



Introduction to Jupyter notebooks

What is a notebook?

- A notebook integrates code and its output into a single document.
- A notebook combines the following:
 - Visualizations
 - Narrative text
 - Mathematical equations
 - Charts and plots
 - Other rich media

Code Cells

A code cell contains code to be run in the kernel. When the code is run, the notebook displays the output below the code cell that generated it.

We can write any Python in a code cell:

```
In [7]: # this is a comment in a code cell
# run this cell to see what it outputs
x = 5
powers_of_five = [x * i for i in range(5)]
print(powers_of_five)

[0, 5, 10, 15, 20]
```

Screenshot from a notebook that you will use later in a lab



A notebook is a software development environment (SDE).

Why are notebooks important for ML?

- If you want to work in data science, notebooks are critical tools to learn and be able to use.
- A notebook can do the following:
 - Accelerate your workflow to work with data because it offers great tools to explore and prepare data
 - Make it easier to communicate and share your results



Using notebooks is now a major part of the data science workflow at companies across the globe.

Introduction to Jupyter Notebook

- Jupyter Notebook is an open-source project that implements notebooks.
- Jupyter is a reference to the three core languages that the project supports: **Julia**, **Python**, and **R**.
- Kernels are available that make Jupyter compatible with dozens of languages, including Ruby, PHP, JavaScript, SQL, and Node.js.



For more information about Project Jupyter, see the Jupyter website at <https://jupyter.org>. You can download the Jupyter Notebook application from the Jupyter website at <https://jupyter.org/install>. You can also download the application as part of the Anaconda data science toolkit from the Anaconda website at <https://www.anaconda.com/products/distribution>.

Although you can use many programming languages in Jupyter Notebook, this course will focus on Python, which is the most common use case.

Among R users, RStudio is a more popular choice. For more information, see Tutorial: Getting Started with R and RStudio at <https://www.dataquest.io/blog/tutorial-getting-started-with-r-and-rstudio>.

What is the Jupyter Notebook application?

- Jupyter Notebook is a server-client application.
- You can edit and run your notebook by using a web browser.
- You can run the application on a PC without internet access.
- You can install Jupyter Notebook on a remote server and access it through the internet.

Main components of Jupyter Notebook

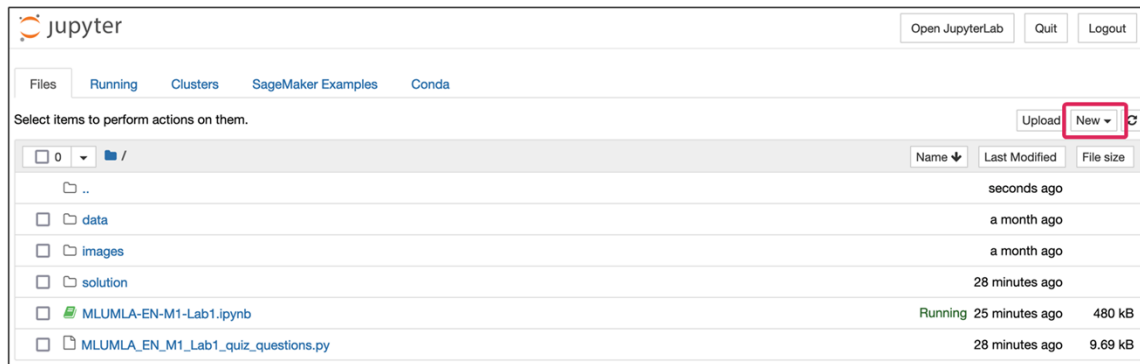
- **Kernel:**
 - A kernel is a program that runs and inspects the user's code.
 - The Jupyter Notebook application has a kernel for Python code.
- **Dashboard:** Use the dashboard to manage both the notebooks and the kernels.

A kernel is a program that runs and inspects the user's code. The Jupyter Notebook application has a kernel for Python code, but kernels are also available for other programming languages.

The dashboard shows you the notebooks that you have made and can open, and you can also use the dashboard to manage your kernels. You can see which notebooks are running and shut them down if necessary.

Creating a Jupyter notebook

On the dashboard, choose **New**.

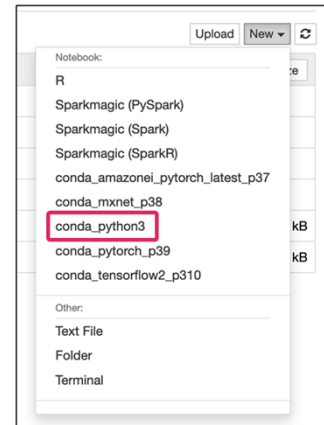


To create a new Jupyter notebook and select a kernel, choose **New** in the upper-right corner of the dashboard.

Selecting the kernel

Select the kernel that you need. A new, empty notebook opens.

For example, you can select a
Python 3 kernel.

