

# Emacs workshop 2022: Day 1

Maikol Solís

2022-01-17

About me

About (GNU) Emacs

About the configuration

Doom-emacs

Basic operations

Let's practice

About me

About (GNU) Emacs

About the configuration

Doom-emacs

Basic operations

Let's practice

- ▶ Maikol Solís, Ph.D
- ▶ Professor Escuela de Matemática, UCR
- ▶ Researcher at CIMPA



# My own journey

**Needs:** L<sup>A</sup>T<sub>E</sub>X, R, email, organization tools, tracking papers, writing.

- ▶ [2002-2015] Pen and paper (still), Word/OpenOffice, TeXstudio/Texmaker, Lyx, gmail/google calendar (still), apps for tasks, Sublime editor, Rstudio (still), and so on.
- ▶ [2016-2017] I started using spacemacs, but I couldn't understand it.
- ▶ [2017] I tried using vim/neovim, but still I was lost.
- ▶ [2018-2020] I discovered VSCode (and like it!). But the ecosystem was too restrictive.
- ▶ [2020-∞] In 2020, I learned about [org-mode](#) and [org-roam](#). I restart using Emacs with Doom-emacs.

Little by little I have all my workflow in Emacs now.

About me

About (GNU) Emacs

About the configuration

Doom-emacs

Basic operations

Let's practice

# What is Emacs?

- ▶ Free/Libre text editor.
- ▶ Extensible and customizable ( $\rightarrow \infty$ ).
- ▶ A platform for text-related applications.

# Emacs History

- 1976 Richard M. Stallman wrote the first Emacs (“Editor MACroS”).
- 1984 RMS started with the GNU Emacs (free opensource) in Lisp and C.
- 1985 Release of first Emacs version (13.0) only supporting console.
- 2001 GNU Emacs has graphical interfaces (an other modern features).
- 2003 Carten Dominik creates [org-mode](#) as an independent package.
- 2006 [org-mode](#) is included by default in Emacs.
- Present New features like [org-babel](#), programming languages, JSON, native-compilation, SQLite, PGTK, etc.

## Emacs features

- ▶ Content aware editing mode.
- ▶ Uses Emacs Lisp dialect (functional programming language).
- ▶ Highly customizable.
- ▶ Has +10,000 built-in commands and macros to create new-ones.
- ▶ Full Unicode support.
- ▶ Real-time display editor.
- ▶ Multiple functions: Project planner, Calendar, Debugger, Mail reader, a more...
- ▶ Exports `org` files to html, pdf, odt, ...

# Emacs philosophy

- ▶ The users have all the control in the editor.
- ▶ They can add, remove or modify any feature (no limits).
- ▶ Emacs doesn't care for "looking" modern or have a check-box for any option.
- ▶ The user decides how to behave or look the editor.

The disadvantage: *The users NEED TO configure the editor.*

The big advantage: *The users CAN configure the editor.*

It's a whole journey, so relax.



About me

About (GNU) Emacs

About the configuration

Doom-emacs

Basic operations

Let's practice

# Let's draw a house

Imagine someone asks to draw a house like this



You can do it in different ways...

# Vanilla Emacs

- ▶ Nothing is configured yet.
- ▶ Steep learning curve.
- ▶ Default and cleanest configuration of Emacs.
- ▶ Definitely not for beginners.



## Doom-emacs

- ▶ With some options preconfigured, but everything can be changed
- ▶ Some learning curve.
- ▶ Thinner, lighter, opinionated.
- ▶ Could be difficult to beginners.



# Spacemacs

- ▶ Almost all is preconfigured and behind abstract layers.  
However, configuration is possible.
- ▶ Easy learning curve.
- ▶ Fast, Mnemonic, consistent.
- ▶ Beginner friendly.



About me

About (GNU) Emacs

About the configuration

Doom-emacs

Basic operations

Let's practice

## Why choosing doom-emacs?

- ▶ Doom-emacs is near to vanilla, but with useful preconfigured packages.
- ▶ As user you will need the freedom to configure Emacs.
- ▶ I have found most of the internet help is written for vanilla Emacs or doom.
- ▶ Doom-emacs is a configuration framework rather than started kit.

# Requirements for the workshop

- ▶ Emacs 27.2
- ▶ Git 2.23+
- ▶ ripgrep 11.0+
- ▶ GNU Find
- ▶ fd 7.3.0+ (fd-find)

Check how to install those packages in your operative system.

# Doom-emacs install

In console

```
git clone https://github.com/hlissner/doom-emacs ~/.emacs.d  
~/.emacs.d/bin/doom install
```

After

```
doom doctor
```

## Doom-emacs main commands

- ▶ `doom sync`: Sync the doom configuration.
- ▶ `doom upgrade`: Update doom and all the packages.

