

Environnement Développement Personnalisé

pour les développeurs



15/02/2023 - Julien Rollin



Ruche 42 !



C'est quoi un environnement de développement ?

“

Integrated Development Environment (IDE)



Visual Studio Code



Sublime Text



ATOM

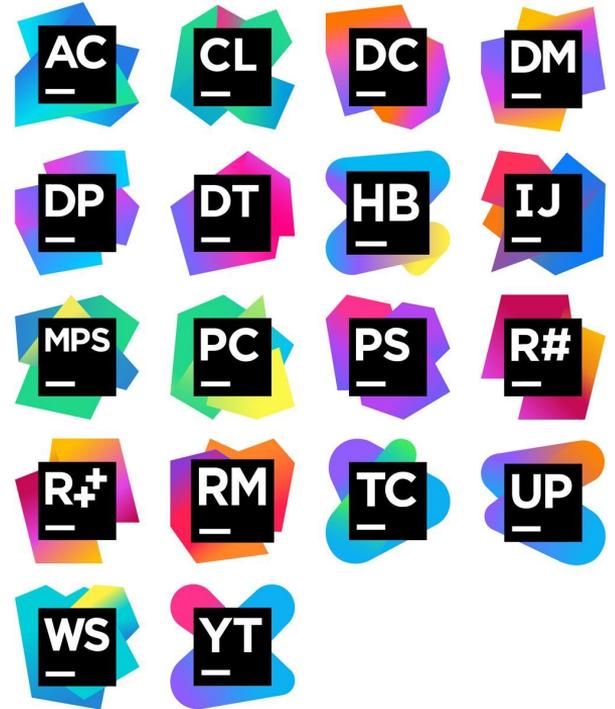


NetBeans

IDE



eclipse





***Qu'est-ce qui nous intéresse
pour coder ?***

“



Quelques fonctionnalités clés

- syntax highlighting
- indentation
- autocomplete
- brace matching

```
32 }
33
34 async fn handler() -> Html<&'static str> {
35     Html("Hello World !")
36 }
37 |
```

```
62 let~
63 } let~
64 let
65 let~
66 asy if let~
67 while let~
68 } if-let~
69 while-let~
70 asy LocalEnterGuard (use tokio::task::LocalEnterGuard)
71 let greeting = name.as_str() -> &str
```

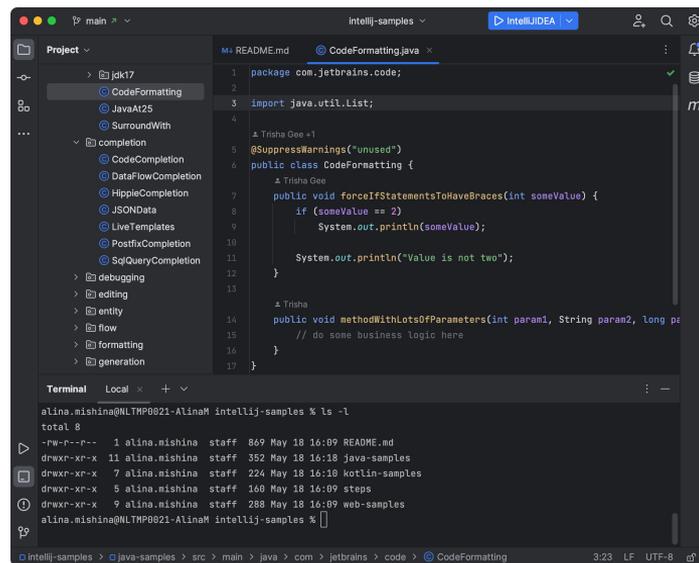
Syntax Error: expected SEMICOLON

let~ Snippet [LSP]
let~ Keyword [LSP]
let~ Snippet [snip]
if let~ Keyword [LSP]
while let~ Keyword [LSP]
if-let~ Snippet [snip]
while-let~ Snippet [snip]
LocalEnterGuard (use tokio::task::LocalEnterGuard) Struct [LSP]



Les IDE simplifient le process

- Graphical User Interface (GUI)
- Composants déjà configurés
- Build Automation Tools
- Interpréteur / compilateur
- Debugger





Oui mais...

“



Ce ne sont pas vos choix..

- souvent “vendor locked” par les choix des éditeurs
- une installation par langage
- rythme des mise à jour d’un plugin (délais dûs aux “surcouches”)

=> on s’adapte plus à l’outil que l’inverse..



