

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・リポジトリーの設定

- git を使う予定であれば、先にリモート・リポジトリー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。
- 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おいてください。

```
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で確実です。

(パワーシェルではなく) コマンドプロンプトを使う。


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

インストール

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)

```
# フロントエンドは、通常はこれだけ
pnpm install
#  node_modules フォルダ等が生成されます。

# それでダメならこれ
#pnpm install --shamefully-hoist
```

(2)

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