



# **tmux :** **an open-source and keyboard- centric terminal multiplexer**

*With some keyboard tips to skyrocket your  
development workflow*

Léo Guillon

- 
- 1 Introduction
  - 2 tmux as your terminal multiplexer
  - 3 Keyboard tips

- 1 Introduction**
- 2 tmux as your terminal multiplexer**
- 3 Keyboard tips**

# Who am I ? What I work on ?

---

# Who am I ? What I work on ?

---

**dec. 2024 → nov. 2025** Research engineer @ Aramis

**sep. 2025 → june 2026** Preparing mathematics  
aggregation @ Sorbonne Université



# Who am I ? What I work on ?

**dec. 2024 → nov. 2025** Research engineer @ Aramis

**sep. 2025 → june 2026** Preparing mathematics  
aggregation @ Sorbonne Université



Dev needs :

- Code : R, Python (+ Julia, HTML/CSS, ...) + Git/GitHub
- Note-taking
- Document writing : L<sup>A</sup>T<sub>E</sub>X
- Some command line interface (CLI) and terminal user interface (TUI) tools.

## My general work and tools philosophy

---

My work philosophy is to use tools that are :

## My general work and tools philosophy

---

My work philosophy is to use tools that are :

- **efficient** : keyboard-centric, local files, ...

---

My work philosophy is to use tools that are :

- **efficient** : keyboard-centric, local files, ...
- **minimalist** : ability to just do what I need, nothing more ; customization if possible

---

My work philosophy is to use tools that are :

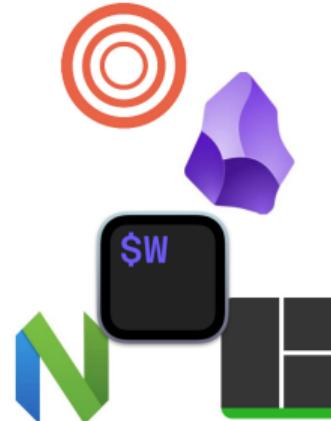
- **efficient** : keyboard-centric, local files, ...
- **minimalist** : ability to just do what I need, nothing more ; customization if possible
- **open** : open-source if possible

My work philosophy is to use tools that are :

- **efficient** : keyboard-centric, local files, ...
- **minimalist** : ability to just do what I need, nothing more ; customization if possible
- **open** : open-source if possible

Examples:

- Zen Browser (web browser);
- Obsidian (note-taking);
- Sioyek (pdf reader);
- WezTerm (terminal), alongside:
  - neovim (code);
  - yazi (file explorer)
  - tmux (terminal multiplexer);



## ① Introduction

## ② tmux as your terminal multiplexer

Use cases

General workflow and vocabulary

Setup and commands

## ③ Keyboard tips

## ① Introduction

## ② tmux as your terminal multiplexer

Use cases

General workflow and vocabulary

Setup and commands

## ③ Keyboard tips

# What is a terminal multiplexer ?

## Terminal multiplexer concept

Instead of working with single terminal, and nuking everything once you close it, you split it in multiple windows.

# Use cases for a terminal multiplexer

---

If you :

## Use cases for a terminal multiplexer

---

If you :

- are already used to the terminal and its commands;

---

If you :

- are already used to the terminal and its commands;
- use several command line interface (CLI) and terminal user interface (TUI) tools;

If you :

- are already used to the terminal and its commands;
- use several command line interface (CLI) and terminal user interface (TUI) tools;
- want to make your terminal sessions persistent;

If you :

- are already used to the terminal and its commands;
- use several command line interface (CLI) and terminal user interface (TUI) tools;
- want to make your terminal sessions persistent;
- are willing to have a streamlined, keyboard-focused workflow,

If you :

- are already used to the terminal and its commands;
- use several command line interface (CLI) and terminal user interface (TUI) tools;
- want to make your terminal sessions persistent;
- are willing to have a streamlined, keyboard-focused workflow,

then, a terminal multiplexer may be the right tool for your day-to-day dev work environment !



