Windows AI 開発環境構築ガイド（入門編）

1. はじめに：AI 開発環境の全体像

このガイドは，**Windows PC 上で AI 開発を始めたいと考えている学生や技術者**を対象としています．本ガイドでは，AI 開発で広く使われるプログラミング言語 Python，基本的なデータ処理ライブラリ，そして開発を助けるコードエディタを中心に，**シンプルかつ実践的な AI 開発環境**を構築することを目指します．このガイドの手順に沿って環境を構築すれば，簡単なデータ分析や機械学習モデルの実行，ディープラーニングの入門的なコードを試すことができるようになります．

対象とする PC 環境は Windows 11 です．NVIDIA 製 GPU がある場合は GPU を活用した開発環境も構築できますが，GPU がない場合でも CPU のみで多くの学習・開発を行うことは可能です．AI 開発は CPU だけでも可能ですが，深層学習では GPU を使用することで計算速度が向上します．

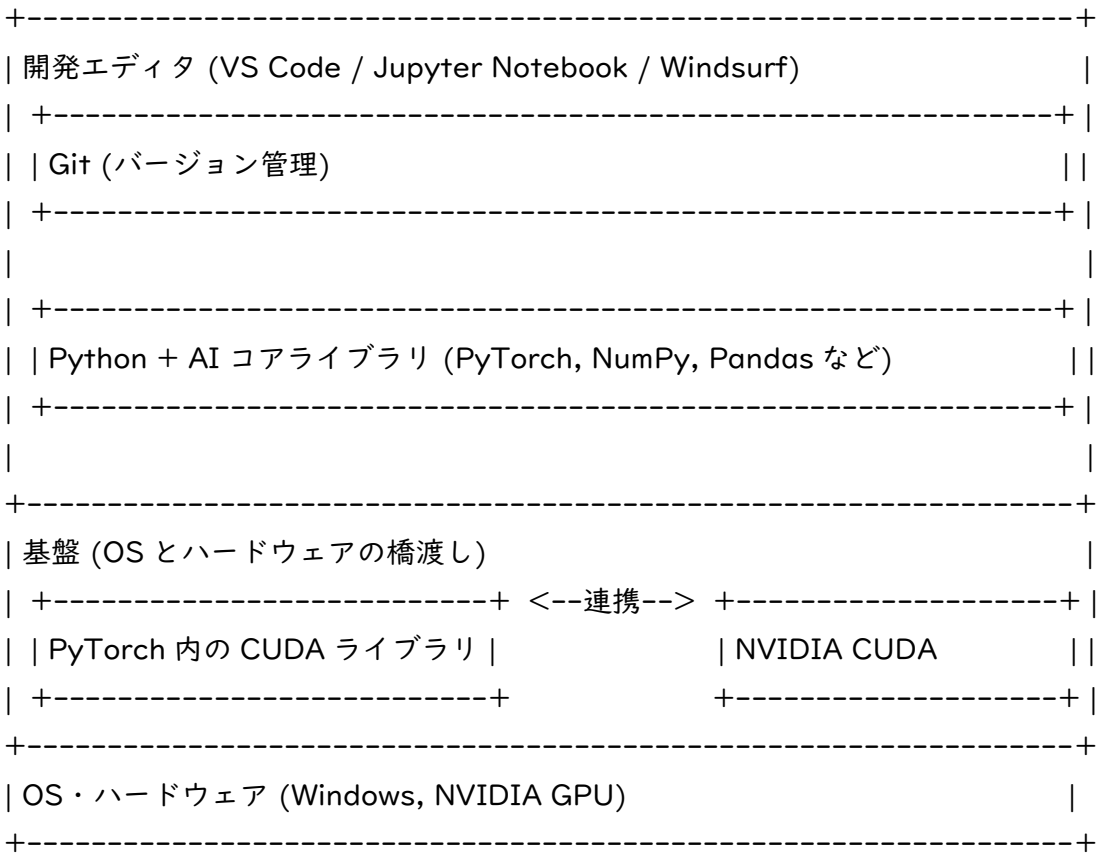
インストールの実行には，PC の SSD／ディスクの空き容量が約 20GB 必要です．事前にご確認ください．空き容量が無い人は実施しないこと．

このガイドが，Windows AI 開発環境構築とその後の学習・開発活動に役立つことを期待しています．

2. ガイドが対象とする AI 開発環境（入門編）

本ガイドでは，AI 開発環境（入門編）として，以下のツール群の役割，メリット，そしてインストール手順を解説します．9 章には，より高度なステップについても説明しています．

- **プログラミング言語:** Python
- **バージョン管理:** Git
- **GPU 利用基盤:** NVIDIA CUDA
- **Python ライブラリ:** PyTorch, NumPy, Pandas, Scikit-learn, Matplotlib, Seaborn など
- **開発エディタ:** Visual Studio Code, Jupyter Notebook, Windsurf
- **ユーティリティ:** 7-Zip



3. 導入にあたっての推奨事項 (段階的アプローチ)

スムーズな環境構築のため，以下の段階的な手順を推奨します．

1. **環境構築の基礎知識を学ぶ:** まず，コマンドプロンプトの基本的な使い方など，Windows での環境構築に必要な基礎知識を確認します（4 章参照）．
2. **AI 開発環境（入門編）のインストール:** Python 本体，Git，7-Zip，開発エディタ必要であれば NVIDIA CUDA を PC にインストールします（5.1 節参照）．
3. **Python ライブラリの導入:** PyTorch などの AI 開発等に必要な Python ライブラリをインストールします（5.2 節参照）．
4. **開発エディタ:** 3 つの開発エディタ（Visual Studio Code, Jupyter Notebook, Windsurf）を知り，状況に応じて活用します（5.3 節参照）．