*Une autre voie existe...
Suivez le guide !*

“

Vous avez dit Terminal ?



● Histoire des terminaux

- jusqu'en 1980 : le coût et la place requise pour un ordinateur nécessite de séparer unité centrale et périphériques dans des pièces distinctes
- terminal : ordinateur sans mémoire ni capacité de stockage et avec juste une carte réseau



[Ordinateur central IBM 702](#)





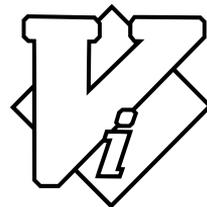
Evolution des terminaux

- les ordinateurs sont de plus en plus petits, plus besoin de séparer les terminaux
- création de système d'exploitation comme unix avec de l'abstraction
- le terminal devient un programme



Evolution de Vi

- Vi => 1976, par Bill Joy (révolution : usage de l'ensemble de l'écran)
- Vim => 1.0 en 1988 Bram Moolenaar (portage en C pour Amiga, Vi improved)
- NeoVim => 2014, idée simple : créer un Vim plus moderne, extensible, et communautaire (v0.8 fin 2022)





Les terminaux sont partout !

- il y a un terminal sur tous les OS
- robustes et éprouvés
- souvent le seul programme disponible sur un serveur distant
- léger par sa conception



Mais pourquoi ?

“

Google

🔍 how to quit



🔍 how to quit **vim**

🔍 how to quit **telnet**

🔍 how to quit **smoking**

🔍 how to quit **vi**

🔍 how to quit **nano**

🔍 how to quit **skyrim**

🔍 how to quit **twitter**

Running joke.. ou pas



● Vim multimodal

- Opérateurs
 - Normal
 - Visuel
 - Insertion
 - Commande

- Mouvements

```
{OPERATOR}{COUNT}{MOTION}
d2w
DELETE 2 WORDS
```



La souris : un faux ami ?

- coder c'est manipuler beaucoup de texte
- éliminer les aller retours à la souris est plus efficace
- les outils en ligne de commandes sont textuels, facile à développer et maintenir

*Ok, mais c'est pas vraiment
pratique..*

Il me manque des choses !



“

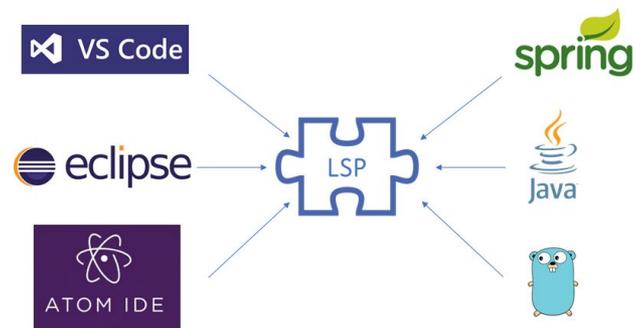
The Game changers !