# Doom-emacs files

All the files are in `~/.doom.d/`

- ▶ `config.el`: Custom configurations.
  - ▶ `(use-package! a-package body)`  
Execute body for `a-package`.
  - ▶ `(after! a-package body)`  
Execute body after load `a-package`.
  - ▶ `(setq variable value)`  
Set a variable to a value.
- ▶ `init.el`: Activate/Deactivate modules.
- ▶ `package.el` Add new packages with `(package!).`

# Where to get help?

- ▶ <https://emacsdocs.org/>
- ▶ <https://discourse.doomemacs.org/t/other-learning-resources/48>

About me

About (GNU) Emacs

About the configuration

Doom-emacs

Basic operations

Let's practice

# Some definitions

SPC Space.

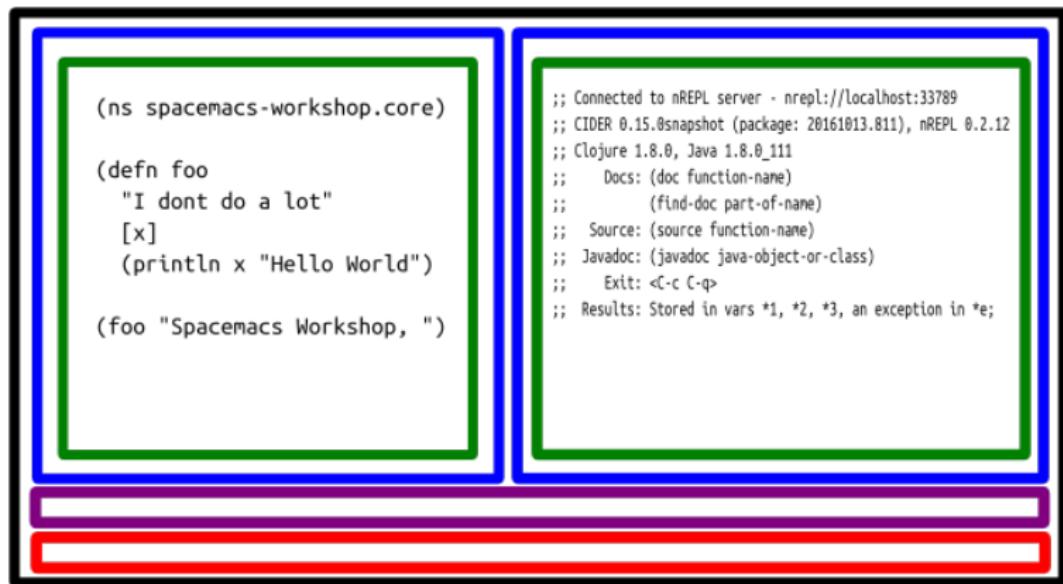
RET Enter.

C-c Press Control + c.

M-x Press Meta (Alt) + x.

C-S-a Press Control + Shift + a.

# Buffers, windows and frames



Frame

Window

Buffer

Status bar

Minibuffer

# Avoiding RSI

- ▶ Default shortcuts in vanilla Emacs starts with **Control** or **Meta**.
- ▶ In doom all basic commands starts with **SPC**.
- ▶ Doom-emacs uses a modal (and easier) system called **evil-mode**. (More later!)



## Basic shortcuts

`SPC` Display the main menu.

`SPC SPC` Open a file in project (current folder or git repository).

`SPC : M-x` execute any command.

`SPC h` The help system.

`SPC h i` Official manuals (look for *Org Mode* and *Emacs*).

`SPC h d h` Doom help.

`SPC h d m` Doom's module help.

## Open and saving

**SPC f f** Opens any file.

**SPC f s o SPC b s** Save file.

=**SPC f r** Rename file.

# Evil (Extensible VI Layer)

**EVI'L**

**Normal**

**Insert**

**Visual**

# Basic evil commands

key	movement	key	action
h	left	ESC	enters normal mode
l	right	i	enters insert mode
j	down	v	enters visual mode
k	up	x	delete char
0	start of line	x	delete char
\$	end of line	r	replace char
		u	undo
		C-r	redo

# Speaking evil

A good tutorial (for spacemacs) here: [https://practical.li/  
spacemacs/spacemacs-basics/working-with-projects/](https://practical.li/spacemacs/spacemacs-basics/working-with-projects/)

## Verbs

**c** (change), **d** (delete), **g** go, **v** visual (select), **y** yank (copy)

## Modifiers

**{ }** beginning/end of paragraph, **a** around, **f** find (includes character), **i** inside, **s** surround, **t** till (just before a character)

## Objects

**b** block/parentheses, **p** paragraph, **s** sentence, **w** word

## Other evil features

- ▶ `evil-tex`: to handle related  $\text{\LaTeX}$  objects.
- ▶ `g` and `z` menus: convenient set of utilities.

About me

About (GNU) Emacs

About the configuration

Doom-emacs

Basic operations

Let's practice

# Preparation

- ▶ Open the latex file.
- ▶ Enable the latex module.
- ▶ Do the exercises.