



# **Streamlining Invoice Processing:** Document Understanding Transformer for Efficient Business Operations in Python

---

Pre-Workshop Guide  
January 2024



## Contents

Introduction	2
Installation	3
<ul style="list-style-type: none"><li>• Installing Python using <i>Miniconda</i></li><li>• Installing Visual Studio Code</li><li>• Installing Python &amp; Jupyter Extension in VS Code</li></ul>	
<b>System Verification</b>	4
<ul style="list-style-type: none"><li>• Verify Miniconda Installation</li><li>• Verify VSCode Installation</li><li>• Verify Python &amp; Jupyter Extension in VS Code Installation</li></ul>	

■ ■ ■



## Introduction

This guide is a resource for students at Algoritma to use in setting up their laptop or environment prior to the scheduled workshops. In this guide, students can find a list of prerequisites that will be consistently used throughout the entire course. These prerequisites are required to be **completed before** the start of the workshop.

For new students, we will run through the installation process to ensure that the necessary programming languages and tools - such as Python - are installed. The next section will then talk about methods on how to verify whether the installs were completed successfully.

For recurring students, we recommend repeating the System Verification section oncemore to confirm past completed installations.

■ ■ ■



## Installation

### Installing Python using *Miniconda*

To install Python, we will use **Miniconda**, a minimal version of Anaconda that includes only *conda*, Python, and the essential packages they rely on. Miniconda provides not only Python but also the required packages (such as numpy and pandas) used in our workshops. Please follow the link below and **select Python version 3 for installation**.

Use this link:

<https://docs.conda.io/en/latest/miniconda.html#latest-miniconda-installer-links>

for Miniconda

More info on *Miniconda*:

<https://docs.conda.io/en/latest/miniconda.html>



## Installing Visual Studio Code

To set up a code editor, we'll be using Microsoft's Visual Studio Code, often referred to as **VS Code**. This robust and versatile code editor supports multiple programming languages and offers a wide range of extensions to enhance its functionality. Follow the steps below to install VS Code:

1. Open your web browser and **go to the official Visual Studio Code downloadpage** using this link: <https://code.visualstudio.com/download>
2. On the download page, you'll find options for Windows, macOS, and Linux. **Select the appropriate download link that matches your operating system.**
3. **Once the installation is complete, launch VS Code.** You'll be greeted with a welcome screen like the picture below.

