"></li>
 <li class="item"></li>
</name>
```

RUBY PLUGINS

RAILS ROUTES

C-c o	Add route at point (using cache)
C-c C-o	Add route at point (refreshing cache)
ga	Go to route

RAILS i18n

C-c i	Add i18n at point (using cache)
C-c C-i	Add i18n at point (refreshing cache)_

Ruby JSON to Hash

SPC m J	Convert JSON at point into hash
SPC m j	Send key of the converted hash to a new let

GIT STUFF

On a file

SPC g r	Revert modification at point (can be used to see the diff)
SPC g t	Time Machine mode (Use C-n and C-p to navigate)
]g	Go to next git hunk
[g	Go to previous git hunk
SPC g R	Revert all modifications on file

On Magit Status

SPC gg	Open Magit Status
f	Fetch
F	Pull
P	Push
cc	Commit
ca	Ammend
Z	Stash
?	See all available options
m	Merge
r	Rebase

It's Magit!

FILE MANAGER (DIRED)

Dired is an awesome file manager integrated on Emacs. To open in some folder, just press **SPC .** And select a folder instead a file. You can also show the current file on dired using **SPC o -**

Operations in DIRED

d	Mark a file to deletion
x	Delete the marked files
m	Mark a file to do some action
u	Unmark a file
U	Unmark all files
C	Copy the file (or files if has marks)
R	Move/Rename file (or files if has marks)
-	Go to parent directory
=	Diff this file with another file
g?	See all commands above
SPC o -	Visit current file on dired.

Super cool tip (please read)

While Renaming / Copying you can press **M-n** to put the current file name on the text box (works in every text box).
While on text box:

M-b	Go previous word
M-f	Go next word
M-d	Delete next word
C-w	Delete previous word

WORKSPACES

Everytime you open a project with SPC p p you create a workspace. To change between workspaces press M-1 to M-9. The prefix for workspace management is SPC TAB.

Workspace Commands

SPC TAB n	New Workspace
SPC TAB N	New Named Workspace
SPC TAB r	Rename Workspace
SPC TAB d	Delete Workspace
SPC TAB .	Change Workspace
SPC TAB s	Save Workspace
SPC TAB l	Load Workspace
M-1 to M-9	Change Workspace

CREATING YOUR OWN SNIPPETS

To create your own snippets, just type M-x yas-new-snippet on the file type you want.

name = Name that will show on autocomplete
key = key pressed to toggle snippet
After the comments, just put the commands that you want to create snippet.

Syntax:

```
def ${1:my_method_name}(${2:my_method_args})
  ${0:}
end
```

What will happen:
After pressing the snippet, you 1: Can change the method name, 2: change the args, and after that, it will put the cursor inside the method.

To persist your snippet, press C-c C-c. You can edit your snippet by searching for it with

SPC f P

IMPORTANT: The file name need to be the same as the key you put.

HARPOON

Harpoon is a plugin to create bookmarks per project / branch. It is awesome to navigate between only the files that you are really working.

Commands

SPC 1 to SPC 9	Navigate between files
C-s	Save file to harpoon
C-SPC	Select one file from harpoon
SPC j f	Edit harpoon file
SPC j c	Clean harpoon file

EPIC TERMINAL MANAGEMENT

This config has a lot of helpers to work with terminal. You can configure single time commands and multiple commands term layouts on **SPC f m** (~/.doom.d/user-settings.el)

Single Time Commands

To configure your single times commands, go to your configs (SPC f m) and search for +add-command-to-term-list. Example:

(+add-command-to-term-list '("docker-compose up; read; exit" . "Docker Compose"))

Command to be run — Command Name

SPC o t	Execute an single time command
SPC ESC	Switch to a terminal
SPC v	Open a quick terminal (empty)

Terminal templates

To configure your single times commands, go to your configs (SPC f m) and search for +add-layout-to-term-list. Example:

(+add-layout-to-term-list '("Rails" . '("rails console" "rails server" nil)))

Layout name — Term 1 — Term 2 — Empty term

SPC T	Create terminals from layout
M-1 to M-9	Switch between workspaces

Terminal quick tips

C-c	(On normal mode) send C-c C-c and create enter in insert-mode
M-n and M-p	Navigate between command history