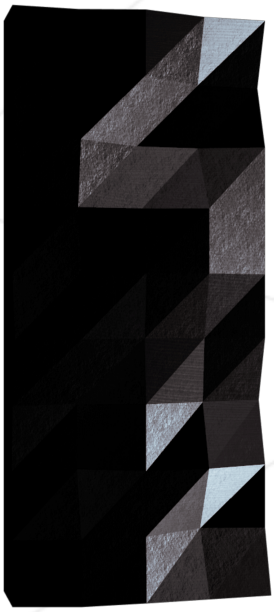


Mexican Standoff

Editeurs de texte - Fonctionnalités avancées

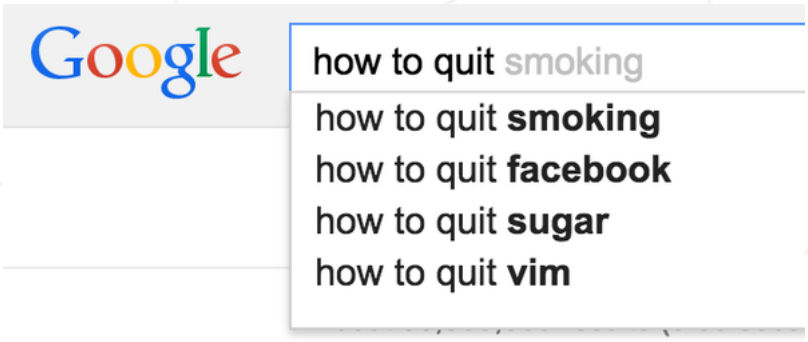
Staff 42 pedago@staff.42.fr

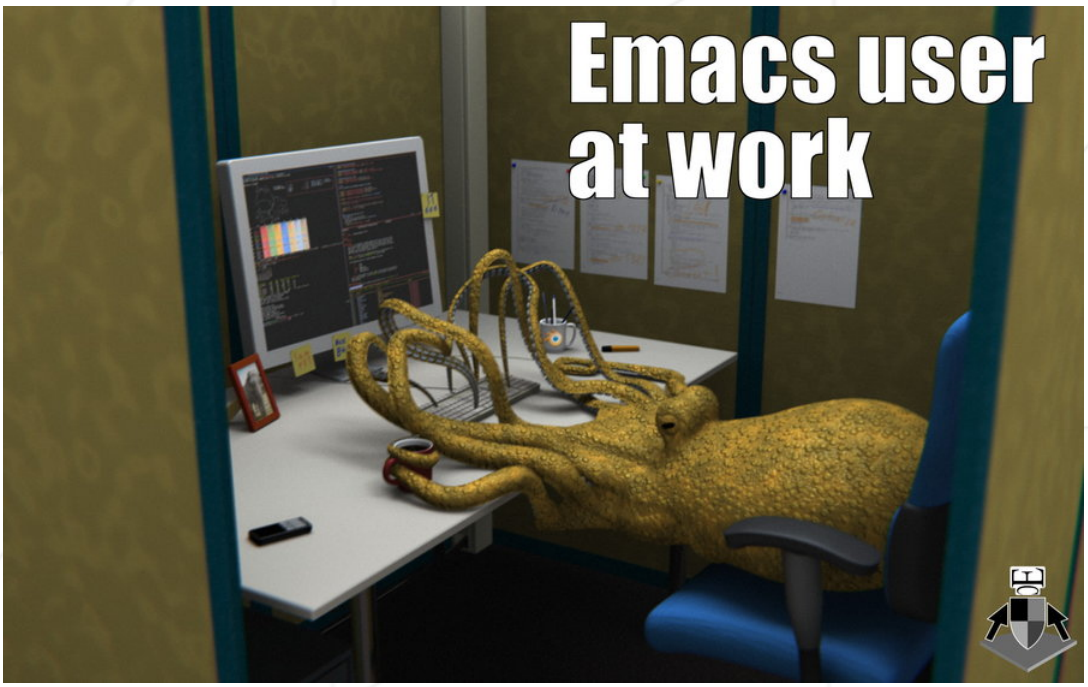
Résumé: Ce rush a pour but de vous faire entrevoir les possibilités (infinies ?) de vos éditeurs de texte.

Table des matières

Chapitre I

Préambule





(Source.)

Chapitre II

Introduction

Ce rush a pour but de vous faire entre-apercevoir les fonctionnalités (infinies ?) des deux mammoths de l'édition de texte : `emacs` et `vim`.

Aujourd'hui, vous allez dépasser les vieilles querelles et utiliser les deux ! L'occasion de vous débarasser, d'une part, de vos préjugés (troquez-les contre des certitudes, c'est mieux), et d'autre part de ne plus jamais avoir besoin de demander comment on sort de ce ****** d'éditeur ouvert par erreur, bordel.

Chapitre III

Objectifs

L'objectif premier de ce rush est de vous encourager à vous approprier votre éditeur de texte, quel qu'il soit. Les éditeurs de texte sont des outils très complets, de base, mais vous allez voir qu'il est possible de les personnaliser pour les faire coller au plus près à vos besoins.