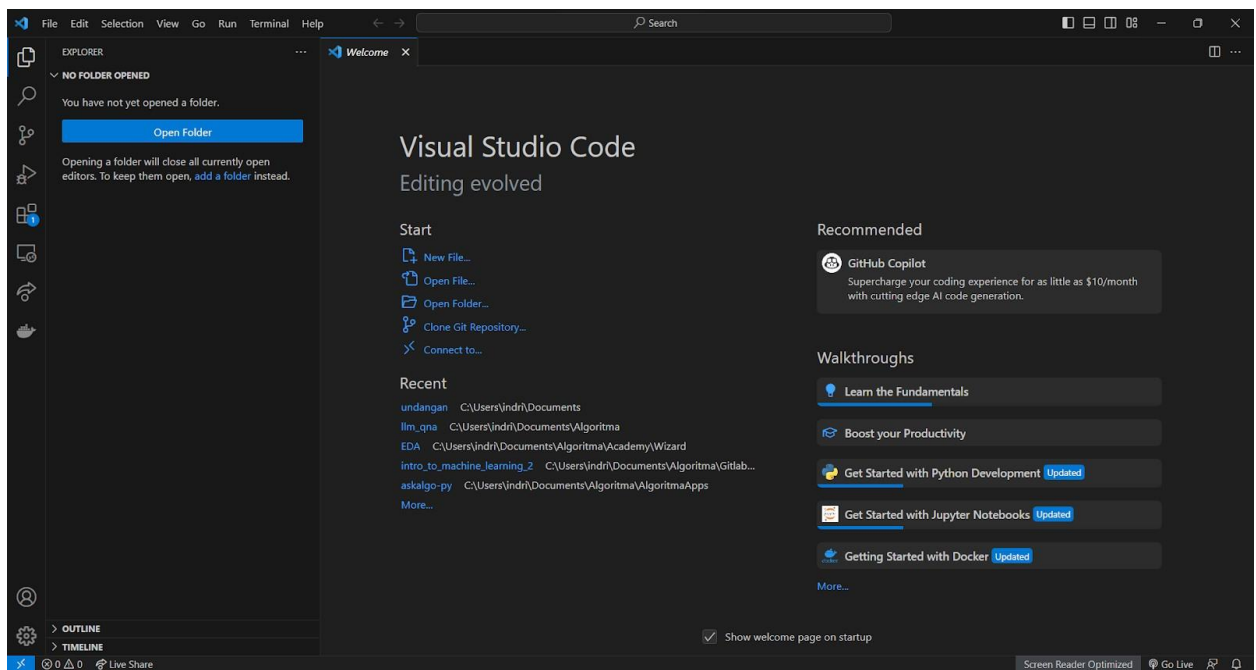


Figure 1: Visual Studio Code's Welcome Page

## Installing Python & Jupyter Extension in VS Code

To set up VS Code with Miniconda, we'll require the 'Python' and 'Jupyter' extensions in VS Code. The 'Python' extension is necessary for managing Python environments and providing code assistance. Moreover, the 'Jupyter' extension is essential for editing, running, and interacting with Jupyter notebooks within VS Code. Follow the steps below to install VS Code:

1. Open VSCode and **navigate to the 'Extension' menu on the sidebar** (highlighted in green boxes in the image below). Alternatively, you can also open this menu using the shortcut **Ctrl+Shift+X** on Windows or **Cmd+Option+X** on macOS.

