

# LLM Engineering - Master AI and LLMs

## Setup instructions for Mac

Welcome, Mac people!

I should confess up-front: setting up a powerful environment to work at the forefront of AI is not as simple as I'd like. For most people these instructions will go great; but in some cases, for whatever reason, you'll hit a problem. Please don't hesitate to reach out - I am here to get you up and running quickly. There's nothing worse than feeling *stuck*. Message me, email me or LinkedIn message me and I will unstick you quickly!

Email: [ed@edwarddonner.com](mailto:ed@edwarddonner.com)

LinkedIn: <https://www.linkedin.com/in/eddonner/>

I use a platform called Anaconda to set up your environment. It's a powerful tool that builds a complete science environment. Anaconda ensures that you're working with the right version of Python and all your packages are compatible with mine, even if our systems are completely different. It takes more time to set up, and it uses more hard drive space (5+ GB) but it's very reliable once its working.

Having said that: if you have any problems with Anaconda, I've provided an alternative approach. It's faster and simpler and should have you running quickly, with less of a guarantee around compatibility.

### Part 1: Clone the Repo

This gets you a local copy of the code on your box.

1. **Install Git** if not already installed (it will be in most cases)
  - Open Terminal (Applications > Utilities > Terminal)
  - Type `git --version` If not installed, you'll be prompted to install it
2. **Navigate to your projects folder:**

If you have a specific folder for projects, navigate to it using the `cd` command. For example: `cd ~/Documents/Projects`

If you don't have a projects folder, you can create one:

```
mkdir ~/Documents/Projects
```

```
cd ~/Documents/Projects
```

3. **Clone the repository:**

Enter this in the terminal in the Projects folder:

```
git clone https://github.com/ed-donner/llm_engineering.git
```

This creates a new directory `llm_engineering` within your Projects folder and downloads the code for the class. Do `cd llm_engineering` to go into it. This `llm_engineering` directory is known as the “project root directory”.

## Part 2: Install Anaconda environment

There is an alternative to Part 2 if this gives you problems.

### 1. Install Anaconda:

- Download Anaconda from <https://docs.anaconda.com/anaconda/install/mac-os/>
- Double-click the downloaded file and follow the installation prompts. Note that it takes up several GB and take a while to install, but it will be a powerful platform for you to use in the future.

### 2. Set up the environment:

- Open a new Terminal (Applications > Utilities > Terminal)
- Navigate to the “project root directory” using `cd ~/Documents/Projects/llm_engineering` (replace this path as needed with the actual path to the `llm_engineering` directory, your locally cloned version of the repo). Do `ls` and check you can see subdirectories for each week of the course.
- Create the environment: `conda env create -f environment.yml`
- Wait for a few minutes for all packages to be installed - in some cases, this can literally take 20-30 minutes if you’ve not used Anaconda before, and even longer depending on your internet connection. Important stuff is happening! If this runs for more than 1 hour 15 mins, or gives you other problems, please go to Part 2B instead.
- You have now built an isolated, dedicated AI environment for engineering LLMs, running vector datastores, and so much more! You now need to **activate** it using this command: `conda activate llms`

You should see `(llms)` in your prompt, which indicates you’ve activated your new environment.

### 3. Start Jupyter Lab:

- In the Terminal window, from within the `llm_engineering` folder, type:  
`jupyter lab`

... and Jupyter Lab should open up in a browser. If you’ve not seen Jupyter Lab before, I’ll explain it in a moment! Now close the `jupyter lab` browser tab, and close the Terminal, and move on to Part 3.

## Part 2B - Alternative to Part 2 if Anaconda gives you trouble

### 1. Open a new Terminal (Applications > Utilities > Terminal)

Run `python --version` to find out which python you’re on. Ideally you’d be using a version of Python 3.11, so we’re completely in sync.

If not, it's not a big deal, but we might need to come back to this later if you have compatibility issues.

You can download python here:

<https://www.python.org/downloads/>

2. Navigate to the “project root directory” using `cd ~/Documents/Projects/llm_engineering` (replace this path with the actual path to the `llm_engineering` directory, your locally cloned version of the repo). Do `ls` and check you can see subdirectories for each week of the course.

Then, create a new virtual environment with this command:

```
python -m venv llms
```

3. Activate the virtual environment with `source llms/bin/activate` You should see `(llms)` in your command prompt, which is your sign that things are going well.
4. Run `pip install -r requirements.txt`  
This may take a few minutes to install.
5. **Start Jupyter Lab:**

From within the `llm_engineering` folder, type: `jupyter lab`

... and Jupyter Lab should open up, ready for you to get started. Open the `week1` folder and double click on `day1.ipynb`. Success! Now close down jupyter lab and move on to Part 3.

