Lab 0

Set up Python, LangChain and LLM API on a local machine(Windows and Mac)

1. Install Python 3.10

- **Download Python 3.10.10**:
    - Visit the official Python website: https://www.python.org/downloads/release/python-31010/
    - Download the **Windows or MacOS** installer based on your system.
- **Install Python**:
    - Run the downloaded installer.
    - **Important(Windows)**: Ensure you check the box labeled `Add Python 3.10 to PATH` before clicking "Install Now". (This will make it the default version)
    - **For Mac** just follow the Steps.

2. Verify Python Installation

- **Windows:**

Open Command Prompt(cmd) and type: This should display Python 3.10.x, confirming the installation.

```
python --version
```

    -
- **Mac:**

Open your terminal and type the following command to check your default Python version.: *Note that sometimes the default Python version on macOS may not be the one you just downloaded*.

```
python --version
```

-
- If it's not the desired version, **you can manually change the default version** by following these steps:

Step 1: Run the following command in your terminal to check all versions of Python installed on your Mac. Look for Python version 3.10.

ls -l /usr/local/bin/python*

1.

Step 2: If you see Python version 3.10 on the list, run the following command to change the default Python version. This will prompt you to enter your password.

sudo ln -s -f /usr/local/bin/python3.10 /usr/local/bin/python

2.
3. **Step 3**: Restart your terminal and use the `python --version` command again to test it. It should now display Python 3.10 as the default version.

3. Install Visual Studio Code (VSCode)

- Download VSCode from: https://code.visualstudio.com/.
- Run the installer and follow the on-screen instructions.

4. Install Python Extension for VSCode

- Open VSCode.

- Navigate to the Extensions view by clicking on the square icon on the sidebar or pressing `Ctrl + Shift + X`(**Windows**) ,or ⌘**+Shift + X(Mac)**

- In the search bar, type the name of each extension below and install them.

  Essential Extensions:

  - **Python** – Official extension by Microsoft for Python development (syntax highlighting, debugging, IntelliSense, linting, etc.).
  - **Pylance** – Provides enhanced IntelliSense, type checking, and autocomplete features for Python.
  - **Jupyter** – Enables Jupyter Notebook support inside VSCode.

    - **Jupyter Notebook Renderers** – Improves Jupyter notebook output visualization.
    - **Jupyter Keymap** – Adds additional shortcuts for working with Jupyter Notebooks.
    - **Database Client JDBC** –Work with database.
- **Optional but Recommended Extensions:**

    - **Rainbow CSV** – Highlights CSV files with color coding for better readability.
    - **Excalidraw** – A lightweight drawing tool inside VSCode for sketching ideas, workflows, or architecture designs.

5. Set Up a Virtual Environment

- **Create a Project Directory**:

    - Open VSCode.
        - Click on `File` > `Open Folder...` and select or create a new folder for your project.
    - **Open Terminal in VSCode**:
        - Navigate to `Terminal` > `New Terminal` or press `Ctrl + ``(windows)or **⌘** + J` (Mac)
        - The `Terminal` should shows the current project folder.
- **Windows:**

    - **Create a Virtual Environment**

In the terminal, run the following command:

```
python -m venv your-env-name
```

        - **Naming Guidelines for `your-env-name`**:

You can replace `your-env-name` with **any valid folder name**. Some common choices:

```
python -m venv venv  # Common convention
python -m venv my_project_env  # More descriptive
python -m venv langchain_env  # Project-specific
```

        -
        - **Avoid spaces** in the name (e.g., `my venv`), as it may cause issues.

        - **Do not use reserved names** like `python` or `env`.
    - This command creates a **new folder** with the virtual environment inside your project directory.
- **Activate the Virtual Environment**
    - **Make sure your terminal is open in the same directory where you created the virtual environment.**

Run the appropriate activation command in the **VSCode terminal**:

your-env-name\\Scripts\\Activate

or
your-env-name\\Scripts\\activate.bat

    -
    - If successful, your terminal prompt should now be prefixed with `(your-env-name)`, indicating that the virtual environment is active.

- **Mac**

  Create VM from terminal

    - **Step 1**: Open your folder in VS Code and click the third icon at the top to open the terminal, or use (⌘**+J**) to directly open the terminal within the current folder.

Step 2: Create a Virtual Environment (VM)

To create a virtual environment, type the following command in the terminal(*with your default python version*) :

python -m venv GenAI
Note: You can replace `GenAI` with a name of your choice for the virtual environment.

Or, If you are targeting a specific Python version, use the following command:

python3.10 -m venv GenAI

- **Note:** Replace `python3.10` with the desired Python version installed on your machine.

Step 3: Activate your VM:

source GenAI/bin/activate

6. Set Up API Keys & Install LangChain Basic Packages

Before installing LangChain, you **must set up an API key** for services like OpenAI.

- **Get an OpenAI API Key**
  - Go to **OpenAI API Keys**
  - Click **"Create new secret key"** and copy it. (You may need to register for an account)
  - **Store this key securely**—you'll need it in the next step.
- **Create a `.env` File to Store API Keys**
  - Inside your project folder, create a new file named **`.env`**
  - Open the `.env` file in VScode and add your OpenAI API key like this:

OPENAI_API_KEY="your-api-key-here" # Replace your-api-key-here with your actual API key within the quotation marks.

- Save the file.

Install LangChain and Required Packages

With the virtual environment activated, install the necessary packages in the terminal:

pip install langchain langchain-core langchain-openai langchain-community langchain-experimental
pip install openai
pip install python-dotenv

Verify the Installation

To ensure everything is set up correctly, create a **new Python file** in your project folder (e.g., `test_installation.py`).

Test: Check Installed Versions

```
import langchain
import openai
from dotenv import load_dotenv

# Load environment variables from .env file
load_dotenv()

# Print package versions
print("LangChain Version:", langchain.__version__)
print("OpenAI Version:", openai.__version__)
```

-

Run the script

Open **VSCode Terminal** and run:

```
python test_installation.py
```

-
- If you see printed **version numbers**, everything is installed correctly.

Deactivate the Virtual Environment

Once you're done working, deactivate the virtual environment by running if needed:

```
deactivate
```

- This will return the terminal to the system's default Python environment.

Now You're Ready! 🚀

✅ Python 3.10 installed

✅ A virtual environment set up

✅ API keys stored securely

✅ LangChain and OpenAI configured

✅ A verified installation