# MSNE Python Workshop
# **Setting Up Your Coding Environment**

*Course Instructor*: Karahan Yilmazer `karahan.yilmazer@tum.de`

Winter Semester 23/24

## Introduction

This tutorial is aimed to help you set up your Python environment in VS Code. Even though it can be a cumbersome process, follow through the whole tutorial and set everything up, so that you do not have to do it again.

In the following, you will learn how to set up

- a Python virtual environment
- Visual Studio Code

After you finish this tutorial, you can follow the other tutorials in the course repository. Without further ado, let's get started!

## 1 Installing Python and Setting up a Virtual Environment

First off, you will have to install Python if you have not done so already. If you do not have a version preference, simply install the latest Python version (3.12.1).

For Windows, you can download the executable directly from the Python website. For Linux, you can refer to this short tutorial for different installation methods.

Once you have installed Python, you can set up a virtual environment for the course. This way all the packages you need will live in one virtual space without affecting your other Python projects. For that, open a terminal (PowerShell for Windows, bash for Linux) and follow the code snippet for your system below.

Setting Up A Virtual Environment (Windows)

```
PS C:\Users\yilma> py --version # Make sure Python is installed
Python 3.12.1
PS C:\Users\yilma> cd ..\..\Programming\Python\ # Navigate to the desired folder
PS C:\Programming\Python> mkdir venvs # Create a folder for virtual environments
PS C:\Programming\Python> cd .\venvs\ # Switch to the venvs folder
PS C:\Programming\Python\venvs> py -m venv neuro # Create a virtual environment
PS C:\Programming\Python\venvs> .\neuro\Scripts\activate # Activate the virtual environment
(neuro) PS C:\Programming\Python\venvs> deactivate # Deactivate the virtual environment
PS C:\Programming\Python\venvs>
```

**Info:**
If you receive the error "File C:\Programming\Python\venvs\neuro\Scripts\Activate.ps1 cannot be loaded because running scripts is disabled on this system.", you should change the execution policy of your system.

You can check your current execution policy using `Get-ExecutionPolicy`. If it is `Restricted`, you can change it to a different policy where you can run scripts on your system. You can do so by typing **`Set-ExecutionPolicy AllSigned`** in a PowerShell with Administrator privileges. You can read more about execution policies here.

**Info:**
In Windows, by using `py` instead of `python`, you automatically use the latest Python installation on your computer. If you want to use a different Python installation, you can do so by replacing `py` with `py -3.X`.

```
Setting Up A Virtual Environment (Linux)

karahan@yilma:~$ python3.12 --version # Make sure Python is installed
Python 3.12.1
karahan@yilma:~$ cd Python/ # Navigate to the desired folder
karahan@yilma:~/Python$ mkdir venvs # Create a folder for virtual environments
karahan@yilma:~/Python$ cd venvs # Switch to the venvs folder
karahan@yilma:~/Python/venvs$ python3.12 -m venv neuro # Create a virtual environment
karahan@yilma:~/Python/venvs$ source neuro/bin/activate # Activate the virtual environment
(neuro) karahan@yilma:~/Python/venvs$ deactivate # Deactivate the virtual environment
karahan@yilma:~/Python/venvs$
```

## 2 Installing Packages With Pip

Assuming that you have successfully set up your virtual environment, you can activate it once again to install the necessary packages for the tutorial. You can either use the `requirements.txt` file to automatically install all the necessary packages or by install all packages one by one.

You can find the `requirements.txt` file in the GitLab repository for the course under the folder `misc`. This is a file that contains a list of all the necessary packages and it can be easily managed by pip. Just make sure that you are in the folder that contains this file when you are running the corresponding line.

```
Installing the Necessary Packages Using requirements.txt

karahan@yilma:~/Python$ source venvs/neuro/bin/activate # Activate the virtual environment
    (Linux)
(neuro) karahan@yilma:~$ cd Python/bci-course-tutorials/misc # Navigate to the project
    folder
(neuro) karahan@yilma:~/Python/msne-python-workshop/misc$ ls # List the contents of the
    folder
requirements.txt
(neuro) karahan@yilma:~/Python/bci-course-tutorials/misc$ pip install --upgrade pip
    setuptools wheel # Update the elementary modules
(neuro) karahan@yilma:~/Python/bci-course-tutorials/misc$ pip install -r requirements.txt #
    Install the necessary packages
```

