

Play Emacs like an instrument

- Teaser -



200ok GmbH

Alain M. Lafon, 2019-10-26

alain@200ok.ch

Outline

Introduction

Impressions

Definition time

Play Emacs like an instrument

Closing words

Introduction

The tiger bites the thrower,
the dog chases the mud-ball

Proposition

- Most work on the computer is based on either text processing or text consumption
- Not having a general text editor at your disposal is like being a carpenter and only having a hammer in the toolbox

The tiger bites the thrower,
the dog chases the mud-ball

Emacs is a great general text processor

- Emacs is a Lisp REPL
- It is self documenting
- It can be changed drastically by the user while it is running
- To me, it is the Swiss Army knife, not of text processors, but ultimately of programs

Original talk

- This **lightning talk** is a teaser of the original talk
 - It'll give you an impression on whether it's worth for you to watch the original talk or not
-

- I'll cover **a lot** of ground in the original talk
 - That's because I spend most of my *computer time* in Emacs
 - Also, there will be lots of background-info up-front
 - It is **not** a lightning talk
 - It's about 2h long \(^_^\)/

Impressions

The faces of Emacs

- Some impressions of the talk
- I call them "The faces of Emacs"

The faces of Emacs

Inception (obviously)

```
- Introduction
** About me...
** Before we begin...
** Right...
** How did you get started with Emacs?...
** Before we begin...
** That's a good point...
** What's your name? Maybe....
** Don't judge me now...
** Questions...
** Definition time...
** Play Emacs like an instrument...
** Closing words...
** PROJECT Incorporate
** TODO Quotes to incorporate
** Frustrations...
```

Play Emacs like an instrument



200ok GmbH

Alain M. Lafon, 23.04.2018

alain@200ok.ch

1

¹org-mode and pdf-tools

The faces of Emacs

200
OK

Mail (Text and HTML)

2

The faces of Emacs

200
OK

Git

³magit (Also: Note how Emacs is configured with literate programming)

The faces of Emacs

Organization

#+SETUPFILE: orgmode-export.conf			
	Processes...		
	INBOX		
	trash		
	mobileorg		
	NEXT...		
	Someday		
	Reference		
	Journal...		
	200 OK		
	INSPOR		
	** New: inspor...		
	** Legal...		
	** Spesen Laebde Zen Tempel		
Datum	Beschreibung	Quittung	Spesen
20.01.2017	Marketing: Fujifilm Zubehör	EN	132,00 JPY
24.04.2017	internationale Academy.com	Mail	10EUR
25.01.2017	inspor.com	Mail	10EUR
25.01.2017	zen-schweiz.com	Mail	10EUR
29.03.2017	zen-temple.net	Mail	3.71EUR;16EUR;6,26EUR
29.03.2017	zen-temple.ch	Mail	1,00EUR;20,92EUR;6,26EUR
26.04.2017	internationale Academy.com	Mail	10EUR
26.07.2017	Charme Talk Airbus	Mail	9EUR
24.01.2018	Ainem Shop	Mail	70,00EUR
31.01.2018	Domain: inspor.com	Mail	10EUR
28.04.2018	Domain: zen-temple.x	Mail	10EUR
07.05.2017	Internationale Yearly Tax	Mail	29,-94USD
07.05.2017	WP NextScripts License	Mail	49,-95USD
14.11.2017	Meetup	Mail	29,-94USD
09.05.2017	Spende Felsentor	EN	150CHF
29.06.2017	Studienreise Flug	Mail	1173,-50CHF;19CHF
24.07.2017	Internationale Fotografie Buch	Mail	130CHF
26.07.2017	Rauchwärmeleider	Mail	72,-27CHF
29.07.2017	Marketing: Affinity Photo	Mail	48CHF
06.08.2017	Airbnb Osaka Studienreise	Mail	77CHF
07.08.2017	Airbnb Tokyo Studienreise	Mail	84CHF
15.08.2017	Reisekosten Fujifilm X70	EN	45CHF
13.11.2017	Schlüchtung	EN	350CHF
27.12.2017	Kurtaxe Glarau 2017	Konto	117,68CHF
22.03.2018	Shikibuton	Mail	153,89CHF
			49355 JPY + 233,96 EUR + 109,83 USD + 2736,26 CHF
#+TBLFM: @+4\$var{g2..g1}			