Difficile de vraiment appréhender toutes les possibilités d'un éditeur tant que vous n'avez pas essayé de le configurer, d'apprendre ses raccourcis, ou de programmer vos propres modes !

Le but est donc de vous lancer sur une démarche. Il vous appartient, le rush fini, de chercher l'éditeur de texte qui vous convient vraiment (ça peut être `emacs`, ça peut être `vim`, , ça peut être l'Evil mode sur `emacs`... ça peut aussi être complètement autre chose), et de vous l'approprier.

Chapitre IV

Consignes

Les trois exercices suivants sont à réaliser pour `emacs` ET pour `vim`. Pas plus, pas moins.

IV.1 Fichiers de configuration

Vous allez d'abord vous attaquer à la configuration de vos éditeurs.

Ecrivez les fichiers `.emacs` et `.vimrc` possédant au moins les configurations suivantes :

- Un code en C est automatiquement indenté avec des tabulations
- Une paire (parenthèse ou accolade) est automatiquement fermée lorsque vous saisissez l'élément ouvrant
- La colonne de position du curseur est affichée
- Deux espaces côte-à-côte sont highlightés
- Un whitespace avant un retour à la ligne est highlighté
- Les fichiers de sauvegarde (se terminant par ~) sont archivés dans un dossier spécifique à l'intérieur des dossiers `~/.emacs.d` et `~/.vim`, respectivement.

`Vous avez le droit d'utiliser des pluggins.`

IV.2 Raccourcis clavier

Pour tirer le meilleur parti de votre éditeur de texte, il est conseillé de passer par l'apprentissage des raccourcis clavier. Utiliser les raccourcis sera laborieux tant que la mémoire musculaire n'aura pas pris le relai, mais dès que les automatismes seront en

place vous y gagnerez en productivité !

Vous trouverez sur la page du projet un fichier `normalize_me.c`.

Vous devez trouver la plus petite combinaison (touches et raccourcis) possible permettant de mettre ce fichier à la norme (la norminette faisant foi). Vous n'avez le droit d'utiliser que les raccourcis de la configuration de base, et non pas de binder des commandes qui font le café (ni dans le rendu, ni en soutenance).

Vous enregistrerez vos séquences de combinaisons dans les fichiers `normalize_emacs` et `normalize_vim`.

IV.3 Header 42

Vous allez maintenant mettre les mains dans la programmation à proprement dite, en recodant la génération du header 42.

Header que voici :

```
/* ************************************************************************** */
/*                                                                            */
/*                                                        :::      ::::::::   */
/*   header.c                                           :+:      :+:    :+:   */
/*                                                    +:+ +:+         +:+     */
/*   By: laurie <laurie@staff.42.fr>                +#+  +:+       +#+        */
/*                                                +#+#+#+#+#+   +#+           */
/*   Created: 2015/12/02 14:40:24 by laurie           #+#    #+#             */
/*   Updated: 2015/12/02 14:40:24 by laurie          ###   ########.fr       */
/*                                                                            */
/* ************************************************************************** */
```

Vous devez respecter les contraintes suivantes :

- L'email est récupéré dans la variable MAIL de l'environnement.
- Vous devez binder la génération du header sur la combinaison <Ctrl-c h> sur `emacs`, et <Ctrl-c Ctrl-h> sur `vim`.
- Presser <Ctrl-c h> doit générer un header UNIQUEMENT si un header correctement formatté n'est pas déjà présent.
- La génération de header ne sera testée que sur des fichiers C. Vous n'avez pas à gérer le mode du fichier.
- La ligne "Updated" ne doit être mise à jour que si le buffer a été modifié depuis la dernière sauvegarde. Sauvegarder sans avoir fait de modification ne doit pas mettre à jour cette ligne.
- Rien n'est interdit. (Dans la limite des cas de triche évidemment, il vous faudra pouvoir expliquer votre code en soutenance.)

Commencez par lire une introduction à Elisp et à Vimscript, vous vous ferez rapidement une idée de ce que vous pouvez utiliser pour réaliser le header.

Chapitre V

Bonus

Les bonus ne seront évalués que si l'intégralité des questions précédentes a été validée, pour les deux éditeurs de texte.

Voici des idées de bonus à réaliser :

- Refaire les exercices précédents avec un autre éditeur de texte
- Customizer le header selon le mode
- Configurer et utiliser un syntax checker (de type `flymake` pour `emacs`)
- Ecrire un minor mode qui vous highlight vos fautes de norme

Chapitre VI

Rendu et peer-évaluation

- Votre rendu ne sera corrigé que par des humains.
- Les fichiers `.emacs`, `.vimrc`, `normalize_emacs` et `normalize_vim` ont des noms imposés, pour les autres vous êtes libres de les nommer comme vous voulez.
- Les fichiers relatifs à `emacs` seront dans un dossier éponyme à la racine du rendu, de même pour les fichiers relatifs à `vim`.
- Vous devrez, en soutenance, commenter et expliquer votre code. De même, vous devez savoir ce que font les raccourcis choisis sans avoir besoin de les utiliser.
- Vous devez rendre à la racine de votre dépôt de rendu un fichier nommé `auteur` contenant les logins des deux membres du groupe suivis d'un '\n', tel que :

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