4. 環境構築の前に知っておきたい Windows の基礎

この章では，環境構築の手順を進める上で必要となる Windows の基本的な操作や概念について解説します．ここで学ぶ知識は，続くインストール手順で頻繁に登場するため，内容を理解しておくとスムーズに進められます．

4.1. フォルダ（ディレクトリ），パス

フォルダ（ディレクトリ：ファイルを整理するための仕組み）は，複数のファイルやフォルダを格納します．そのことで，大量のファイルを整理して格納できます．

パスは，コンピュータ上でファイルやフォルダの位置を示す文字列です．例えば，「C:¥Users¥金子さん¥Documents¥研究¥データ.xlsx」は，C ドライブ（コンピュータ内の記憶装置）に Users フォルダがあり，その中に金子さんフォルダがあり，更に，その中に Documents フォルダがあり，その中に「データ.xlsx」があります．このファイル「データ.xlsx」のパスは「C:¥Users¥金子さん¥Documents¥研究¥データ.xlsx」になります．

4.2. 環境変数とユーザープロファイル

環境変数はシステム全体で共有される設定値を格納するものです．**ユーザープロファイル**は各ユーザーアカウントに割り当てられる専用ディレクトリ（C:¥Users¥ユーザー名）であり，個人データと設定が格納されます．

4.2.1. 環境変数

環境変数は，システム全体で共有される設定値を保存する変数です．例えば，環境変数%USERPROFILE%には現在のユーザーのプロファイルディレクトリ名（例：C:¥Users¥金子さん）が格納されています．コマンドプロンプトなどで「%USERPROFILE%¥Desktop」と記述することで，そのユーザーのデスクトップフォルダを参照できます．

4.2.2. ユーザープロファイル

ユーザープロファイルは，個人専用のフォルダ領域であり，Windows の各ユーザーアカウント（コンピュータを使用する個人の登録情報）に対して個別に割り当てられる専用ディレクトリです．ユーザープロファイルにはユーザー固有のデータと設定が格納されます．

- Windows 11 ではユーザープロファイルは，**C:¥Users¥ユーザー名**になります（ユーザー「金子さん」の場合は C:¥Users¥金子さん）．

- **コマンドプロンプト**（コマンドでシステムを操作するツール）を起動すると，作業ディレクトリ（コマンド実行時の基準となる場所）が，「C:¥Users¥金子さん>」のように表示されます．コマンドプロンプトの起動直後は，作業ディレクトリがユーザープロファイルに設定されます．コマンドプロンプトの作業ディレクトリは，**cd コマンド**で変更できます．

4.3. コマンドプロンプトの基本

コマンドプロンプトはコマンドでシステムを操作するツールです．システム管理，ファイル操作，設定変更，インストール作業などの様々な作業をコマンドで実行でき，GUI（グラフィカルインターフェース）では手順が多くなる操作や，自動化したい操作の実行に便利です．通常起動と管理者権限起動が可能で，cd コマンドで作業ディレクトリを変更できます．**ワイルドカード**（*.tmp 等）でファイルを一括指定し，dir，copy，del 等のコマンドでファイル操作を実行できます．

4.3.1. コマンドプロンプトとは

- **別の言葉で例えると:** 文字で PC と会話するための「画面」のようなものです．dir（ファイル一覧表示）や cd（フォルダ移動）といった「コマンド」を入力して，PC に直接命令を送ります．
- **コマンドプロンプトとは:** コマンドと呼ばれる文字列を入力して OS の機能を直接実行するための，キャラクタユーザインターフェース（CUI）です．GUI（Graphical User Interface）と異なり，キーボード入力が中心になります．
- **使用理由:** 複雑な作業を効率的に行ったり，繰り返しの作業を自動化したりするのに役立ちます．ソフトウェアのインストールや設定変更など，GUI では対応が難しい操作もコマンドで行えます．

4.3.2. 一般ユーザーと管理者

一般ユーザー権限では自分のプロファイル内操作が可能であり，**管理者権限**ではシステム全体を制御できます．このように，一般ユーザー権限であるか，管理者権限であるかによって，Windows におけるアクセス制御（誰が何を操作できるかを管理する仕組み）が行われます．つまり，実行できる操作が異なります．アクセス制御は，重要なシステムファイルが誤って変更されることを防ぐのに役立ちます．

- **一般ユーザー権限:** 日常的な作業に必要な基本的な操作権限です．自分のユーザープロファイル内でのファイル操作，インストール済みアプリケーションの実行などが可能です．
- **管理者権限:** システム全体を制御できる最高レベルの権限であり，システムの重要な設定やファイルを変更できる特別な権限です．一般ユーザー権限では C:¥Windows¥System32 などのシステムディレクトリへのファイル作成や，システム設定の変更ができませんが，管理者権限では可能です．重要なシステム操作には管理者権限が必要となります．

4.3.3. コマンドプロンプトの起動

- **コマンドプロンプトの起動（通常起動）**

  いくつかの方法があります．「cmd」は，コマンドプロンプトを起動するためのプログラム名です．

  - 方法① Windows キーまたはスタートメニュー，cmd と入力，コマンドプロンプトを選ぶ，Enter
  - 方法② Windows キー ＋ R（同時押し），cmd と入力，Enter

