Emacs Workshop for Beginners

Karl Voit

April 28, 2017

License

Emacs Distributions

Emacs Distributions = Emacs base files + modes + install routines + goodies

- XEmacs: historical
- GNU/Emacs: this is where the music plays
  - GNU/Linux
    - easy (see next section)
  - macOS
    - GNU/Emacs for OS X
    - Aquamacs (historical)
  - Microsoft Windows
    - GNU/Emacs

Installing Emacs

Download GNU/Emacs (general URL)

- Windows
  - download latest ZIP from a mirror
    - e.g., `emacs-25.2-x86_64.zip`
  - unpack ZIP file
  - start installation exe
  - create start icon
- GNU/Linux
  - ask your local package management
    - e.g., `apt install emacs`
- macOS
  - 3 options
    - `brew install emacs --with-cocoa`
    - `sudo port install emacs-app`
    - Emacs for OSX

Starting Emacs

- `emacs`
- run as server + `emacsclient`
  - `(server-start)`
- plain GNU/Emacs: `emacs -q` or `emacs --no-init-file`
- debugging startup issues: `emacs --debug-init`

UI Elements

UI Elements Overview

- Menu Bar
- Windows != Buffers
- Mode Line(s)
- Mini Buffer

Commands

- Icons
- Menus
- Shortcuts
- `M-x` → type in commands in Mini Buffer

Abbrev	Key
`C`	`Ctrl`
`M`	`Alt` or `ESC`
`S`	`Shift`
`C-M-t`	press `t` while holding `Ctrl` and `Alt`

Help

- `C-h ?` help of help
- `C-h v` help variable
- `C-h f` help function

Example: sort-lines → `C-h f` = help for this function

`M-x help-with-tutorial` (`C-h t`)

Basic movement and text manipulation. Just do it!

Cheatsheets

- GNU Emacs Reference Card
- Sacha Chua:
  - How to Learn Emacs
  - How to learn Emacs keyboard shortcuts
  - Map for learning Org Mode
  - Use Dired to manage files; Emacs movements
- Org-Mode Reference Card

Modes

Many modes are built-in so that you can run them out-of-the-box:

- Org-mode
- mail-mode
- ...

Thousands of modes can be installed on-top for your Emacs.

Some modes are loaded according to file extensions: `FOOBAR.org` → Org-mode

Geting Even More

- elisp-files
  - GitHub
  - web pages
  - simple snippets
- Package manager
  - configure package repositories
    - ELPA
    - MELPA
    - Marmalade

  - M-x list-packages

init.el

System startup file: /etc/emacs/site-start.el

Each user has its startup file that holds personalizations:

- $HOME/.emacs or $HOME/.emacs.d/init.el
  - On Windows, it's more complicated
- My startup files (for getting ideas)
  - (load "~/.emacs.d/main.el") loads my main.el
  - best used within Emacs:
    - C-u TAB on the first asterisk → fold headings starting with ;;*
    - TAB fold/unfold a single heading

use-package

Installing packages within your init.el: use-package

Installing via M-x list-packages (and mouse):

- installs only on current host
- more detailed configuration has to be written in init.el in any case

Advantage of use-package: installs most current version of a package on each host that uses same (shared) init.el.

One-time setup in init.el to be able to use use-package:

From cachestocaches.com:

```
(require 'package)
(setq package-enable-at-startup nil)
(add-to-list 'package-archives '("melpa" . "http://melpa.org/packages/"))
(add-to-list 'package-archives '("marmalade" . "http://marmalade-repo.org/packages/"))
(add-to-list 'package-archives '("gnu" . "http://elpa.gnu.org/packages/"))
(package-initialize)

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

(eval-when-compile
  (require 'use-package))
(require 'diminish)
(require 'bind-key)
```

Examples of installing packages via `use-package`:

```
;; Nyan cat as a horizontal scroll bar
(use-package nyan-mode
  :ensure t ;; install package if not found OR: (setq use-package-always-ensure t)
  :config ;; executed after loading package
  (nyan-mode t)
)

;; http://www.emacswiki.org/emacs/UndoTree
(use-package undo-tree
  :ensure t
  :diminish undo-tree-mode
  :config ;; executed after loading package
  (autoload 'undo-tree "undo-tree.el")
)

;;** yafolding
;; https://github.com/zenozeng/yafolding.el
;; Folding based on identation
(use-package yafolding
  :ensure t
  :defer 10
  :mode ("\\.xml\\'" . yafolding-mode)
  :config ;; executed after loading package
  (add-to-list 'auto-mode-alist '("\\.xml$" . nxml-mode))
  :bind (("<M-S-return>" . yafolding-toggle-all)
         ("<M-return>" . yafolding-toggle-element)
         ("<C-Tab>" . yafolding-toggle-element))
)
```