Installing the Necessary Packages Manually

```
karahan@yilma:~/Python$ source venvs/neuro/bin/activate # Activate the virtual environment
    (Linux)
(neuro) karahan@yilma:~/Python$ pip install --upgrade pip setuptools wheel # Update the
    elementary modules (you can use py instead of python for Windows)
(neuro) karahan@yilma:~/Python$ pip install numpy matplotlib sklearn scipy screeninfo mne
    pyxdf pylsl pyqt5 pygame ... # Install the necessary packages
```

## 3 Setting up Visual Studio Code

Visual Studio Code is a powerful code editor, that can also be used as an IDE (integrated development environment). This means you can edit, run and debug your Python code using only VS Code. It has a marketplace for extensions, which further advances its usability.

To be able to run Python scripts in VS Code, you will have to install the **official Python extension** from the VS Code extensions marketplace. For that, click on the icon with four squares on the left side bar or press Ctrl+Shift+X to open the marketplace. Search for "python" and install the extension provided by Microsoft. You can find a screenshot of this procedure in Figure 1.

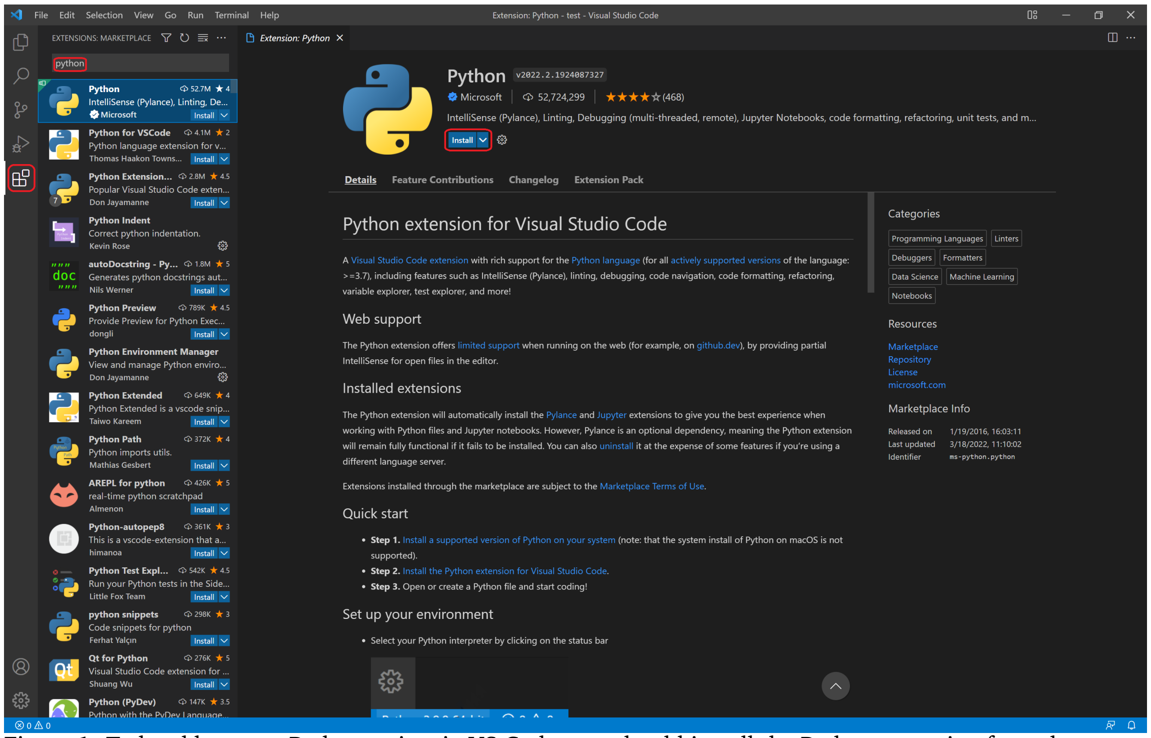
Next up, you will have to connect your virtual environment with VS Code. Click on **`File > Preferences > Settings`**. Then type "venvs" in the search bar. If nothing shows up you may have to restart VS Code. Copy and paste the path to your `venvs` folder to **`Python: Venv Path`**. You can refer to Figure 2 for this procedure.

