

# NEW - Prise en main depuis Linux

🕒 Ce tutoriel suit le processus d'installation de ZSH, OH-MY-ZSH, Anaconda/MiniConda, Poetry, Pyenv et la configuration de VSCode

CONFIGURATION UTILISÉE : Debian GNU/Linux 12 (bookworm)

## Index

1. [Installation de VIM](#)
2. [Installation de ZSH et Oh-My-ZSH](#)
3. [Installation de Pyenv](#)
4. [Installation de Poetry](#)
5. [Installation de Anaconda](#)
6. [Installation de Miniconda](#)
7. [Installation de VSCode](#)

## Installation de VIM

Si VIM n'est pas déjà présent sur votre machine, installez le avec la commande:

```
sudo apt-get install vim
```

## Installation de ZSH et Oh-My-ZSH

Pour comprendre ce qu'est ZSH et Oh-My-ZSH, connaître leur spécificités et leurs améliorations qu'ils apportent, je vous renvoie aux vidéos dédiées dans la partie "Se familiariser avec le terminal".

### ZSH

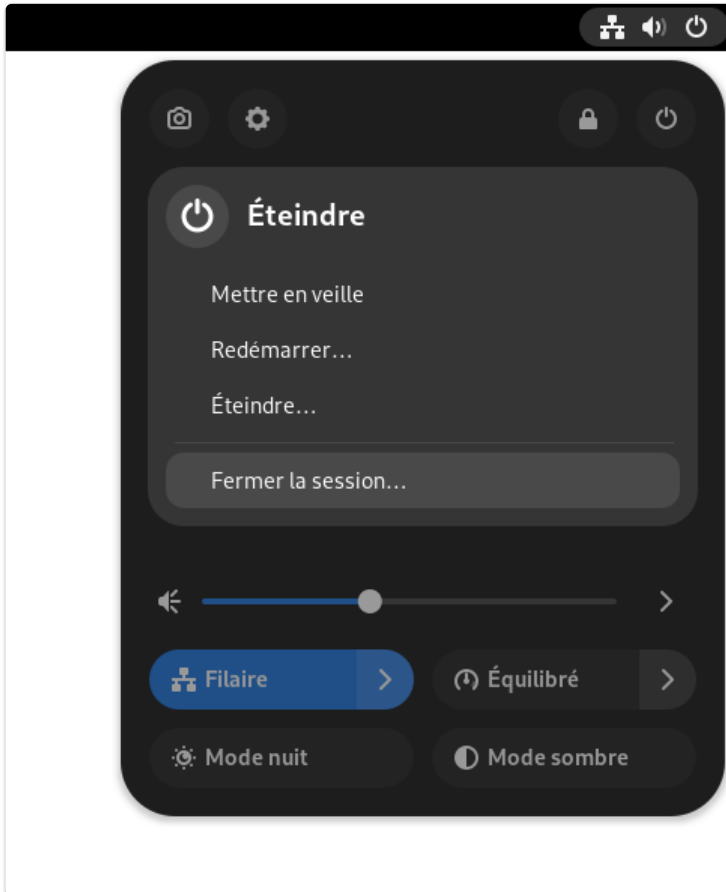
Dans votre terminal, vous allez taper cette commande pour installer zsh, qui est un terminal plus avancé que bash :

```
sudo apt update && sudo apt install zsh -y
```

Pour mettre ZSH en tant que terminal par défaut, il faudra taper:

```
chsh -s $(which zsh)
```

Puis vous devez vous déconnecter de votre machine (Fermer la session), et vous reconnecter de nouveau. Cela dans le but de prendre en compte les modifications de changement de terminal par défaut.



Ensuite relancez le terminal, et cette fois-ci c'est ZSH qui s'ouvre, et qui vous demande de configurer votre environnement ZSH :

```
Terminal
This is the Z Shell configuration function for new users,
zsh-newuser-install.
You are seeing this message because you have no zsh startup files
(the files .zshenv, .zprofile, .zshrc, .zlogin in the directory
~). This function can help you with a few settings that should
make your use of the shell easier.

You can:

(q) Quit and do nothing. The function will be run again next time.

(0) Exit, creating the file ~/.zshrc containing just a comment.
    That will prevent this function being run again.

(1) Continue to the main menu.

(2) Populate your ~/.zshrc with the configuration recommended
    by the system administrator and exit (you will need to edit
    the file by hand, if so desired).

--- Type one of the keys in parentheses --- |
```

Tapez `0` pour créer le fichier de configuration `.zshrc`

#### Info

Pour savoir quel est le terminal par défaut utilisé, tapez la commande `echo $SHELL`. Cela affichera quelque chose comme ça pour bash : `usr/bin/bash` et quelque chose comme ça pour zsh : `usr/bin/zsh`

Pour savoir quel est le terminal courant, tapez `echo $0`, qui affichera `zsh` ou `bash`

## OH-MY-ZSH

[Documentation officielle](#)

Installons maintenant oh-my-zsh (depuis le terminal Linux):

```
sh -c "$(curl -fsSL
https://raw.githubusercontent.com/ohmyzsh/ohmyzsh/master/tools/install.
sh)"
```

```
antonin@DESKTOP-ASSH4VC:~  
DESKTOP-ASSH4VC% sh -c "$(curl -fsSL https://raw.githubusercontent.com/ohmyzsh/ohmyzsh/master/tools/install.sh)"  
Cloning Oh My Zsh...  
remote: Enumerating objects: 1431, done.  
remote: Counting objects: 100% (1431/1431), done.  
remote: Compressing objects: 100% (1368/1368), done.  
remote: Total 1431 (delta 44), reused 1021 (delta 35), pack-reused 0 (from 0)  
Réception d'objets: 100% (1431/1431), 3.27 Mio | 11.22 Mio/s, fait.  
Résolution des deltas: 100% (44/44), fait.  
Depuis https://github.com/ohmyzsh/ohmyzsh  
* [nouvelle branche] master -> origin/master  
la branche 'master' est paramétrée pour suivre 'origin/master'.  
Déjà sur 'master'  
/home/antonin  
  
Looking for an existing zsh config...  
Found /home/antonin/.zshrc. Backing up to /home/antonin/.zshrc.pre-oh-my-zsh  
Using the Oh My Zsh template file and adding it to /home/antonin/.zshrc.  
  
.....is now installed!  
  
Before you scream Oh My Zsh! look over the '.zshrc' file to select plugins, themes, and options.  
  
• Follow us on X: @ohmyzsh  
• Join our Discord community: Discord server  
• Get stickers, t-shirts, coffee mugs and more: Planet Argon Shop  
  
-> ~
```

Cela va ajouter la configuration OH-MY-ZSH dans le fichier `.zshrc`.

Maintenant, dès que vous ouvrirez le terminal, vous aurez un terminal ZSH avec OH-MY-ZSH.