If there are any problems, contact me!

### **Part 3 - OpenAI key (OPTIONAL but recommended)**

Particularly during weeks 1 and 2 of the course, you'll be writing code to call the APIs of Frontier models (models at the forefront of AI).

For week 1, you'll only need OpenAI, and you can add the others if you wish later on.

1. Create an OpenAI account if you don't have one by visiting: <https://platform.openai.com/>
2. OpenAI asks for a minimum credit to use the API. For me in the US, it's \$5. The API calls will spend against this \$5. On this course, we'll only use a small portion of this. I do recommend you make the investment as you'll be able to put it to excellent use. But if you'd prefer not to pay for the API, I give you an alternative in the course using Ollama.

You can add your credit balance to OpenAI at Settings > Billing:

<https://platform.openai.com/settings/organization/billing/overview>

I recommend you disable the automatic recharge!

3. Create your API key

The webpage where you set up your OpenAI key is at <https://platform.openai.com/api-keys> - press the green ‘Create new secret key’ button and press ‘Create secret key’. Keep a record of the API key somewhere private; you won’t be able to retrieve it from the OpenAI screens in the future. It should start `sk-proj-`.

In week 2 we will also set up keys for Anthropic and Google, which you can do here when we get there.

- Claude API at <https://console.anthropic.com/> from Anthropic - Gemini API at <https://ai.google.dev/gemini-api> from Google

Later in the course you’ll be using the fabulous HuggingFace platform; an account is available for free at <https://huggingface.co> - you can create an API token from the Avatar menu » Settings » Access Tokens.

And in Week 6/7 you’ll be using the terrific Weights & Biases at <https://wandb.ai> to watch over your training batches. Accounts are also free, and you can set up a token in a similar way.

#### **PART 4 - .env file**

When you have these keys, please create a new file called `.env` in your project root directory. The filename needs to be exactly the four characters “`.env`” rather than “`my-keys.env`” or “`.env.txt`”. Here’s how to do it:

1. Open Terminal (Applications > Utilities > Terminal)
2. Navigate to the “project root directory” using `cd ~/Documents/Projects/llm_engineering` (replace this path with the actual path to the `llm_engineering` directory, your locally cloned version of the repo).
3. Create the `.env` file with

```
nano .env
```

4. Then type your API keys into nano, replacing `xxxx` with your API key (starting `sk-proj-`).

```
OPENAI_API_KEY=xxxx
```

If you have other keys, you can add them too, or come back to this in future weeks:

```
GOOGLE_API_KEY=xxxx
ANTHROPIC_API_KEY=xxxx
HF_TOKEN=xxxx
```

5. Save the file:

Control + O

Enter (to confirm save the file)

Control + X to exit the editor

6. Use this command to list files in your project root directory:

```
ls -a
```

And confirm that the `.env` file is there.

This file won't appear in Jupyter Lab because jupyter hides files starting with a dot. This file is listed in the `.gitignore` file, so it won't get checked in and your keys stay safe.

### Part 5 - Showtime!!

- Open Terminal (Applications > Utilities > Terminal)
- Navigate to the “project root directory” using `cd ~/Documents/Projects/llm_engineering` (replace this path with the actual path to the `llm_engineering` directory, your locally cloned version of the repo). Do `ls` and check you can see subdirectories for each week of the course.
- Activate your environment with `conda activate llms` (or `source llms/bin/activate` if you used the alternative approach in Part 2B)
- You should see `(llms)` in your prompt which is your sign that all is well. And now, type: `jupyter lab` and Jupyter Lab should open up, ready for you to get started. Open the `week1` folder and double click on `day1.ipynb`.

And you're off to the races!

Note that any time you start jupyter lab in the future, you'll need to follow these Part 5 instructions to start it from within the `llm_engineering` directory with the `llms` environment activated.

For those new to Jupyter Lab / Jupyter Notebook, it's a delightful Data Science environment where you can simply hit shift+return in any cell to run it; start at the top and work your way down! I've included a notebook called 'Guide to Jupyter' that shows you more features. When we move to Google Colab in Week 3, you'll experience the same interface for Python runtimes in the cloud.

If you have any problems, I've included a notebook in week1 called `troubleshooting.ipynb` to figure it out.

Please do message me or email me at [ed@edwarddonner.com](mailto:ed@edwarddonner.com) if this doesn't work or if I can help with anything. I can't wait to hear how you get on.