



Configuration de macOS pour la masterclass Data

Cette fiche présente les étapes essentielles pour configurer un Mac avec **Python**, **pyenv**, **TensorFlow**, **pandas**, **Homebrew**, et **Jupyter**.

1. 🍺 Installation de Homebrew

Homebrew est un gestionnaire de paquets pour macOS, nécessaire pour installer plusieurs outils.

- Ouvrir le terminal et exécuter :

```
/bin/bash -c "$(curl -fsSL
https://raw.githubusercontent.com/Homebrew/install/HEAD/install.sh)"
```

- Mettre à jour Brew:

```
brew update
```

Par la suite installer les package suivants:

```
brew upgrade git          || brew install git
brew upgrade jq           || brew install jq
brew upgrade openssl     || brew install openssl
brew upgrade xz           || brew install xz
brew upgrade zlib         || brew install zlib
brew upgrade readline    || brew install readline
```

Attention ⚠

Il est possible que vous ayez une erreur de la sorte :

```
Error: Not a valid ref: refs/remotes/origin/master :
fatal: ambiguous argument 'refs/remotes/origin/master': unknown revision
or path not in the working tree.
```

Si c'est le cas cette ligne de commande doit régler le problème:

```
rm -fr $(brew --repo homebrew/core) # because you can't `brew untap
homebrew/core`
```

```
brew tap homebrew/core
```

[Solution Github](#)

2. Installer les outils en ligne de commande

Xcode CLI tools sont essentiels pour faire fonctionner plusieurs outils.

Installe-les avec cette commande magique 🧙 :

```
xcode-select --install
```

3. Installation de python

Comment vérifier si Python est déjà installé sur votre Mac

Si votre macOS est plus ancien que Catalina, il se peut que Python soit déjà installé. Voici comment le vérifier :

1. Allez dans **Applications > Utilitaires > Terminal**.
2. Tapez : `python --version`
3. Appuyez sur **Entrée** ↵
4. Si Python est installé, un message indiquera la version installée.
5. Si Python n'est pas installé, vous verrez le message : `command not found: python`

Comment installer Python dans le cas où il n'est pas installé

Pour installer Python sur votre Mac, vous devrez d'abord télécharger l'installateur officiel.

1. Ouvrez un navigateur web et allez sur : python.org/downloads
2. Vous serez automatiquement dirigé vers la page de téléchargement. Cliquez sur **Télécharger la dernière version pour macOS**.
3. Une fois le téléchargement terminé, ouvrez le fichier d'installation pour le lancer.
4. Cliquez sur **Continuer** et suivez les instructions à l'écran.
5. Une fois l'installation terminée, un dossier Python apparaîtra dans votre dossier **Applications**.

✅ Pour vous assurer que l'installation s'est bien déroulée...

1. Ouvrez le dossier **Python** dans vos applications et double-cliquez sur **IDLE** pour vérifier que tout est installé correctement.
2. Quand IDLE s'ouvre, une interface en ligne de commande apparaîtra. Tapez la commande suivante pour tester le bon fonctionnement :

```
print('Hello world')
```

Installation de pyenv pour un environnement virtuel

Installer Pyenv

Pyenv te permet de jongler avec différentes versions de Python, et donc de créer plusieurs environnements sécuriser qui n'auront pas d'impact les uns sur les autres.

Installe Pyenv via Homebrew :

```
brew install pyenv
```

Configure ton terminal pour qu'il utilise Pyenv automatiquement:

- Pour zshrc:

```
echo 'export PYENV_ROOT="$HOME/.pyenv"' >> ~/.zshrc
echo 'export PATH="$PYENV_ROOT/bin:$PATH"' >> ~/.zshrc
echo 'eval "$(pyenv init --path)"' >> ~/.zshrc
exec zsh
```

- Pour bash:

```
echo 'export PYENV_ROOT="$HOME/.pyenv"' >> ~/.bashrc
echo 'export PATH="$PYENV_ROOT/bin:$PATH"' >> ~/.bashrc
echo 'eval "$(pyenv init --path)"' >> ~/.bashrc
source ~/.bashrc
```

Installer la bonne version de Python avec Pyenv

Maintenant nous allons créer notre environnement de travail.