### 🔥 Important

Pour éviter des erreurs de manipulations, faites une sauvegarde du fichier `.zshrc` dans un autre fichier (à chaque fois que vos modifications sont stables et vérifiées), par exemple dans le fichier `.zshrc_backup` en utilisant la commande suivante: `cp .zshrc .zshrc_backup` (depuis le terminal linux)  
Cela permettra de revenir à un état stable du fichier

## Plugins oh-my-zsh

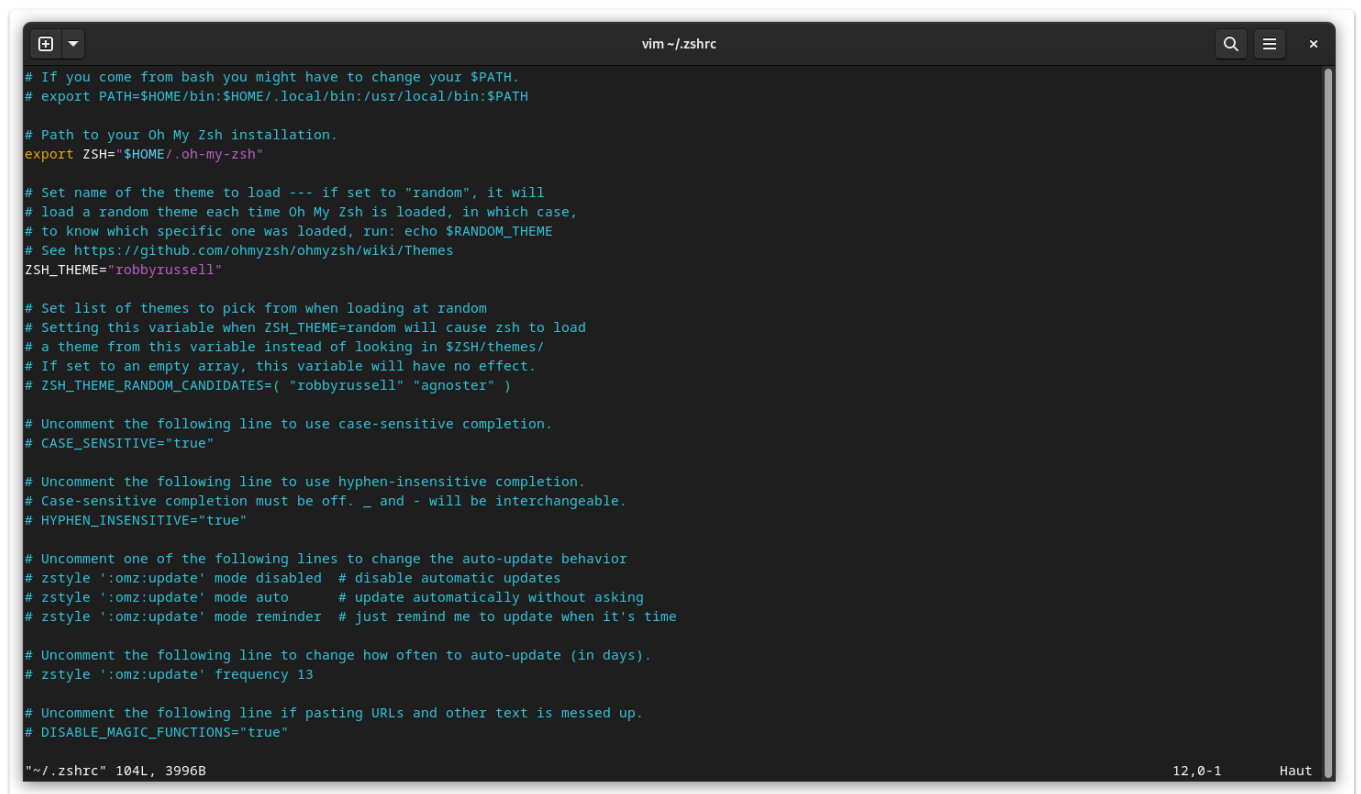
Oh-my-zsh vient avec un grand nombre de plugins très utiles, comme par exemple pour git, vscode, poetry ou conda.

Pour ajouter un ou plusieurs plugins on va modifier le fichier `.zshrc` qui contient notamment la configuration de oh-my-zsh.

**Ouvrez le fichier `.zshrc` avec Vim** (Pour plus de détails sur l'utilisation de VIM, je vous renvoie à la vidéo dédiée) :

```
vim ~/.zshrc
```

Le fichier va s'ouvrir dans votre terminal :



```
vim ~/.zshrc

# If you come from bash you might have to change your $PATH.
# export PATH=$HOME/bin:$HOME/.local/bin:/usr/local/bin:$PATH

# Path to your Oh My Zsh installation.
export ZSH="$HOME/.oh-my-zsh"

# Set name of the theme to load --- if set to "random", it will
# load a random theme each time Oh My Zsh is loaded, in which case,
# to know which specific one was loaded, run: echo $RANDOM_THEME
# See https://github.com/ohmyzsh/ohmyzsh/wiki/Themes
ZSH_THEME="robbyrussell"

# Set list of themes to pick from when loading at random
# Setting this variable when ZSH_THEME=random will cause zsh to load
# a theme from this variable instead of looking in $ZSH/themes/
# If set to an empty array, this variable will have no effect.
# ZSH_THEME_RANDOM_CANDIDATES=( "robbyrussell" "agnoster" )

# Uncomment the following line to use case-sensitive completion.
# CASE_SENSITIVE="true"

# Uncomment the following line to use hyphen-insensitive completion.
# Case-sensitive completion must be off. _ and - will be interchangeable.
# HYPHEN_INSENSITIVE="true"

# Uncomment one of the following lines to change the auto-update behavior
# zstyle ':omz:update' mode disabled # disable automatic updates
# zstyle ':omz:update' mode auto # update automatically without asking
# zstyle ':omz:update' mode reminder # just remind me to update when it's time

# Uncomment the following line to change how often to auto-update (in days).
# zstyle ':omz:update' frequency 13

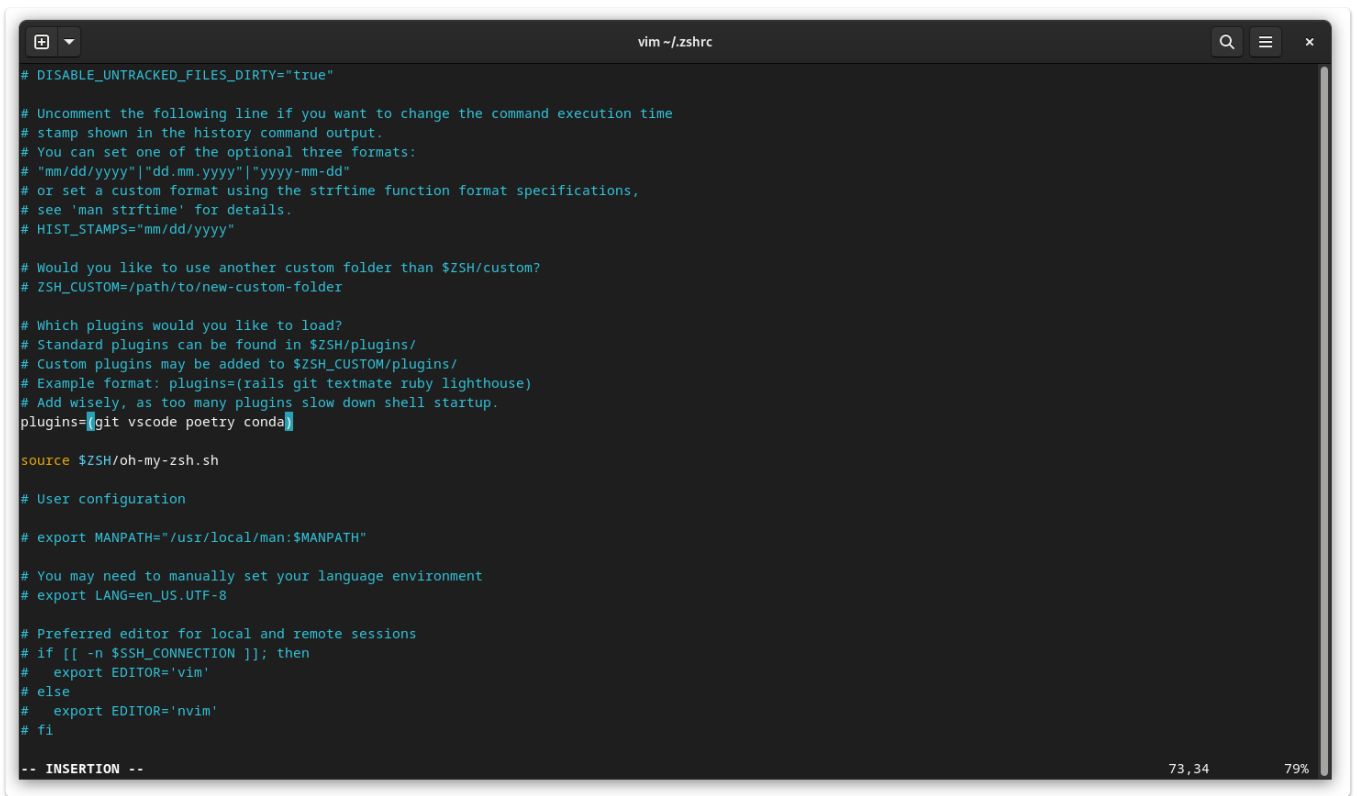
# Uncomment the following line if pasting URLs and other text is messed up.
# DISABLE_MAGIC_FUNCTIONS="true"

"~/.zshrc" 104L, 3996B 12,0-1 Haut
```

