

# Workshop Prerequisites: A Guide to Setting Up Your Local AI Development Environment

Welcome to the workshop! To ensure we can dive straight into the practical sessions without delays, please follow this guide to install the necessary software and libraries on your laptop **at least two days before the event**.

This guide explains what each tool does, why it's important, and how to install and configure it correctly.


**Estimated Time & Disk Space:** \* **Time:** 30–60 minutes, depending on your internet speed. \* **Disk Space:** Approximately 15–20 GB for all tools and models.

---

## Quick Setup Checklist

For experienced users who just need to verify their setup, here is a quick checklist.

- [ ] **Python 3.9+** with `pip` or `conda`.
- [ ] **Windsurf IDE:** The primary code editor for the workshop.
- [ ] **PyTorch 2.4+:** Core deep learning framework (latest CUDA version).
- [ ] **Hugging Face Libraries:** `transformers`, `accelerate`, `bitsandbytes`.
- [ ] **Ollama:** Local LLM runner.
- [ ] **Ollama Models:** `ollama pull phi3` and `ollama pull llama3`.
- [ ] **TabbyML:** Self-hosted code assistant (Docker recommended).
- [ ] **Git & Command-Line Tools:** For version control and terminal access.

 *Total disk space after installation and model downloads: ~10–15 GB.*

---

## 1. Understanding Your Toolkit: What and Why

Here's a breakdown of the software you will be installing. Understanding their roles will help you see how they fit together.

- **Python (version 3.9 or newer)**
  - **What it is:** A versatile programming language that is the standard for machine learning.
  - **Why we need it:** All our scripts and libraries are Python-based. You'll use `pip` (Python's package installer) to manage dependencies.
  - **Official Docs:** [python.org](https://python.org)
- **Windsurf**
  - **What it is:** An AI-powered IDE based on VS Code, designed for advanced coding assistance.

- **Why we need it:** It provides powerful, context-aware code generation and a seamless interface for interacting with local models. Windsurf automatically detects Python and Conda environments.
- **Official Docs:** [windsurf.ai](https://windsurf.ai)

#### • PyTorch

- **What it is:** A powerful, open-source machine learning framework.
- **Why we need it:** It is the engine that executes our AI models. **Transformers** provides the model architectures, **Accelerate** helps them run efficiently, and **PyTorch** performs the underlying calculations.
- **Official Docs:** [pytorch.org](https://pytorch.org)

#### • Hugging Face Libraries ( `transformers` , `accelerate` , `bitsandbytes` )

- **What they are:** A suite of libraries to download, run, and optimize state-of-the-art models.
- **Why we need them:** `transformers` gives us the models, `accelerate` optimizes their execution, and `bitsandbytes` reduces memory usage on GPUs (optional on CPU systems).
- **Official Docs:** [Hugging Face Docs](https://huggingface.co/docs)

#### • Ollama

- **What it is:** A tool that lets you easily run large language models (LLMs) locally.
- **Why we need it:** It provides a simple command-line interface and an API to interact with powerful models privately on your machine.
- **Official Docs:** [ollama.com](https://ollama.com)

#### • TabbyML

- **What it is:** A self-hosted AI coding assistant, like a private GitHub Copilot.
- **Why we need it:** It provides intelligent code completions directly in your editor while keeping your code private. Can run with or without GPU.
- **Official Docs:** [tabby.tabbyml.com](https://tabby.tabbyml.com)

#### • Git / Command-Line Tools

- **What they are:** Git is for version control; command-line tools are for interacting with your system.
- **Why we need them:** We'll use Git to access workshop code and the command line for all installations.
- **Official Docs:** [git-scm.com](https://git-scm.com)

---

## 2. Hardware & Driver Notes

- **CPU vs. GPU:** All tools will work on a modern CPU. However, performance is significantly better with a dedicated GPU.

- **NVIDIA GPU Users:** Update GPU drivers to version **550+** for full CUDA 12.4+ compatibility. Use PyTorch builds with CUDA 12.4 or newer.
  - **Apple Silicon (M1/M2/M3) Users:** No extra drivers are needed. PyTorch and other libraries automatically use Apple's Metal Performance Shaders (MPS) for acceleration.
- 

### 3. Installation Instructions

Please follow the instructions for your operating system.

