Bogdan's EMACS Config

Bogdan Popa

2014-04-27

Contents

1 Installation **2**
- 1.1 Linux . . . 2
- 1.2 Mac OS X . . . 2
- 1.3 Windows . . . 2

2 Setup **2**
- 2.1 Unix . . . 2
- 2.2 Windows . . . 3

3 Bindings **3**

4 EVIL **3**
- 4.1 Differences From Standard EVIL . . . 3
- 4.2 Normal Mode Bindings . . . 3
- 4.3 Visual Mode Bindings . . . 4
- 4.4 Insert Mode Bindings . . . 4

5 Magit **4**
- 5.1 Generic Bindings . . . 4
- 5.2 Status Bindings . . . 4
- 5.3 Branch Bindings . . . 4

6 Elisp **5**

7 Haskell **5**

8 Python **5**
8.1 Flycheck 5
8.1.1 Ignoring certain errors 5
8.1.2 Bindings 5
8.2 Jedi 5
8.2.1 Bindings 5

9 Scala **5**

10 Scheme **5**
10.1 Setup 5
10.2 Geiser mode 5
10.2.1 Buffer Bindings 6
10.2.2 REPL Bindings 6

1 Installation

1.1 Linux

This depends on the distribution. Most of them will have EMACS in their repositories.

```
sudo apt-get install emacs-snapshot-gtk
```

1.2 Mac OS X

Download and install EMACS for OS X from http://emacsformacosx.com.

1.3 Windows

Download and install EMACS for Windows from http://ftp.gnu.org.ua/gnu/emacs/windows/.

2 Setup

2.1 Unix

Clone the repo into your home directory and then fetch its submodules.

```
cd ~/
git clone https://github.com/Bogdanp/.emacs.d.git
cd .emacs.d
git submodule init
git submodule update
```

2.2 Windows

Windows makes things a bit harder. Add an environment variable called `HOME` and make it point to `C:/Users/Username/`. Clone the `windows` branch of the repository in that folder and fetch its submodules.

3 Bindings

The most important EMACS binding is `C-g`. Use it as an escape hatch for when you mess up your key combinations.

- `C-p` and `C-n` goto next and prev line.
- `C-a` and `C-e` goto the beginning and end of the current line.
- `C-c M-a` toggle between the current window state and a fullscreen terminal.

4 EVIL

4.1 Differences From Standard EVIL

My EVIL mode fork makes some changes to the way registers are handled. Most importantly, while you are inside an EVIL mode buffer you can only access the clipboard through the `+` register and nothing else, this prevents EMACS from dirtying the clipboard whenever you cut things.

The copy-on-motion bug is fixed.

`C-w` works in the minibuffer and in the search buffer.

4.2 Normal Mode Bindings

- `SPC` bring up ace jump mode.
- `S-SPC` bring up ace char jump mode.
- `C-w f` toggle between making the current window fullscreen or not.

4.3 Visual Mode Bindings

The `C-a`, `C-e`, `C-p`, `C-n` bindings are available in visual mode.

4.4 Insert Mode Bindings

The `C-a`, `C-e`, `C-p`, `C-n` bindings are available in insert mode.

- `C-w` deletes the previous word.
- `C-r` inserts whatever is in a given buffer (`C-r +` will

insert whatever is on the clipboard).

5 Magit

Magit is fucking great. You can find its official manual at http://magit.github.io/master/magit.html.

5.1 Generic Bindings

- `n` goto next object.
- `p` goto previous object.
- `TAB` expand/collapse object.

5.2 Status Bindings

Use `C-c m` to bring `magit-status` up.

- `S` stage everything.
- `s` stage object under point.
- `u` unstage object under point.
- `cc` begin commit.
- `FF` pull.
- `PP` push.

5.3 Branch Bindings

Use `bv` inside `magit-status` to bring up the visual branch manager.

6 Elisp

7 Haskell

8 Python

8.1 Flycheck

Install `flake8` to use it as a backend for Flycheck.

8.1.1 Ignoring certain errors

Create a `.flake8rc` file in your `HOME` directory. For example:

```
[flake8]
ignore = E501,F403,E712
```

8.1.2 Bindings

- `C-c !n` and `C-c !p` goto next and prev error.
- `C-c !l` list errors.

8.2 Jedi

Install `virtualenv` and `epc` and then run `M-x jedi:install-server`.

8.2.1 Bindings

- `C-c .` goto definition (and `C-c ,=). - =C-c ?` show documentation of the object at point.

9 Scala

10 Scheme

10.1 Setup

Link `mzscheme` and `racket` so that they can be found in `PATH`.

10.2 Geiser mode

Start it up with `M-x run-geiser`.

10.2.1 Buffer Bindings

- `C-c C-a` to switch to the REPL and enter the current module.
- `C-c C-z` to switch between the buffer and the REPL.
- `C-M-x` eval definition around point.
- `C-c M-e` eval definition around point and switch to REPL.
- `C-c C-x` eval sexp before point.
- `C-c C-r` eval region (also `C-c M-r`).
- `C-c C-b` eval buffer (also `C-c M-b`).
- `M-g n` and `M-g p` to jump to next and prev error.

10.2.2 REPL Bindings

- `C-c C-q` kill Scheme process.
- `C-c M-o` clear REPL.