This installation guide complements the **README** file in the GitHub repository. For Windows users, ensure that Python is available on your machine before downloading or cloning the GitHub repository.

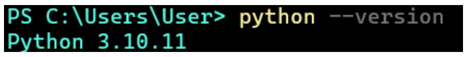
Check the version of Python:

1. Open PowerShell and type the following command:

   ```
   python —version
   ```

   Note: Use a double dash before ”version”.

2. If Python is installed, you will obtain the version, as shown in the example below:



```
PS C:\Users\User> python --version
Python 3.10.11
```

Figure 1: Version of python currently installed

3. If nothing shows, please install Python as follows:

   (a) Open Microsoft Store and search for **Python 3.8**.

   (b) Click **Get**. **Note**: Pip will be automatically installed.

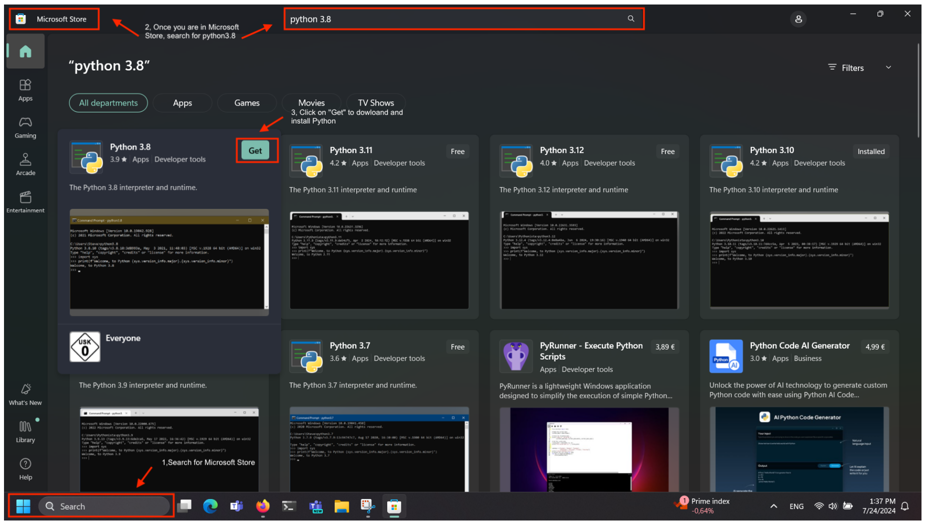


Figure 2: Install Python from Microsoft Store

4. Now, open PowerShell and check that Python is installed using the previous command.

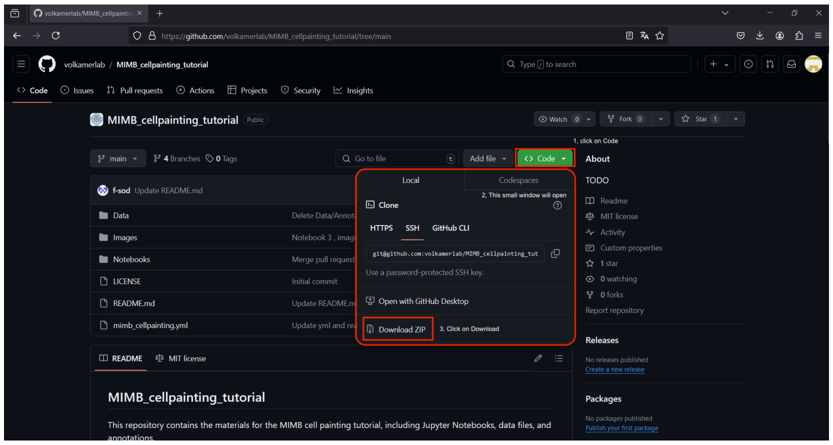
Getting Started

To clone the repository to your machine using the Git command, follow the instructions from the Readme file. If you never used Git before and it is not already installed, you can download the repository instead. The output will be the same.

1. Click the green button labelled **Code**, a small window will open. Click on **Download ZIP**.

2. The download will take between 5 to 10 minutes.

3. Once downloaded, navigate to your **Downloads** folder and unzip (right-click > unzip) the file, to your desktop or any location in your documents. Remember the path.



```
git@github.com:volkamerlab/MIMB_cellpainting_tut
```

MIMB_cellpainting_tutorial

This repository contains the materials for the MIMB cell painting tutorial, including Jupyter Notebooks, data files, and annotations.