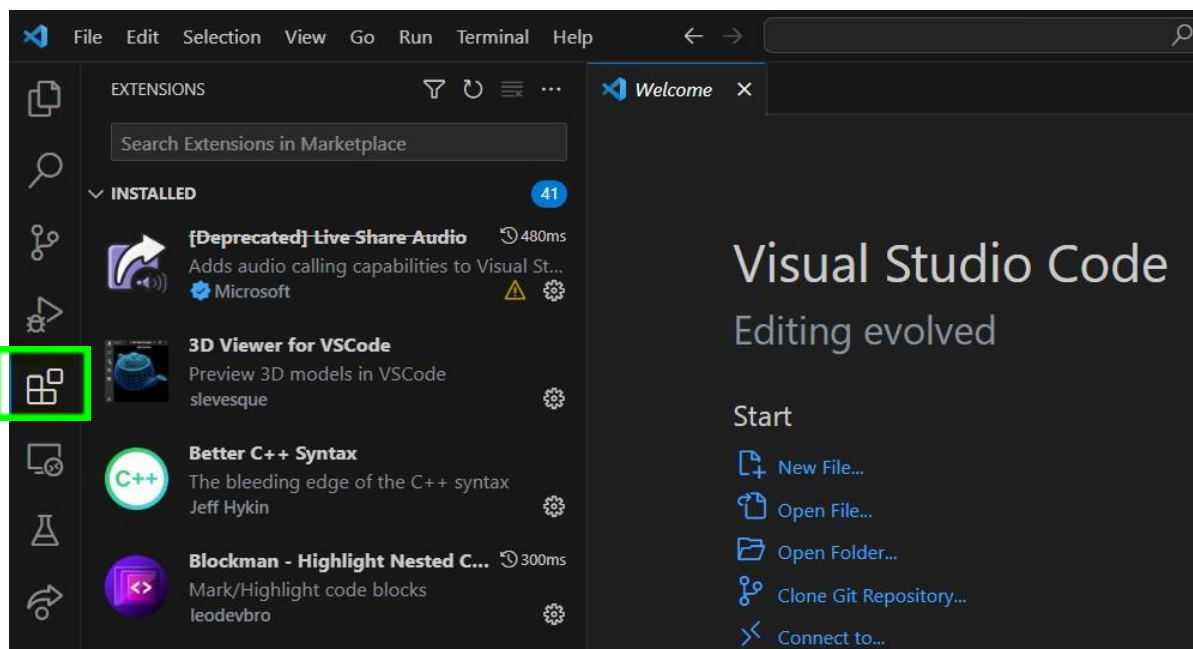


Figure 2: VS Code Extensions on the Activity Bar

### Python Extension Installation

2. **Search 'Python'** on Extension search bar. Then, press **Install**

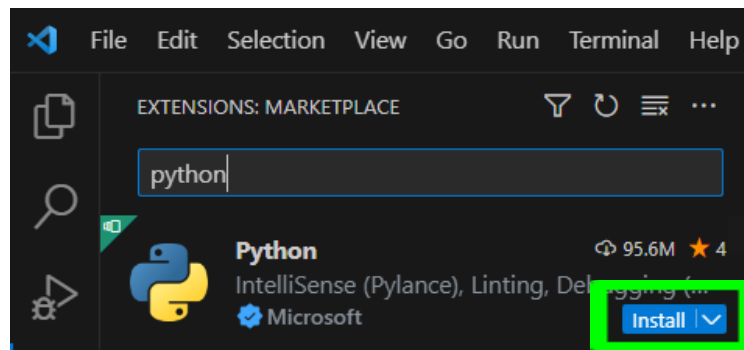


Figure 3: Searching for **Python** Extension in VS Code Extensions

3. Once the 'Python' extension installation is complete, the page view is like picture below



Figure 4: **Python** Extension Installed in VS Code Extensions

## Jupyter Extension Installation

4. Search 'Jupyter' on Extension search bar. Then, click **Install**

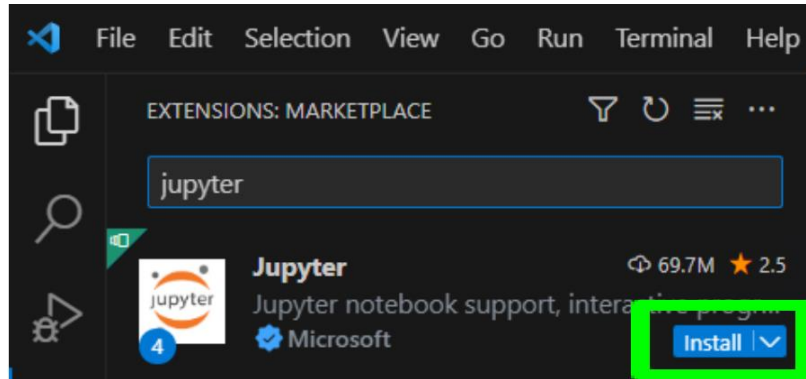


Figure 5: Searching for **Jupyter** Extension in VS Code Extensions

5. Once the 'Jupyter' extension installation is complete, the page view is like picture below

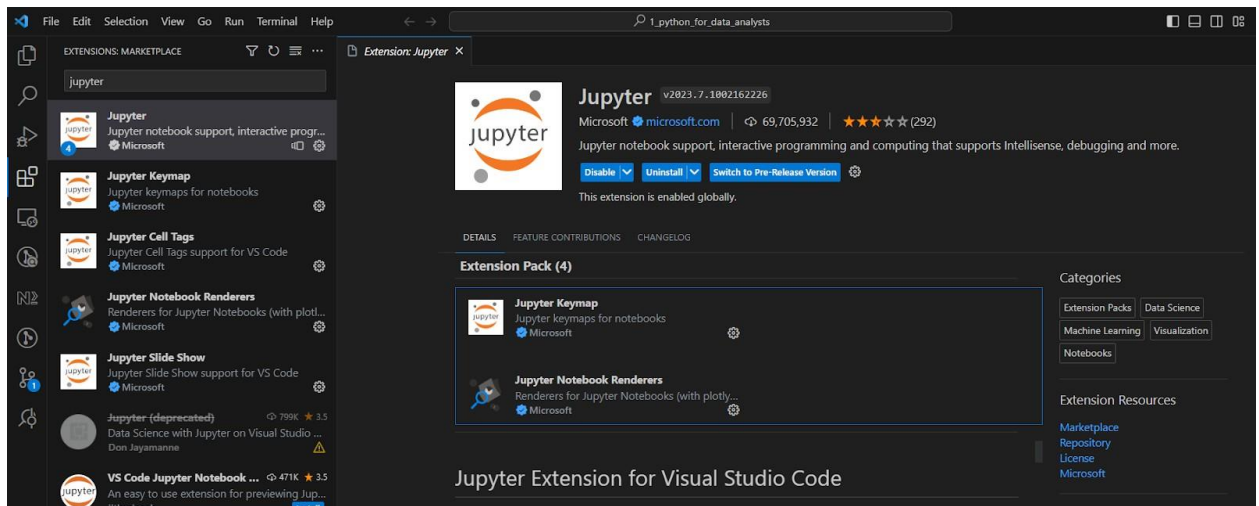


Figure 6: **Jupyter** Extension Installed in VS Code Extensions

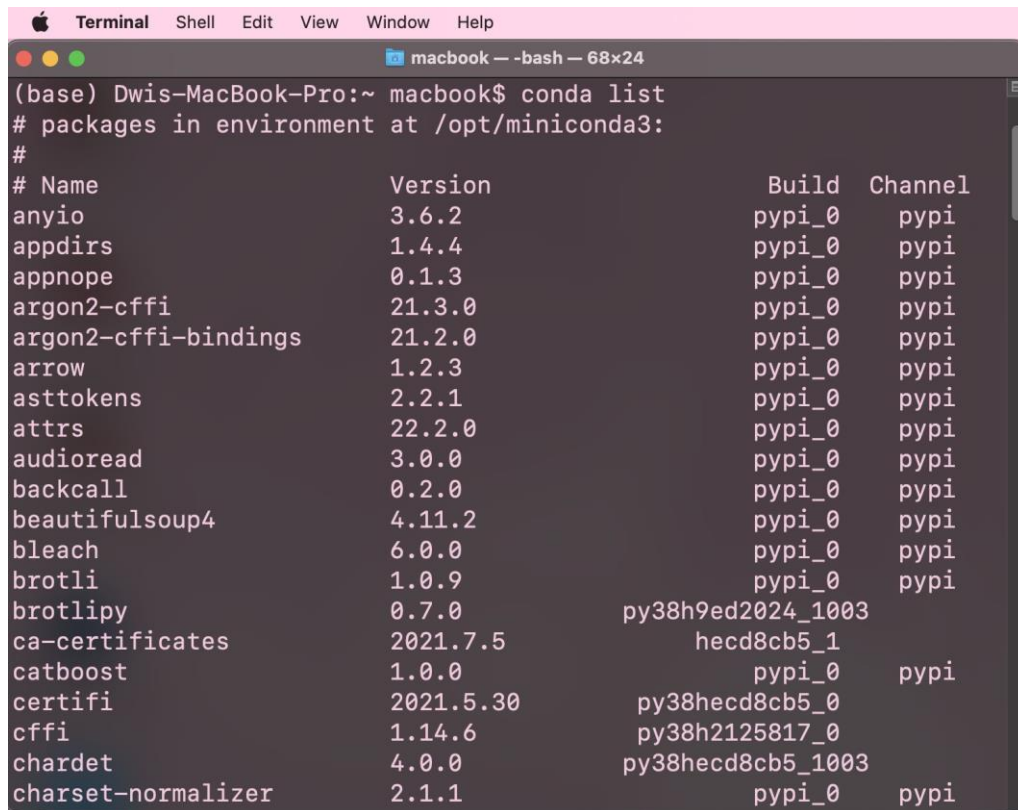


## System Verification

- **For Mac OS X and Linux-based OS:** Open “Terminal”
- **For Windows:** Open “Anaconda Prompt (miniconda3)”

### Verify Anaconda Installation

1. Type the command `conda list` in your “Terminal” or “Anaconda Prompt(miniconda3)”.
2. If the installation was completed successfully, your terminal will give a response of list of packages like the example below.
3. If your terminal does not give any response, please check the installation section's **Warning**, if the problem still persists, kindly reach out for further assistance via email at [mentor@algorit.ma](mailto:mentor@algorit.ma).