#+TBLFM: @+4\$var{g2..g1}

File things.org Top (2022-06-20) Git:master (Org v1.10 Fly Projectile[long] FlyC- emacs-lisp-tree (+) ARev Full) 7:09PM 1.02 Mail

⁴ Spreadsheets, project planning, time tracking, etc with org-mode

The faces of Emacs

Browsing the web (distraction/tracking free)

[200ok - Consultancy, Research Lab, Troubulator; http://200ok.ch/posts/200ok-in-the-media.html](http://200ok.ch/posts/200ok-in-the-media.html)

Hacken muss nicht negativ sein

Am Freitagabend begannen die Teilnehmer, an ihren Projekten zu arbeiten. «Das Ziel ist es, mit den Projekten einen Mehrwert für das Glarnerland zu schaffen», sagt Schötzellus. Von einer WLAN-Richtantenne, die 15 Kilometer weit sendet, über das schlüssellose Öffnen einer Haustür bis zu automatisierten Bestelllösungen für Getränke aus Essen: Hier wurde vieles gepräsentiert. Eine davon war ein von Lukas Pfeifer und Max Lüthi gegründetes Unternehmen, das am 200ok Hackathon teilnahm. Er ist Geschäftsführer der 200ok GmbH aus Zürich, die auch in Glarus einen Standort hat. Am «Hackathon» half er als Co-Organisator mitgewirkt. «Der Begriff ‐Hack‐ bedeutet, etwas zu versuchen. Zu schauen, was überhaupt möglich wäre.» Der Begriff sei mittlerweile besonders wegen der Medien negativ konnotiert.

«Uns ist es wichtig, dass man an diesem Wochenende spielerisch lernen kann. Es braucht keine abschliessende Lösung», erklärt Lafon. «Genau, das Projekt muss nicht perfekt sein. Aber das Konzept sollte gut und umsetzbar sein. Das ist genau hier der Beweis, dass es funktioniert. Ich denke, dass die ohne technisches Vorwissen mitgemacht hätten. «Sie wollten einfach etwas lernen und gehen nun mit einem Mehrwissen in die Welt hinaus.» Das nütze ihnen selbst, aber auch der Gesellschaft, da sie ihr Wissen nun in anderen Situationen einbringen könnten.]]



Oben: Ein Teilnehmer am 200ok Hackathon arbeitet an seinem Projekt. Unten links: Die beiden Gründer des 200ok-Gesellschaften, Lukas Pfeifer und Max Lüthi.

5

⁵Browsing 200ok.ch with eww

The faces of Emacs



Someone might even use Emacs for coding

The screenshot shows the Emacs interface with two main panes. The left pane displays Clojure test code using CIDER, while the right pane shows the test results.

```
(deftest add-callback
  (is (= (c/add-callback {:a 1} :b identity)
         {:a 1 :callbacks {:b [identity]}}))

(deftest resolve-strategy
  (is (= (c/*/resolve-strategy {} :a) nil))
  (is (= (c/*/resolve-strategy {:a_id 1} :a) :belongs-to)))

(deftest resolve-missing
  (is (= (c/resolve-missing {} :a) "no value for :a"))
  (is (= (c/resolve-missing {:a_id 1} :a) "resolving associated :a")))

(deftest entity-map
  (is (= ((c/*/entityMap {:a_id 1}) :a) "resolving associated :a"))

(deftest run-callback
  (is (= (c/*/run-callback
           (fn [& x] true) nil nil {:a 24})
         {:a 42} [])))
  (is (= (c/*/run-callback
           (fn [& x] 23) nil nil {:a 42})
         {:a 23} [])))
  (is (= (c/*/run-callback
           (fn [& x] false) nil nil {:a 42}))
        {:a 42} [[{:error "Unspecified Error"}]))
  (is (= (c/*/run-callback
           (fn [& x] "Volmposten") nil nil {:a 42})
        {:a 42} [[{:error "Volmposten"}])))
  (is (= (c/*/run-callback
           (fn [& x] 23) nil nil {:a 42})
        {:a 42} [[{:error "Unknown Error" :object 23}]])))

(def registry
  {:petunias
   {:callbacks
    {:drop [(fn [& _] false)]}
    :more-petunias
    {:callbacks
     {:drop [(fn [& _] false)
             (fn [& _] false)]}
    :even-more-petunias
    {:callbacks
     {:drop [(fn [& _] [:c 23])
             (fn [& _] "Volmposten")]}}
    :whale
    {:cider-test-clj 33 (:c 40) :nrepl 615/master (:closure >1 MB to Show Projectile) (:cider-test-report) All Li <H> (Test Report +1 NK) Projectile(collecture) Fly
  6
```