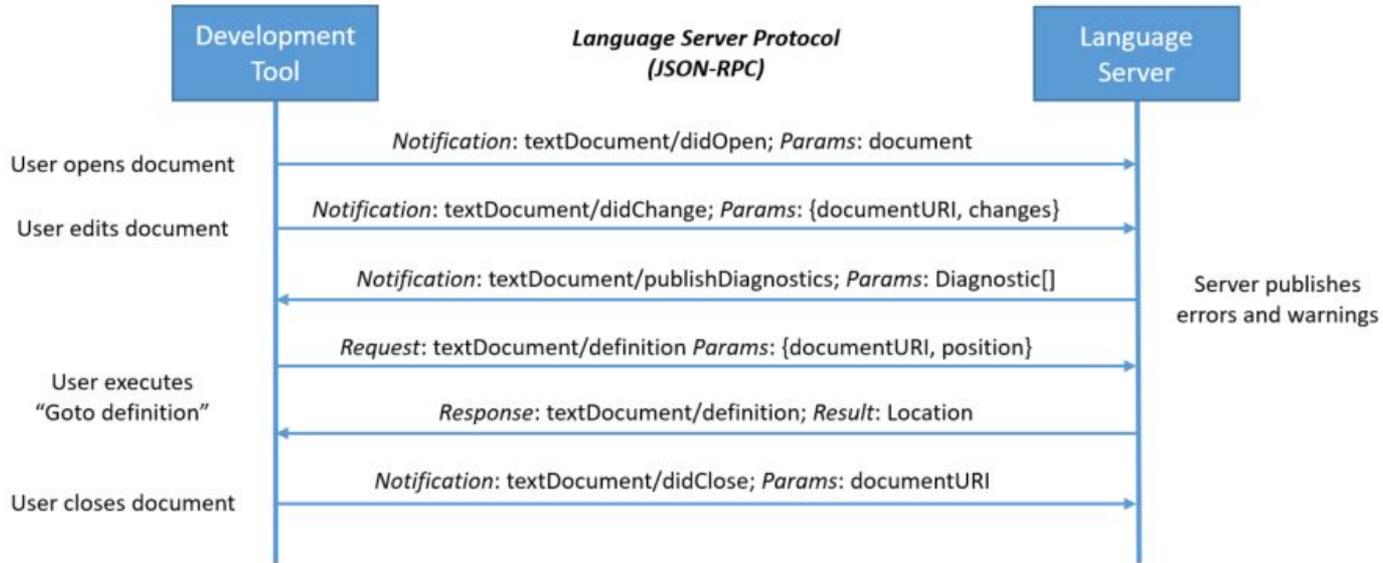




Language Server Protocol (LSP)

- Créé par Microsoft pour Vscode en 2017
- Open Standard sur Github
- Définit un protocole standard JSON-RPC d'intercommunication entre un client et un "language server"
- Support facilité des fonctionnalités
 - autocomplete
 - go to definition
 - code diagnostic
 - find references
 - etc





LSP protocole





Debug Adapter Protocol (DAP)

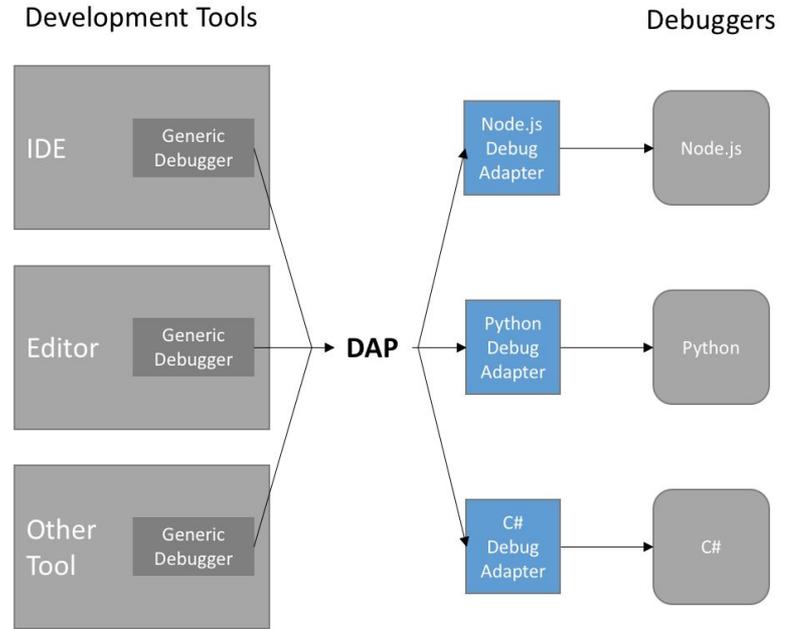
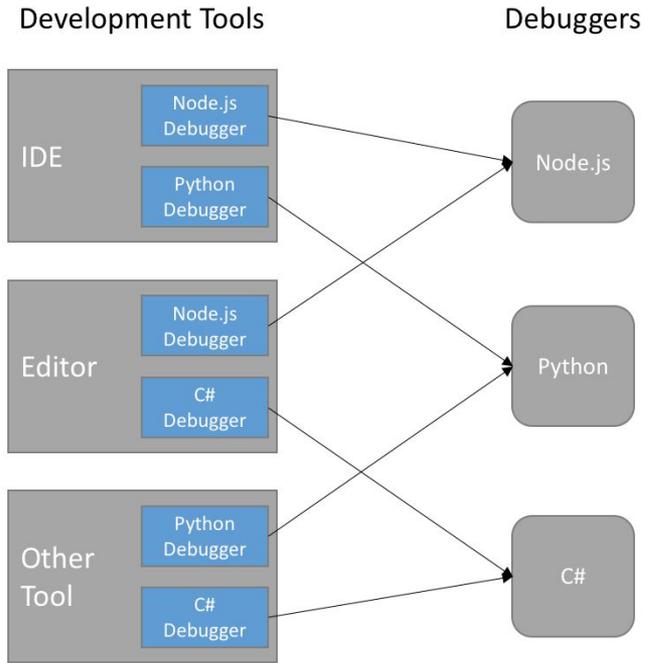
- ◉ projet porté par Microsoft
- ◉ abstraction de la communication des outils avec les debugger / runtime via un protocole
- ◉ permet de d'implémenter un debugger générique qui peut communiquer avec des debuggers via des adaptateurs
- ◉ réduit le temps d'implémentation pour les clients et fournisseurs !

```
14 →   console.log(`worker ${worker.process.pid} died`);  
15 →   });  
16 → } else {  
17 →   // Workers can share any TCP connection  
18 →   // In this case it is an HTTP server  
19 →   http.createServer((req, res) => {  
20 →     res.writeHead(200);  
21 →     res.end('hello world\n');  
22 →   }).listen(8000);  
23 → }
```