With this, you will be able to choose your virtual environment in the bottom status bar. This way, you can quickly switch between different virtual environments. By switching to a virtual environment, you will be able to use all the packages installed in that virtual environment in your Python scripts.

The Python scripts you will find in the course repository will have special comments **`# %%`**. These special comments divide the code into cells which can be run separately in VS Code, just like a Jupyter Notebook. To be able to use this feature, you have to install the **official Jupyter extension** provided by Microsoft from the marketplace as well. You can simply type "jupyter" in the search bar to install it just like the Python extension.

Now, when you click on "Run Cell" or press Shift+Enter in a cell, a pop up might appear. It will read: "Running cells with 'Python 3.11.3' requires ipykernel package.". Simply click 'Install'. This will open up a terminal to install all the necessary pip packages. As a side note, you can use this built-in terminal, for example, to install different packages, without leaving VS Code.

Congratulations! With this you have finished the boring setup part of your project ;)



| | | | |
|---|---|---|---|
| Python | IntelliSense (Pylance), Linting, De... | Microsoft | 52.7M ★ 4 |
| Python for VSCode | Python language extension for v... | Thomas Haakon Towns... | 4.1M ★ 2 |
| Python Extension... | Popular Visual Studio Code exten... | Don Jayamanne | 2.8M ★ 4.5 |
| Python Indent | Correct python indentation. | Kevin Rose | |
| autoDocstring - Py... | Generates python docstrings aut... | Nils Werner | 1.8M ★ 5 |
| Python Preview | Provide Preview for Python Exec... | dongli | 789K ★ 4.5 |
| Python Environment Manager | View and manage Python enviro... | Don Jayamanne | |
| Python Extended | Python Extended is a vscode snip... | Taiwo Kareem | 649K ★ 4 |
| Python Path | Python imports utils. | Mathias Gesbert | 372K ★ 4 |
| AREPL for python | real-time python scratchpad | Almenon | 426K ★ 5 |
| Python-autopep8 | This is a vscode-extension that a... | himanoa | 361K ★ 3 |
| Python Test Expl... | Run your Python tests in the Side... | Little Fox Team | 542K ★ 4.5 |
| python snippets | Code snippets for python | Ferhat Yalçın | 298K ★ 3 |
| Qt for Python | Visual Studio Code extension for ... | Shuang Wu | 276K ★ 5 |
| Python (PyDev) | Python with the PyDev Language... | | 147K ★ 3.5 |

Python v2022.2.1924087327

Microsoft | 52,724,299 | ★★★★☆ (468)

IntelliSense (Pylance), Linting, Debugging (multi-threaded, remote), Jupyter Notebooks, code formatting, refactoring, unit tests, and m...

Python extension for Visual Studio Code

A Visual Studio Code extension with rich support for the Python language (for all actively supported versions of the language: >=3.7), including features such as IntelliSense (Pylance), linting, debugging, code navigation, code formatting, refactoring, variable explorer, test explorer, and more!

Web support

The Python extension offers limited support when running on the web (for example, on github.dev), by providing partial IntelliSense for open files in the editor.

Installed extensions

The Python extension will automatically install the Pylance and Jupyter extensions to give you the best experience when working with Python files and Jupyter notebooks. However, Pylance is an optional dependency, meaning the Python extension will remain fully functional if it fails to be installed. You can also uninstall it at the expense of some features if you're using a different language server.

Extensions installed through the marketplace are subject to the Marketplace Terms of Use.

Quick start

- **Step 1.** Install a supported version of Python on your system (note: that the system install of Python on macOS is not supported).
- **Step 2.** Install the Python extension for Visual Studio Code.
- **Step 3.** Open or create a Python file and start coding!