```
(base) Dwis-MacBook-Pro:~ macbook$ conda list
# packages in environment at /opt/miniconda3:
#
# Name                          Version          Build       Channel
anyio                            3.6.2            pypi_0     pypi
appdirs                          1.4.4            pypi_0     pypi
appnope                          0.1.3            pypi_0     pypi
argon2-cffi                     21.3.0           pypi_0     pypi
argon2-cffi-bindings            21.2.0           pypi_0     pypi
arrow                            1.2.3            pypi_0     pypi
asttokens                        2.2.1            pypi_0     pypi
attrs                            22.2.0           pypi_0     pypi
audioread                       3.0.0            pypi_0     pypi
backcall                        0.2.0            pypi_0     pypi
beautifulsoup4                  4.11.2           pypi_0     pypi
bleach                           6.0.0            pypi_0     pypi
brotli                           1.0.9            pypi_0     pypi
brotlipy                         0.7.0            py38h9ed2024_1003
ca-certificates                 2021.7.5         hecd8cb5_1
catboost                         1.0.0            pypi_0     pypi
certifi                          2021.5.30        py38hecd8cb5_0
cffi                             1.14.6           py38h2125817_0
chardet                         4.0.0            py38hecd8cb5_1003
charset-normalizer               2.1.1            pypi_0     pypi
```