DEBUG CONSOLE

- writable
- write
- writeContinue
- writeHead
- writeHeader
- writeRaw

> res.wri



Debug Adapter Protocol





Pimp my Neovim !

“



Lua : un langage simple et puissant

```
function map(mode, shortcut, command)
  vim.api.nvim_set_keymap(mode, shortcut, command,
    { noremap = true, silent = true })
end

map('', '<C-D>', ':Telescope find_files<CR>')
map('', '<C-F>', ':Telescope grep_string<CR>')
map('', '<C-X>', ':NvimTreeToggle<CR>')
```



nginx, apache, jeux, etc



Packer : plugin manager

- Développé et configuré en Lua
- Gère les dépendances
- Beaucoup d'options d'installation
- Installations au travers de tâches asynchrones
- Support de Git branch/ tag

```
}  
-- LSP  
use({ -- LSP Configuration & Plugins  
  "neovim/nvim-lspconfig",  
  requires = {  
    -- Automatically install LSPs to stdpath  
    "williamboman/mason.nvim",  
    "williamboman/mason-lspconfig.nvim",  
  },  
})  
-- autocomplete and snippets  
use({  
  "hrsh7th/nvim-cmp"
```



UI des plugins : Mason

```
- headers
mason.nvim
press g? for help
https://github.com/williamboman/mason.nvim
(1) ALL (2) LSP (3) DAP (4) Linter (5) Formatter
Language Filter: press <C-f> to apply filter

Installed (14)
  clang
  codeLLdb
  cpplint
  cpptools
  css-lsp    new version available: vscode-langservers-extracted 4.5.0 -> 4.6.0
  elm-format
  elm-language-server
  gopls
  html-lsp   new version available: vscode-langservers-extracted 4.5.0 -> 4.6.0
  lua-language-server  new version available: sumneko/vscode-lua v3.6.8 -> v3.6.10
  prettier   new version available: prettier 2.8.3 -> 2.8.4
  rust-analyzer  new version available: rust-lang/rust-analyzer 2023-01-30 -> 2023-02-06
  tailwindcss-language-server  new version available: @tailwindcss/language-server 0.0.10 -> 0.0.11
  typescript-language-server  new version available: typescript-language-server 3.0.3 -> 3.1.0

Available (259)
  actionlint
  ada-language-server
  alex
  angular-language-server
  ansible-language-server
  apex-language-server
  arduino-language-server
  asm-lsp
```



Telescope : Smart File Explorer

The screenshot displays the Telescope IDE interface. On the left, a file explorer shows a project structure with files like `src/main.rs`, `README.md`, `Cargo.toml`, `Cargo.lock`, `.gitignore`, `.github/workflows/ci.yml`, and `.env.template`. The `Cargo.toml` file is selected. The main area is split into two panes: 'Results' and 'File Preview'. The 'File Preview' pane shows the contents of `Cargo.toml`:

```
[package]
name = "oodini"
version = "0.1.0"
edition = "2021"

# See more keys and their definitions at https://doc.rust-lang.org/cargo/reference/manifest.html

[dependencies]
axum = "0.6.4"
dotenv = "0.15.0"
tokio = { version = "1.25.0", features = ["full"] }
```

At the bottom of the interface, there is a search bar containing `>[]` and a page indicator showing `7 / 7`. The status bar at the very bottom shows the number `51`.