Avec pyenv tu peux maintenant lister l'ensemble des versions disponibles au téléchargements :

```
pyenv install --list
```

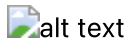
Pour notre cours nous allons utiliser la versions 3.12.3

```
pyenv install 3.12.3
```

POur être sûr que tu peux continuer, vérifie que ton installation c'est bien passé:

- Lance la commande

```
pyenv versions
```



Si tu vois bien la version dans la réponse du terminal tu peux continuer.

🌐 Créer un environnement virtuel

On va créer maintenant un environnement virtuel à partir de cette version de python.

Installe le plugin pyenv-virtualenv :

```
brew install pyenv-virtualenv  
source ~/.zshrc (si zsh)  
source ~/.bashrc (si bash)
```

Créer ton premier environnement virtuel 🧙 :

```
pyenv virtualenv 3.12.3 masterclassData  
pyenv activate masterclassData
```

6. 📦 Installer les packages essentiels

Une dernière étape à suivre. Maintenant vous allez devoir installer l'ensemble des paquets python nécessaires avec votre nouvel environnement virtuel fraîchement installé. Pour ce faire rien de plus simple:

- Vérifiez bien que vous êtes dans votre environnement virtuel :

```
3.12.3/envs/test  
langchain --> /Users/christoph  
langchain_courses --> /Users/c  
* masterclassData --> /Users/chr  
riable)  
my-env --> /Users/christopherd  
scrapy --> /Users/christopherd
```

- Par la suite si vous êtes sur un apple silicon:

```
pip install -r silicon.txt
```

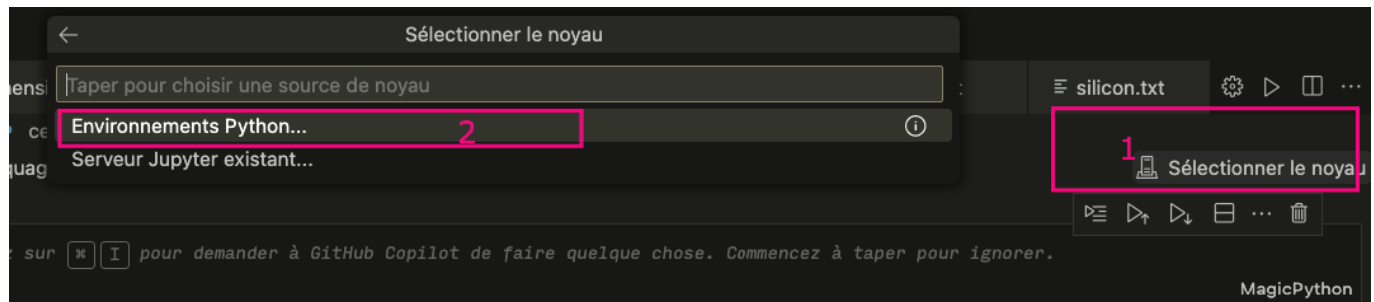
- Dans le cas ou vous êtes sur intel

```
pip install -r intel.txt
```

Good Job 🍷 Ta machine est prête maintenant pour commencer. Tu peux dès à présent dans vsCode créer un fichier **.ipynb** et choisir le bon kernel afin de te familiariser avec Jupyter.

7. 📖 Configurer le noyau Jupyter Notebook dans VS Code

Dans vsCode, tu vas devoir sélectionner le Kernel que tu souhaites utiliser. C'est à dire l'environnement de python avec lequel ton notebook doit lancer ses cellules.



Comme sur l'image précédente, tu dois cliquer sur :

1. Sélectionner un noyau
2. Environnements Python
3. Et enfin choisir l'environnements que tu as créé avec Pyenv

8. 📚 Des ressources à préparer

- [Utilisation de Jupyter](#)
- [Les bases de Python](#)
- [Choisir le Kernel de VsCode](#)