  実行例．次により，環境変数 %USERPROFILE% の値が表示されます．

  1. コマンドプロンプトを起動（手順：Windows キーまたはスタートメニュー，cmd と入力，コマンドプロンプトを選ぶ，Enter）
  2. 次のコマンドを実行．echo を用いて環境変数 USERPROFILE の値を表示．

     ```
     echo %USERPROFILE%
     ```

- **コマンドプロンプトを管理者として実行**

  システム設定やソフトウェアのインストールなど，システムレベルの変更が必要な操作をコマンドプロンプトで実行するときは，Windows のコマンドプロンプトを管理者として実行します（手順：Windows キーまたはスタートメニュー，「cmd」と入力，右クリックメニューなどで「管理者として実行」を選択）．コマンドプロンプトの通常実行では，Windows のアクセス制御により，一部の操作が制限される場合があります．

  コマンドプロンプトの起動（管理者権限で起動）

  1. Windows キーまたはスタートメニュー，cmd と入力

2. 「コマンドプロンプト」を右クリックし，「管理者として実行」を選択．あるいはメニューに「管理者として実行」が表示されている場合は，それを選択
3. セキュリティ警告：「このアプリがデバイスに変更を加えることを許可しますか？」というメッセージが表示されたら「はい」をクリックします．

4.3.4. コマンド実行のパラメータ（オプション）

パラメータや**オプション**（コマンドの動作を細かく制御するための追加指示）により，コマンドに条件や設定を追加できます．例えば「dir /a」コマンドにおいて，dir（dir：ディレクトリの内容を表示するコマンド）は「ディレクトリ内容の表示」という指示であり，/a（/a：すべてのファイルを「隠しファイルも含めて全て表示する」という追加指示）です．

4.3.5. ワイルドカード

コマンド実行では，**ワイルドカード**と呼ばれる「何でも当てはまる」という意味を持つ特殊文字を用いて，複数のファイルを一度に指定できます．例えば，「*.tmp」は「拡張子が.tmpであるすべてのファイル」を意味し，テスト.tmp，12345.tmp などのファイル名とマッチします．別の例としては「data_*.xlsx」は「data_で始まるすべての Excel ファイル」を指定します．これは，ファイル名の一部について，任意の文字列にマッチする**パターンマッチング**（条件に合致するものを探す仕組み）を行うものです．

4.3.6. コマンドプロンプトによるファイル操作

コマンド入力後，Enter キーで実行が開始されます．プロンプトには現在の作業ディレクトリ（現在操作対象となっているフォルダ）のパスが表示されます．コマンドプロンプトでは，現在作業しているフォルダを「**カレントディレクトリ**」と呼びます．

基本的なファイル操作コマンド

- **dir**: ディレクトリ（フォルダ）の内容を表示するコマンド．中のファイルやサブディレクトリ（サブフォルダ）などを一覧表示します．
  - dir: 現在のディレクトリ内容を詳細表示します
  - dir /a: 隠しファイルとシステムファイルを含む全ファイルを表示します
  - dir /s: サブディレクトリを含む階層的な内容を表示します
  - dir /w: ファイル名を横方向に整列表示します
- **cd**: カレントディレクトリを確認，移動するコマンド

- 使用例: cd Documents または cd ..（cd ..：一つ上の親ディレクトリへ移動するための特殊な指定）
- cd ディレクトリパス: カレントディレクトリを変更します．例：cd C:¥Users
- cd ..: カレントディレクトリを，一つ上の階層のディレクトリに移動します．
- cd（引数なし）：現在のカレントディレクトリを表示（ただし，プロンプトに既に表示されている情報と同じ）

- **mkdir**: 新規ディレクトリ（フォルダ）を作成するコマンド．
  - 使用例: mkdir project_folder または mkdir "My Project"
- **rmdir**: ディレクトリ（フォルダ）を削除するコマンド
  - 使用例: rmdir /s /q old_project（/s：サブディレクトリも含めて削除するオプション，/q：確認メッセージを表示しないオプション）
  - rmdir ディレクトリ名: 空のフォルダーを削除します．（/s オプションで中身ごと削除）
- **del**: 指定したファイルを削除するコマンド．
  - 使用例: del *.tmp （拡張子が「tmp」であるすべてのファイルを削除）
  - del ファイル名: ファイルを完全削除します（ごみ箱には移動されません）．/f オプションで読み取り専用ファイルも削除

ファイル内容の表示

- **type ファイル名**: テキストファイルの内容を表示します
- **more ファイル名**: 内容をページ単位で表示します
  - スペースキー：次ページへ移動
  - Enter キー：次行を表示
  - Q キー：表示を終了

ファイルとフォルダーの操作

- **move 元名 新名**: ファイルまたはフォルダーの名称変更や移動
- **copy 元名 先名**: ファイルの複製を作成
- **xcopy /s /e /h 元フォルダー 先フォルダー**
  - /s: 非空のサブディレクトリを含めて複製
  - /e: 空のサブディレクトリを含めて複製

- /h: 隠しファイルとシステムファイルも複製

実行例

1. コマンドプロンプトを管理者として実行（手順：Windows キーまたはスタートメニュー，「cmd」と入力，「管理者として実行」を選択）

2. 次のコマンドを実行．notepad はテキストエディタ（メモ帳）を起動するコマンド．

```
dir
cd ..
mkdir test_folder
notepad
```

4.4. エラーメッセージとの付き合い方

エラーメッセージは「失敗」ではなく，**「問題解決へのヒント」**です．エラーメッセージには，どのファイル（File）の何行目（line）で，どのような種類のエラー（例: ModuleNotFoundError）が起きたかなどが記載されています．エラーメッセージの一部や全体をコピーしてインターネットで検索したり，AI アシスタントで調査すると，解決策が見つかることが多いです．