## ① Introduction

## ② tmux as your terminal multiplexer

Use cases

General workflow and vocabulary

Setup and commands

## ③ Keyboard tips

# tmux's hierarchy

---

---

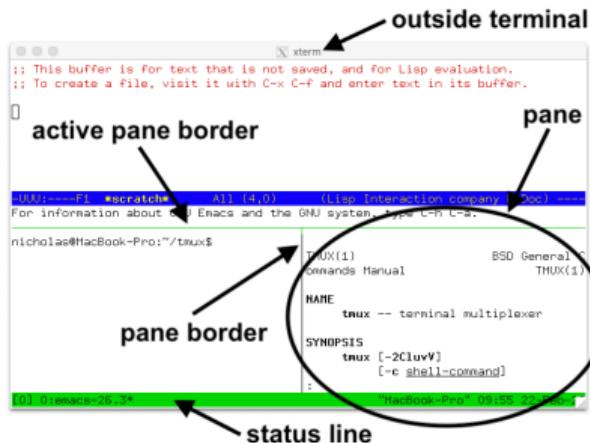
**pane** a single terminal instance

**pane** a single terminal instance

**window** a collection of panes, arranged in a defined layout

**pane** a single terminal instance

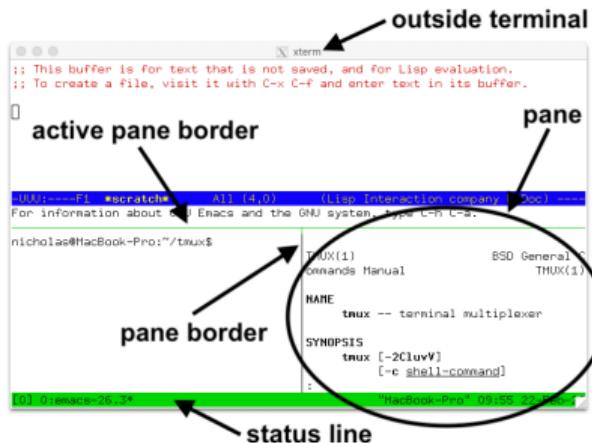
**window** a collection of panes, arranged in a defined layout



**pane** a single terminal instance

**window** a collection of panes, arranged in a defined layout

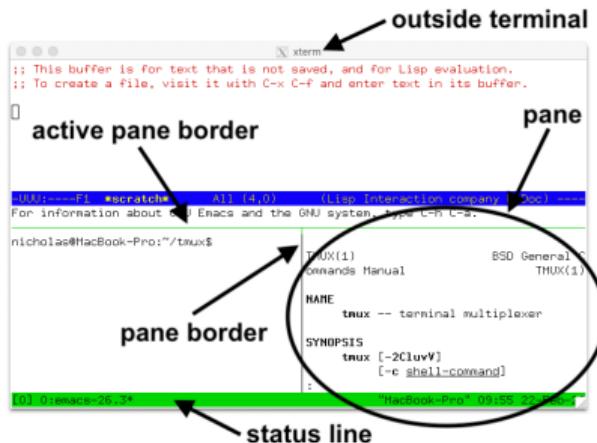
**session** a collection of windows



**pane** a single terminal instance

**window** a collection of panes, arranged in a defined layout

**session** a collection of windows



# Workflow with tmux

---

- 
- ① Terminal startup
  - ② You attach to an already existing tmux session, or create a new one
  - ③ In a session, you go in your working window (or create a new one)
  - ④ In a window, you choose or create a pane
  - ⑤ Have a nice working session !

- ① Terminal startup
- ② You attach to an already existing tmux session, or create a new one
- ③ In a session, you go in your working window (or create a new one)
- ④ In a window, you choose or create a pane
- ⑤ Have a nice working session !

Once you're done :

- ① You can kill your pane, window and/or session if you want to
- ② Detach from your session, and done !

Use case for my current work :

### **L<sup>A</sup>T<sub>E</sub>X custom commands**

Writing a L<sup>A</sup>T<sub>E</sub>X package for my custom mathematical commands, as well as a L<sup>A</sup>T<sub>E</sub>X document with a big table to present them, as well as a .json file to have autocompletion snippets in Obsidian.