Tree Sitter : Syntax Parser

- Générique, peut parser tous les langages
- Suffisamment rapide pour parser à chaque frappe
- Robuste et fiable même en cas d'erreur de syntaxe
- Pas de dépendance (pur C)



```
line_comment [17, 4] - [17, 32]
let_declaration [18, 4] - [18, 70]
  pattern: identifier [18, 8] - [18, 12]
  type: primitive_type [18, 14] - [18, 17]
  value: call_expression [18, 20] - [18, 69]
    function: field_expression [18, 20] - [18, 39]
      value: call_expression [18, 20] - [18, 32]
        function: field_expression [18, 20] - [18, 30]
          value: identifier [18, 20] - [18, 24]
            field: field_identifier [18, 25] - [18, 30]
              arguments: arguments [18, 30] - [18, 32]
            field: field_identifier [18, 33] - [18, 39]
              arguments: arguments [18, 39] - [18, 69]
              string_literal [18, 40] - [18, 68]
          line_comment [19, 4] - [19, 41]
        let_declaration [20, 4] - [23, 55]
          pattern: identifier [20, 8] - [20, 11]
          value: call_expression [20, 14] - [23, 54]
            function: field_expression [20, 14] - [23, 14]
              value: call_expression [20, 14] - [22, 50]
                function: field_expression [20, 14] - [22, 14]
                  value: call_expression [20, 14] - [21, 33]
                    function: field_expression [20, 14] - [21, 14]
                      value: call_expression [20, 14] - [20, 27]
                        function: scoped_identifier [20, 14] - [20, 25]
                          path: identifier [20, 14] - [20, 20]
                          name: identifier [20, 22] - [20, 25]
                        arguments: arguments [20, 25] - [20, 27]
                        field: field_identifier [21, 9] - [21, 14]
                      arguments: arguments [21, 14] - [21, 33]
                      string_literal [21, 15] - [21, 18]
                      call_expression [21, 20] - [21, 32]
                        function: identifier [21, 20] - [21, 23]
                        arguments: arguments [21, 23] - [21, 32]
                          identifier [21, 24] - [21, 31]
                      field: field_identifier [22, 9] - [22, 14]
                    arguments: arguments [22, 14] - [22, 50]
                    string_literal [22, 15] - [22, 29]
                    call_expression [22, 31] - [22, 49]
                      function: identifier [22, 31] - [22, 34]
                        response::html, IntoResponse,
                        routing::get,
                        Router,
10 };
11
12 #[tokio::main]
13 async fn main() {
14     // loads the environment variables from the ".env" file.
15     dotenv().ok();
16     // get listening port
17     let port = std::env::var("PORT").unwrap_or("3000".to_string);
18     // ensure port is valid type
19     let port: u16 = port.parse().expect("Port should be valid");
20     // build our application with a route
21     let app = Router::new() => Router
22         .route("/", get(handler))
23         .route("/hello/:name", get(hello_handler))
24         .route("/status/:status", get(status_handler));
25     // run it
26     let addr = SocketAddr::from(([127, 0, 0, 1], port)); => SocketAddr
27     println!("listening on {addr}");
28     axum::Server::bind(&addr)
29         .serve(app.into_make_service())
30         .await
31         .unwrap();
32 }
33
34 async fn handler() -> Html {
35     Html("Hello World!")
36 }
37
38 async fn hello_handler(Path(name): Path<String>) -> impl IntoResponse {
39     let greeting = name.as_str(); => &str
40     let hello = String::from("Hello "); => String
41
42     (StatusCode::OK, Html(hello + greeting))
43 }
44
45 async fn status_handler(Path(name): Path<String>) -> impl IntoResponse {
46     match name.as_str() {
```

TreeSitter Playground



Au-delà de l'éditeur ?





Workspaces : Tri mental

- un espace mental par workspace
 - ex: i3 ou autre workspace
- des projets par “session”
 - ex: tmux, zellij, etc
- granularité plus fine avec les “onglets”
 - ex: tmux pane, vim pane



i3

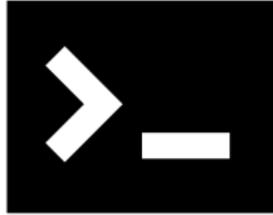
improved tiling wm



tmux



neovim

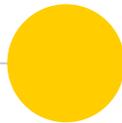


Démo !





*Neovim n'est pas la
réponse à la grande
question sur la vie,
l'univers et le reste...*





Je fonce ou pas ?

“



Pour qui ?

- ⦿ comprendre comment ça marche
- ⦿ adapter les outils plutôt que l'inverse
- ⦿ utiliser sa mémoire musculaire
=> oublier la souris



Freins ?