Ajoutez les plugins suivants dans la section des plugins (il faut un peu descendre dans le fichier .zshrc en utilisant les flèches) en passant en mode insertion en appuyant sur **i** (Vous savez que vous êtes en mode insertion s'il est écrit **-- Insert --** tout en bas de la fenêtre)

Modifiez la ligne des plugins avec celle-ci:

```
plugins=(git vscode poetry conda)
```



```
# DISABLE_UNTRACKED_FILES_DIRTY="true"

# Uncomment the following line if you want to change the command execution time
# stamp shown in the history command output.
# You can set one of the optional three formats:
# "mm/dd/yyyy"|"dd.mm.yyyy"|"yyyy-mm-dd"
# or set a custom format using the strftime function format specifications,
# see 'man strftime' for details.
# HIST_STAMPS="mm/dd/yyyy"

# Would you like to use another custom folder than $ZSH/custom?
# ZSH_CUSTOM=/path/to/new-custom-folder

# Which plugins would you like to load?
# Standard plugins can be found in $ZSH/plugins/
# Custom plugins may be added to $ZSH_CUSTOM/plugins/
# Example format: plugins=(rails git textmate ruby lighthouse)
# Add wisely, as too many plugins slow down shell startup.
plugins=(git vscode poetry conda)

source $ZSH/oh-my-zsh.sh

# User configuration

# export MANPATH="/usr/local/man:$MANPATH"

# You may need to manually set your language environment
# export LANG=en_US.UTF-8

# Preferred editor for local and remote sessions
# if [[ -n $SSH_CONNECTION ]]; then
#   export EDITOR='vim'
# else
#   export EDITOR='nvim'
# fi

-- INSERTION --
```

**Enregistrez et fermez Vim** : Appuyez sur `Esc` pour quitter le mode insertion (la mention `– Insert --` disparaît), tapez `:wq`, puis appuyez sur `Entrée`.

Si vous voulez annuler les modifications que vous êtes en train de faire, faites `:q!` pour quitter sans enregistrer.

Vous pouvez relancer le terminal pour que la configuration de OH-MY-ZSH soit prise en compte.

## Installation de Pyenv

[Documentation officielle](#)

On commence par l'installation de pyenv:

```
curl https://pyenv.run | bash
```

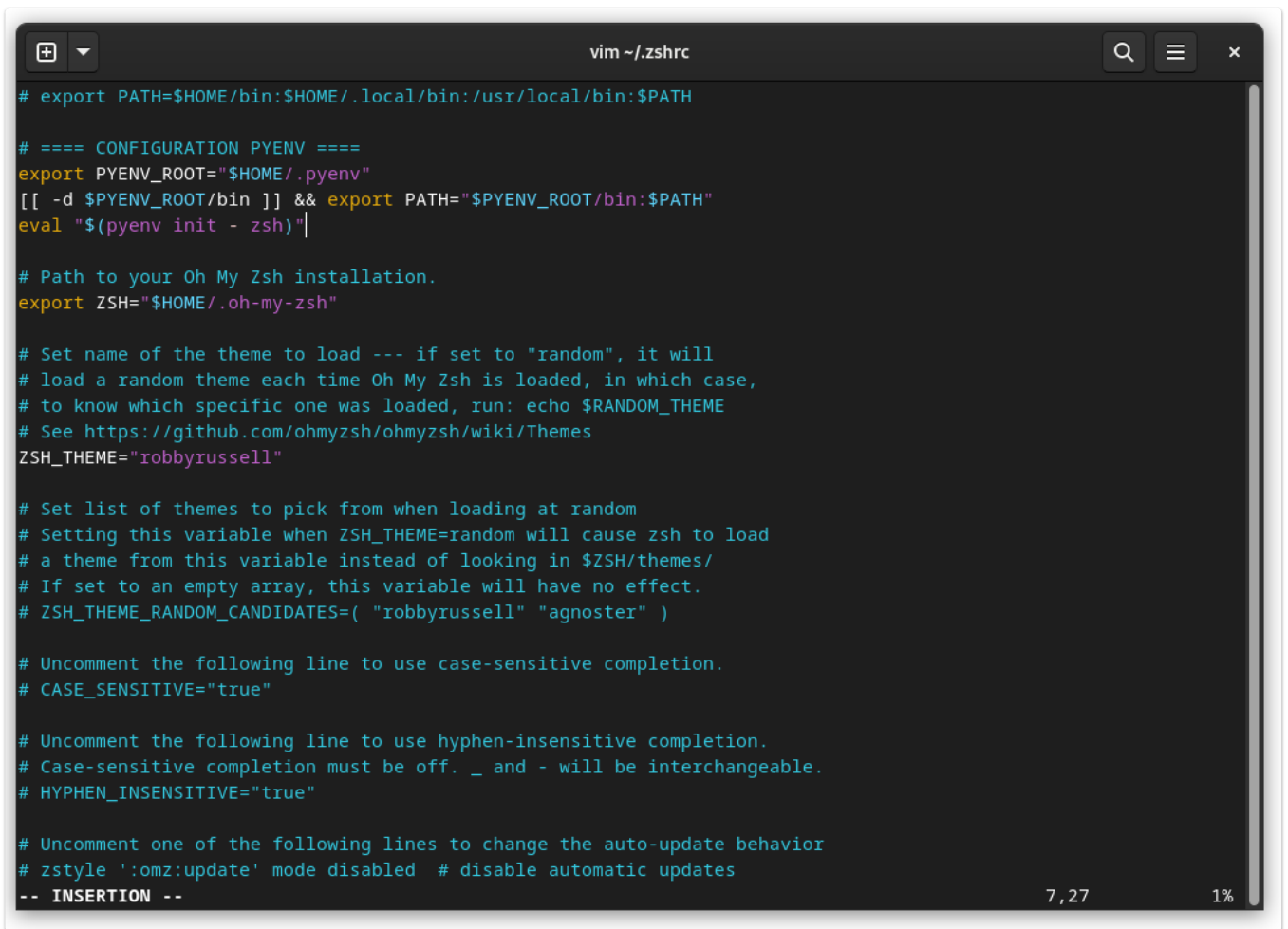
```
antonin@DESKTOP-ASSH4VC:~  
+ ~ curl https://pyenv.iun | bash  
% Total % Received % Xferd Average Speed Time Time Time Current  
Dload Upload Total Spent Left Speed  
100 270 100 270 0 0 665 0 --:--:-- --:--:-- --:--:-- 665  
Clonage dans '/home/antonin/.pyenv'...  
remote: Enumerating objects: 1330, done.  
remote: Counting objects: 100% (1330/1330), done.  
remote: Compressing objects: 100% (726/726), done.  
remote: Total 1330 (delta 796), reused 777 (delta 471), pack-reused 0 (from 0)  
Réception d'objets: 100% (1330/1330), 1.13 Mio | 8.36 Mio/s, fait.  
Résolution des deltas: 100% (796/796), fait.  
Clonage dans '/home/antonin/.pyenv/plugins/pyenv-doctor'...  
remote: Enumerating objects: 11, done.  
remote: Counting objects: 100% (11/11), done.  
remote: Compressing objects: 100% (9/9), done.  
remote: Total 11 (delta 1), reused 5 (delta 0), pack-reused 0 (from 0)  
Réception d'objets: 100% (11/11), 38.72 Kio | 1016.00 Kio/s, fait.  
Résolution des deltas: 100% (1/1), fait.  
Clonage dans '/home/antonin/.pyenv/plugins/pyenv-update'...  
remote: Enumerating objects: 10, done.  
remote: Counting objects: 100% (10/10), done.  
remote: Compressing objects: 100% (6/6), done.  
remote: Total 10 (delta 1), reused 5 (delta 0), pack-reused 0 (from 0)  
Réception d'objets: 100% (10/10), fait.  
Résolution des deltas: 100% (1/1), fait.  
Clonage dans '/home/antonin/.pyenv/plugins/pyenv-virtualenv'...  
remote: Enumerating objects: 64, done.  
remote: Counting objects: 100% (64/64), done.  
remote: Compressing objects: 100% (57/57), done.  
remote: Total 64 (delta 10), reused 23 (delta 0), pack-reused 0 (from 0)  
Réception d'objets: 100% (64/64), 43.06 Kio | 1.79 Mio/s, fait.  
Résolution des deltas: 100% (10/10), fait.  
  
WARNING: seems you still have not added 'pyenv' to the load path.  
  
# Load pyenv automatically by appending  
# the following to  
# ~/.bash_profile if it exists, otherwise ~/.profile (for login shells)  
# and ~/.bashrc (for interactive shells) :  
  
export PYENV_ROOT="$HOME/.pyenv"  
[[ -d $PYENV_ROOT/bin ]] && export PATH="$PYENV_ROOT/bin:$PATH"  
eval "$(pyenv init - bash)"  
  
# Restart your shell for the changes to take effect.  
  
# Load pyenv-virtualenv automatically by adding  
# the following to ~/.bashrc:  
  
eval "$(pyenv virtualenv-init -)"  
  
+ ~ |
```

