

Emacs

Nicolas Lamirault

Me, Myself and I



Software Engineer



nlamirault.github.io



Open source fan



@nlamirault



github.com/nlamirault



gitlab.com/nicolas-lamirault

Introduction

History

- ❶ Ecrit en 1976 par **Richard Stallman**
- ❷ **E** ditor **MAC** ro **S** for TECO
- ❸ Utilise le langage d'extension **Emacs Lisp**
- ❹ Premier projet **GNU**

Ce qu'il permet

- 💡 Ecrire du texte :)
- 💡 Développer
- 💡 Lire ses mails
- 💡 Communiquer (Irc, Xmpp, ...)
- 💡 ...

What else

“ *"I'm using **Linux**. A library that **emacs** uses to communicate with Intel hardware."* — heard on *#emacs on Freenode*

Ecosysteme

Packages

 Depuis la version 24

 Plusieurs sources :

1. GNU elpa
2. **Melpa**
3. Marmelade

Package manager

 Inclus dans Emacs : package.el

 Addons :

1. paradox
2. cask
3. quelpa
4. el-get

Package	Version	Status	Archive	Description
1 bpr	20151106	new	nelpa	11 Background Process Runner
2 caroline-theme	20151030	new	nelpa	1 A trip down to New Orleans.
3 comint	20151104	new	nelpa	2 Toggle grouping commas in numbers
4 dracula-theme	1.1.0	new	nelpa-stable	1316 Dracula Theme
5 dracula-theme	20151102	new	nelpa	1316 Dracula Theme
6 ewm	0.4	new	gnu	Emacs X Window Manager
7 helm-grepint	20151030	new	nelpa	0 Gene Helm interface to grep
8 live-py-mode	2.8.1	new	nelpa-stable	43 No description available
9 live-py-mode	20151105	new	nelpa	43 Live Coding in Python
10 pastelabc-theme	1.0	new	nelpa-stable	1 a soothing theme with a pastel color palette
11 pastelabc-theme	20151030	new	nelpa	1 a soothing theme with a pastel color palette
12 rbt	20151030	new	nelpa	3 Integrate reviewboard with emacs
13 replace-symbol	1.0	new	nelpa-stable	0 Rename symbols in expressions or buffers
14 replace-symbol	20151030	new	nelpa	0 Rename symbols in expressions or buffers
15 termbright-theme	1.0	new	nelpa-stable	1 a more usable theme for white-on-black terminals
16 termbright-theme	20151030	new	nelpa	1 a more usable theme for white-on-black terminals
17 oblavoat	20151021	available	nelpa	29 Layout grouping with ease
18 2048-game	20151026	available	nelpa	play 2048 in Emacs
19 4closure	20131014	available	nelpa	Open and evaluate 4closure.com questions
20 d	20140707	available	nelpa	35 multiple-inheritance prototype-based objects DSL
21 abc-mode	20140225	available	nelpa-stable	6 Major mode for editing abc music files
22 abc-mode	20140225	available	nelpa	6 Major mode for editing abc music files
23 abl-mode	20140724	available	nelpa	25 Python TDD minor mode
24 abyss-theme	20151007	available	nelpa	2 A dark theme with contrasting colours
25 ac-chemist	0.1	available	nelpa-stable	6 auto-complete source for alchemist
26 ac-chemist	20150907	available	nelpa	6 auto-complete source for alchemist
27 ac-anaconda	0.1.0	available	nelpa-stable	5 Anaconda sources for auto-complete-mode
28 ac-anaconda	20150912	available	nelpa	5 Anaconda sources for auto-complete-mode
29 ac-headers	20151021	available	nelpa	15 auto-complete source for C headers
30 ac-cake	1.0.0	available	nelpa-stable	3 CakePHP Minor Mode auto-complete.el source
31 ac-cake	20140315	available	nelpa	3 CakePHP Minor Mode auto-complete.el source
32 ac-cake2	1.0.1	available	nelpa-stable	3 CakePHP 2 Minor Mode auto-complete.el source
33 ac-cake2	20140320	available	nelpa	3 CakePHP 2 Minor Mode auto-complete.el source
34 ac-capp	0.1	available	nelpa-stable	5 auto-complete source with completion-at-point
35 ac-capp	20151031	available	nelpa	5 auto-complete source with completion-at-point
36 ac-cider	0.2.1	available	nelpa-stable	55 Clojure auto-complete sources using CIDER
37 ac-cider	20151012	available	nelpa	55 Clojure auto-complete sources using CIDER
38 ac-clang	1.6.0	available	nelpa-stable	17 Auto Completion source for clang
39 ac-clang	20150906	available	nelpa	17 Auto Completion source for clang
40 ac-dabbrev	20130905	available	nelpa	6 auto-complete.el source for dabbrev
41 ac-dcd	0.4	available	nelpa-stable	6 Auto Completion source for dcd for GNU Emacs
42 ac-dcd	20150702	available	nelpa	6 Auto Completion source for dcd for GNU Emacs
43 ac-enmet	20131015	available	nelpa	9 auto-complete sources for enmet-mode's snippets
44 ac-enmet	0.2	available	nelpa-stable	4 auto-complete source of Enmet
45 ac-etag	20150823	available	nelpa	4 auto-complete source of Etag
46 ac-etas	0.6	available	nelpa-stable	18 etags/ctags completion source for auto-complete
47 ac-etas	20151031	available	nelpa	18 etags/ctags completion source for auto-complete
48 ac-getter	1.1	available	nelpa-stable	10 Auto-complete backend for getter
49 ac-getter	20130928	available	nelpa	10 Auto-complete backend for getter
50 ac-haskell-process	0.7	available	nelpa-stable	10 Haskell auto-complete source which uses the current haskell process
51 ac-haskell-process	20150423	available	nelpa	10 Haskell auto-complete source which uses the current haskell process
52 ac-helm	2.4	available	nelpa-stable	12 Auto Complete with Helm
53 ac-helm	20140919	available	nelpa	12 Helm interface for auto-complete
54 ac-html	0.31	available	nelpa-stable	15 auto complete source for html tags and attributes
55 ac-html	20151005	available	nelpa	15 auto complete source for html tags and attributes
56 ac-html-bootstrap	0.9.3	available	nelpa-stable	6 auto complete bootstrap/fontawesome classes for ac-html and comanv-web

