Jupyter (formerly IPython Notebook) is an open-source project that lets you easily combine Markdown text and executable Python source code on one canvas called a notebook. All of the workshop's modules will be distributed via this files, setting up the environment will be vital to continue from this point on. In the following sections, you will see a couple options that you can pick from: Intel devcloud setup (1) and VS Code setup (2).

In case anyone is stalled here, please do not hesitate to ask for help to the CE CoP core team.

1 Intel Devcloud Setup

The Intel Devcloud provides a strightforward way to create and use Jupyter Notebooks with no extra steps required to setup an environment. Once you have the access to the Intel Devcloud that is everything you need to start working on it.

1.1 Importing the files into the Devcloud

In order to have the files as part of the devcloud, the following steps will be required:

1. Unzip the .ipynb files provided for this workshop in a known location.

2. Go to https://jupyter.oneapi.devcloud.intel.com/hub/login?next=%2Fhub%2F.

3. Press "Click Here to Sign In" button.

4. Provide your intel's email and press "Continue" button

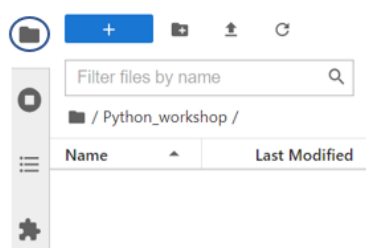5. Select the file browser view by clicking the folder icon on the left as shown in Figure 1.

Figure 1: File browser

6. Select "New Folder" option as presented in Figure 2.

7. Enter "Python_workshop" as folder's name.

8. Open the "Python_workshop" folder by double clicking over its name.

9. Select "Upload Files" option. See Figure 3 for reference.

Figure 2: New folder creation

Figure 3: Uploading the files

10. Select all the .ipynb files and press the "Open" button.

11. Open the files by double clicking over its name.

2 VS Code Setup

Visual Studio Code supports working with Jupyter Notebooks natively, and through Python code files. To work with Python in Jupyter Notebooks, you must activate an Anaconda environment in VS Code, or another Python environment in which you've installed the Jupyter package.

2.1 Automatically setting up your environment

VS Code can handle the installation of those packages automatically. Once you request to create a new file and it does not know how to deal with it, it will suggest to install what is required to execute that specific file extension. In Figure 4 you will see where to create a new file.
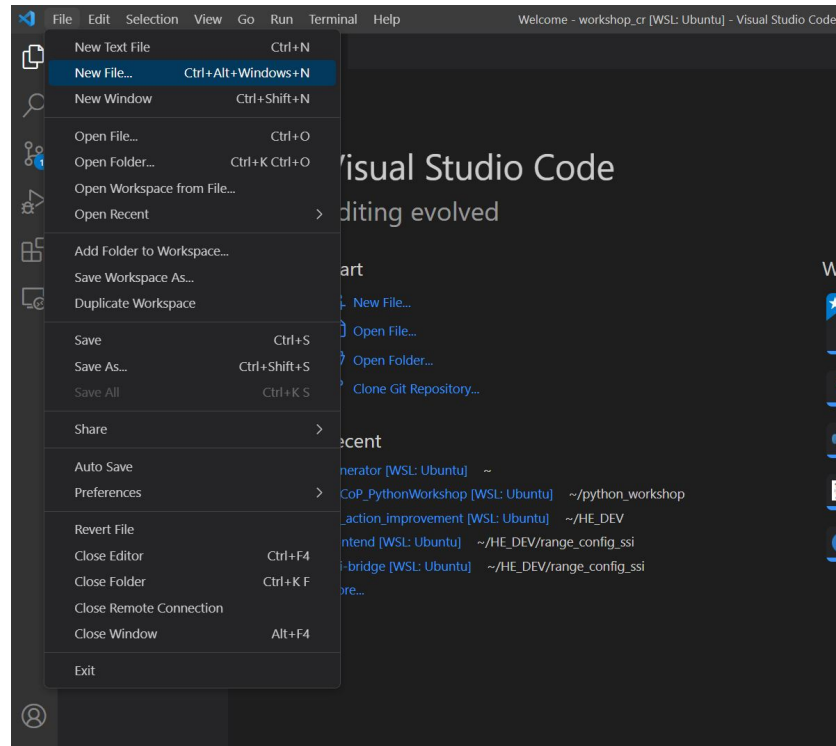