Figure 3: Download from Github

Setting Up the Environment

This part is similar to what is described in the **README** file.

1. **Install MINICONDA**: To install and use the Miniconda installer, follow the instructions on the official Conda installation page for your operating system.

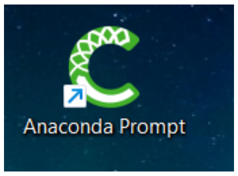


Figure 4: Anaconda Prompt icon

2. Once installed, you will have the **Anaconda Prompt** available on your machine.
3. **Pip Installation**: Pip is installed automatically when you download Python. If Python was already available on your machine, you can check if Pip is installed using one of the commands below:

```
pip3
which pip
pip —version
```

   If Pip is not installed, follow the instructions on the official Pip installation page.

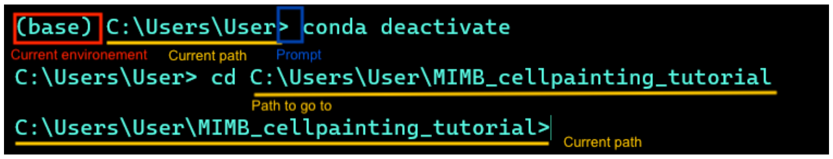
4. **Setting up the Environment**: Now that Conda is installed, you should see (base) before the prompt in your PowerShell, indicating that Conda is activated, and you are currently in the 'base' environment. To deactivate this environment, type:

```
conda deactivate
```

   Now, in your terminal, change the directory to where the `MIMB_cellpainting_tutorial` folder is located using the `cd` (change directory) command:

```
cd path
```

   To verify you are in the correct directory, use the `ls` command to list all elements in your repository. You should see the relevant files and folders.



```
(base) C:\Users\User> conda deactivate
C:\Users\User> cd C:\Users\User\MIMB_cellpainting_tutorial
C:\Users\User\MIMB_cellpainting_tutorial>
```

Figure 5: Terminal change directory, and deactivate base environment

5. To install the Conda environment needed to run the Jupyter Notebook, execute:

```
conda env create -f mimb_cellpainting.yml
```

Note: Pressing the `Tab` key will auto-complete your command and prevent mistakes in file names.

For example, if you type:

```
conda env create -f m
```

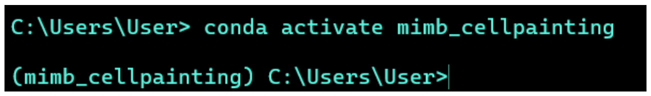
Pressing `Tab` will complete it to:

```
conda env create -f mimb_cellpainting.yml
```

This process will take some time to download and install all the libraries. It will be successful once you see a completion message. You can then activate your environment using:

```
conda activate mimb_cellpainting
```

Note: You will see the name of your current environment (mimb_cellpainting) on the left of your prompt. Only in this environment can you run the Notebooks. When you turn off and then on your computer, regardless of whether you deactivate this environment, you will see the default environment (base) next to the prompt. Deactivate it and activate the MIMB environment as needed.



```
C:\Users\User> conda activate mimb_cellpainting

(mimb_cellpainting) C:\Users\User>
```

Figure 6: Activation of the mimb_cellpainting environment

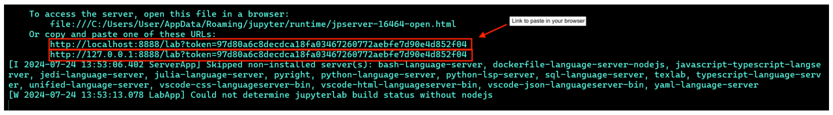
Getting started: Launch JupyterLab

JupyterLab is an integrated development environment (IDE) for interactive computing, it supports programming in various languages, with a primary focus on Python and can be used for data analysis, scientific research, machine learning, and more.