Developpement

Go

i go-mode

i company-go

i go-code

i go-errcheck

i go-tests :)

```
1 package main
2
3 import (
4     "fmt"
5     "github.com/mitchellh/cli"
6     "io/ioutil"
7     "log"
8     "os"
9 )
10
11 func main() {
12     os.Exit(realMain())
13 }
14
15 func realMain() int {
16     log.SetOutput(ioutil.Discard)
17
18     // Get the command line args. We shortcut "--version" and "-v" to
19     // just show the version.
20     os.Args = []string{"", "--version"}
21     for ModeNull, args {
22         ModeAppend      args {
23             ModeCharDevice  "--" {
24                 ModeDevice  "sk"
25                 ModeDir      ""
26                 ModeExclusive "-v" || arg == "--version" {
27                     Args := make([]string, len(args)+1)
28                     Args[0] = "version"
29                     Args[1:] = (newArgs[1:], args)
30                     s = newArgs
31                     break
32                 }
33             }
34         }
35     }
36     cli := &cli.CLI{
37         Args:      args,
38         Commands:  Commands,
39         HelpFunc:  cli.BasicHelpFunc("consul"),
40     }
41     exitCode, err := cli.Run()
42     if err != nil {
43         fmt.Fprintf(os.Stderr, "Error executing CLI: %s\n", err.Error())
44         return 1
45     }
46     return exitCode
47 }
48 }
```

20 11 ~*-[consul]main.go All -master Go Compiling WK [realMain() int] [Onat|actarus|decks|entgna|nat|scale|vision]
const DevNull

Example

```
(add-hook 'before-save-hook 'gofmt-before-save)
(add-hook 'go-mode-hook
  (lambda ()
    (local-set-key (kbd "C-x g r") 'go-remove-unused-imports)
    (local-set-key (kbd "C-x g i") 'go-goto-imports)
    (local-set-key (kbd "C-x g d") 'godoc)
    (local-set-key (kbd "C-x g j") 'godef-jump)))
```

Communauté

Web

 [Redit](#)

 [StackExchange](#)

 [awesome-emacs](#)

Liste de diffusions

 [emacs-devel](#)

 [help-gnu-emacs](#)

Autre

 IRC: #emacs on freenode

 [Prelude](#)

 [Spacemacs](#)

 [Graphene](#)

 [Scame](#) :)

Conclusion

References

- 📌 Slides are accessible here nlamirault.github.io/decks/emacs
- 📌 Slides source github.com/nlamirault/decks/emacs
- 📌 Slides generated with **Asciidoctor** and **DZSlides** backend
- 💡 Original slide template - **Dan Allen & Sarah White**
- 📌 License of this doc: [CC BY-SA 4.0](https://creativecommons.org/licenses/by-sa/4.0/)

Nicolas Lamirault



@nlamirault