Ensuite on ouvre le fichier `.zshrc` en utilisant vim:

```
vim ~/.zshrc
```

On passe en mode insertion en appuyant sur `i` et ici j'ajoute une section `# =====`  
`CONFIGURATION PYENV =====` en ajoutant les lignes suivantes :

```
export PYENV_ROOT="$HOME/.pyenv"  
[[ -d $PYENV_ROOT/bin ]] && export PATH="$PYENV_ROOT/bin:$PATH"  
eval "$(pyenv init - zsh)"
```



```
vim ~/.zshrc

# export PATH=$HOME/bin:$HOME/.local/bin:/usr/local/bin:$PATH

# ==== CONFIGURATION PYENV ====
export PYENV_ROOT="$HOME/.pyenv"
[[ -d $PYENV_ROOT/bin ]] && export PATH="$PYENV_ROOT/bin:$PATH"
eval "$(pyenv init - zsh)"

# Path to your Oh My Zsh installation.
export ZSH="$HOME/.oh-my-zsh"

# Set name of the theme to load --- if set to "random", it will
# load a random theme each time Oh My Zsh is loaded, in which case,
# to know which specific one was loaded, run: echo $RANDOM_THEME
# See https://github.com/ohmyzsh/ohmyzsh/wiki/Themes
ZSH_THEME="robbyrussell"

# Set list of themes to pick from when loading at random
# Setting this variable when ZSH_THEME=random will cause zsh to load
# a theme from this variable instead of looking in $ZSH/themes/
# If set to an empty array, this variable will have no effect.
# ZSH_THEME_RANDOM_CANDIDATES=( "robbyrussell" "agnoster" )

# Uncomment the following line to use case-sensitive completion.
# CASE_SENSITIVE="true"

# Uncomment the following line to use hyphen-insensitive completion.
# Case-sensitive completion must be off. _ and - will be interchangeable.
# HYPHEN_INSENSITIVE="true"

# Uncomment one of the following lines to change the auto-update behavior
# zstyle ':omz:update' mode disabled # disable automatic updates
-- INSERTION --
```

Puis on sort du mode insertion avec Echap et on appuie sur `:wq`

Reste plus qu'à recharger le terminal pour prendre en compte la nouvelle configuration de Pyenv et du PATH:

```
exec "$SHELL"
```

À présent on va installer les dépendances pour Pyenv, avec les commandes suivantes:

```
sudo apt update; sudo apt install build-essential libssl-dev zlib1g-dev \
libbz2-dev libreadline-dev libsqlite3-dev curl git \
libncursesw5-dev xz-utils tk-dev libxml2-dev libxmlsec1-dev libffi-dev
```

```
sudo apt install liblzma-dev
```

Testez si Pyenv est bien installé avec la commande :

```
pyenv --version
```

Qui retournera la version de Pyenv.