## ① Introduction

## ② tmux as your terminal multiplexer

Use cases

General workflow and vocabulary

Setup and commands

## ③ Keyboard tips

## tmux's config

```
~/  
└── .config/  
    └── tmux/  
        └── tmux.conf
```

## tmux's config

```
~/  
└── .config/  
    └── tmux/  
        ├── config/  
        │   ├── tmux.keybindings.conf  
        │   ├── tmux.options.conf  
        │   └── tmux.theme.conf  
        ├── plugins/  
        └── ...  
        └── tmux.conf
```

## Philosophy of the keys succession

Instead of having keyboard shortcuts as keys combination, inputs are keys *succession*.

## Philosophy of the keys succession

Instead of having keyboard shortcuts as keys combination, inputs are keys *succession*.

- not compete with usual OS or app keyboard shortcuts, TUI specific commands
- it's easier to enter a succession than a combination of keystrokes

## **Philosophy of the keys succession**

Instead of having keyboard shortcuts as keys combination, inputs are keys *succession*.

- not compete with usual OS or app keyboard shortcuts, TUI specific commands
- it's easier to enter a succession than a combination of keystrokes

## **tmux command formula**

**tmux command = prefix + command keystroke**

# The default prefix, and the better prefix

Default prefix : C - b



# The default prefix, and the better prefix

Default prefix : C - b

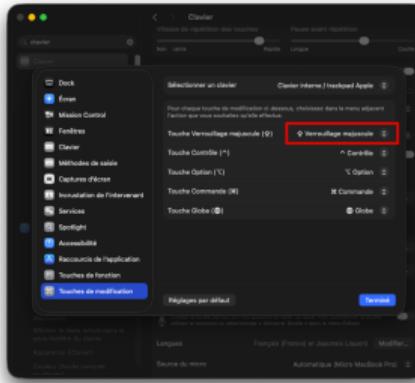


My recommandation : setup C- on Caps Lock (in your MacBook settings), and change it to C-a (on a QWERTY layout) or C-q (on a AZERTY layout)

# The better prefix: how to

Caps lock to Control :

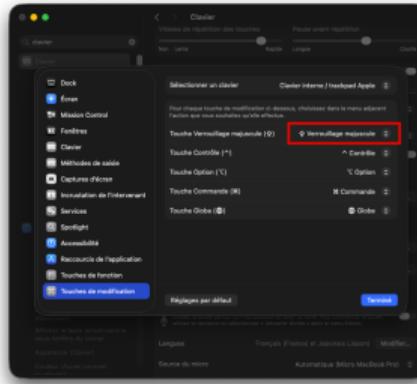
- ① Settings
- ② Keyboard
- ③ Keyboard shortcuts
- ④ Modifier keys



# The better prefix: how to

Caps lock to Control :

- ① Settings
- ② Keyboard
- ③ Keyboard shortcuts
- ④ Modifier keys



In tmux.conf:

```
set-option -g prefix C-a
unbind C-b
bind C-a send-prefix
```

In your command line :

- New session : tmux new-session
- Attach to an existing session : tmux attach -t
- List sessions : tmux list-sessions

Advice : create aliases in your .zshrc (tn, ta, tls, ...)

In your command line :

- New session : tmux new-session
- Attach to an existing session : tmux attach -t
- List sessions : tmux list-sessions

Advice : create aliases in your .zshrc (tn, ta, tls, ...)