5. Windows での AI 開発環境（入門編）インストール手順

注意:

- 以下の手順では主に**コマンドプロンプト（cmd.exe）**を使用します（4.3 節参照）．一部の操作では「管理者として実行」が必要です（4.3.2 節参照）．
- winget コマンドが利用できない場合は，Microsoft Store から「アプリ インストーラー」をインストールする必要があります．
- 以下のコマンド例では，コメントを示すために REM を使用しています．

5.1. AI 開発環境（入門編）のインストールと設定

5.1.1. AI 開発環境（入門編）の一括インストール

Python は AI 開発で広く使われる言語です．winget は Windows のコマンドラインからアプリをインストールできるツールです．

Python バージョンについて： 2025 年 6 月現在，Python 3.13.4 が最新安定版です．ただし，AI 開発の入門としては Python 3.12 系列の利用を推奨します．これは多くの AI ライブラリが Python 3.12 での動作確認が行われており，安定性が確保されているためです．Anaconda や Miniconda など，他の配布版もありますが，このガイドでは，winget を用いて Python を本体をインストールします．

インストールは，ソフトウェアをコンピュータで実行可能な状態にするための処理のことです．プログラムファイルの配置，システムへの登録，設定などを行います．

winget（Windows Package Manager：Microsoft 公式のソフトウェア管理ツール）は，ソフトウェアのインストールや管理を行うための**コマンドラインツール**（文字コマンドで操作するプログラム）です．インストール作業を簡単なコマンドで実行可能になります．

インストールスコープは，ソフトウェアインストール時の対象ユーザーの範囲の指定です．winget を用いたインストールにおいても，重要な設定です． 例えば，winget では，「--scope machine オプション（コマンドの動作を変更するための追加設定）」を指定して Python をインストールすると，そのコンピュータのすべてのユーザーが Python を使用できます．このオプションを指定しない場合，インストールを実行したユーザーのみが Python を使用可能となります．

- --scope machine: すべてのユーザーが使用可能（インストール時に管理者権限必須）
- --scope user: 現在のユーザーのみが使用可能

以下のコマンドで，AI 開発環境（入門編）を一括でインストールします。

AI 開発環境（入門編）の一括インストール

1. コマンドプロンプトを管理者として実行（手順：Windows キーまたはスタートメニュー，「cmd」と入力，「管理者として実行」を選択）
2. 次のコマンドを 1 つずつ実行（長いコマンドは，表示が複数行に分かれていますが，1 つのコマンドとして実行してください）

```
REM Windows のパス長制限を緩和
reg add "HKEY_LOCAL_MACHINE¥SYSTEM¥CurrentControlSet¥Control¥FileSystem" /v
LongPathsEnabled /t REG_DWORD /d 1 /f
reg query "HKLM¥SYSTEM¥CurrentControlSet¥Control¥FileSystem" /v LongPathsEnabled

REM winget コマンドは必ず 1 つずつ実行すること．（実行完了を待つため）
REM Nvidia CUDA 12.6 をシステム領域にインストール
winget install --scope machine --id Nvidia.CUDA --version 12.6 -e

REM Python, Git, 7zip, Visual Studio Code, Windsurf をシステム領域にインストール
winget install --scope machine --id Python.Python.3.12 --id Python.Launcher --id Git.Git --
id 7zip.7zip --id Microsoft.VisualStudioCode --id Codeium.Windsurf -e --silent
```

コマンドの説明：

- **パス長制限の緩和:** Windows の既定では 260 文字のパス長制限がありますが，AI 開発では深いフォルダ階層を使うことがあるため，この制限を緩和します．
- **CUDA 12.6:** NVIDIA GPU を使用する場合に必要な計算基盤．GPU がない場合はスキップ可能です．
- **–scope machine:** システム全体（全ユーザー）で使えるようにインストールします．
- **–silent:** インストール中の対話的な質問をスキップし，既定の設定で自動的にインストールします．

- -e: パッケージ ID の完全一致検索（曖昧な検索を避けます）.

よくある間違いの回避策

- コマンドの途中で改行しない
- 管理者権限で実行していることを確認
- インストール中に PC をスリープモードにしない
- Python がすでにインストール済みのときは，「winget install --scope machine --id Python.Python.3.12...」の実行のときに，「Modify Repair Uninstall」画面が表示されることがあります．この画面が出た時は，この画面では「Cancel」をクリックして次に進んでください.
- winget コマンドが認識されない場合は，Windows 10 の場合，Microsoft Store から App Installer をインストールする必要があります.

次に，パスを通す設定，環境変数の設定，Python 開発に必要な VS Code 拡張機能のインストールを行います．最初に**必ず新しいコマンドプロンプトを管理者として実行**します.