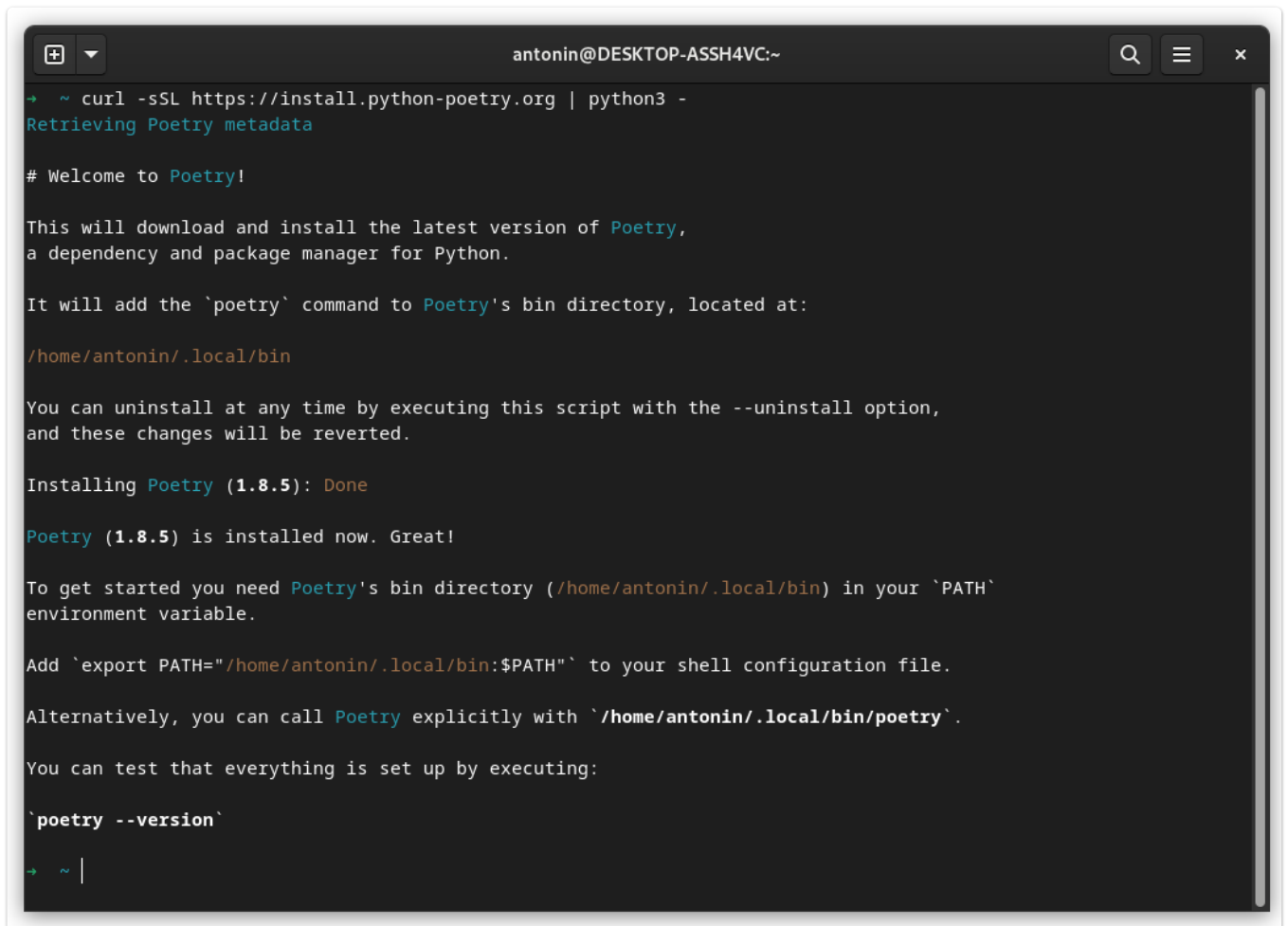


# Installation de Poetry

[Documentation officielle](#)

On installe Poetry avec la commande suivante:

```
curl -sSL https://install.python-poetry.org | python3 -
```

A screenshot of a terminal window titled 'antonin@DESKTOP-ASSH4VC:~'. The terminal shows the execution of the command 'curl -sSL https://install.python-poetry.org | python3 -'. The output includes: 'Retrieving Poetry metadata', '# Welcome to Poetry!', 'This will download and install the latest version of Poetry, a dependency and package manager for Python.', 'It will add the `poetry` command to Poetry's bin directory, located at: /home/antonin/.local/bin', 'You can uninstall at any time by executing this script with the --uninstall option, and these changes will be reverted.', 'Installing Poetry (1.8.5): Done', 'Poetry (1.8.5) is installed now. Great!', 'To get started you need Poetry's bin directory (/home/antonin/.local/bin) in your `PATH` environment variable.', 'Add `export PATH="/home/antonin/.local/bin:\$PATH"` to your shell configuration file.', 'Alternatively, you can call Poetry explicitly with `/home/antonin/.local/bin/poetry`.', 'You can test that everything is set up by executing: `poetry --version`'. The prompt '~ |' is visible at the bottom.

```
antonin@DESKTOP-ASSH4VC:~  
→ ~ curl -sSL https://install.python-poetry.org | python3 -  
Retrieving Poetry metadata  
  
# Welcome to Poetry!  
  
This will download and install the latest version of Poetry,  
a dependency and package manager for Python.  
  
It will add the `poetry` command to Poetry's bin directory, located at:  
  
/home/antonin/.local/bin  
  
You can uninstall at any time by executing this script with the --uninstall option,  
and these changes will be reverted.  
  
Installing Poetry (1.8.5): Done  
  
Poetry (1.8.5) is installed now. Great!  
  
To get started you need Poetry's bin directory (/home/antonin/.local/bin) in your `PATH`  
environment variable.  
  
Add `export PATH="/home/antonin/.local/bin:$PATH"` to your shell configuration file.  
  
Alternatively, you can call Poetry explicitly with `/home/antonin/.local/bin/poetry`.  
  
You can test that everything is set up by executing:  
  
`poetry --version`  
→ ~ |
```

Puis il faut ajouter poetry au PATH.

Pour cela on ouvre le fichier `.zshrc` avec VIM:

```
vim ~/.zshrc
```

Puis, après avoir appuyé sur `i` pour passer en mode insertion, on ajoute cette ligne dans le fichier `.zshrc`:

```
export PATH="$HOME/.local/bin:$PATH"
```

Ici, j'ai ajouté une section `# ==== CONFIGURATION POETRY =====`

```
vim ~/.zshrc

# If you come from bash you might have to change your $PATH.
# export PATH=$HOME/bin:$HOME/.local/bin:/usr/local/bin:$PATH

# ==== CONFIGURATION PYENV ====
export PYENV_ROOT="$HOME/.pyenv"
[[ -d $PYENV_ROOT/bin ]] && export PATH="$PYENV_ROOT/bin:$PATH"
eval "$(pyenv init - zsh)"

# ==== CONFIGURATION POETRY ====
export PATH="$HOME/.local/bin:$PATH"

# Path to your Oh My Zsh installation.
export ZSH="$HOME/.oh-my-zsh"

# Set name of the theme to load --- if set to "random", it will
# load a random theme each time Oh My Zsh is loaded, in which case,
# to know which specific one was loaded, run: echo $RANDOM_THEME
# See https://github.com/ohmyzsh/ohmyzsh/wiki/Themes
ZSH_THEME="robbyrussell"

# Set list of themes to pick from when loading at random
# Setting this variable when ZSH_THEME=random will cause zsh to load
# a theme from this variable instead of looking in $ZSH/themes/
# If set to an empty array, this variable will have no effect.
# ZSH_THEME_RANDOM_CANDIDATES=( "robbyrussell" "agnoster" )

# Uncomment the following line to use case-sensitive completion.
# CASE_SENSITIVE="true"

# Uncomment the following line to use hyphen-insensitive completion.
# Case-sensitive completion must be off. _ and - will be interchangeable.
-- INSERTION --
```

Puis vous tapez Echap puis `:wq` pour enregistrer le fichier.

Et ensuite `exec "$SHELL"` pour appliquer la modification dans le terminal actuel.

Si l'installation et l'ajout au PATH a réussi, la commande `poetry --version` devrait vous donner la version de Poetry.