Test Summary
collecture.entities-test
Tested 1 namespaces
Run 17 assertions, in 14 test functions
0 failures

Results
collecture.entities-test
1 non-passing tests:
Fail in run-callback
expected: [[{:a 24} []]]
actual: [[{:a 42} []]]
diff: - [[{:a 24} []]]
+ [[{:a 42} []]]

⁶Test Clojure code with cider

The faces of Emacs

200
OK

Infinite diversity in infinite combinations

Ok then, but what **is** Emacs?

Definition time

What even is Emacs?

C-h i (Emacs FAQ -> Status of Emacs)

Next: [Latest version of Emacs](#), Up: [Status of Emacs](#)
(efaq)Top > [Status of Emacs](#) > [Origin of the term Emacs](#)

4.1 Where does the name “Emacs” come from?

Emacs originally was an acronym for Editor MACRoS. The says he “picked the name because it was not the name of an abbreviation at MIT at the time.” The first Emacs was a set of macros written in 1976 at MIT by RMS for the editor TECO (Text Editor and COrrector, originally Tape Editor and COrrector) under ITS (the Incompatible Timesharing System) on a PDP-10. He had already invented what would become a “readline” facility, which made with programmable keys. These were inserted by the user as a prefix to enter the new command. This command will end the function in RCT, and completed by RCT.

So... Emacs is software from 1976?!



Every other old software is **young** compared to that

- The first VIM release was 1991
 - Says Wikipedia⁷, I couldn't find it in the VIM in-editor docs, on vim.org the original VIM 1.0 source of the git repo of current VIM (8.0).
 - Linux appeared in 1991, as well⁸
 - Only C is (slightly) older (1972)⁹

⁷[https://en.wikipedia.org/wiki/Vim_\(text_editor\)](https://en.wikipedia.org/wiki/Vim_(text_editor))

⁸<https://en.wikipedia.org/wiki/Linux>

⁹[https://en.wikipedia.org/wiki/C_\(programming_language\)](https://en.wikipedia.org/wiki/C_(programming_language))

Emacs is the mother of all **Free Software**

M-x describe-gnu-project

- GCC
- GNU
- Emacs
- GPL
- FSF
- GNU/Linux

GNU Emacs

The Emacs tutorial (C-h t) dates the copyright to 1985.

What is GNU?¹⁰

GNU is an operating system that is free software—that is, it respects users' freedom. The development of GNU made it possible to use a computer without software that would trample your freedom.