#### macOS (Apple Silicon & Intel)

1. **Install Windsurf:** Download and install from the [official Windsurf website](https://windsurf.sh/).
2. **Install Homebrew:**

```
/bin/bash -c "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/HEAD/install.sh)"
```

3. **Install Core Tools:**

```
brew install python ollama
```

4. **Create and Activate a Virtual Environment:**

```
python3 -m venv workshop_env && source workshop_env/bin/activate
```

5. **Install Python Libraries:**

```
pip install --upgrade pip
pip install torch torchvision torchaudio
pip install "transformers==4.41.2" "accelerate==0.30.1"
"bitsandbytes==0.43.1"
```

6. **Install TabbyML (Docker Recommended):**

```
docker run -it -p 8080:8080 -v ~/.tabby:/data tabbyml/tabby serve --model
DeepSeek-Coder-V2-Lite-Instruct
```

7. **Pull Ollama Models:**

```
ollama pull phi3 && ollama pull llama3
```

## Linux (Debian/Ubuntu)

1. **Install Windsurf:** Download the `.deb` or AppImage file from the [Windsurf website](#).
2. **Install Dependencies:**

```
sudo apt update && sudo apt install -y git curl build-essential python3-venv python3-pip
```

3. **Install Ollama:**

```
curl -fsSL https://ollama.com/install.sh | sh
```

4. **Install Docker and TabbyML:**

```
sudo apt install -y docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-compose-plugin
sudo systemctl enable --now docker
sudo usermod -aG docker $USER # Re-login required
docker run -it --gpus all -p 8080:8080 -v $HOME/.tabby:/data tabbyml/tabby
serve --model DeepSeek-Coder-V2-Lite-Instruct
```

5. **Create and Activate Virtual Environment:**

```
python3 -m venv workshop_env && source workshop_env/bin/activate
```

6. **Install Python Libraries:**

```
pip install --upgrade pip
pip install torch torchvision torchaudio --index-url https://download.pytorch.org/whl/cu124
pip install "transformers==4.41.2" "accelerate==0.30.1"
"bitsandbytes==0.43.1"
```

7. **Pull Ollama Models:**

```
ollama pull phi3 && ollama pull llama3
```

## Windows

1. **Install Windsurf:** Download and install from the [Windsurf website](#).
2. **Install WSL2:** Follow [Microsoft's WSL2 Guide](#).
3. **Install Python 3.9+:** From [python.org](#) (check "Add Python to PATH").
4. **Install Ollama:** From [ollama.com](#).

5. **Install Docker and TabbyML (inside WSL2):** Use the Linux instructions above.
6. **Create and Activate Virtual Environment:**

```
python -m venv workshop_env  
.\workshop_env\Scripts\activate
```

#### 7. Install Python Libraries:

```
pip install --upgrade pip  
pip install torch torchvision torchaudio --index-url https://  
download.pytorch.org/whl/cu124  
pip install "transformers==4.41.2" "accelerate==0.30.1"  
pip install bitsandbytes || echo "bitsandbytes optional"
```

#### 8. Pull Ollama Models:

```
ollama pull phi3 && ollama pull llama3
```

---

## 4. Configuring Your Environment

Ensure your IDE (Windsurf) and terminal use the `workshop_env` Python interpreter.

1. Open Windsurf and the project folder containing `workshop_env`.
2. Confirm the bottom-right status bar shows **Python 3.x ('workshop\_env')**.
3. If not, select it manually via **Select Interpreter**.

### Jupyter Users

```
pip install notebook ipykernel  
python -m ipykernel install --user --name=workshop_env
```

Then select `workshop_env` as the kernel in your notebook interface.

### Hugging Face Accelerate Configuration (Optional)

```
accelerate config
```

---

## 5. Post-Install Verification

Run these tests inside Windsurf (with `workshop_env` activated):

### Check Python and Libraries:

```
import sys, torch, transformers, accelerate
print(f"Python: {sys.executable}")
print(f"PyTorch: {torch.__version__}")
print(f"Transformers: {transformers.__version__}")
print(f"Accelerate: {accelerate.__version__}")
print(f"GPU: {torch.cuda.get_device_name(0) if torch.cuda.is_available() else 'CPU/MPS detected'})")
```

### Test Ollama:

```
ollama run phi3 "Hello! Who are you?"
```

### Test Transformers:

```
from transformers import pipeline
gen = pipeline("text-generation", model="TinyLlama/TinyLlama-1.1B-Chat-v1.0")
print(gen("The future of AI is", max_new_tokens=20))
```

---

## 6. Troubleshooting Tips

**pip install permission errors:** - Ensure your virtual environment is activated.

**bitsandbytes installation fails:** - Safe to skip (`pip uninstall bitsandbytes`).

**CUDA errors:** - Update NVIDIA drivers (v550+). Ensure CUDA 12.4 build is installed.

**Ollama connection error:** - Start Ollama manually: `ollama serve`.

**Docker permission denied:** - Re-login after adding to `docker` group or use `sudo`.

**Proxy/SSL issues:** - For institutional networks:

```
export HTTPS_PROXY=https://username:password@proxy.icgeb.org:8080
```

**Apple MPS performance:** - Upgrade to macOS 14+ and PyTorch 2.3+ for improved GPU utilization.

---

Prepared for **AI & Data Science Workshop – November 2025**