**c** [c]reate a new window

" Horizontal split

% Vertical split

**d** [d]etach current session

# tmux commands : moving

---

## tmux commands : moving

---

**hjk1/←↓↑→** move from pane to pane (Vim logic for arrows)

## tmux commands : moving

**hjkl**/ $\leftarrow \downarrow \uparrow \rightarrow$  move from pane to pane (Vim logic for arrows)



Figure 1: hjkl and arrow keys on an old keyboard

## tmux commands : moving

**hjkl/←↓↑→** move from pane to pane (Vim logic for arrows)  
**q+1-9** display pane number, then move to another pane

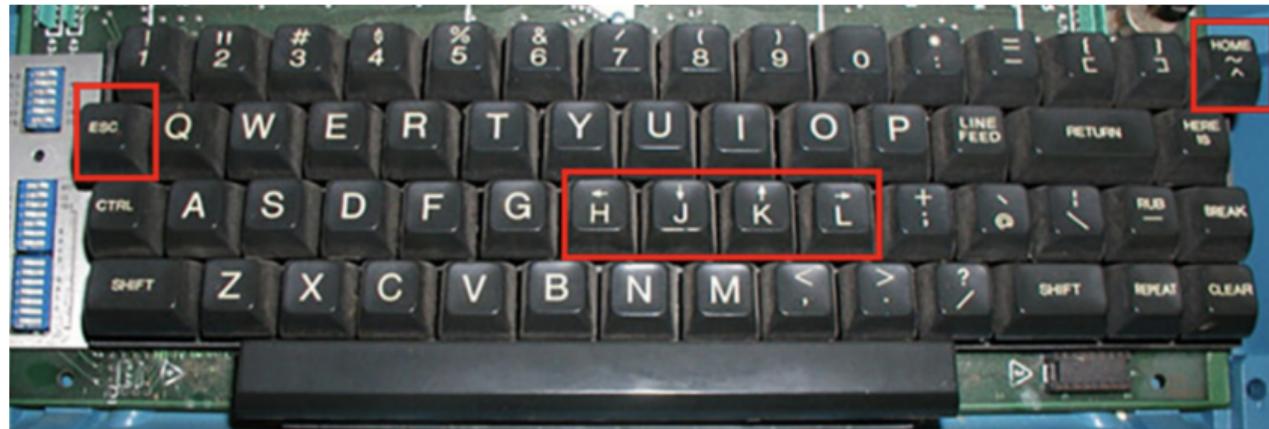


Figure 1: hjkl and arrow keys on an old keyboard

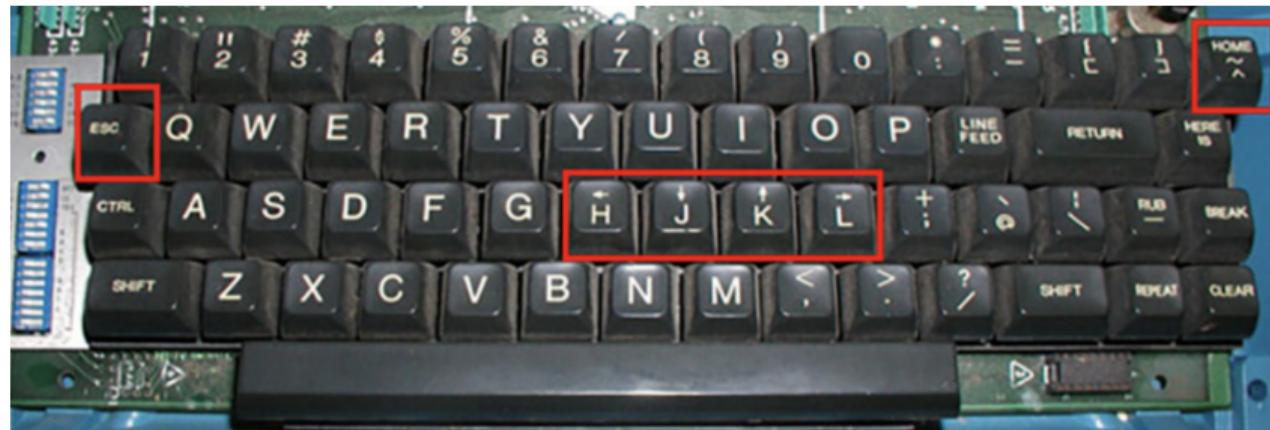
## tmux commands : moving

**hjkl/←↓↑→** move from pane to pane (Vim logic for arrows)

**q+1-9** display pane number, then move to another pane

**n/p** move to [n]ext/[p]revious window

**1-9** move to window of the corresponding number



**Figure 1:** hjkl and arrow keys on an old keyboard

## tmux commands : moving

**hjk1/←↓↑→** move from pane to pane (Vim logic for arrows)

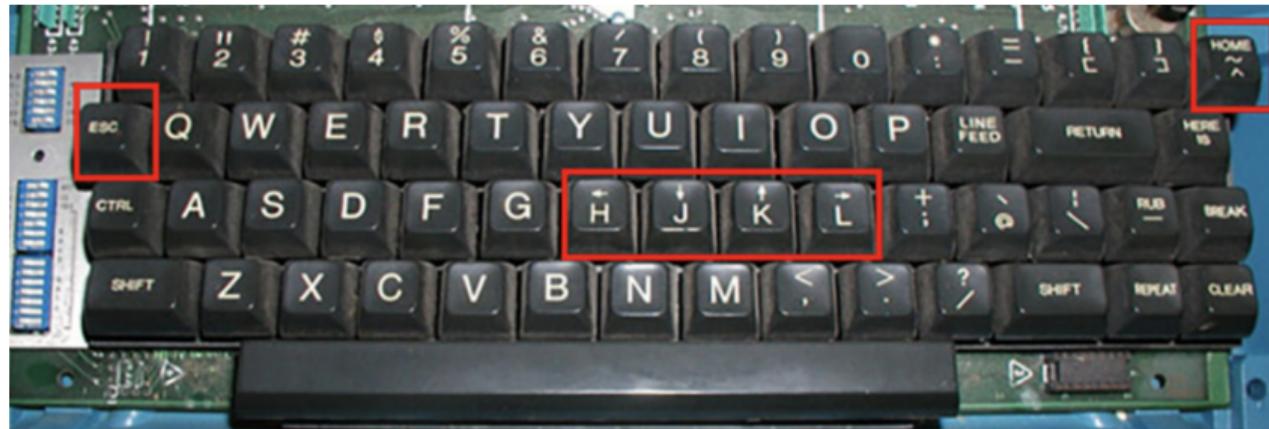
**q+1-9** display pane number, then move to another pane

**n/p** move to [n]ext/[p]revious window

**1-9** move to window of the corresponding number

**s** list [s]essions

**f** [f]ind among sessions



**Figure 1:** hjkl and arrow keys on an old keyboard

, rename current window

\$ rename current session

: open tmux's command line

? list all key binding

- , rename current window