Figure 7: `conda list` Response on Mac OS X Terminal

```
Anaconda Prompt (miniconda) X + v
(base) C:\Users\user>conda list
# packages in environment at C:\Users\user\miniconda3:
#
# Name                          Version      Build      Channel
boltons                          23.0.0       py311haa95532_0
brotlipy                        0.7.0        py311h2bbff1b_1002
bzip2                           1.0.8        he774522_0
ca-certificates                 2023.05.30   haa95532_0
certifi                         2023.5.7     py311haa95532_0
cffi                            1.15.1       py311h2bbff1b_3
charset-normalizer              2.0.4        pyhd3eb1b0_0
colorama                        0.4.6        py311haa95532_0
conda                           23.5.2       py311haa95532_0
conda-content-trust             0.1.3        py311haa95532_0
conda-libmamba-solver          23.5.0       py311haa95532_0
conda-package-handling         2.1.0        py311haa95532_0
conda-package-streaming        0.8.0        py311haa95532_0
console_shortcut_miniconda     0.1.1        haa95532_1
cryptography                   39.0.1       py311h21b164f_2
fmt                             9.1.0        h6d14046_0
idna                           3.4          py311haa95532_0
jsonpatch                      1.32         pyhd3eb1b0_0
jsonpointer                    2.1          pyhd3eb1b0_0
libarchive                     3.6.2        hb62f4d4_2
libcurl                        8.1.1        h86230a5_0
libffi                         3.4.4        hd77b12b_0
libiconv                      1.16         h2bbff1b_2
libmamba                      1.4.1        h77c03ed_1
libmambapy                    1.4.1        py311h77c03ed_1
```

Figure 8: `conda list` Response on Anaconda Prompt (miniconda3)

## Verify 'Python' and 'Jupyter' Extension on VS Code Installation

1. Open VS Code and press **Ctrl+Shift+P** on Windows or **Cmd+Option+P** on MacOS until a modal appears like the one pictured below. This modal is called the Command Palette, and from here, we can access all the functionality of VS Code.