- ⦿ désapprendre la souris
- ⦿ progression pas facile... mais progressive !
- ⦿ les habitudes sont coriaces...
- ⦿ ne pas abandonner



Une communauté immense et très active

- vim extension browser
- vim keys : extension vscode / idea
- l'aide intégrée de neovim :)



Vimium



IdeaVim



VSCodeVim



R/unixporn - the home for *NIX

The image displays a collage of terminal windows and system monitoring dashboards, showcasing various Linux distributions and system configurations.

Top Left: neofetch Output

```
ax1thinkpad
OS: Arch Linux x86_64
Host: 90URUNME-ThinkPad T480s
Kernel: 6.1.0-arch1-1
Uptime: 16 hours
Packages: 1325 (pacman)
Shell: bash 5.1.16
Resolution: 1920x1080
DE: Hyprland
WM: Catppuccin-Mocha-Standard-Peach-Dark [GTK2/3]
Icons: Papirus-Dark [GTK2/3]
Terminal: alacritty
Terminal Font: Fira Code
CPU: Intel i7-9750H (4) @ 4.00GHz
GPU: Intel UHD Graphics G5
```

Top Right: System Status Dashboard

stop for archlinux Load Average: 2.94 2.14 1.04 21:21:22
CPU usage: 99%

Memory usage: 50%
Process list:

Command	CPU %	Count	Memory %
node	3.4	1	1.2
kitty	3.1	14	17.4
firefox	2.6	1	8.7
bash	2.0	1	0.0

Middle Left: File Manager

```
source ~/.config/hypr/mocha.conf
# For all categories, see https://wiki.hyprland.org/Configuring
input {
  kb_layout = us
  kb_variant =
  kb_model =
  kb_options =
  kb_rules =
  follow_mouse = 1
  touchpad {
    natural_scroll = 1
    clickfinger_behavior = 1
    tap-to-click = 1
    middle_button_emulation = 1
  }
  sensitivity = 8 * 1.0 - 1.0, 1.0, 8 means no modification.
}
general {
  @ see https://wiki.hyprland.org/Configuring/Variables/ for more
  gaps_in = 3
  gaps_out = 10
  border_size = 2
  collective_border = 0
  col_inactive_border = rgb(50,50,50)
  layout = master
}
[[type]]
url = utf-8 | conf | 20x | 3 | 10 | 8
```

Middle Right: System Status Dashboard

stop for archlinux Load Average: 2.94 2.14 1.04 21:21:22
CPU usage: 99%

Memory usage: 50%
Process list:

Command	CPU %	Count	Memory %
node	3.4	1	1.2
kitty	3.1	14	17.4
firefox	2.6	1	8.7
bash	2.0	1	0.0

Bottom Left: File Manager

```
source ~/.config/hypr/mocha.conf
# For all categories, see https://wiki.hyprland.org/Configuring
input {
  kb_layout = us
  kb_variant =
  kb_model =
  kb_options =
  kb_rules =
  follow_mouse = 1
  touchpad {
    natural_scroll = 1
    clickfinger_behavior = 1
    tap-to-click = 1
    middle_button_emulation = 1
  }
  sensitivity = 8 * 1.0 - 1.0, 1.0, 8 means no modification.
}
general {
  @ see https://wiki.hyprland.org/Configuring/Variables/ for more
  gaps_in = 3
  gaps_out = 10
  border_size = 2
  collective_border = 0
  col_inactive_border = rgb(50,50,50)
  layout = master
}
[[type]]
url = utf-8 | conf | 20x | 3 | 10 | 8
```

Bottom Right: System Status Dashboard

stop for archlinux Load Average: 2.94 2.14 1.04 21:21:22
CPU usage: 99%

Memory usage: 50%
Process list:

Command	CPU %	Count	Memory %
node	3.4	1	1.2
kitty	3.1	14	17.4
firefox	2.6	1	8.7
bash	2.0	1	0.0



Questions ? Suggestions ?

“



Ressources utiles

[Neovim website](#)

[Neovim kickstart config](#)

[TJ DE vries - Youtube](#)

[TJ De Vries - Personalized Development Environment](#)

[The Primeagen - Youtube](#)

[The valuable dev](#)



Terminal resources

[A Guide to the Terminal, Console, and Shell](#)

[TTY: under the hood](#)

[Programming TTY](#)

Environnement Développement Personnalisé

♥ Merci !

*N'hésitez pas à laisser des
commentaires !*