- \$ rename current session
- : open tmux's command line
- ? list all key binding

How to change or add a keybinding : bind [key] [command]  
Example : bind - split-vertical

- **Enabling the mouse :** :set -g mouse on
- **Start windows indexs at 1 :** set -g base-index 1
- **Automatically renumber windows :** set -g renumber-windows on
- **Stop windows auto-renaming :** set -g allow-rename off

- **tpm** tmux plugin manager, mandatory
- **tmux-continuum+tmux-resurrect** allows saving and resurrection of sessions even when your machine shuts down
- **vim-tmux navigator** seamless navigation between tmux panes and vim panes
- lot of others plugins, and color themes (Catppuccin <3) !

## Summary and ressources

### General advices

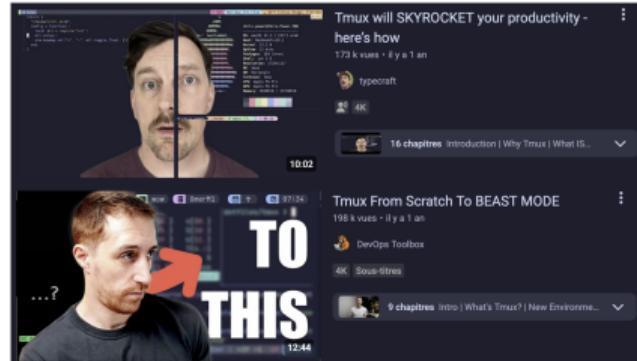
- *Start small!*
- Get used to tmux's philosophy and commands.
- Add relevant keybindings and plugins that suits your workflow over time.

## General advices

- *Start small!*
- Get used to tmux's philosophy and commands.
- Add relevant keybindings and plugins that suits your workflow over time.

Some useful tutos and ressources :

- tmux official startup guide
- Video tutorials : *Typecraft*, *DevOps toolbox*, *Henry Misc*, ...



## 1 Introduction

## 2 tmux as your terminal multiplexer

## 3 Keyboard tips

Touch-typing

Learn your keyboard shortcuts !

Text expander

Some more rabbit holes

## 1 Introduction

## 2 tmux as your terminal multiplexer

## 3 Keyboard tips

Touch-typing

Learn your keyboard shortcuts !