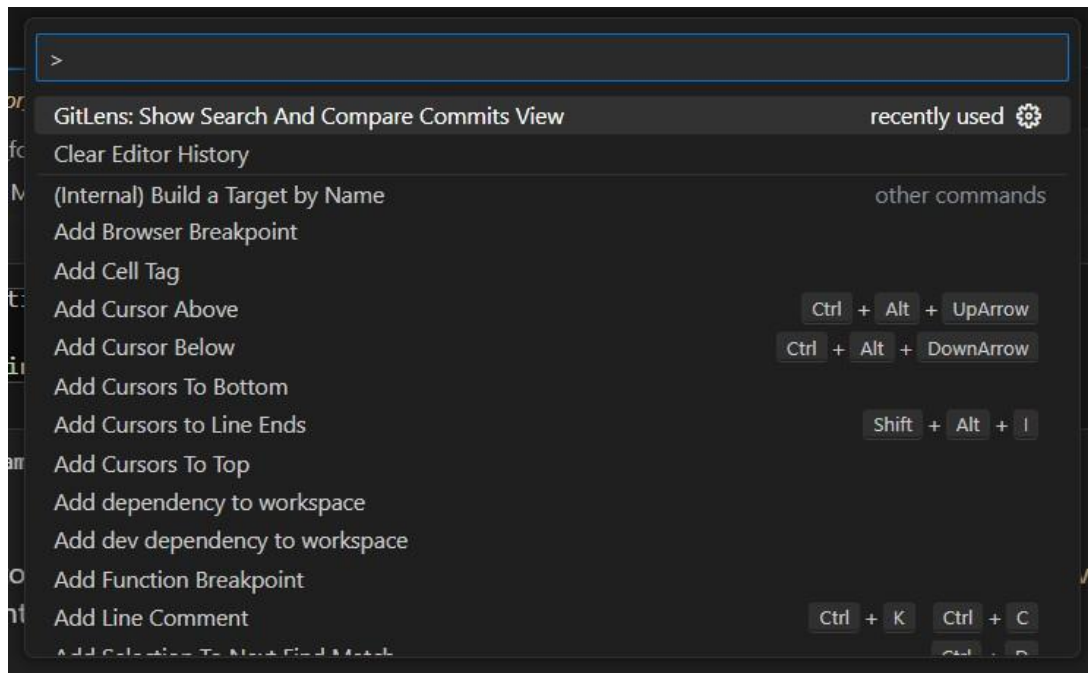


Figure 9: Opening the Command Palette in VS Code

2. Type 'interpreter' to the search bar

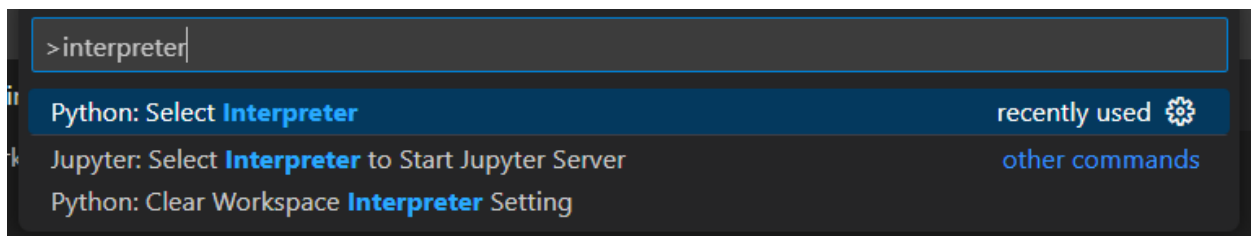


Figure 10: Typing 'Interpreter' to Find the 'Python: Select Interpreter' Menu

3. Click on "Python: Select Interpreter" then you can select the **base** environment (~\miniconda3\python.exe).

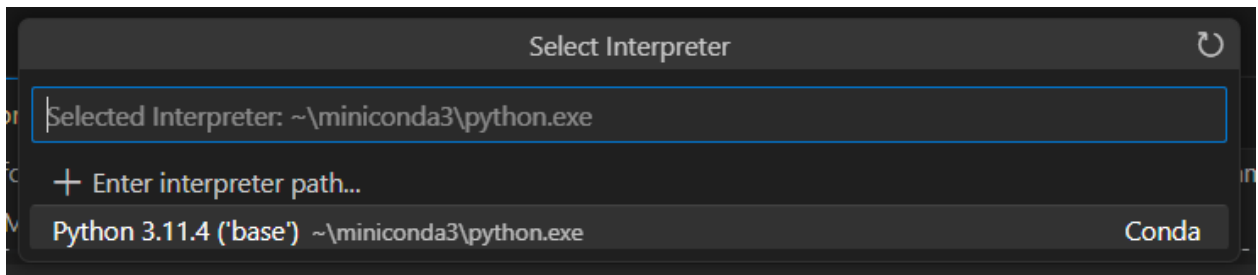


Figure 11: Selecting 'Python 3 ('base') ~\miniconda\python.exe'

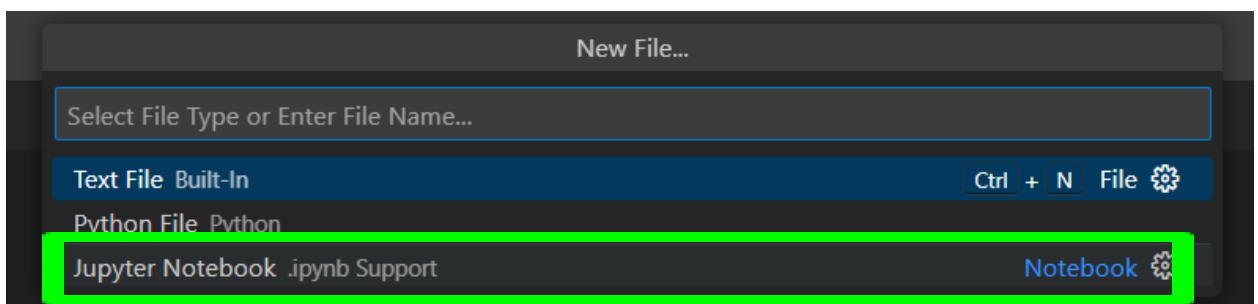
## Alternative Kernel Selection

Instead of manually using Ctrl+Shift+P or Cmd+Shift+P in VSCode, you can select the kernel by clicking on "Select Kernel" in the upper right-hand corner of your notebook

