Emacs Configuration

Maxfield Lehman

March 30, 2017

Contents

1 General emacs setup

Neither scrollbars nor a tool/menu bar are relevant when you have a keyboard.

```
(tool-bar-mode -1)
(menu-bar-mode -1)
(scroll-bar-mode -1)
(global-hl-line-mode)
(set-default-font "Operator Mono 12")
(setq ring-bell-function 'ignore)
```

2 Set personal information

```
(setq user-full-name "Maxfield Lehman")
```

3 Utility function definitions

```
(defun kill-matching-lines (regexp &optional rstart rend interactive)
"Kill lines containing matches for REGEXP.

See ‘flush-lines’ or ‘keep-lines’ for behavior of this command.

If the buffer is read-only, Emacs will beep and refrain from deleting
the line, but put the line in the kill ring anyway.  This means that
you can use this command to copy text from a read-only buffer.
\(If the variable ‘kill-read-only-ok’ is non-nil, then this won't
even beep.)"
(interactive
(keep-lines-read-args "Kill lines containing match for regexp"))
(let ((buffer-file-name nil)) ;; HACK for ‘clone-buffer’
    (with-current-buffer (clone-buffer nil nil)
    (let ((inhibit-read-only t))
        (keep-lines regexp rstart rend interactive)
        (kill-region (or rstart (line-beginning-position))
                     (or rend (point-max))))
    (kill-buffer)))
(unless (and buffer-read-only kill-read-only-ok)
    ;; Delete lines or make the "Buffer is read-only" error.
    (flush-lines regexp rstart rend interactive)))
```

4 Package setup

```
(require ’package)

(add-to-list ’package-archives ’("org" . "http://orgmode.org/elpa/"))
(add-to-list ’package-archives ’("melpa" . "http://melpa.org/packages/"))
(add-to-list ’package-archives ’("melpa-stable" . "http://stable.melpa.org/packages/"))

(setq package-enable-at-startup nil)
(package-initialize)

(unless (package-installed-p ’use-package)
(package-refresh-contents)
(package-install ’use-package))

(eval-when-compile
    (require ’use-package))
```

5 Include sensible-defaults.el

The settings (not keybindings) seem mostly inocuous and better choices.

```
(load-file "~/.emacs.d/sensible-defaults.el")
(sensible-defaults/use-all-settings)
```

6 Quality of life improvements

```
(setq-default indent-tabs-mode nil)
(setq tab-width 4)
(setq-default c-basic-offset ’tab-width)
(defvaralias ’cperl-indent-level ’tab-width)
(electric-indent-mode 1)
(electric-pair-mode 1)
(global-linum-mode)
(setq linum-relative-current-symbol "")
(with-eval-after-load ’adaptive-wrap
(setq-default adaptive-wrap-extra-indent 2))

(add-hook ’visual-line-mode-hook
(lambda ()
    (adaptive-wrap-prefix-mode +1)
    (diminish 'visual-line-mode)))

(global-visual-line-mode +1)
```

7 Version control highlighting

```
(global-diff-hl-mode)
(diff-hl-flydiff-mode)
(add-hook 'magit-post-refresh-hook 'diff-hl-magit-post-refresh)
```

8 Set up evil-mode

Evil is a necessity for any real editing.

```
(global-evil-leader-mode)
(evil-leader/set-leader "<SPC>")
(require 'evil)
(evil-mode t)
(evil-commentary-mode)
(global-evil-surround-mode t)
```

8.1 Leader bindings

```
(evil-leader/set-key "k" 'kill-buffer)
(evil-leader/set-key "g" 'magit-status)
; Okay what? why do these have to be swapped?
(evil-leader/set-key "<up>" 'evil-numbers/dec-at-pt)
(evil-leader/set-key "<down>" 'evil-numbers/inc-at-pt)
```

9 Use powerline

Cool fonts upset the nice themes though.

```
(require 'powerline)
(powerline-center-evil-theme)
```

10 Set up org-mode

```
(add-to-list 'load-path "~/dev/github/org-mode/lisp" t)
(add-hook 'org-mode-hook (lambda () (org-bullets-mode 1)))
(add-hook 'org-mode-hook (lambda () (linum-mode 0)))
(setq org-src-fontify-natively t)
(setq org-export-latex-listings t)
(setq org-ellipsis "")
(setq org-src-tab-acts-natively t)
(setq org-bullets-bullet-list '("" "" "" "" "" "" "" "" "∘" "◯" "" "" "" "" "" "" "" "
```

11 Set up helm

```
(use-package helm
    :ensure t)
(global-set-key (kbd "M-x") 'helm-M-x)
(global-set-key (kbd "C-x C-f") #'helm-find-files)
(helm-mode 1)
```

12 Use company-mode

```
(add-hook 'after-init-hook 'global-company-mode)
(setq company-dabbrev-downcase 0)
(setq company-idle-delay 0)
```

13 Language specific setup

13.1 Rust

It's always good to be optimistic about learning Rust.

```
(add-hook 'rust-mode-hook 'cargo-minor-mode)
(evil-leader/set-key-for-mode 'rust-mode "f" 'rust-format-buffer)
(setq racer-cmd "~/.cargo/bin/racer") ;; Rustup binaries PATH
(setq racer-rust-src-path "/Users/Max/.multirust/toolchains/stable-x86_64-apple-darwin/

(add-hook 'rust-mode-hook #'racer-mode)
(add-hook 'racer-mode-hook #'eldoc-mode)
(add-hook 'racer-mode-hook #'company-mode)
(add-hook 'flycheck-mode-hook #'flycheck-rust-setup)
```

13.2 Python

```
(add-hook 'python-mode-hook 'anaconda-eldoc-mode)
(add-hook 'python-mode-hook 'anaconda-mode)
(eval-after-load "company"
'(add-to-list 'company-backends 'company-anaconda))
```

13.3 Java

```
(add-hook 'java-mode-hook (lambda ()
                    (setq c-basic-offset 4
                            tab-width 4
                            indent-tabs-mode t)))
```