

Building an LLM from Zero

Chapter 1: Environment Setup

Module 1: Foundations

Truong (Jack) Luu · jackluu.io/book

Where we are

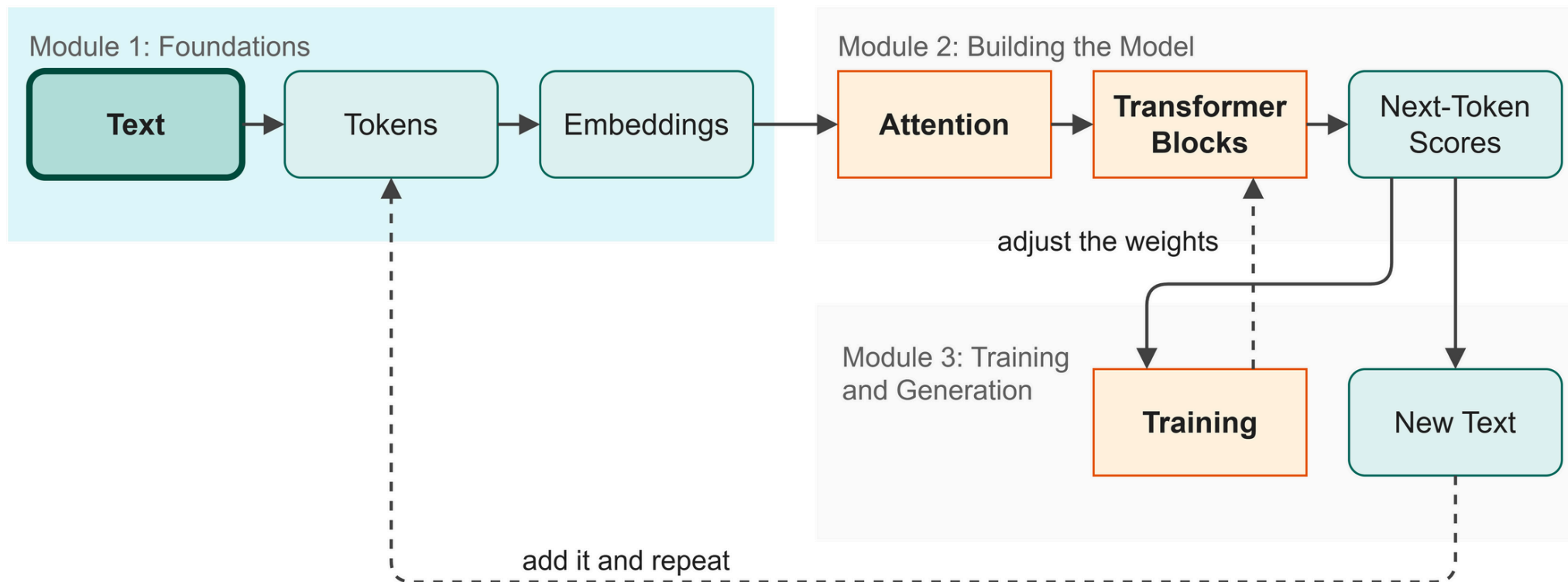


Figure 1.1: Before you build, you set up your tools.

Nothing on this map runs until your laptop can run Python and PyTorch. Today we make that true, step by step.

Why set up carefully

"It works on my machine" is the oldest problem in software teams.

- A company building a house-style email assistant standardizes every developer's environment
- **Virtual environments** (an isolated toolbox for one project's packages) and **fixed package versions** make the code run the same on every laptop and on the server
- New hires start working on day one instead of debugging their setup

The same discipline makes this book work: the code runs for you exactly as it runs in the book.

In this chapter you will

1. Install **Python** and a terminal to run commands
2. Download the project code using **Git**
3. Set up an isolated workspace for your packages
4. Download the training dataset

Words to know (each is glossed again where it first appears):

- **Terminal**: a text-based window where you type commands
- **Repository**: a folder of code stored online
- **Virtual environment**: an isolated toolbox for a single project's packages
- **Package manager**: a tool that downloads and installs code libraries