AI 開発環境（入門編）の各種設定

1. コマンドプロンプトを管理者として実行（手順：Windows キーまたはスタートメニュー，「cmd」と入力，「管理者として実行」を選択）
2. 次のコマンドを実行

```
set "INSTALL_PATH=C:¥Program Files¥Python312"
echo %PATH% | find /i "%INSTALL_PATH%" >nul
if errorlevel 1 setx PATH "%PATH%;%INSTALL_PATH%" /M >nul
echo %PATH% | find /i "%INSTALL_PATH%¥Scripts" >nul
if errorlevel 1 setx PATH "%PATH%;%INSTALL_PATH%¥Scripts" /M >nul

set "CUDA_PATH=C:¥Program Files¥NVIDIA GPU Computing Toolkit¥CUDA¥v12.6"
if exist "%CUDA_PATH%" setx CUDA_PATH "%CUDA_PATH%" /M >nul
if exist "%CUDA_PATH%" setx CUDNN_PATH "%CUDA_PATH%" /M >nul

set "NEW_PATH=C:¥Program Files¥Git¥cmd"
if exist "%NEW_PATH%" echo %PATH% | find /i "%NEW_PATH%" >nul
```

```
if exist "%NEW_PATH%" if errorlevel 1 setx PATH "%PATH%;%NEW_PATH%" /M >nul

set "NEW_PATH=C:¥Program Files¥7-Zip"
if exist "%NEW_PATH%" echo %PATH% | find /i "%NEW_PATH%" >nul
if exist "%NEW_PATH%" if errorlevel 1 setx PATH "%PATH%;%NEW_PATH%" /M >nul

set "NEW_PATH=C:¥Program Files¥Microsoft VS Code"
if exist "%NEW_PATH%" echo %PATH% | find /i "%NEW_PATH%" >nul
if exist "%NEW_PATH%" if errorlevel 1 setx PATH "%PATH%;%NEW_PATH%" /M >nul

set "NEW_PATH=C:¥Program Files¥Windsurf"
if exist "%NEW_PATH%" echo %PATH% | find /i "%NEW_PATH%" >nul
if exist "%NEW_PATH%" if errorlevel 1 setx PATH "%PATH%;%NEW_PATH%" /M >nul


if exist "C:¥Program Files¥Microsoft VS Code¥bin" cd "C:¥Program Files¥Microsoft VS Code¥bin"
if exist "C:¥Program Files¥Microsoft VS Code¥bin" code --install-extension ms-python.python
if exist "C:¥Program Files¥Microsoft VS Code¥bin" code --install-extension ms-python.vscode-pylance
if exist "C:¥Program Files¥Microsoft VS Code¥bin" code --install-extension MS-CEINTL.vscode-language-pack-ja
if exist "C:¥Program Files¥Microsoft VS Code¥bin" code --install-extension dongli.python-preview
```

PATH 設定コマンドの説明

- これらのコマンドは，各ツールの実行ファイルが置かれているフォルダを PATH 環境変数に追加します．
- if exist と if errorlevel を使って，既に設定されている場合は重複して追加しないようにしています．
- /M オプションによりシステム環境変数として設定されます．

- これらのコマンドを実行において，特に結果は表示されません．エラーメッセージが出ない場合は実行に成功しています．

Python と Git のインストール確認

新しいコマンドプロンプトを開き，以下のコマンドでバージョンが表示されれば成功です．

```
python --version
git --version
```

簡単な Python の動作確認

コマンドプロンプトで python と入力すると，Python の対話モードが起動します．以下のコードを入力してエンターキーを押し，結果が表示されるか確認してみます．

```
print("Hello, AI World!")
```

>>>と表示されている行は Python の入力受付中であることを示します．print("Hello, AI World!")と入力してエンターキーを押すと，その下の行に Hello, AI World!と表示されます．確認できたら exit()と入力してエンターキーを押すと対話モードを終了できます．もしここでエラーが出る場合は，8 章のトラブルシューティングを参照してください．

5.1.2. 各ツールの概要

- **Python:** AI やデータサイエンスの世界で最も広く使われているプログラミング言語です．
- **Git:** プログラムコードの変更履歴を記録・管理するシステム．GitHub との同期にも使用されます．
- **7-Zip:** ファイルの圧縮・展開を行うツール．
- **Visual Studio Code (VS Code):** 多機能なコードエディタ．
- **Windsurf:** AI 支援機能を特徴とする開発環境．

5.2. Python ライブラリの一括インストール

pip を使って，Python の AI 開発でよく使うライブラリを一括でインストールします．**pip** とは: Python パッケージマネージャ（Python の追加機能を管理するためのツール）です．ライブラリ（プログラムで使用できる機能の集合）のインストールや管理を行うためのツールです．

- 使用例: pip install numpy
- このガイドでは pip install コマンドは，管理者権限で実行することを推奨します．Windows のユーザ名が日本語であった場合に発生する可能性があるトラブルを防止するためです．

pip 基本コマンド

- pip install [パッケージ名]: 指定したパッケージ（機能をまとめたソフトウェア部品）をインストールします．
- pip install --upgrade [パッケージ名]（--upgrade：既存パッケージを最新版に更新するためのオプション）: 指定したパッケージを更新します．
- pip uninstall [パッケージ名]: 指定したパッケージを削除します．
- pip list: インストール済みパッケージを一覧表示します．
- pip show [パッケージ名]: 指定したパッケージの詳細情報を表示します．

重要：「pip install -U torch torchvision torchaudio --index-url https://download.pytorch.org/whl/cu126」の実行では、PyTorch 公式サイト（https://pytorch.org/get-started/locally/）で，自身の環境（Windows, **Pip**, Python, CUDA）に合った**最新のインストールコマンドを必ず確認**すること．特に GPU を利用する場合は，適切な CUDA バージョンの選択が重要です（7.3 節参照）