Text expander

Some more rabbit holes

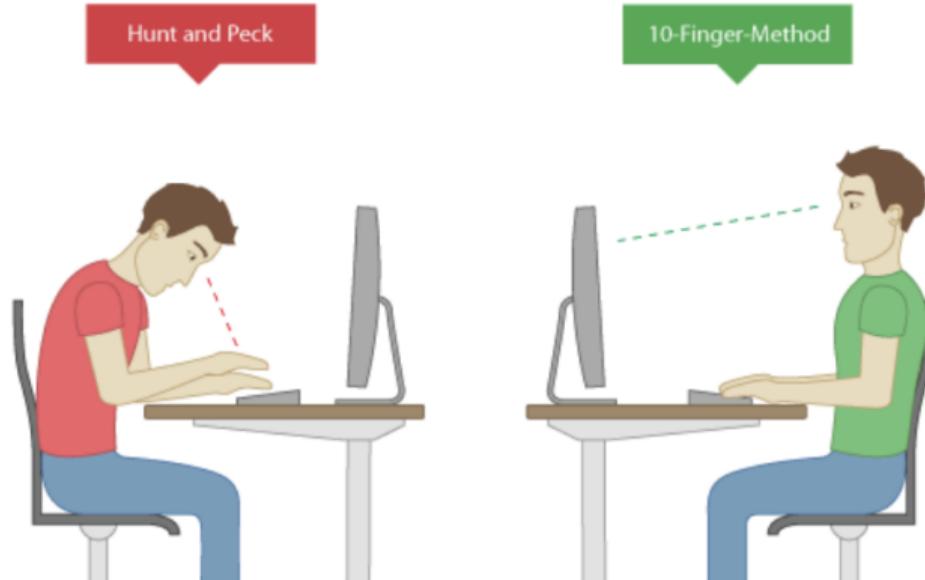
## *Hunt and peck vs. touch-typing*

---

## *Hunt and peck vs. touch-typing*

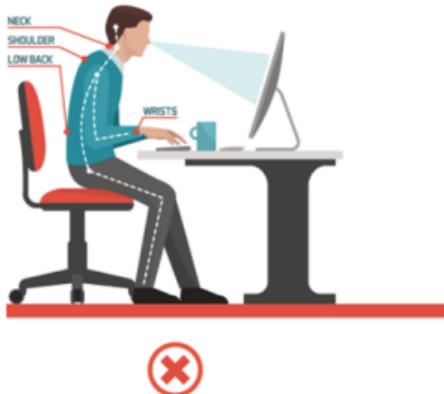
**hunt and peck** looking at your keys, and searching them with your eyes

**touch-typing** one key = one finger, typing without looking at your keyboard

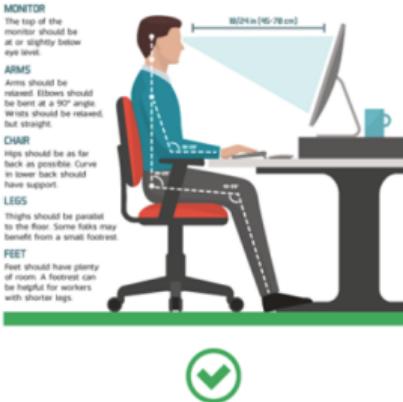


# Touch-typing : general posture

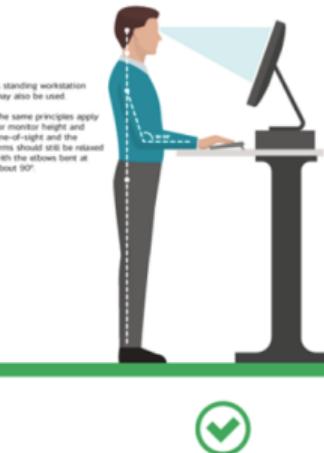
General rule of thumb : 90° angles



WRONG SITTING POSTURE



CORRECT SITTING POSITION



CORRECT STANDING POSITION

**MONITOR**  
The top of the monitor should be at or slightly below eye level.

**ARMS**  
Arms should be relaxed. Elbows should be bent at a 90° angle. Wrists should be relaxed, but straight.

**CHAIR**  
Hips should be as far back as possible. Curve in lower back should have support.

**LEGS**  
Knees should be parallel to the floor. Some folks may benefit from a small footrest.

**FEET**  
Feet should have plenty of room. A footrest can be helpful for workers with shorter legs.

A standing workstation may also be used.  
The same principles apply for standing height and line-of-sight and the arms should still be relaxed with the elbows bent at about 90°.

