ARAMIS
LAB
BRAIN DATA SCIENCE

09/12/2025

tmux
an open-source terminal multiplexer

With some keyboard tips to skyrocket your development workflow

Léo Guillon

Presentation Outline

1. Introduction
2. `tmux` as your terminal multiplexer
3. Keyboard tips

Section outline

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

Who am I, and my general work philosophy

My work philosophy : use tools that are :

- **efficient**
- **minimalist**
- **open**

Examples of tools I use :

- Zen Browser
- Obsidian
- Sioyek
- Wezterm (terminal), with `neovim` (code), `yazi` (file explorer), `tmux`

Section outline

1. Introduction

2. tmux as your terminal multiplexer
   - Use cases
   - General workflow and vocabulary
   - Commands and setup

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 :

- are already used to the terminal and its commands ;
- use several command line interface (CLI) and terminal user interface (TUI) tools, and simultaneously ;
- make your terminal sessions persistant even when shutting down ;
- 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 !

tmux’s hierarchy

pane a single terminal instance

window a collection of panes, arranged in a defined layout (similar to a tab)

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 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

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

tmux’s commands startup point : the prefix key

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

Commands worth considering to begin with

d	Detach current session
c	Create a new window
C	Create a new session with a window
hjkl/←↓↑→	move from pane to pane (Vim logic for arrows)
1–9	move to window of the corresponding number
s	list Sessions
,	rename current window
$	rename current session
:	open `tmux`’s command line

Bonus setup : *plugins*

- **`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

My config is available at …

Section outline

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

Touch-typing : general posture

Touch-typing : advices

MacOS usual keyboard shortcuts

App-specific shortcuts

What is a text expander ?

Espanso : an open-source text expander

Some more rabbit holes

Section outline

1. Introduction

2. `tmux` as your terminal multiplexer

3. Keyboard tips

Conclusion