

09/12/2025

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

*With some typing tips to skyrocket your  
development workflow*

Léo Guillon

## 1 Introduction

- 1 Introduction**
- 2 tmux as your terminal multiplexer**

- 
- 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 ?

---

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, Julia, Python
- Note-taking
- Document writing : L<sup>A</sup>T<sub>E</sub>X

## 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 ?

---

General idea : instead of working with single terminal, and nuking everything once you close it, you split it in multiple windows, as you can do in VSCode for example.

# 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

---

# tmux's hierarchy

---

pane a single terminal instance

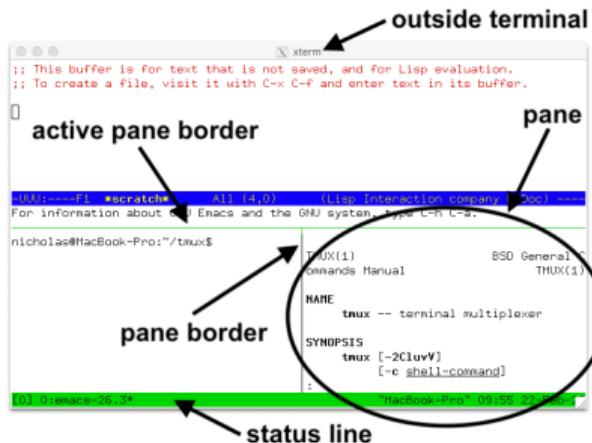
pane a single terminal instance

window a collection of panes, arranged in a defined layout

# tmux's hierarchy

**pane** a single terminal instance

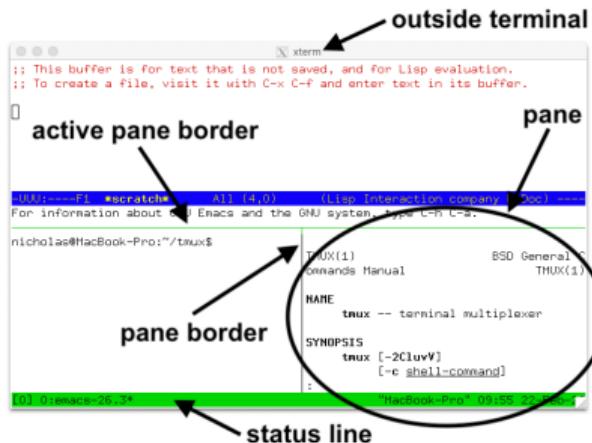
**window** a collection of panes, arranged in a defined layout



**pane** a single terminal instance

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

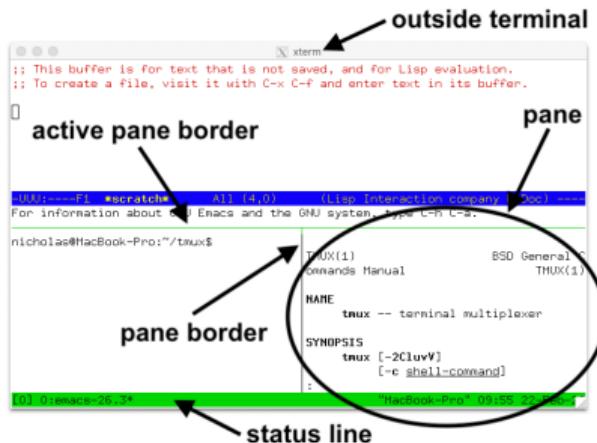
**session** a collection of windows



**pane** a single terminal instance

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

**session** a collection of windows



- 
- 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 !

## Example with my personal use

---

Use case for my current work :

### $\text{\LaTeX}$ custom commands

Writing a  $\text{\LaTeX}$  package for my custom mathematical commands, as well as a  $\text{\LaTeX}$  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

tmux works with a config file, `tmux.conf`, in your `/ .config/tmux` folder  
You can also split up the config as you want, to make it more modular

Different philosophy for commands : 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

Different philosophy for commands : 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 = 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 recommendation : 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 default prefix, and the better prefix

Default prefix : C - b



My recommendation : 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 :

Settings → keyboard → keyboard shortcuts → modifier keys

Caps lock to Control :

Settings → keyboard → keyboard shortcuts → modifier keys

In tmux.conf:

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

d Detach current session

” Horizontal split

Vertical split

c Create a new window

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

1–9 move to window of the corresponding number

n/b move to next/previous window

s list Sessions

, rename current window

\$ rename current session

: open tmux's command line

? list all key binding

How to change or add a keybinding : ...

- **Enabling the mouse :** :set -g mouse on
- **Start windows indexes 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 !

General advice : start small, and add relevant keybindings and plugins for your specific workflow.