# Touch-typing : advices



- Type with your 10 fingers.
- One key = one finger.



- Type with your 10 fingers.
- One key = one finger.
- Always come back to the default position.



- Type with your 10 fingers.
- One key = one finger.
- Always come back to the default position.
- Never look at the keyboard, mask it if needed.



- Type with your 10 fingers.
- One key = one finger.
- Always come back to the default position.
- Never look at the keyboard, mask it if needed.
- Typo : use opt/ctrl+backspace.



- Type with your 10 fingers.
- One key = one finger.
- Always come back to the default position.
- Never look at the keyboard, mask it if needed.
- Typo : use opt/ctrl+backspace.
- **Accuracy over speed !!** (aim for  $\geq 96\%$  accuracy)

## How to practice touch-typing ?

---

On a day to day basis : apply previous slides principles.

# How to practice touch-typing ?

On a day to day basis : apply previous slides principles.

Websites to practice :

- ① **keybr** : to learn to touch-type with your keyboard layout



# How to practice touch-typing ?

On a day to day basis : apply previous slides principles.

Websites to practice :

- ① **keybr** : to learn to touch-type with your keyboard layout
- ② **monkeytype (<3)** : to improve your accuracy (first) and then your speed



# How to practice touch-typing ?

On a day to day basis : apply previous slides principles.

Websites to practice :

- ① **keybr** : to learn to touch-type with your keyboard layout
- ② **monkeytype (<3)** : to improve your accuracy (first) and then your speed



## 1 Introduction

## 2 tmux as your terminal multiplexer

## 3 Keyboard tips

Touch-typing

Learn your keyboard shortcuts !

Text expander

Some more rabbit holes

# MacOS usual keyboard shortcuts

cmd + ...

**H** hide app

**W** close window

**Q** quit app

## 1 Introduction

## 2 tmux as your terminal multiplexer

## 3 Keyboard tips

Touch-typing

Learn your keyboard shortcuts !

Text expander

Some more rabbit holes

# What is a text expander ?

## Text expander

A piece of software that expands text shortcuts snippets to longer text snippets.

You can see it as a system-wide code snippets !

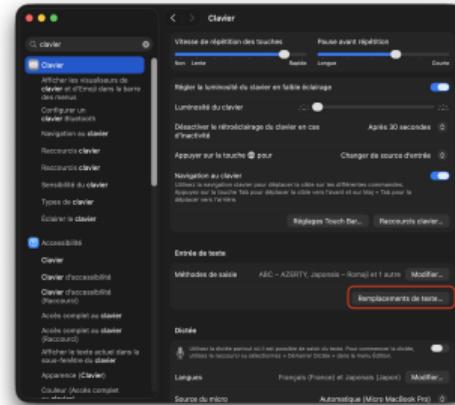
# What is a text expander ?

## Text expander

A piece of software that expands text shortcuts snippets to longer text snippets.

You can see it as a system-wide code snippets !

There is a default text expander available in MacOS, but quite limited.



- **Open-source**
- **Multi-platform** : Windows, Linux, MacOS
- **File-based** : all your shortcuts are consigned in .yml files
- **Powerful and modular**
- **Available on the ICM AppStore !**



- **Open-source**
- **Multi-platform** : Windows, Linux, MacOS
- **File-based** : all your shortcuts are consigned in .yml files
- **Powerful and modular**
- **Available on the ICM AppStore !**



Let's see some examples in action !

## Example of practical uses

- **ligatures** : -> = →
- **symbols** : :cmd:, :heart:
- **word shortcuts** : cdlt = cordialement
- **typos** : latex = LaTeX
- **custom infos** : :mpro = leo.guillon@mailo.com
- **automatic** : :today = current date
- **useless, but fun** : :flip, ...