### ⚡ Important

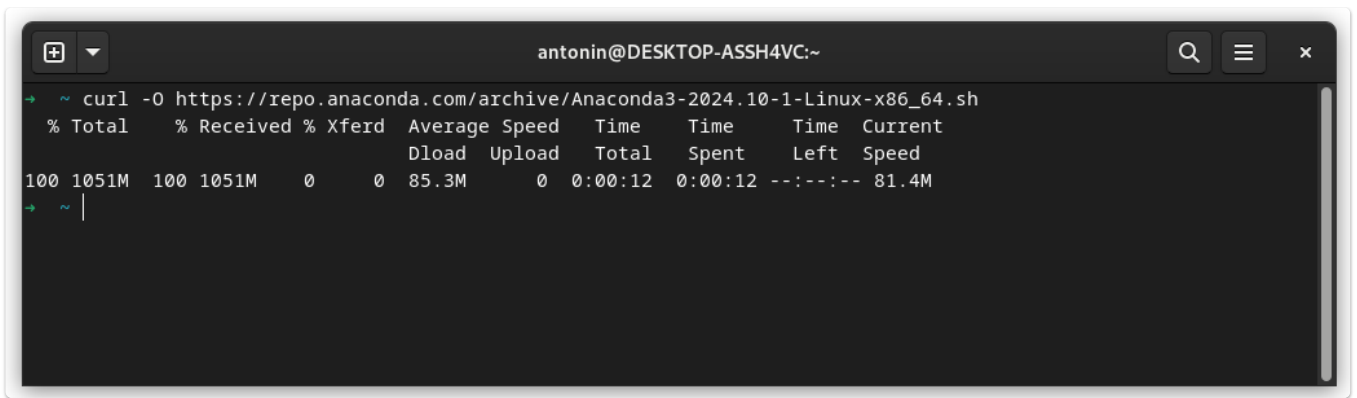
Pour les deux parties qui suivent, sur Anaconda et Miniconda, il faudra choisir quelle distribution choisir entre les deux. Pour cela, je vous renvoie à la vidéo sur la comparaison de Anaconda/Miniconda/Conda

## Installation de Anaconda

[Documentation officielle](#)

On commence par télécharger la dernière version de Anaconda:

```
curl -O https://repo.anaconda.com/archive/Anaconda3-2024.10-1-Linux-  
x86_64.sh
```

A terminal window titled 'antonin@DESKTOP-ASSH4VC:~' showing the execution of a curl command to download the Anaconda3 installer script. The command is 'curl -O https://repo.anaconda.com/archive/Anaconda3-2024.10-1-Linux-x86\_64.sh'. The output shows a progress bar and a table of statistics. The download is complete at 100% with a current speed of 81.4M.

```
antonin@DESKTOP-ASSH4VC:~  
→ ~ curl -O https://repo.anaconda.com/archive/Anaconda3-2024.10-1-Linux-x86_64.sh  
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current  
           Dload  Upload   Total     Spent    Left     Speed  
100 1051M  100 1051M    0     0  85.3M      0  0:00:12  0:00:12 --:--:-- 81.4M  
→ ~ |
```

Puis on installe Anaconda en utilisant le fichier téléchargé précédemment:

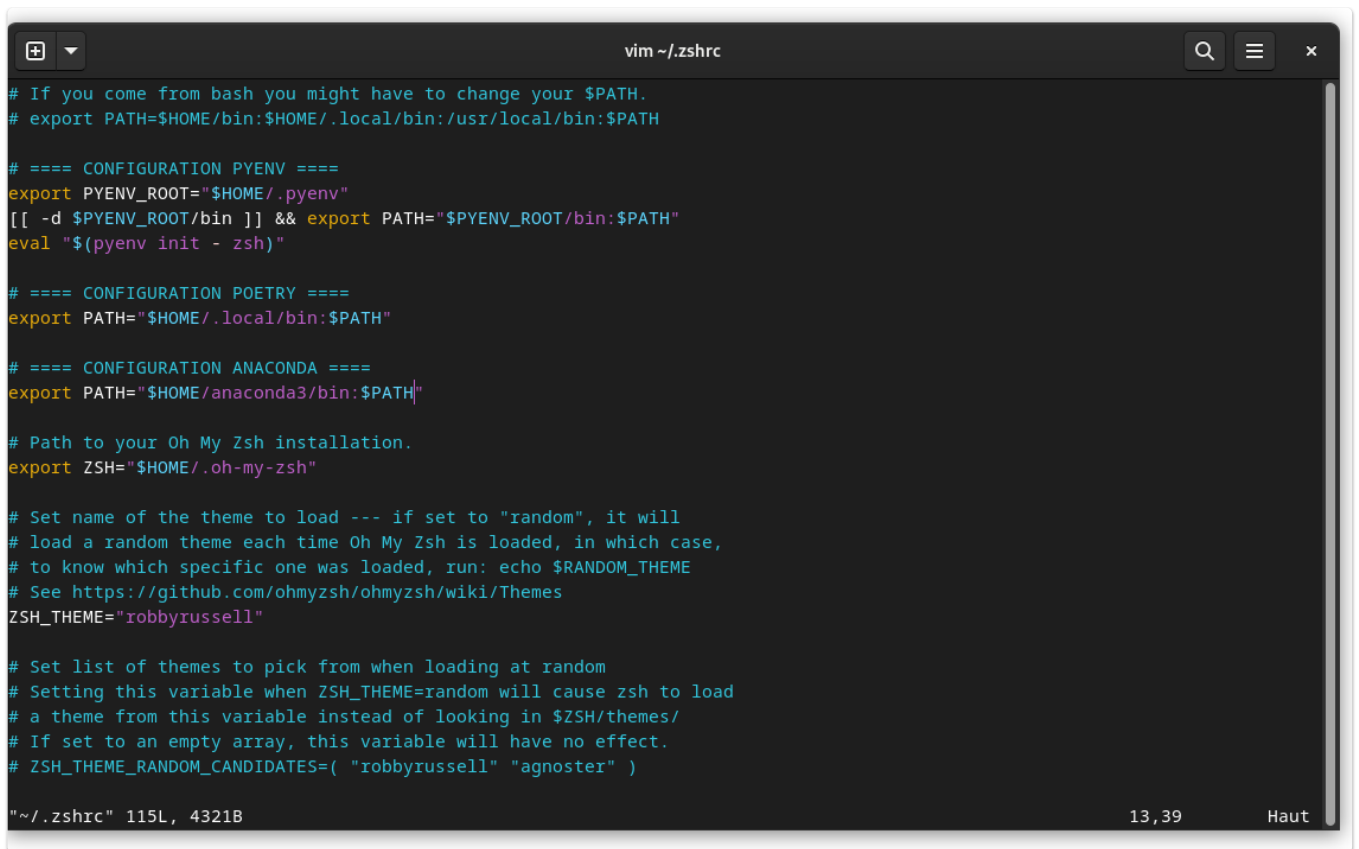
```
bash ~/Anaconda3-2024.10-1-Linux-x86_64.sh
```

Il faudra appuyer sur **Entrée** pour afficher la licence, puis il faudra aller tout en bas et taper **q** pour sortir de la Licence et ensuite **yes** pour accepter. Il faudra une nouvelle fois appuyer sur **Entrée** pour confirmer l'installation à l'endroit indiqué. L'installation se lance, et ensuite on vous demande "Do you wish to update your shell profile to automatically initialize conda?". Il faudra dire **yes**.

Une fois terminée, le message **Thank you for installing Anaconda3!** apparait et il faudra relancer le terminal.