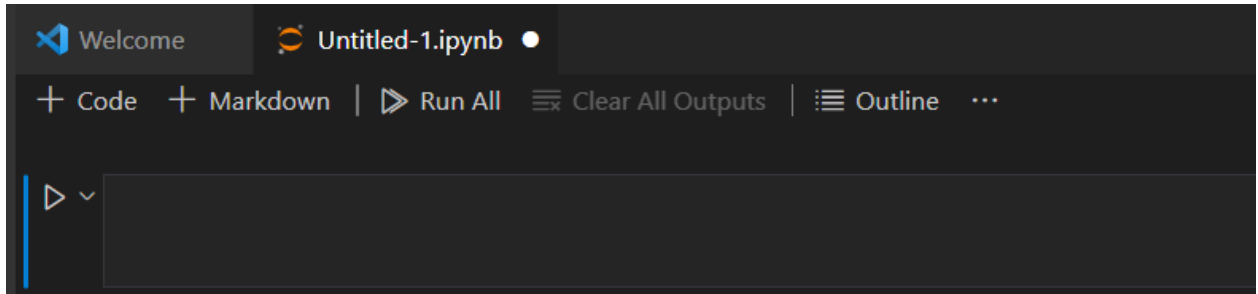
1. Create **a new notebook** by clicking the '**New File**' button.

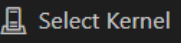


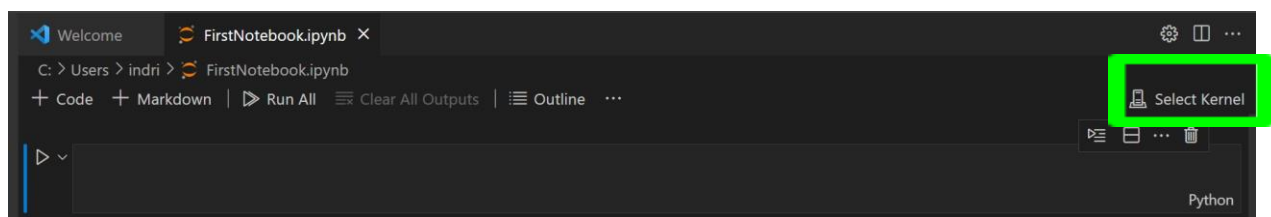
2. A modal will appear, prompting you to choose a file type. **Select the 'Jupyter Notebook .ipynb Support' file.**



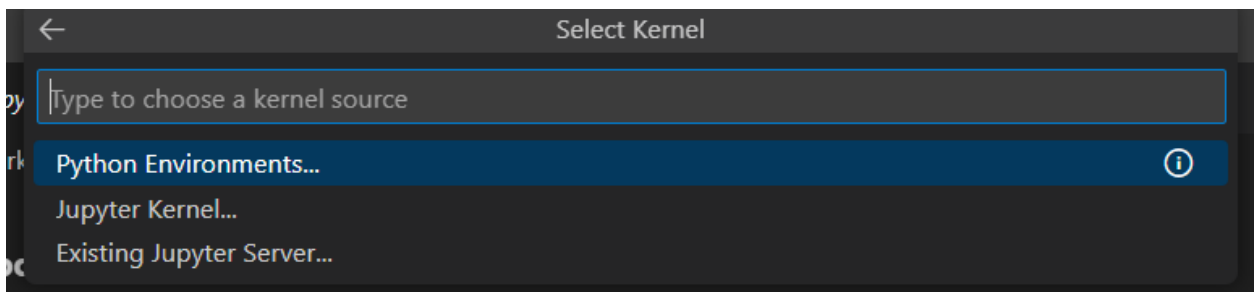
- Afterward, an 'Untitled.ipynb' file will appear as shown below. Save it as **'FirstNotebook.ipynb.'**



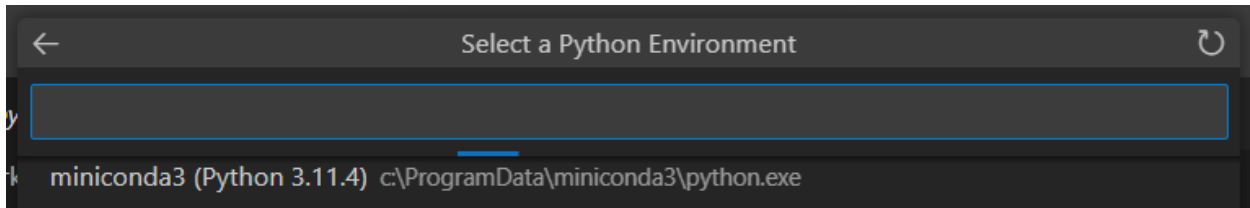
- Look for the  option on the upper right-hand corner of your notebook interface. It's typically a dropdown menu or a button with a kernel icon.