以下のコマンドで，AI 開発に必要なライブラリを一括でインストールします（2025 年 6 月版）：

主要な Python ライブラリの一括インストール

1. コマンドプロンプトを管理者として実行（手順：Windows キーまたはスタートメニュー，「cmd」と入力，「管理者として実行」を選択）
2. 次のコマンドを 1 つずつ実行（長いコマンドは，表示が複数行に分かれていますが，1 つのコマンドとして実行してください）

```
REM pip を最新版に更新
python -m pip install -U pip
```

```
REM PyTorch のインストール（CUDA 12.6 対応版）
pip install -U torch torchvision torchaudio --index-url https://download.pytorch.org/whl/cu126

REM その他の AI ライブラリを一括インストール
pip install jupyter ipykernel numpy pandas scikit-learn matplotlib japanize-matplotlib
seaborn scipy opencv-python pillow timm pygame streamlit
```

注意： 上記のコマンドは CUDA 12.6 を使用する場合の例です．2025 年 6 月現在，
PyTorch 2.6.0 および 2.7 系で CUDA 12.6 が正式サポートされています．

インストールされる Python ライブラリの説明

- **PyTorch:** ディープラーニング（深層学習）モデルを効率的に作るための主要なライブラリです．
- **torchvision:** 画像処理用の PyTorch ライブラリ
- **torchaudio:** 音声処理用の PyTorch ライブラリ
- **jupyter:** 開発エディタの Jupyter Notebook
- **NumPy, Pandas, Scikit-learn:** 数値計算（NumPy），表形式データの操作・分析（Pandas），古典的な機械学習（Scikit-learn）のための定番ライブラリ
- **Matplotlib / Seaborn:** データの分布や学習結果などをグラフで可視化するためのライブラリで
- **japanize-matplotlib:** Matplotlib で日本語を表示するためのライブラリ
- **scipy**: 科学計算ライブラリです（SciPy：高度な科学計算機能を提供）．
- **opencv-python**: 画像処理・コンピュータビジョン用ライブラリ．
- **pillow**: 画像処理ライブラリです（Pillow：画像の読み込み，編集，保存機能を提供）．
- **timm:** PyTorch Image Models（事前学習済みモデル集）
- **pygame:** ゲーム開発・インタラクティブアプリケーション用ライブラリ
- **streamlit**: Web ベースの対話アプリ開発ツール

インストール後の確認コマンド

```
# GPU 情報確認
wmic path win32_VideoController get name
```

```
# Python バージョン確認
python --version

# PyTorch と CUDA 対応確認
python -c "import torch; print(torch.__version__, torch.cuda.is_available())"
```

ライブラリの動作確認

インストールしたライブラリが正しく使えるか，簡単な Python コードで確認してみます．コマンドプロンプトで python と入力し，対話モードに入って以下のコードを実行します．

```
# PyTorch のバージョン確認と GPU 利用可能性の確認
import torch
print("PyTorch Version:", torch.__version__)
print("CUDA Available:", torch.cuda.is_available()) # GPU が使えるか（True なら OK）
if torch.cuda.is_available():
    print("CUDA Version:", torch.version.cuda) # PyTorch が対応している CUDA のバージョン

# NumPy のバージョン確認
import numpy as np
print("NumPy Version:", np.__version__)

# Pandas のバージョン確認
import pandas as pd
print("Pandas Version:", pd.__version__)

# Matplotlib のバージョン確認
import matplotlib
print("Matplotlib Version:", matplotlib.__version__)
```

エラーが出ずにバージョン情報などが表示されれば成功です．特に CUDA Available: True と表示されていれば，PyTorch が GPU を認識しており，GPU を使った計算が可能です．False と表示された場合は，GPU が搭載されていないか，CUDA ドライバや PyTorch のインストールに問題がある可能性があります．確認できたら exit()と入力して対話モードを終了します．動作確認でエラーが出た場合は，8 節のトラブルシューティングを参照してください．

6. 開発エディタの活用

5.1 でインストール・設定した Visual Studio Code と Windsurf，および 5.2 でインストールした Jupyter Notebook について，それぞれの特徴と適した用途を説明します．開発エディタにはそれぞれ特徴があり，併用も可能です．

6.1. 開発エディタの概要

① Visual Studio Code (VS Code)

特徴: Microsoft が開発するコードエディタです．「拡張機能」により，Python 開発支援など様々な機能を追加してカスタマイズできます．

適した用途: プログラム開発，複数のファイルで構成されるプロジェクトの管理．

主な AI 機能: GitHub Copilot や Claude などの拡張機能を導入することで，コード補完やチャットによる質問応答，コード生成といった AI 支援機能を利用できます．

5.1.1 でインストールした拡張機能の説明

- **ms-python.python:** Python 言語の基本サポート（シンタックスハイライト，デバッグ，リンティングなど）
- **ms-python.vscode-pylance:** Python の高度な言語サポート（型チェック，自動補完，定義へのジャンプなど）を提供する言語サーバー
- **MS-CEINTL.vscode-language-pack-ja:** VS Code の日本語化パック
- **dongli.python-preview:** Python コードの実行結果をプレビュー表示する拡張機能

② Jupyter Notebook

Web ブラウザ上で動作する対話的な開発環境です．「セル」と呼ばれるブロック単位でコードを書き，実行するとその結果（数値，表，グラフなど）がすぐ下に表示されます．シンプルなインターフェースになっています．より多くの機能が必要になったら JupyterLab への移行も検討できます．

適した用途: データ分析，アルゴリズム，学習過程や結果の可視化，デモンストレーション．

主な AI 機能

- **コードの自動生成:**「# titanic データセットを読み込み，最初の 5 行を表示」といったコメント（自然言語）をセルに書くと，AI が対応する Python コードを自動で生成してくれます．
- **コードと結果の解説:** 実行したコードや表示されたグラフが何をしているのか，自然言語で解説を生成させることができ，学習の助けになります．

