

Assemblage of AI Dev Environment

By Eugene Asahara

Last Updated: January 19, 2026

This document walks you through setting up the development environment required to follow along with the tutorials in the virtual book, [Assemblage of Artificial Intelligence](#). The environment includes Neo4j, Visual Studio Code (VS Code), and Python. Follow the instructions step-by-step.

Prerequisites

This tutorial assumes you are working on a personal or work laptop. Most of the setup will take place locally.

Minimum Requirements

- Local admin rights – Needed to install:
 - Neo4j Desktop
 - [Visual Studio Code \(VS Code\)](#) and [Python \(3.10.2 or later\)](#)
 - [Protégé](#)
 - Git for Windows (includes Git Bash)
- **Internet access** – Required for downloading installers, extensions, cloning the GitHub repo, and accessing the sample database files.
- **Disk space** – At least 10 GB free for SQL Server, database files, Neo4j, and Python packages.
- **GitHub account**– For cloning the assemblage-of-ai GitHub repository and participating in any future updates.
 - [GitHub Desktop](#) (optional) – A graphical interface for cloning and managing repositories.

Alternative Arrangements

If you're working on a **restricted corporate laptop** or already have parts of the stack available, you may not need to install everything:

- **Python:**
 - If Python is already installed and you're using an IDE like PyCharm, Anaconda, or JupyterLab, you can use that instead of VS Code.
- **Neo4j:**
 - If you're already using Neo4j (Desktop, Server, or Aura), you may skip Neo4j Desktop installation.
 - **Important:** You must be able to load plugins (e.g., APOC and n10s) and import data into the database.

Clone the Assemblage of AI Repository

Before setting up Python and VS Code, you'll need to clone the Assemblage of AI GitHub repository to your local machine. This will give you access to all the scripts, notebooks, and configuration files used throughout the setup.

a. Create a Local Folder

I recommend creating a base directory to hold the project: `C:\MapRock\`

If this folder doesn't exist, create it manually or let Git do it during the clone step.

b. Install Git (if not already installed)

Download Git for Windows, which includes Git Bash:

- <https://git-scm.com/download/win>

Follow the installation prompts. Leave most options at their defaults.

c. Clone the MapRock/assemblage-of-artificial-intelligence Repository

Open **Git Bash** (or Command Prompt if Git is on your PATH), and run:

```
git clone https://github.com/MapRock/assemblage-of-artificial-intelligence.git  
C:/MapRock/assemblage-of-artificial-intelligence
```

This will download the full repository contents into *C:\MapRock\assemblage-of-artificial-intelligence*.

Python and VS Code Setup

a. Install Python (version 3.10.2 or later)

- Visit: <https://www.python.org/downloads/>
- Download and install version 3.10.2 or later.
- Add Python to PATH during installation.

b. Install Visual Studio Code (VS Code)

- Visit: <https://code.visualstudio.com/>
- Download and install VS Code.

c. Add VS Code Extensions

- Launch VS Code.
- Click the Extensions icon in the Activity Bar on the side (or press Ctrl+Shift+X).
- In the Extensions pane, type the following names one by one into the search bar:
 - Jupyter by Microsoft: Enables support for Jupyter notebooks (.ipynb files).
 - Neo4j for VS Code by neo4j.com: Provides database integration, query execution, and exploration for Neo4j.
 - Cypher Query Language: Adds syntax highlighting for Cypher if not already covered by the Neo4j extension.
- Click Install on each relevant result.
- Reload VS Code when prompted after installation.

d. Open Project in VS Code

- Launch VS Code.
- From the File menu, select Open Folder...
- Navigate to:
C:\MapRock\assemblage-of-artificial-intelligence\book_code\src
and click *Select Folder*.
- This folder contains all the Python scripts and configuration files needed to run the Assemblage of AI examples.
- If prompted to install Python or related extensions, accept the suggestions.
- Make sure the Explorer pane (left sidebar) is open so you can see the file structure.
- You'll be editing files like .env, *.ipynb notebooks, and Python modules here.

e. Set Up the Environment Variables

This step configures essential settings required by the Assemblage of AI Python code to connect to external services like OpenAI and your local Neo4j instance. These settings are stored in a special file named `.env`, short for "environment." The `.env` file is not part of the code itself but is automatically loaded at runtime to provide secure and customizable configuration values—such as API keys and file paths—withoudt hardcoding them. By renaming the provided `.env.example` and editing its contents, you create a personal configuration file tailored to your system.

- In the src folder of the cloned repository, locate the file named `.env.example`.
 - Rename the file to `.env`
- Select the `.env` file in the src folder and set the following contents:
 - `OPENAI_API_KEY=[Your OpenAI key]"`
`CHATGPT_MODEL="gpt-4.1-mini" # gpt-4.1-nano will work for simpler prompts.`
`CHATGPT_EMBEDDING_MODEL="text-embedding-3-large"`
`CHATGPT_MAX_RESPONSE_TOKENS=800`
 - `NEO4J_URI="bolt://localhost:7687" # default URI for Neo4j`
`NEO4J_User="[Your UserID]"`
`NEO4J_Password="[Your password]"`

Note: The dbms-xxxxxxxx folder name is unique to your Neo4j installation. You can find it by opening Neo4j Desktop → Manage → Open Folder → Import.