- Click on it to see a list of available kernels. Kernels are associated with specific programming languages or environments, so choose the one that matches your requirements.
  - Select **"Python Environments"**



- After installing Miniconda for the first time, you can select the **'miniconda3'** kernel, which is the default Python 3.X.X kernel provided by Miniconda



6. Once you've selected the desired kernel, your notebook will start using that kernel for code execution.

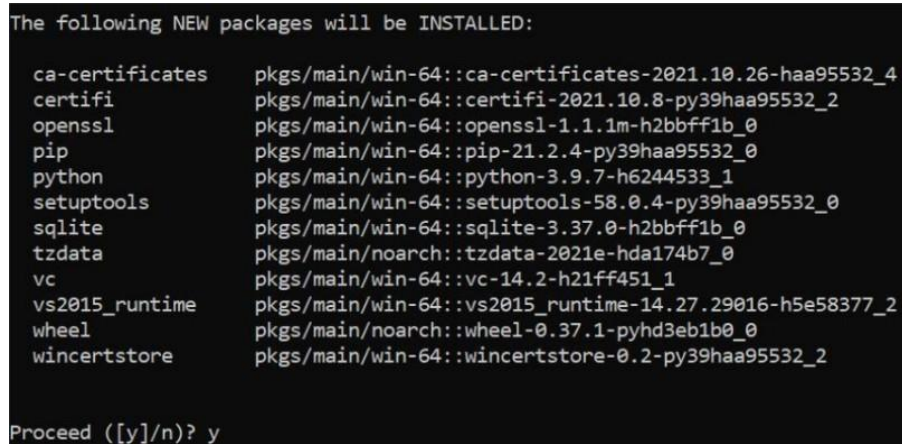
*This makes it easy to switch between different programming languages or environments within a Jupyter Notebook without having to remember keyboard shortcuts.*

## Virtual Environment and Package Preparation \*

1. Search and open **Anaconda Prompt (miniconda3)** application for windows, or **Terminal** for Mac
2. Create a new virtual environment named **dss\_donut** using Python version **3.10**

```
conda create -n dss_donut python=3.10
```

3. Proceed the installation by typing **y**



```
The following NEW packages will be INSTALLED:

ca-certificates      pkgs/main/win-64::ca-certificates-2021.10.26-haa95532_4
certifi              pkgs/main/win-64::certifi-2021.10.8-py39haa95532_2
openssl              pkgs/main/win-64::openssl-1.1.1m-h2bbff1b_0
pip                  pkgs/main/win-64::pip-21.2.4-py39haa95532_0
python               pkgs/main/win-64::python-3.9.7-h6244533_1
setuptools           pkgs/main/win-64::setuptools-58.0.4-py39haa95532_0
sqlite               pkgs/main/win-64::sqlite-3.37.0-h2bbff1b_0
tzdata               pkgs/main/noarch::tzdata-2021e-hda174b7_0
vc                   pkgs/main/win-64::vc-14.2-h21ff451_1
vs2015_runtime       pkgs/main/win-64::vs2015_runtime-14.27.29016-h5e58377_2
wheel                pkgs/main/noarch::wheel-0.37.1-pyhd3eb1b0_0
wincertstore         pkgs/main/win-64::wincertstore-0.2-py39haa95532_2

Proceed ([y]/n)? y
```


4. Activate the newly created virtual environment, namely **dss\_donut**

```
conda activate dss_donut
```

5. Change your terminal directory to the path where the **requirements.txt** is located. For example, if your txt file is in the Downloads directory, use: `cd Downloads`
6. For standardization packages and libraries installation, please install using the **requirements.txt** shared to you.

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

Wait until the installation is complete, and after that, you are good to go.



Notes: If you choose not to install the package via the shared requirements.txt, make sure you have the following package version for standardization:

1. `git+https://github.com/huggingface/transformers.git`
2. `torchvision==0.16.1`
3. `gradio==4.7.1`
4. `sentencepiece==0.1.99`
5. `protobuf==4.23.4`
6. `pandas==2.1.4`

\* : will be discussed later in day 1 class

■ ■ ■

■ ■ ■