③ Windsurf

特徴: AI 統合開発環境（IDE）です．エディタ機能と AI との連携が特徴です．

主な AI 機能

- **Cascade（エージェント型チャットボット）:**「# ファイルアップロード機能を持つ Web アプリの雛形を作って」のような指示からコード生成やプロジェクト構築を行うなど，対話によるコード生成が可能です．
- **Context Awareness（文脈認識）:** 開いているプロジェクト全体のコードを AI が自動的に理解し，別のファイルに存在する関数や変数を考慮した上で，適切なコード補完や修正案を提示します．

6.2. AI エディタ Windsurf の活用

Windsurf は、AI 機能を統合したコードエディタです。VS Code をベースにしており、VS Code と操作性が似ています。。

メリットと機能

- **無料で利用可能**: 学生や個人開発者向けに無料プランが提供されています。
- **VS Code ベースで VS Code 拡張がそのまま利用可能**: VS Code の拡張機能と高い互換性を持つため、多くの拡張機能をそのまま利用できます。
- **Windsurf Tab**: Tab キーを活用したコード補完および提案機能を提供します。

Cascade 機能

Cascade はコード生成や実行を支援する AI 機能です：

- **開始方法:** Cmd/Ctrl + L キーで Cascade パネルを開きます。
- **使用例:** 例えば、「Python で折れ線グラフのサンプルコードを作成して」といった自然言語でリクエスト可能です。
- **Turbo モード:** 生成したコマンドを自動実行し、ワークフローをシームレスに進める機能です。

無料プランの機能（2025 年 6 月現在）

- **プロンプトクレジット:** 月 25 クレジット（GPT-4.1 プロンプト約 100 回分に相当）
- **AI モデル:** GPT-4.1（0.25 クレジット/プロンプト）、Claude 3.7 Sonnet（1 クレジット/プロンプト）、DeepSeek-V3-0324（無料）など、複数のモデルを利用できます。
- **その他機能:** 無制限の Fast Tab、SWE-1 Lite、Command などが含まれます。

API Key 要件

- **API Key 不要:** サードパーティの API キーは不要です。
- **設定:** Windsurf アカウントの作成のみが必要です（無料）。
- **利用開始:** ダウンロード → アカウント登録 → 利用開始できます。

Cursor と Windsurf の比較

Claude 3.7 Sonnet の利用回数では、Cursor 無料版は月 50 回、Windsurf 無料版は月約 25 回利用可能です。ただし Windsurf では、無料で利用できる他のモデル（DeepSeek-V3-0324 など）が無制限で利用できる点が特徴です。

Windsurf の起動と初回設定

1. Windsurf を起動（スタートメニューまたは「windsurf」コマンド）
2. Get Started をクリック
3. VS Code から設定を引き継ぎたいときに限り「Import from VS Code」，ふつうは「Start fresh」
4. 設定を続行
5. 「Log in to Windsurf」の画面で、「Sign up」 をクリックし，アカウントを新規作成。この時の登録を覚えておき、次回からは「Log in」.

このとき、**Google アカウントを用いて、Windsurf のアカウント登録可能**。以上でインストールと初期設定は完了

6. 動作確認のため、Ctrl+L で Cascade を開き，「折れ線グラフを描くコードを出して」などと入力.

Windsurf の推奨モデル（クレジット節約のため）

1. DeepSeek-V3-0324 (0 クレジット・恒久的)
2. SWE-1 (0 クレジット・期間限定)
3. SWE-1-lite (0 クレジット・軽量版)

DeepSeek-V3-0324 モデルの選択手順

1. Cascade パネル（Ctrl + L）を開く
2. 入力欄上部のモデル選択ドロップダウンをクリック
3. 「DeepSeek-V3-0324」を選択

Python コードの実行手順

① "Hello World"プログラム

1. 新しいファイルを作成し、hello.py という名前で保存.
2. print("Hello World")と入力.
3. ターミナルを開く（View > Terminal）.
4. 実行方法：
    - ターミナルで python hello.py と入力して実行.
    - または Cmd/Ctrl+I で「実行して」とリクエスト.

② 折れ線グラフのプロットプログラム

1. plot_graph.py というファイルを作成し，以下のコードを入力.

```
import matplotlib.pyplot as plt
import numpy as np
# データの生成
x = np.linspace(0, 10, 100)
y = np.sin(x)
# グラフのプロット
plt.figure(figsize=(8, 6))
plt.plot(x, y, 'b-', linewidth=2, label='sin(x)')
plt.title('Sine Wave')
plt.xlabel('x')
plt.ylabel('sin(x)')
plt.grid(True)
```

```
plt.legend()

# グラフの保存と表示

plt.savefig('sine_wave.png')

plt.show()
```

2. 実行するとグラフウィンドウが表示され，同時にプロジェクトディレクトリに画像ファイルも保存される．

7. このガイドの AI 環境開発環境（入門編）のメリットと注意点

7.1. メリット

- **シンプルさ:** インストールするツールの数を絞り，AI 開発を始めやすい構成です．
- **標準的な構成:** Python + pip の構成は広く利用されています．
- **開発エディタの選択肢:** 開発スタイルに合わせて，VS Code, Jupyter Notebook, Windsurf といったエディタから選択できます．