On ajoute ensuite anaconda3 dans le PATH en ajoutant la ligne suivant dans le fichier **.zshrc**:

```
export PATH="$HOME/anaconda3/bin:$PATH"
```



```
# If you come from bash you might have to change your $PATH.
# export PATH=$HOME/bin:$HOME/.local/bin:/usr/local/bin:$PATH

# ==== CONFIGURATION PYENV ====
export PYENV_ROOT="$HOME/.pyenv"
[[ -d $PYENV_ROOT/bin ]] && export PATH="$PYENV_ROOT/bin:$PATH"
eval "$(pyenv init - zsh)"

# ==== CONFIGURATION POETRY ====
export PATH="$HOME/.local/bin:$PATH"

# ==== CONFIGURATION ANACONDA ====
export PATH="$HOME/anaconda3/bin:$PATH"

# Path to your Oh My Zsh installation.
export ZSH="$HOME/.oh-my-zsh"

# Set name of the theme to load --- if set to "random", it will
# load a random theme each time Oh My Zsh is loaded, in which case,
# to know which specific one was loaded, run: echo $RANDOM_THEME
# See https://github.com/ohmyzsh/ohmyzsh/wiki/Themes
ZSH_THEME="robbyrussell"

# Set list of themes to pick from when loading at random
# Setting this variable when ZSH_THEME=random will cause zsh to load
# a theme from this variable instead of looking in $ZSH/themes/
# If set to an empty array, this variable will have no effect.
# ZSH_THEME_RANDOM_CANDIDATES=( "robbyrussell" "agnoster" )

"~/.zshrc" 115L, 4321B 13,39 Haut
```

Et on met à jour la configuration du terminal actuel avec `exec "$SHELL"` .

Si tout est bien installé, vous devriez voir la version de conda avec la commande :

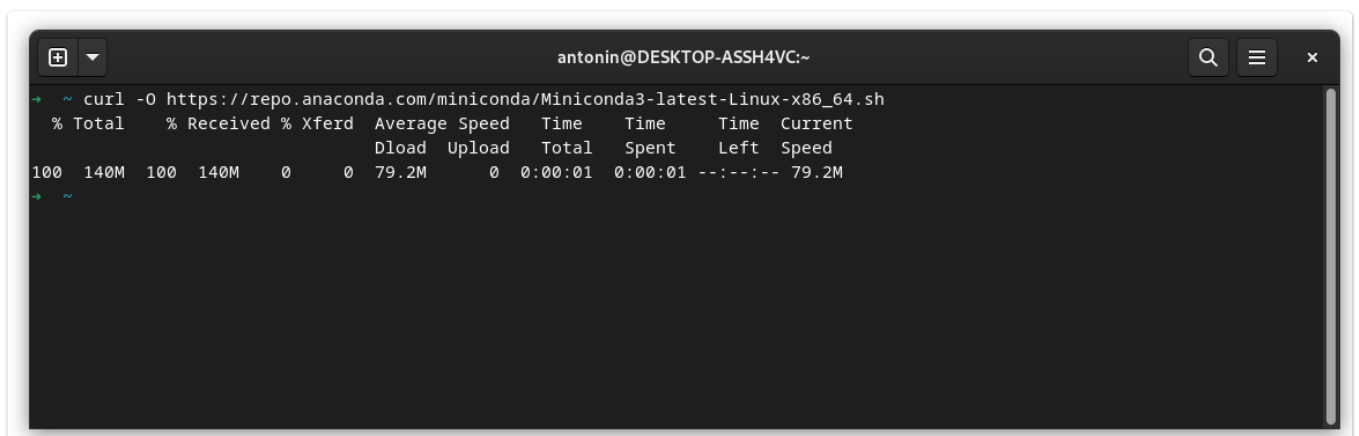
```
conda --version
```

## Installation de Miniconda

[Documentation officielle](#)

On commence par télécharger la dernière version de Miniconda :

```
curl -O https://repo.anaconda.com/miniconda/Miniconda3-latest-Linux-x86_64.sh
```



```
antonin@DESKTOP-ASSH4VC:~
→ ~ curl -O https://repo.anaconda.com/miniconda/Miniconda3-latest-Linux-x86_64.sh
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
           Dload  Upload   Total   Spent    Left  Speed
100 140M  100 140M    0     0  79.2M    0  0:00:01  0:00:01 --:--:--  79.2M
→ ~
```

Puis on installe Miniconda en utilisant le fichier téléchargé précédemment:

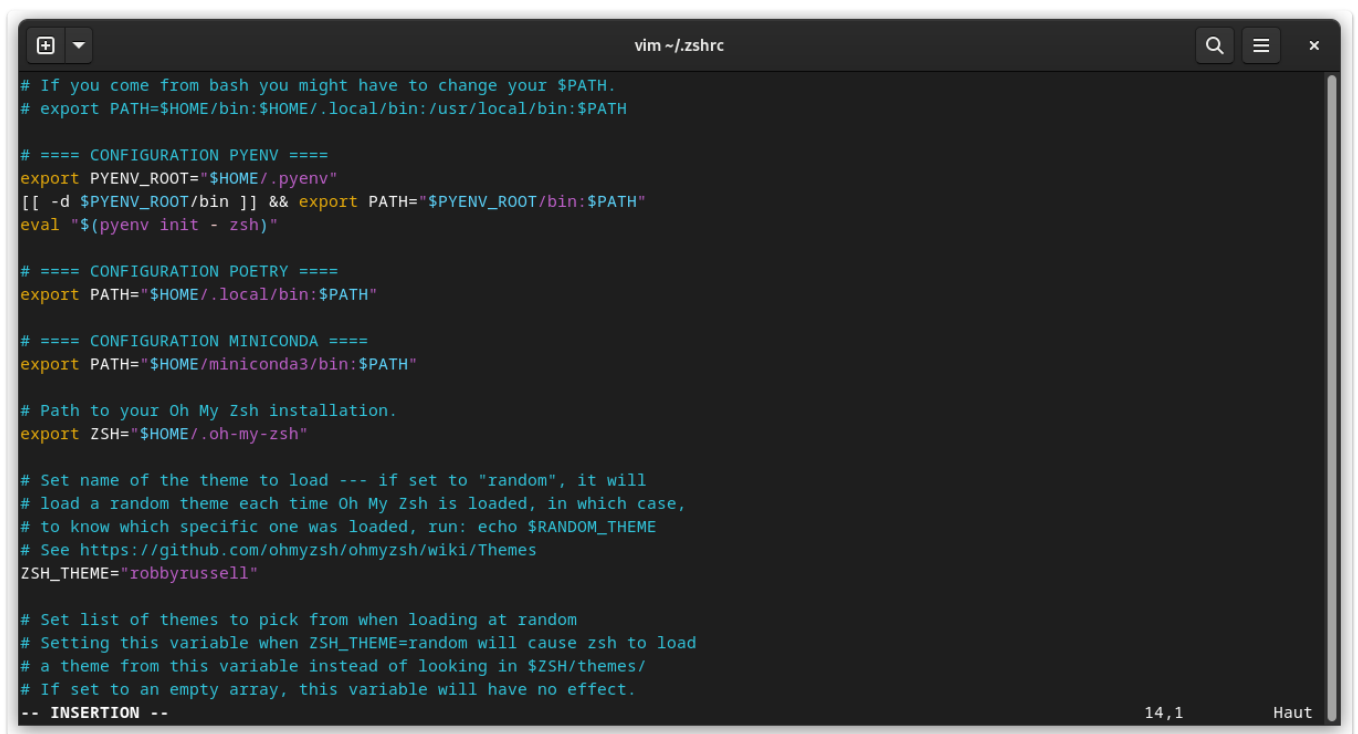
```
bash ~/Miniconda3-latest-Linux-x86_64.sh
```

Il faudra appuyer sur **Entrée** pour afficher la licence, puis il faudra aller tout en bas et taper **q** pour sortir de la Licence et ensuite **yes** pour accepter. Il faudra une nouvelle fois appuyer sur **Entrée** pour confirmer l'installation à l'endroit indiqué. L'installation se lance, et ensuite on vous demande "Do you wish to update your shell profile to automatically initialize conda?". Il faudra dire **yes**.