Figure 4: Creating a new file

Then you will require to select the file extension for VS Code to trigger the automatic setup. In Figure 5, once you type "Jup" to look for the file extension, it will suggest the type you are looking for.
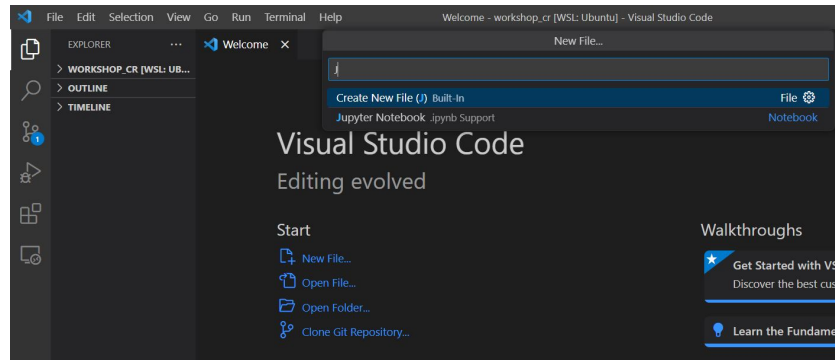

Figure 5: Picking the file extension

As soon as you click on the type, an empty file will be created such as the one presented in Figure 6.
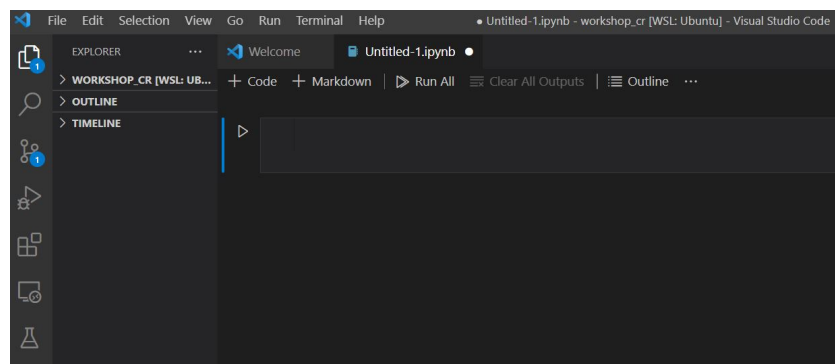

Figure 6: Empty file created

There are a couple places where to execute the file, but whenever you hit the play button in a similar fashion than in Figure 7.a, VS Code will automatically try to execute the code, if it does not have an interpreter an the required packages, it will suggest you to proceed with the installations as in Figure 7.b.
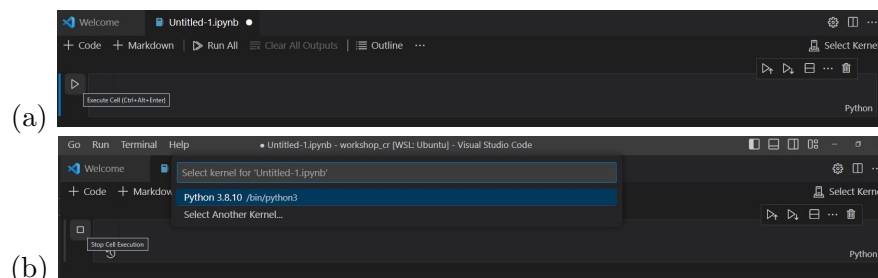

Figure 7: Play button on a new file

4

2.2   Manual installation/setup

To select an environment, use the Python: Select Interpreter command from the Command Palette (Ctrl+Shift+P). Once the appropriate environment is activated, you can create and open a Jupyter Notebook, connect to a remote Jupyter server for running code cells, and export a Jupyter Notebook as a Python file.

In case you want to have a better understanding on how this is managed, please visit Jupyter Notebooks in VS Code.

2.3   Importing the files into VS Code

Having the files placed into VS Code is quite simple, please follow these steps:

1. Unzip the .ipynb files provided for this workshop in a known location.

2. Have VS Code opened.
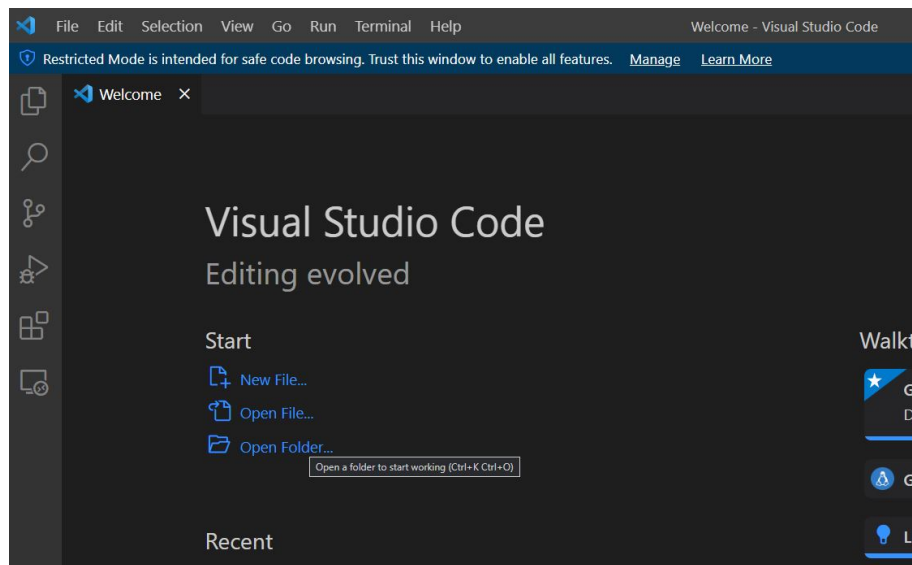
3. As shown in Figure 8, hit on "Open Folder...".

Figure 8: Importing the workshop directory.

4. On the "Open Folder" view, select the known path where the files were extracted. Figure 9 is a reference.
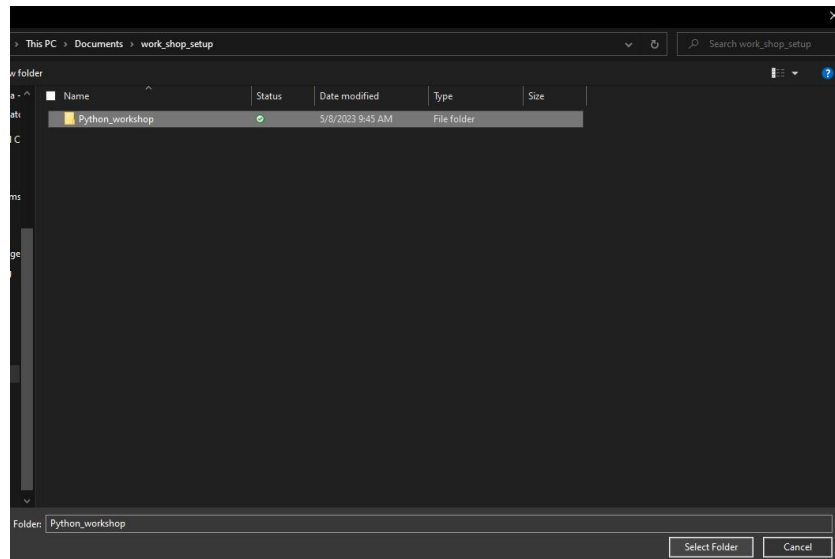

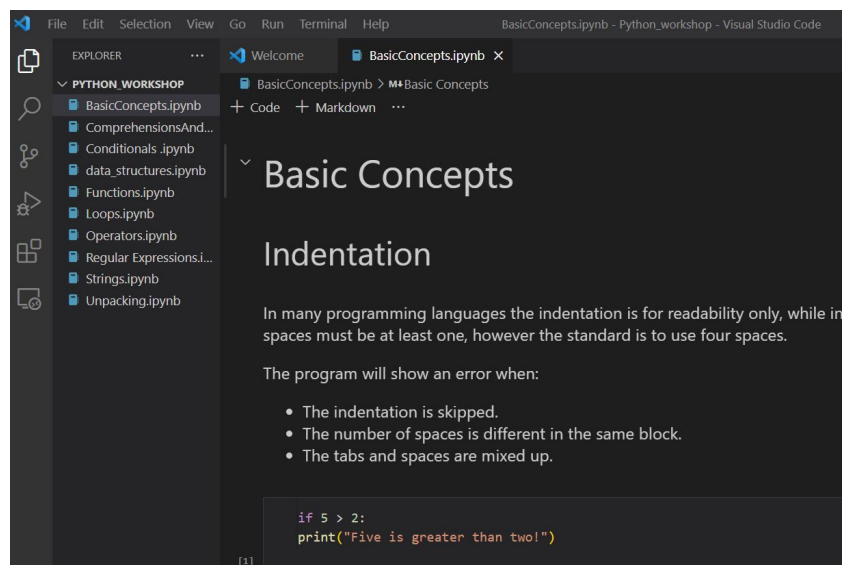
Figure 9: Opening files.

5. Click on any file that has been imported to open it. See Figure 10.

Figure 10: Opening files.