- **ligatures** : -> = →
- **symbols** : :cmd:, :heart:
- **word shortcuts** : cdlt = cordialement
- **typos** : latex = LaTeX
- **custom infos** : :mpro = leo.guillon@mailo.com
- **automatic** : :today = current date
- **useless, but fun** : :flip, ...

For more options : community-driven packages !

## 1 Introduction

## 2 tmux as your terminal multiplexer

## 3 Keyboard tips

Touch-typing

Learn your keyboard shortcuts !

Text expander

Some more rabbit holes

# Making your keyboard more ergonomic

## Making your keyboard more ergonomic

---

Imagine that you had *both* Ctrl and esc on Caps Lock !

Imagine that you had *both* Ctrl and esc on Caps Lock !

## General idea with keyboard optimization

Putting useful keys, layers and shortcuts in more accessible positions

Imagine that you had *both* Ctrl and esc on Caps Lock !

## General idea with keyboard optimization

Putting useful keys, layers and shortcuts in more accessible positions

- **Tons of concepts** : angle-mod, layers and tap-layers, homerow mods, ...
- **Software** : Karabiner-Elements (MacOS), Kanata (multi-platform), ...
- **Inspiration** : Karabiner's modifications community, Arsenik (<3), ...

Imagine that you had *both* Ctrl and esc on Caps Lock !

## General idea with keyboard optimization

Putting useful keys, layers and shortcuts in more accessible positions

- **Tons of concepts** : angle-mod, layers and tap-layers, homerow mods, ...
- **Software** : Karabiner-Elements (MacOS), Kanata (multi-platform), ...
- **Inspiration** : Karabiner's modifications community, Arsenik (<3), ...



**Figure 2:** Arsenik keyboard, QWERTY layout, all options enabled

# Changing your keyboard layout

---

## Ergonomic keyboard layout

Putting the most frequent letters in your language under the most accessible keys according to the « default position »

## Ergonomic keyboard layout

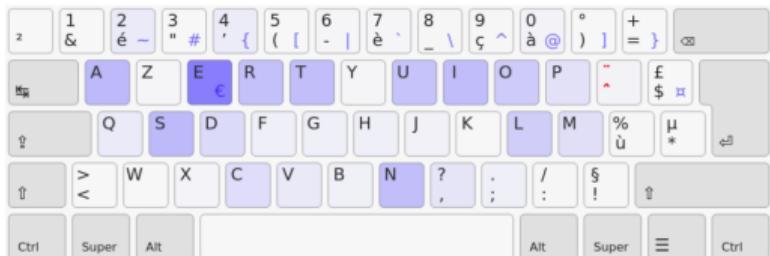
Putting the most frequent letters in your language under the most accessible keys according to the « default position »



**Figure 3:** Azerty layout french heatmap

## Ergonomic keyboard layout

Putting the most frequent letters in your language under the most accessible keys according to the « default position »



**Figure 3:** Azerty layout french heatmap

**English** Qwerty → Dvorak, Colemak, Workman, ...

**French** Azerty → Bépo → Ergo-L (<3 <3 <3)

## Ergonomic keyboard layout

Putting the most frequent letters in your language under the most accessible keys according to the « default position »



**Figure 3:** Azerty layout french heatmap



**Figure 4:** Ergo-L layout french heatmap

**English** Qwerty → Dvorak, Colemak, Workman, ...

**French** Azerty → Bépo → Ergo-L (<3 <3 <3)

# Switching to an ergonomic keyboard

Warning : **Ultra-deep endless rabbit hole!**

# Switching to an ergonomic keyboard

Warning : **Ultra-deep endless rabbit hole!**



# Conclusion

---

## Key points

- tmux is a nice tool to streamline your terminal workflow ;
- practicing your typing skill is an important asset as a dev ;
- lots of potential rabbit holes to fall into !

## Key points

- tmux is a nice tool to streamline your terminal workflow ;
- practicing your typing skill is an important asset as a dev ;
- lots of potential rabbit holes to fall into !

Config (tmux, espanso, karabiner, ...) : [github.com/LeoGuillon/dotfiles](https://github.com/LeoGuillon/dotfiles)  
Contact : [leo.guillon@mailo.com](mailto:leo.guillon@mailo.com)

Questions ?