Step 1: Install Python

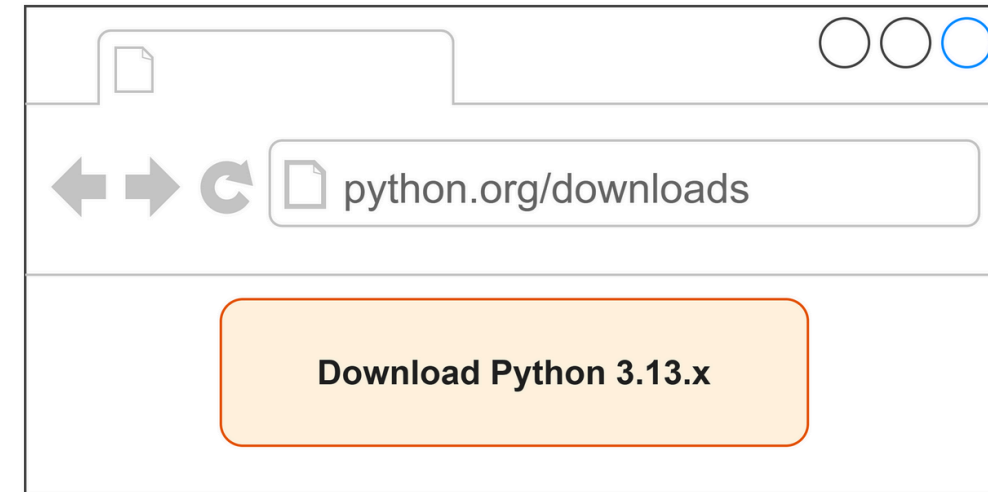
You need Python **3.10 or newer**.

Windows

1. Go to **python.org/downloads**, download Python 3.13.x
2. Run the `.exe` installer
3. **Check "Add Python to PATH"** on the first screen
(PATH: where the computer looks for programs)
4. Click "Install Now"

macOS: run the `.pkg` installer and follow the prompts.

Miss the PATH box and Python will not be found when you type commands.



Step 2: Open a terminal

The **terminal** (a text window where you type commands) is where you run every script in this book.

Windows

Press `Win + R`, type `cmd`, press Enter. A black window with a blinking cursor appears.

Check that Python is installed:

```
$ python --version
```

You should see Python 3.10 or newer. On macOS, if `python` is not found, type `python3 --version` instead.

macOS

Press `Cmd + Space`, type `Terminal`, press Enter.

Step 3: Install Git

Git downloads the project code from the internet.

Windows

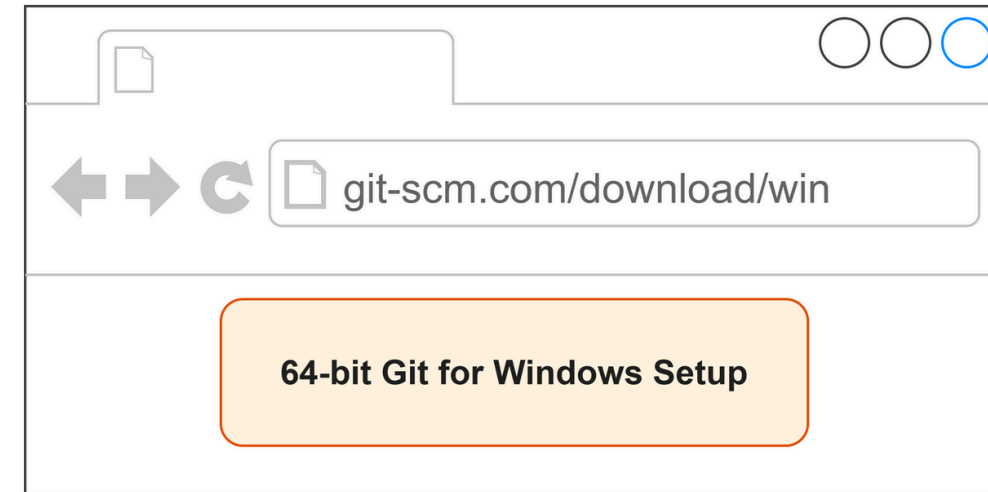
1. Go to **git-scm.com/download/win**
2. Download the "64-bit Git for Windows Setup"
3. Run the installer and click "Next" through every screen

macOS: type `git --version` in the terminal. If Git is missing, macOS offers to install the "Command Line Developer Tools". Click Install.

Verify (open a new terminal window on Windows):

```
$ git --version
```

You should see a message confirming your Git version.



Step 4: Download the project

Download the whole **repository** (a folder of code stored online) with one command:

```
$ git clone https://github.com/jackluucoding/build-llm-from-zero.git
```

Then move into the project folder:

```
$ cd build-llm-from-zero
```

Every command in this book is typed from inside this folder. If a script is "not found", check where you are first.

Step 5: Create a virtual environment

Different projects need different versions of the same library. A **virtual environment** gives this project its own isolated box of packages.

From inside the `build-llm-from-zero` folder:

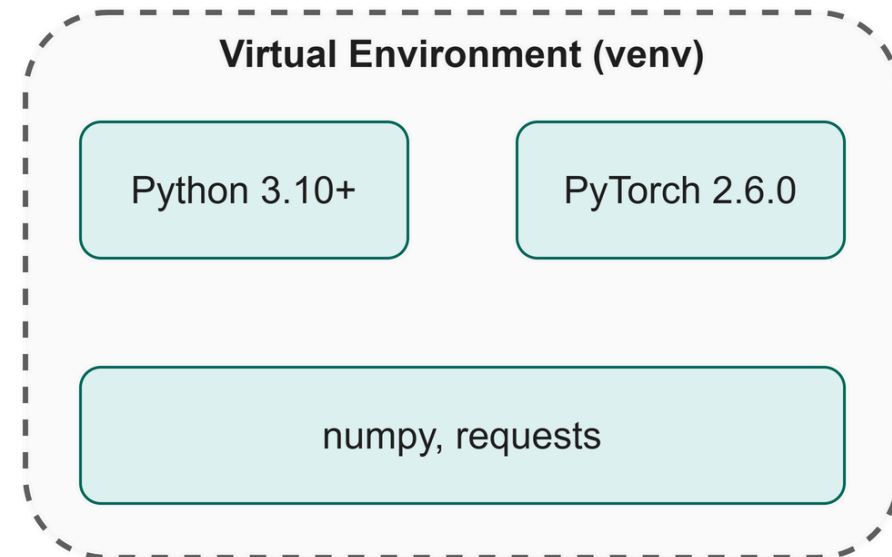
Windows

```
$ python -m venv venv  
$ venv\Scripts\activate
```

macOS

```
$ python3 -m venv venv  
$ source venv/bin/activate
```

(venv) appears at the start of your prompt. Run the activate command every time you open a new terminal window.



Step 6: Install PyTorch and dependencies

With the virtual environment active, use `pip`, Python's **package manager** (a tool that downloads and installs code libraries).

First PyTorch, the CPU-only build:

```
$ pip install torch==2.6.0 --index-url https://download.pytorch.org/whl/cpu
```

Then the other libraries:

```
$ pip install numpy requests matplotlib
```

CPU-only is deliberate. This book never needs a GPU, so nobody needs a special laptop.

Step 7: Download the Shakespeare dataset

The model will learn to write from the complete works of Shakespeare (the Tiny Shakespeare text).

```
$ python src/utils/download_data.py
Dataset ready: src/data/shakespeare.txt (1089 KB)
```

```
Dataset statistics:
```

```
  Total characters : 1,115,394
```

```
  Unique characters: 65
```

```
First 200 characters:
```

```
-----
```

```
First Citizen:
```

```
Before we proceed any further, hear me speak.
```

```
...
```

Two numbers to remember: **65** different characters become the vocabulary in Chapter 4, and **1,115,394** characters become the training data.

Step 8: Verify everything works

```
$ python src/ch00_setup_check.py
Chapter 1: Checking your environment...

[OK] Python: 3.13.13
[OK] PyTorch: 2.6.0+cpu
[OK] NumPy: 2.5.3
[OK] Requests: 2.34.2
[OK] Shakespeare dataset found: 1,115,394 characters
[OK] Quick tensor test passed

All checks passed! You are ready to start Chapter 2.
```

If you see this, you are fully ready to begin. Your version numbers may differ; every line must say [OK].

Watch out: "No module named 'torch'"

If a script fails with

```
ModuleNotFoundError: No module named 'torch'
```

your virtual environment is **not active**.

- Windows: `venv\Scripts\activate`
- macOS: `source venv/bin/activate`

Look for `(venv)` at the start of the prompt. No prefix, no PyTorch.

Discussion

Why not just install everything once, globally?

In pairs, 3 minutes:

1. You work on two projects. One needs PyTorch 2.6.0, the other an older version. What happens without a virtual environment?
2. Your teammate says "it works on my machine". Name two things that could differ between your machines.

Take two answers from the room.

Recap: today in 4 points

1. You installed **Python** (the language), **Git** (the code downloader), **PyTorch** (the math library) and a **terminal**.
2. A **virtual environment** keeps this project's packages isolated.
3. Always **activate** the virtual environment before working on this project.
4. The dataset is **1,115,394 characters** with **65 different characters**, and the setup check says **All checks passed!**

Your workbench is ready. From here on, every chapter builds one stage of the map.

Check your understanding

1. What does a virtual environment do?
2. How do you know if your virtual environment is active?
3. What is the command to move into the project folder?

Next up: Chapter 2, What Is an LLM?

- The one rule every language model follows: **predict the next token**
- No code to build next time; we look at the whole map before zooming in

Before next class: run `python src/ch00_setup_check.py` once more from a fresh terminal. It must print **All checks passed!** with the virtual environment active.

References

Book and figures

- Luu, T. (2026). *Building an LLM from Zero*, Chapter 1. <https://jackluu.io/book/>
- Figures 1.1 to 1.4 are the book's own diagrams (draw.io sources in the book repository, `diagrams/`). The browser windows are generic sketches, not screenshots.
- Terminal output: `src/utls/download_data.py` and `src/ch00_setup_check.py`, run on the author's laptop.

Further reading (from the chapter)

- Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., ... Chintala, S. (2019). *PyTorch: An imperative style, high-performance deep learning library* (arXiv:1912.01703). arXiv.

Dataset

- Tiny Shakespeare, from Andrej Karpathy's char-rnn project: <https://github.com/karpathy/char-rnn>