

## TP 0 : installation des outils nécessaires Windows

### 1. Installation de FNM

---

FNM (pour *Fast Node Manager*) est un utilitaire de gestion des versions de Node qui vous permet d'installer en parallèle plusieurs versions du moteur d'exécution. Vous pourrez alors utiliser la version que vous désirez pour chaque projet, vous assurant une expérience et une compatibilité optimales.

Pour installer l'utilitaire `fnm`, exécutez simplement la commande suivante dans PowerShell :

```
winget install Schniz.fnm
```

Cette commande télécharge un script d'installation sur un serveur distant, et l'exécute dans votre terminal. Fermez et relancez ensuite votre terminal pour avoir accès à **fnm**. Pour tester que tout fonctionne une fois votre terminal relancé, exécutez la commande :

```
fnm list
```

Cette commande devrait vous renvoyer comme résultat « \* system ». Cela indique qu'elle fonctionne, qu'elle liste les versions de NodeJS installées, mais qu'elle n'en trouve aucune. Pour installer une nouvelle version de NodeJS, passez à l'étape 2.

### 2. Installation de NodeJS

---

Pour le reste des séances, nous utiliserons NodeJS v20.18.1 qui est la version LTS (*Long Time Support*) actuelle (à la date du 19/01/2025). Choisir cette version nous offre les meilleures possibilités de support dans le futur en cas de problème, il est conseillé d'utiliser au maximum les versions LTS lorsqu'elles sont disponibles.

Pour installer NodeJS, une fois **fnm** installé, lancez un terminal, puis exécutez la commande :

```
fnm install 20
```

Lorsque la barre de chargement sera complétée, et que le terminal vous aura rendu la main, exécutez la commande :

```
fnm list
```

Le retour de la commande devrait maintenant contenir « \* v20.18.1 default » dans la liste des versions installées. Cela indique que vous avez installé la version 20 avec succès, et qu'elle est désormais la version par défaut si vous n'en précisez pas d'autre ouvertement.

### 3. Configuration de fnm

---

Si, lorsque vous utilisez la commande suivante :

```
fnm use 20
```

Après avoir installé Node, vous avez un message d'erreur qui vous dit que les variables d'environnement nécessaires ne peuvent pas être trouvées, vous devrez configurer deux choses sur votre système.

Tout d'abord, vous devrez changer les autorisations d'exécution dans le terminal pour pouvoir lire le fichier de configuration que nous créerons à l'étape suivante. Pour cela lancez un terminal powershell **en mode administrateur** Et lancez la commande suivante :

```
Set-ExecutionPolicy -ExecutionPolicy RemoteSigned
```

Ensuite, vous devrez créer le fichier de configuration de Microsoft Powershell :

```
if (-not (Test-Path $profile)) { New-Item $profile -Force }
```

Pour l'éditer, exécutez la commande suivante :

```
Invoke-Item $profile
```

Enfin, ajoutez dans ce fichier la ligne suivant pour configurer **fnm** au lancement :

```
fnm env --use-on-cd --shell powershell | Out-String | Invoke-Expression
```

## 4. PNPM

---

Pour installer de nouvelles dépendances dans nos projets VueJS, nous utiliserons le gestionnaire de dépendances **pnpm**, plus performant que le gestionnaire par défaut **npm** sur de nombreux points. Nous allons tout d'abord devoir installer **pnpm** globalement, pour cela ouvrez un terminal, et après avoir sélectionné la bonne version de NodeJS via **fnm** (voire les étapes précédentes), exécutez la commande suivante :

```
npm install --global pnpm
```

Vérifiez l'installation avec la commande :

```
pnpm --version
```

Le résultat devrait être « 9.15.4 ». Félicitations, tous les outils nécessaires sont désormais installés, vous pouvez passer au prochain TP et créer vos premières applications VueJS archaïques.