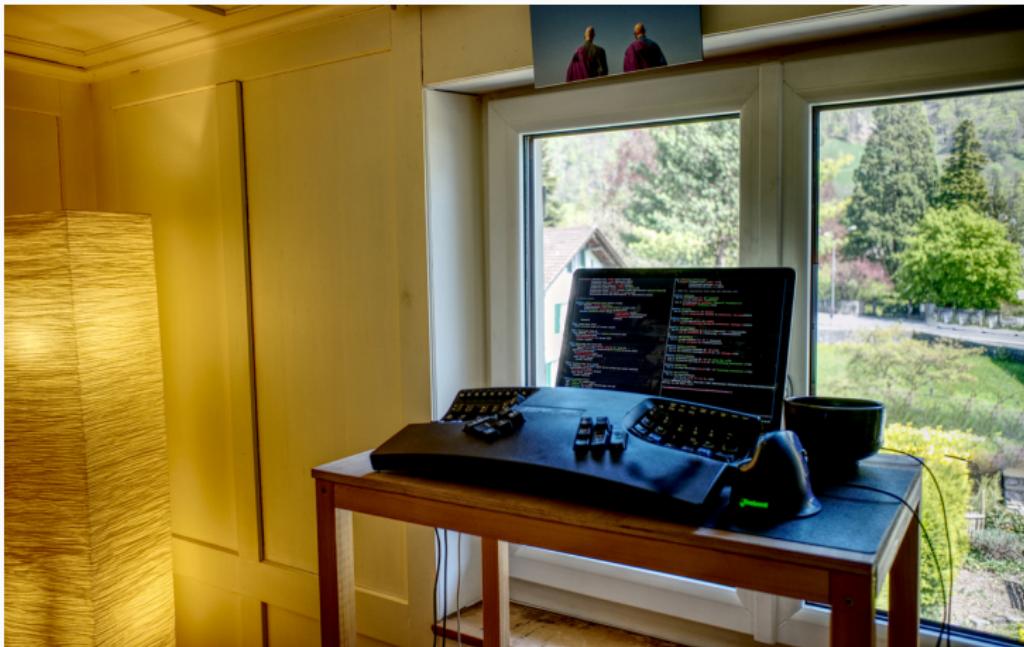
¹⁰<https://www.gnu.org/>

Play Emacs like an instrument

The meat part

- Switching gears now into live Emacs usage
- I plan to announce all the new features that I'm touching
- If I'm loosing you, forget to mention the name of a feature or you have any other questions, don't hesitate to ask!

Live Coding - what can go wrong?



Closing words

If you liked this short teaser, please check out the original talk here: https://200ok.ch/posts/2018-04-27_Play_Emacs_like_an_Instrument.html

Further reading



If you want to continue on your own:

- Emacs Tutorial: C-h t || M-x help-with-tutorial
- Emacs Manual: C-h r || M-x info-emacs-manual
- All info manuals: C-h i || M-x info
 - org-mode
 - magit
 - mu4e
 - Actually all info-manuals from your system and Emacs
- 200ok.ch/category/emacs.html
- 200ok.ch/atom.xml

Further reading

Configuration

The screenshot shows a web browser window displaying the README.org page for the `munen/emacs.d` repository on GitHub. The page title is "Emacs configuration". It contains sections for "Installation" and "Dependency management". The "Installation" section includes instructions for cloning the repository and managing dependencies via Org mode. The "Dependency management" section provides code snippets for defining package repositories and installing packages.

Emacs configuration

This repository contains my Emacs configuration. It is written and documented in literate programming style.

Installation

Initial

Emacs configuration is usually done in the home directory in the `~/.emacs.d` folder. This holds true for Unix and Linux systems. For Windows, look it up [here](#).

```
git clone git@github.com:munen/emacs.d.git ~/.emacs.d
```

Dependencies

Emacs dependencies/libraries are managed via the internal [package management system](#). To initially install packages, open `~/.emacs.d/init.el`, refresh your package list with `M-x package-refresh-contents` and install everything using `M-x eval-buffer`.

Dependency management

Define package repositories(archives)

```
(require 'package)
(package-archives '(("gnu" . "http://elpa.gnu.org/packages/")
                    ("org" . "https://marmalade-repo.org/packages/")
                    ("elpa" . "https://elpa.org/packages/")))
```

Define packages that are to be installed

```
List all used third-party packages. Most will be configured further down in this file, some are used with the default
```

11

¹¹Literate Configuration/Programming through org-mode

Give yourself a chance, use Emacs!

- If you liked this talk, head over to <https://github.com/munen/emacs.d/>, go through the README and enjoy magical powers
- Ah, yes - and put a star on the repo, would ya?(:

