

# Setting Up a Python Environment

## Introduction

- This presentation will guide you through setting up a Python environment for development.
- We will cover:
  - Python installation
  - Setting an environment variable
  - Creating a virtual environment
  - Installing Jupyter Notebook
  - Cloning a GitHub repository
  - Installing dependencies from a `requirements.txt` file
  - Starting a Jupyter Notebook

## Installing Python

### Windows

1. Download the Python installer from [python.org](https://python.org).
2. Run the installer and check “Add Python to PATH”.
3. Click “Install Now”.

### macOS

1. Check if Python is already installed:

```
python --version
```

2. If not, install Python using Homebrew: “`bash /bin/bash -c “$(curl -fsSL https://raw.githubusercontent.com/Homebrew/homebrew-core/master/installs/macos/install.sh)`”

## Setting an Environment Variable

### Windows (Command Line)

1. Open Command Prompt.
2. Set the environment variable for the current session:

```
set OPENAI_KEY=your_openai_api_key
```

3. To make the variable persistent across sessions, use:

```
setx OPENAI_KEY "your_openai_api_key"
```

### Windows (System Properties)

1. Open the Start Search, type in “env”, and choose “Edit the system environment variables”.
2. In the System Properties window, click on the “Environment Variables...” button.
3. In the Environment Variables window, click “New...” under the “User variables” section.
4. Enter `OPENAI_KEY` as the variable name and your OpenAI API key as the variable value.
5. Click OK to save the changes.

### macOS

1. Open Terminal.
2. Open your `~/.bash_profile` or `~/.zshrc` file in a text editor:

```
nano ~/.bash_profile
```

3. Add the following line to the file:

```
export OPENAI_KEY="your_openai_api_key"
```

4. Save the file and exit the text editor.
5. Apply the changes by running:

```
source ~/.bash_profile
```

- Replace `your_openai_api_key` with your actual OpenAI API key.

- This environment variable will be available in all new terminal sessions.

## Creating a Virtual Environment

### Windows

1. Open Command Prompt.
2. Navigate to your project directory:

```
cd path\to\your\project
```

3. Create a virtual environment:

```
python -m venv myenv
```

### macOS

1. Open Terminal.
2. Navigate to your project directory:

```
cd /path/to/your/project
```

3. Create a virtual environment:

```
python3 -m venv myenv
```

## Activating the Virtual Environment

### Windows

```
myenv\Scripts\activate
```

### macOS

```
source myenv/bin/activate
```

## Installing Jupyter Notebook

```
pip install notebook
```

## Cloning a GitHub Repository

```
git clone https://github.com/abigailhaddad/LMGradingRubric.git
```

## Installing Dependencies

Navigate to the cloned repository and install the requirements:

```
cd LMGradingRubric  
pip install -r requirements.txt
```

## Starting Jupyter Notebook

### From the Command Line

Activate your virtual environment and start Jupyter Notebook:

```
jupyter notebook
```

### Using a File Browser

- Double-clicking an .ipynb file might not open it in the virtual environment.
- To ensure it uses the correct environment, start Jupyter Notebook from the command line and open the file from the Jupyter dashboard.

### Changing the Kernel (if needed)

- If your notebook is not using the kernel associated with your virtual environment:
  - Click on “Kernel” > “Change kernel” in the Jupyter Notebook menu.
  - Select the kernel that corresponds to your virtual environment.

## Conclusion

- You now have a complete Python environment set up for development.
- You can use this environment to work on projects, run Jupyter Notebooks, and more.