## 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

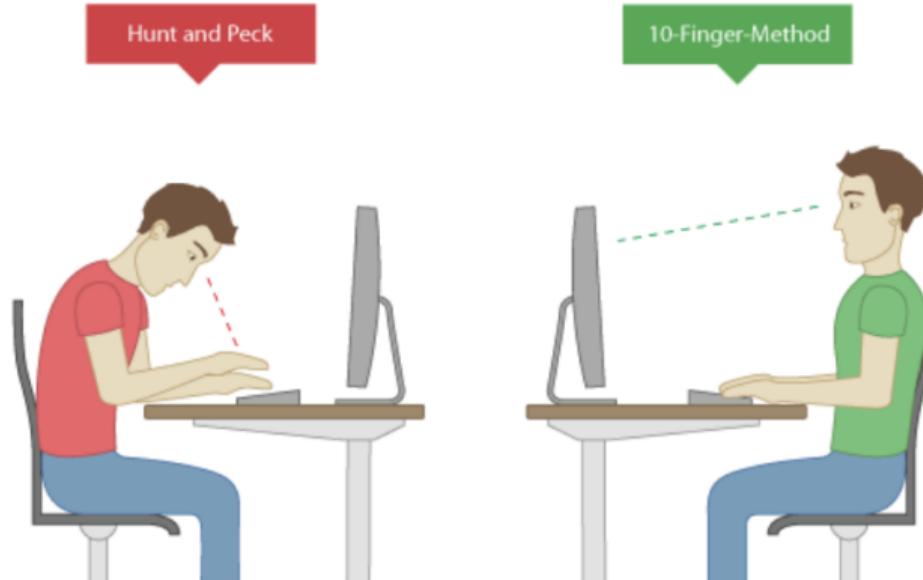
## *Hunt and peck vs. touch-typing*

---

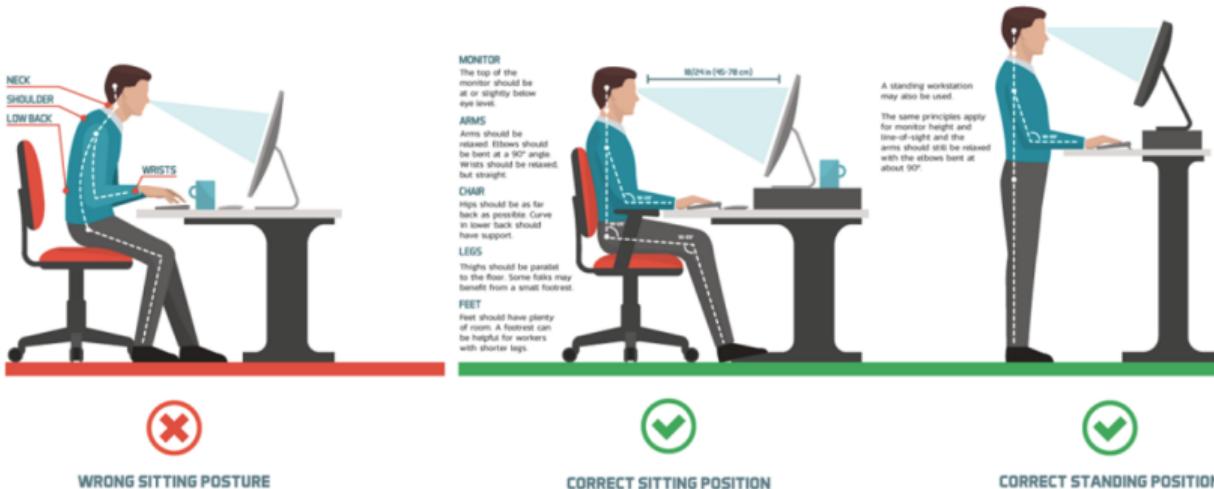
## *Hunt and peck vs. touch-typing*

**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



# Touch-typing : advices

---

- Type with your 10 fingers.

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

- Type with your 10 fingers.
- One key = one finger.
- Never look at the keyboard, mask it if needed

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

- Type with your 10 fingers.
- One key = one finger.
- Never look at the keyboard, mask it if needed
- Always come back to the default position
- **Accuracy over speed !!** Aim for  $\geq 96\%$  accuracy, speed will build up over time.

- Type with your 10 fingers.
- One key = one finger.
- Never look at the keyboard, mask it if needed
- Always come back to the default position
- **Accuracy over speed !!** Aim for  $\geq 96\%$  accuracy, speed will build up over time.
- Error : use opt+backspace (or ctrl+backspace on Windows) to delete an entire word.

## 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** : 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** : to improve your accuracy (first) and then your speed



Other alternatives : 10fastfingers, typingrace, ...

## 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

## 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 ?

---

# Espanso : an open-source text expander

## Example of practical uses

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

## 1 Introduction

## 2 tmux as your terminal multiplexer

## 3 Keyboard tips

Touch-typing

Learn your keyboard shortcuts !

Text expander

Some more rabbit holes

Useful keys closer to the default position : enter, backspace, etc.

Ctrl on Caps Lock is only the first step : imagine that you had both Ctrl *and* esc on Caps Lock.

Putting useful shortcuts

Tons of concepts : layer-taps, double-taps, homerow mods, tap-hold, etc.

Software and github repos : Karabiner (MacOS), Kanata (multi-platform), Arsenik/Miryoku/... for inspirations



**Figure 1:** Arsenik keyboard, all options enabled

## Changing your keyboard layout

---

General idea : putting the most frequent letters in your language under the most accessible keys according to the « default position »

# Changing your keyboard layout

---

General idea : putting the most frequent letters in your language under the most accessible keys according to the « default position »

English Qwerty → Dvorak, Colemak, ...

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

# Changing your keyboard layout

General idea : putting the most frequent letters in your language under the most accessible keys according to the « default position »

English Qwerty → Dvorak, Colemak, ...

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



**Figure 2:** Azerty layout heatmap

# Changing your keyboard layout

General idea : putting the most frequent letters in your language under the most accessible keys according to the « default position »

English Qwerty → Dvorak, Colemak, ...

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



**Figure 2:** Azerty layout heatmap



**Figure 3:** Ergo-L layout heatmap

# Switching to an ergonomic keyboard

---

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

# Conclusion

---