Set up your environment

- Select your Python interpreter by clicking on the status bar

Categories

Programming Languages, Linters, Debuggers, Formatters, Data Science, Machine Learning, Notebooks

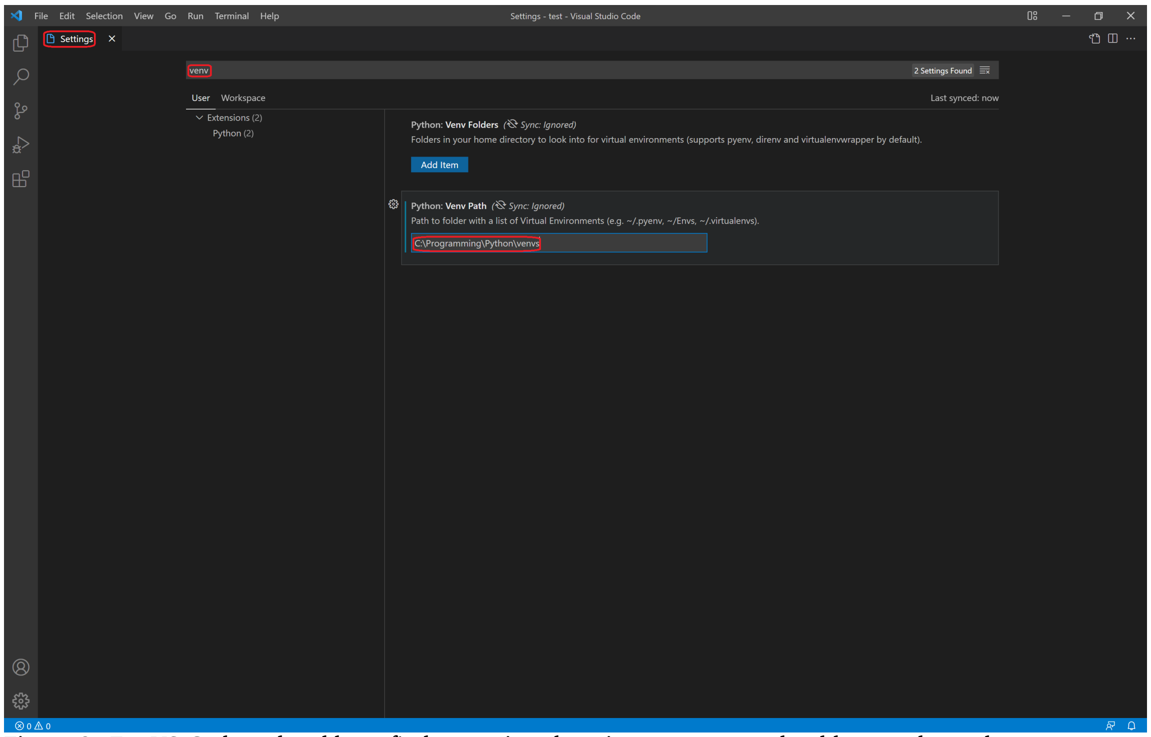
Resources

Marketplace
Repository
License
microsoft.com

Marketplace Info

| | |
|---|---|
| Released on | 1/19/2016, 16:03:11 |
| Last updated | 3/18/2022, 11:10:02 |
| Identifier | ms-python.python |

Figure 1: To be able to run Python scripts in VS Code, you should install the Python extension from the extensions marketplace. The red box on the left side bar shows the icon for the marketplace. By typing in "python" in the search bar, you can find and install the official Python extension.



venv

Python: **Venv Folders** *(Sync: Ignored)*
Folders in your home directory to look into for virtual environments (supports pyenv, direnv and virtualenvwrapper by default).

Python: **Venv Path** *(Sync: Ignored)*
Path to folder with a list of Virtual Environments (e.g. ~/.pyenv, ~/Envs, ~/.virtualenvs).

C:\Programming\Python\venvs

Figure 2: For VS Code to be able to find your virtual environments, you should paste the path to your venvs folder to `Python: Venv Path` in Settings.