Une fois terminée, le message **Thank you for installing Miniconda3!** apparait et il faudra redémarrer le terminal.

On ajoute ensuite miniconda3 dans le PATH en ajoutant la ligne suivant dans le fichier `.zshrc`:

```
export PATH="$HOME/miniconda3/bin:$PATH"
```

A screenshot of a terminal window with a dark background. The title bar at the top says "vim ~/.zshrc". The terminal shows the contents of the .zshrc file, which includes comments and configuration for PYENV, POETRY, and MINICONDA. The MINICONDA section contains the line "export PATH=\"\$HOME/miniconda3/bin:\$PATH\"". At the bottom right of the terminal window, it shows "14,1" and "Haut".

```
# If you come from bash you might have to change your $PATH.
# export PATH=$HOME/bin:$HOME/.local/bin:/usr/local/bin:$PATH

# ==== CONFIGURATION PYENV ====
export PYENV_ROOT="$HOME/.pyenv"
[[ -d $PYENV_ROOT/bin ]] && export PATH="$PYENV_ROOT/bin:$PATH"
eval "$(pyenv init - zsh)"

# ==== CONFIGURATION POETRY ====
export PATH="$HOME/.local/bin:$PATH"

# ==== CONFIGURATION MINICONDA ====
export PATH="$HOME/miniconda3/bin:$PATH"

# Path to your Oh My Zsh installation.
export ZSH="$HOME/.oh-my-zsh"

# Set name of the theme to load --- if set to "random", it will
# load a random theme each time Oh My Zsh is loaded, in which case,
# to know which specific one was loaded, run: echo $RANDOM_THEME
# See https://github.com/ohmyzsh/ohmyzsh/wiki/Themes
ZSH_THEME="robbyrussell"

# Set list of themes to pick from when loading at random
# Setting this variable when ZSH_THEME=random will cause zsh to load
# a theme from this variable instead of looking in $ZSH/themes/
# If set to an empty array, this variable will have no effect.
-- INSERTION --
```

Et on met à jour la configuration du terminal actuel avec `exec "$SHELL"`.

Si tout est bien installé, vous devriez voir la version de conda avec la commande :

```
conda --version
```

## Installation de VSCode

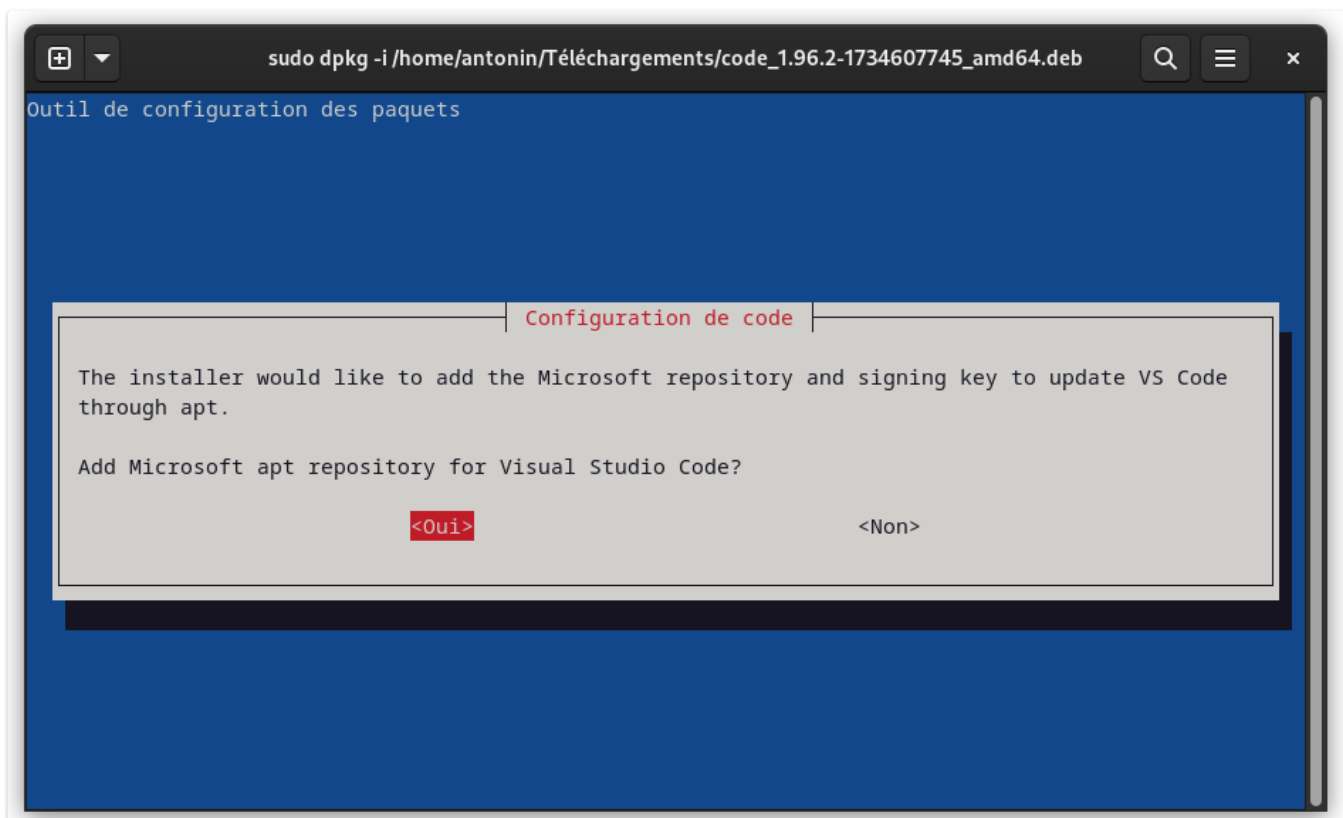
## [Documentation Officielle](#)

On commence par télécharger le fichier .deb de VSCode [ICI](#) puis on fait:

```
sudo dpkg -i <file_path>
```

en remplaçant `<file_path>` par le chemin du fichier téléchargé précédemment (le fichier .deb ).

Sélectionnez `oui` quand on vous demande l'ajout du Repository Microsoft et des clés:



Ensuite l'installation se termine. Pour ouvrir VSCode vous pouvez soit le lancer depuis les applications (et choisir ensuite quel projet ouvrir), soit depuis le terminal à partir du dossier d'un projet en tapant `code .` . Si vous tapez uniquement `code` cela ouvrira VSCode comme s'il était lancé depuis les applications.

## Extensions VSCode

🕒 Les extensions suivantes sont détaillées dans une vidéo dédiées dans ce module.

- **Python Extension Pack** : Fournit un support complet pour Python. (De Don Jayamanne)
- **GitHub Copilot** : Un assistant de codage IA.
- **Path Intellisense** : Autocomplétion des chemins de fichiers.

- **Material Icon Theme** : Améliore les icônes de fichiers et dossiers.
- **Jupyter**