



Mexican Standoff

Text Editors - Advanced Features

Staff 42 pedago@staff.42.fr

Summary: This rush is to make you see the possibilities (in finite?) Of your text editors.

Contents

I	Preamble	2
II	Introduction	3
III	objectives	4
instructions IV		5
IV.1	of fi configuration files.	5
IV.2	Keyboard Shortcuts.	5
Header 42 IV.3.	6
V	bonus	8
VI	Rendering and peer-assessment	9

Chapter I

Preamble

Google

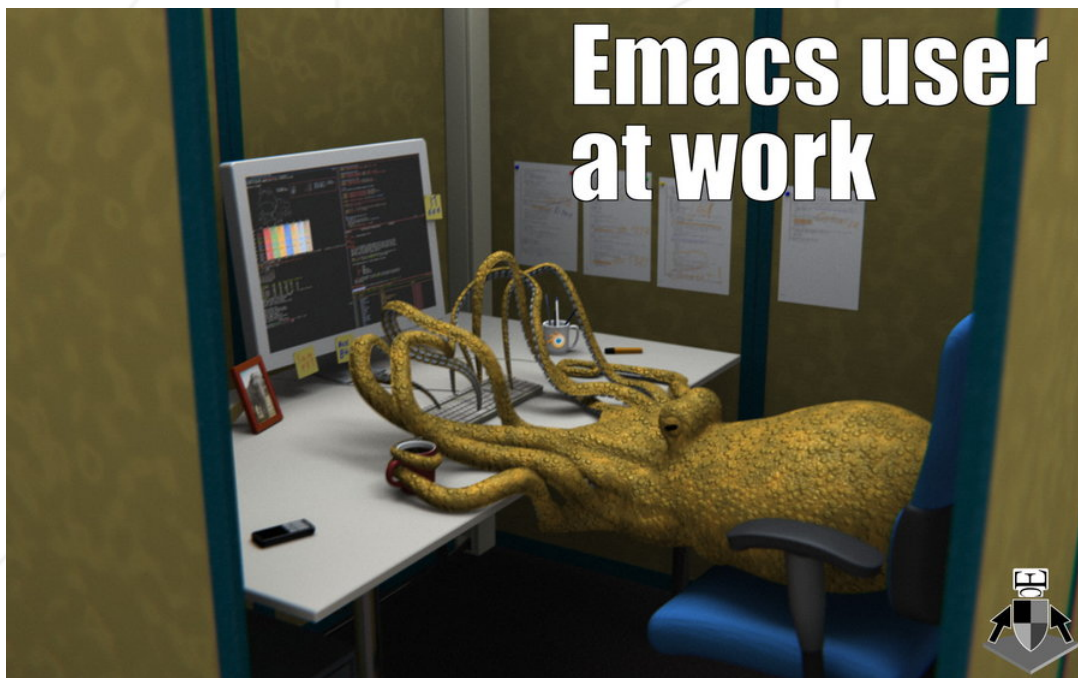
how to quit smoking

how to quit **smoking**

how to quit **facebook**

how to quit **sugar**

how to quit **vim**



([Source.](#))

Chapter II

Introduction

This rush is to make you see between-functionality (in fi nite?) Of the two mammoths of text editing: emacs and vim.

Today you will go beyond the old arguments and use both! The opportunity removal, firstly, your prejudices (barter them against certainty is better), and secondly to never need to ask how to spell it *** *** open by mistake editor brothel.

Chapter III

Objectives

The primary objective of this rush is to encourage you to take ownership of your text editor, whatever it is. Text editors are very comprehensive tools, basic, but you will see that it is possible to customize them to make them stick closer to your needs.

Di ffi cult to really understand all the possibilities of an editor as you have not tried to con fi gure, learn its shortcuts, or program your own modes!

The aim is therefore embarking on a process. It is your rush fi ned to seek the text editor that really suits you (it can be emacs it can be vim, , it may be the Evil mode emacs ... it can also be something else entirely), and make it your own.

Chapter IV

Guidelines

The next three years are to be made to emacs And for vim. No more no less.

IV.1 configuration files

First you are going to tackle the configuration of your publishers. Write fi les. emacs and. vimrc having at least the following configurations:

- A C code is automatically indented with tabs
- A pair (bracket or brace) is automatically closed when the opening element sea- sissez
- The cursor position column is a ffi chée
- Two spaces side by side are highlightés
- A whitespace before a newline is highlighté
- Backup fi les (ending with ~) are saved in a specific folder within the folder ~/ . emacs.d and ~/ . vim, respectively.



You have the right to use plug-ins.

IV.2 Keyboard Shortcuts

To get the most out of your text editor, it is advisable to go through the learning keyboard shortcuts. Use shortcuts will be laborious as muscle memory will not have taken the baton, but as soon as the automation will

instead you will gain in productivity!

You will find on the project page a file `normalize_me.c`.

You need to find the smallest combination (shortcut keys) can perform putting to this file to the standard (the norminette authentic). You have the right to use the basic configuration shortcuts, not of the binder commands that make coffee (or in the record or in defense).

You will record your sequence combinations in files `normalize_emacs` and `normalize_vim`.

IV.3 Header 42

You will now reach into the programming itself, recoding the generation of the header 42.

Header as follows:

```

/* ***** */
/*
/*      header.c
/*
/*      By: laurie <laurie@staff.42.fr>
/*
/*      Created: 2015/12/02 2:40:24 p.m. by laurie
/*      Updated: 2015/12/02 2:40:24 p.m. by laurie
/*
/* ***** */

```

You must comply with the following constraints:

- The email is recovered in the variable `MAIL` environment.
- You must generate the header on the `<Ctrl-c h>` on **emacs** and `<Ctrl-c Ctrl-h>` on **vim**.
- Press `<Ctrl-c h>` must generate a header ONLY if a correctly formatted header is not already present.
- The generation of header shall be tested only on file `C`. You do not have to manage file mode.
- The line "Updated" should only be updated if the buffer has been modified since the last backup. Save without making a modification does not update this line.
- Nothing is forbidden. (In the limit cases of cheating, of course, you need to explain your code defense.)



Start by reading an introduction to ELisp and Vimscript, you will quickly get an idea of what you can use to make the header.

Chapter V

Bonus

Bonuses will be evaluated only if all of the above questions has been validated for both text editors.

Here bonus ideas to achieve:

- Repeat the previous years with another text editor
- Customizer the header according to the mode
- **Con fi gure and use a syntax checker (type flymake for emacs)**
- Write a minor mode that you highlight your faults standard

chapter VI

Rendering and peer-assessment

- Your rendering will be corrected only by humans.
- Files `.emacs`, `.vimrc`, `normalize_emacs` and `normalize_vim` have imposed names, for others you are free to name them as you want.
- Files relating to emacs will be in an eponymous folder at the root of the rendering, even for the files relating to vim.
- You will, in defense, comment and explain your code. Likewise, you should know what the selected shortcuts without having to use them.
- You must get to the root of your rendering deposit a file named `author` containing logins followed the two members of a group '\ n', such that:

```
$> Cat -e ./auteur login1 $$$  
login2>
```