7.2. 注意点と潜在的な課題

- **バージョン互換性:** GPU を使う場合，NVIDIA CUDA と PyTorch（が要求する CUDA バージョン）の組み合わせには注意が必要です（7.3 節参照）．
- **コマンド操作への慣れ:** ライブラリのインストールなど，基本的なコマンドプロンプトの操作に慣れる必要があります（4 章参照）．
- **複数プロジェクトの管理:** このシンプルな構成では，すべてのライブラリが PC 全体で共有されます．複数のプロジェクトで異なるバージョンのライブラリを使いたい場合は，環境管理（例: Python の venv 機能や conda）が必要になる点に留意してください．

7.3. 注意点：GPU 利用時の PyTorch のバージョン互換性

1. 使用したい PyTorch のバージョンを決める: 最新安定版が推奨されます．
2. PyTorch 公式サイトのインストールページを確認する．
    - PyTorch 公式サイト（https://pytorch.org/get-started/locally/）にアクセスします．
    - 「PyTorch Build」（Stable），「Your OS」(Windows)，「Package」(**Pip** を選択)，「Language」(Python)，「Compute Platform」を選択します．
    - 「Compute Platform」で，利用したい CUDA のバージョン（例: CUDA 11.8, CUDA 12.6）を選択します．
    - 選択すると，下にインストールコマンドが表示されます（例: pip install torch ... --index-url .../cu126）．このコマンドの末尾（cu126 など）が，PyTorch が対応する CUDA バージョンを示します．

3. 必要な CUDA のバージョンを確認する.

   最新の互換性情報は https://docs.nvidia.com/cuda/cuda-toolkit-release-notes/ で確認してください.

8. 一般的なトラブルシューティング

○　問題: python や pip コマンドが認識されない

エラーメッセージ例: 'python' is not recognized as an internal or external command, operable program or batch file.

原因: winget でのインストールに失敗したか，パス（PAT）が正しく設定されていない可能性があります．

対処法: PC を再起動してみてください．それでも解決しない場合は，Windows の「設定」→「システム」→「バージョン情報」→「システムの詳細設定」→「環境変数」を開き，システム環境変数の Path に Python のインストールフォルダ（例: C:¥Program Files¥Python312）が含まれているか確認してください．

○　問題: ライブラリが見つからない

エラーメッセージ例: ModuleNotFoundError: No module named 'xxx'

原因: 必要なライブラリ（例: torch, numpy, pandas など）が正しくインストールされていないか，コマンドを実行している Python 環境とは異なる環境にインストールされている可能性があります．

対処法: pip install <ライブラリ名>コマンドを再実行してみてください．また，pip list コマンドでインストール済みのライブラリ一覧を確認できます．

○　問題: GPU が認識されない，または GPU 関連のエラーが出る

エラーメッセージ例: CUDA is not available など

原因: CUDA 12.6 と，pip でインストールした PyTorch の CUDA バージョンの間に互換性がないことが主な原因です（7.3 節参照）．その他，CUDA のインストール失敗，GPU ハードウェアの問題なども考えられます．

対処法:

1. 7.3 節の手順を参考に，インストールされている CUDA 12.6 と，pip でインストールした PyTorch の CUDA バージョン（コマンドの--index-url 部分や PyTorch 公式サイト

で確認できる）の互換性を再確認してください．必要であれば，CUDA または PyTorch を再インストールしてください．

2. コマンドプロンプトで nvidia-smi を実行し，NVIDIA ドライバが正しくインストールされ，GPU がシステムに認識されているか確認してください．GPU の状態やドライババージョンが表示されます．

3. Python インタープリタを起動し，以下のコードを実行して PyTorch が GPU（CUDA）を認識しているかプログラムで確認してください．

   ```
   import torch
   print(torch.cuda.is_available())
   print(torch.version.cuda) # PyTorch が対応している CUDA バージョンを表示
   ```

   True が返ってくれば PyTorch は GPU を認識しています．False の場合は，バージョン不整合の可能性が高いです（7.3 節参照）．pip uninstall torch で一度アンインストールし，PyTorch 公式サイトで確認した正しいコマンドで PyTorch を再インストールすることを検討してください．

4. PC の再起動も試してみてください．

5. 次の手順で，NVIDIA ドライバの更新も試みてください．

   - NVIDIA 公式サイトのドライバダウンロードページにアクセス．
   - GPU モデル、OS、ドライバタイプを選択．
   - 最新版をダウンロードし、インストーラーを実行（「クリーンインストール」推奨）．
   - 確認 (コマンドプロンプト): nvidia-smi

9. このガイドを終えたら：次のステップへ

これで，Windows PC 上に AI 開発の基本的な環境が構築できました．

この後，AI 開発やプログラミング学習をさらに進めるためのステップをいくつか案内します．

- **Python の基本を学ぶ:** まだ Python に慣れていない場合は，変数，オブジェクト，メソッド，クラス，インポート，関数，データ型，制御構文（if 文，for 文）といった基本的な文法や概念を学ぶのが最初のステップです．
- **Python の基本的なライブラリを使ってみる:** インストールした NumPy や Pandas などを使って，データの読み込み，表示，集計などの操作を試してみましょう．データに触れることは AI 開発の基礎になります．
- **開発環境を使いこなす:** Windsurf の AI 機能やデバッグ機能を使ってみたり，Jupyter Notebook でデータ分析を進めてみたりなど，開発環境の便利な機能を積極的に使ってみましょう．
- **仮想環境について学ぶ:** 複数のプロジェクトで異なるライブラリのバージョンを使いたいなど，より進んだ環境管理が必要になったら，Python の仮想環境（venv や conda など）について学ぶことをお勧めします．

このガイドで構築した環境を使い，様々なコードを試すことができます．AI 開発の学習が充実したものとなることを期待しています．