

インストール方法

✖ 次の場所を開発ディレクトリーに選ぶことは止めておいた方がいいです

- NAS - UNCパスに対応していないライブラリーがあり、インストールに失敗する。vite や esbuild など。

(必要であれば) ローカル・リポジトリの設定

- git を使う予定であれば、先にリモート・リポジトリとの接続を確立しておいた方が楽です。

パッケージ・マネージャーの用意

- 本書では、パッケージ・マネージャーは pnpm を使うとします。

最近（2025年）のお勧めは npm より pnpm らしいんで、 pnpm を積極的に使っていくことにします。

□ pnpm は npm と何が違うのか

- 以下のコマンドを打鍵して、既にインストールされていないか調べてください。

```
pnpm -v  
# もしインストールされていれば、バージョン番号が出る。例：  
# 10.15.0
```

- pnpm のインストール方法
 - pnpm は npm を使ってインストールします。
 - npm は Node.JS に付属しています。

Windows の場合、Node.JS のホームページから .msi インストーラーをダウンロードして Node.JS をインストールする。

```
# Node.JS のバージョン確認  
node -v  
# 例：  
# v22.19.0
```

Windowsにて、npm, pnpm の両方のバージョンアップ：

```
npm install -g npm  
npm install -g pnpm
```

```
pnpm -v  
# 例：  
# 10.15.0
```

(必要であれば) 既存フォルダー、ファイルの削除

- （再インストールであれば）以下のフォルダー、ファイルを削除しておくと確実です。

（パワーシェルではなく）コマンドプロンプトを使う。

- └ node_modules フォルダーを削除。
- └ src-tauri/target フォルダーを削除。
- └ pnpm-lock.yaml ファイルを削除。

インストール

Tauri をインストールしておいてください。（説明略）

```
tauri --version  
# 例：  
# tauri-cli 2.0.4
```

以下の2つをインストールする方法を説明します。

1. フロントエンドの TypeScript 環境
2. バックエンドの Rust 環境

(1)

（パワーシェルではなく）コマンドプロンプトを使う。

```
# フロントエンドは、通常はこれだけ  
pnpm install  
# └ node_modules フォルダー等が生成されます。  
  
# それでダメならこれ  
# pnpm install --shamefully-hoist  
  
# ( 依存関係を満たしていないくとも、強制的に構築する場合 )  
# pnpm install --force
```

(2) Rust のインストール

└ Rust

VSCodeを再起動してください。

```
# バックエンドで Rust も使うので、これも。  
cd src-tauri  
cargo check  
cargo build
```