To start JupyterLab. Use either:

1. `jupyter-lab`
2. `jupyter-notebook`

It will either directly open a browser window or give you an `http://` link to copy and paste into your browser. If it occurs that using one of the above commands, the browser window opens but you cannot move from one folder to another or when clicking nothing happens, close the browser window, go back to your Power-Shell and use the `ctrl+c` command (from your keyboard) to shut down the Jupyter server.



```
    To access the server, open this file in a browser:
        file:///C:/Users/User/AppData/Roaming/jupyter/runtime/jpserver-16464-open.html
    Or copy and paste one of these URLs:
        http://localhost:8888/lab?token=97d80a6c8decdca18fa03467260772aebfe7d90e4d852f04
        http://127.0.0.1:8888/lab?token=97d80a6c8decdca18fa03467260772aebfe7d90e4d852f04
[I 2024-07-24 13:53:06.402 ServerApp] Skipped non-installed server(s): bash-language-server, dockerfile-language-server-nodejs, javascript-typescript-langse
rver, jedi-language-server, julia-language-server, pyright, python-language-server, python-lsp-server, sql-language-server, texlab, typescript-language-serv
er, unified-language-server, vscode-css-languageserver-bin, vscode-html-languageserver-bin, vscode-json-languageserver-bin, yaml-language-server
[W 2024-07-24 13:53:13.078 LabApp] Could not determine jupyterlab build status without nodejs
```

Figure 7: Run Jupyter in your terminal

JupyterLab Interface

Overview of the Interface

JupyterLab provides an interactive development environment for working with notebooks, code, and data. The interface is organized into several key components:

1. **JupyterLab Window Segmentation:**
   - (a) **Left Folder Pane:** This pane displays the file directory of your project, allowing you to navigate and manage files.
   - (b) **Main Page:** The central area where notebooks and other documents are opened and edited.
   - (c) **Command Palette**
   - (d) **Upper Right:** This section shows the status and controls of the currently active kernel.
2. **Command Palette:**
   - (a) **Save Disk Icon:** Saves the current state of the notebook.
   - (b) **Add Icon (+):** Adds a new cell below the currently selected cell.
   - (c) **Cut Icon (Scissors):** Cuts the selected cell.
   - (d) **Copy Icon (Clipboard):** Copies the selected cell.
   - (e) **Paste Icon (Clipboard with Arrow):** Pastes the copied or cut cell below the currently selected cell.
   - (f) **Run Icon (Play Button):** Executes the code in the selected cell.
   - (g) **Interrupt Icon (Square):** Interrupts the execution of the current cell.
   - (h) **Restart Icon (Circular Arrow):** Restarts the kernel, clearing all variables and outputs.
   - (i) **Cell Type(Markdown):** Changes the cell type between Code, Markdown, Raw etc.
   - (j) **Kernel Selector:** Allows selection of the kernel to use for executing the code (e.g., Python, R).

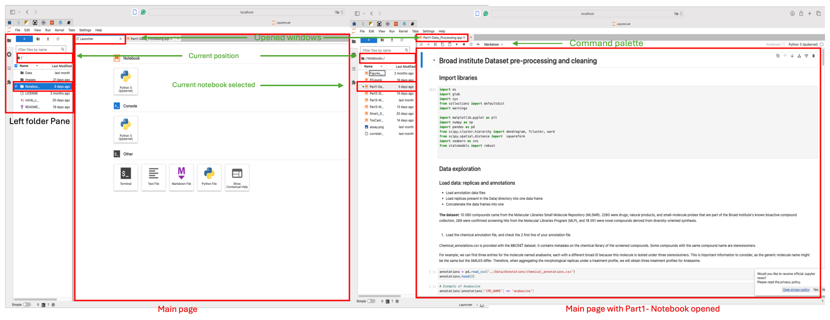

Opened windows
Current position
Current notebook selected
Left folder Pane
Main page
Command palette
Broad institute Dataset pre-processing and cleaning
Import libraries
Data exploration
Main page with Part1- Notebook opened

Figure 8: JupyterLab Interface

Working with Notebooks

Notebooks Basics: A notebook in JupyterLab consists of cells, which can be either code cells or markdown cells.

- **Code Cells:**
  - *Writing and Executing Python Code:* Write your Python code in these cells and execute them to see the output below the cells.
  - *Displaying Output:* The results of the executed code appear directly under the code cell.
  - *Keyboard Shortcuts:* Use shortcuts like `Ctrl + Enter` to run the current cell, and `Shift + Enter` to run the current cell and move to the next one.
- **Markdown Cells:**
  - *Writing Formatted Text:* Use Markdown syntax to create formatted text.
  - *Basic Markdown Syntax:* Includes headers, lists, bold/italic text, links, and images.

Exporting Notebooks

JupyterLab allows you to export your notebooks to various formats for sharing and presentation purposes:

- *HTML:* Export your notebook as an HTML file for web viewing.
- *PDF:* Export your notebook as a PDF for printing or distribution.
- *Python Script:* Export the notebook as a Python script for running in any Python environment.