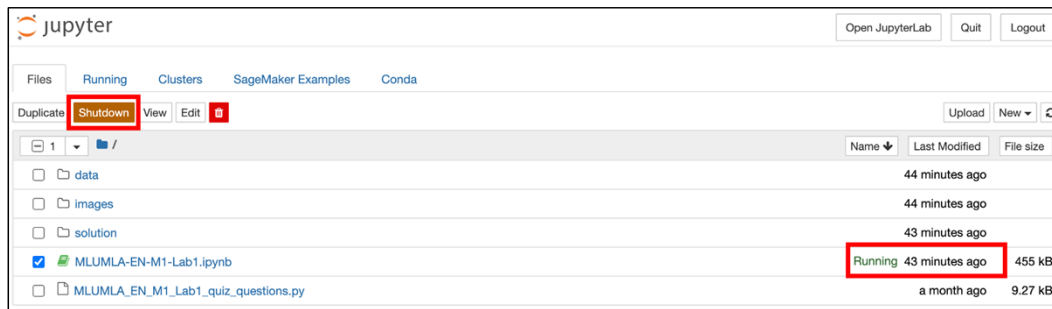


After you choose **New**, select the kernel that you need. A new, empty notebook opens.

Managing a Jupyter notebook kernel

On the dashboard, you can also manage the kernels of existing notebooks.

**Shut down
the kernel.**



The screenshot shows the Jupyter dashboard interface. At the top, there are buttons for 'Open JupyterLab', 'Quit', and 'Logout'. Below these are tabs for 'Files', 'Running', 'Clusters', 'SageMaker Examples', and 'Conda'. The 'Running' tab is active. In the toolbar, the 'Shutdown' button is highlighted with a red box. Below the toolbar is a table of notebooks. The notebook 'MLUMLA-EN-M1-Lab1.ipynb' is selected, and its status 'Running 43 minutes ago' is highlighted with a red box. The table also shows other notebooks like 'data', 'images', 'solution', and 'MLUMLA_EN_M1_Lab1_quiz_questions.py'.

Name	Last Modified	File size
data	44 minutes ago	
images	44 minutes ago	
solution	43 minutes ago	
MLUMLA-EN-M1-Lab1.ipynb	Running 43 minutes ago	455 kB
MLUMLA_EN_M1_Lab1_quiz_questions.py	a month ago	9.27 kB

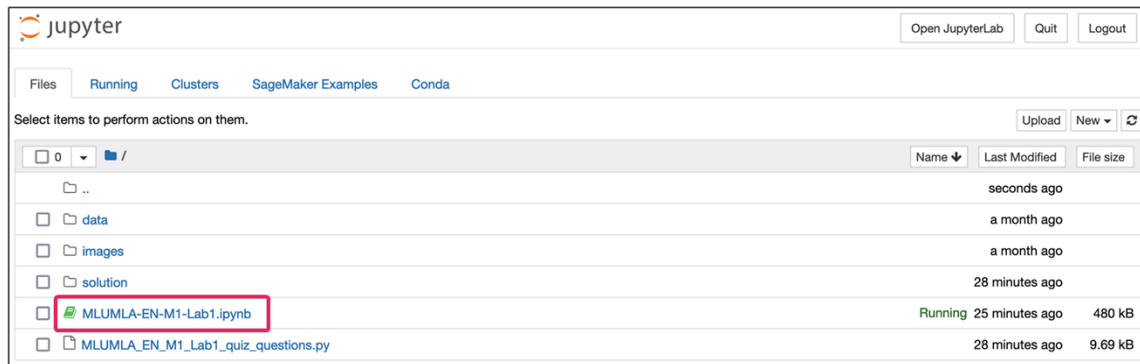
**See which
notebooks are
running.**



On the dashboard, you can also manage the kernels of existing notebooks. You can see which ones are running and shut them down if necessary.

Opening an existing Jupyter notebook

On the dashboard, choose the link for the notebook name.



The screenshot shows the Jupyter dashboard interface. At the top, there's a header with the Jupyter logo and buttons for 'Open JupyterLab', 'Quit', and 'Logout'. Below the header, there are tabs for 'Files', 'Running', 'Clusters', 'SageMaker Examples', and 'Conda'. The 'Files' tab is active, showing a file browser view. The file browser displays a list of files and folders. The file 'MLUMLA-EN-M1-Lab1.ipynb' is highlighted with a red box. The file is a Jupyter notebook, indicated by the icon. The status of the notebook is 'Running', and it was last modified '25 minutes ago' with a size of '480 kB'. Other files in the list include 'data', 'images', 'solution', and 'MLUMLA_EN_M1_Lab1_quiz_questions.py'.

Name	Last Modified	File size
..	seconds ago	
data	a month ago	
images	a month ago	
solution	28 minutes ago	
MLUMLA-EN-M1-Lab1.ipynb	Running 25 minutes ago	480 kB
MLUMLA_EN_M1_Lab1_quiz_questions.py	28 minutes ago	9.69 kB

To open an existing Jupyter notebook from the dashboard, choose the link for the notebook name.

Inside a Jupyter notebook

In the next two labs, you will learn how a Jupyter notebook works.

Try it Yourself!



Activity

Select the cell below and then use the **Run** button in the Toolbar (or press Ctrl/Cmd+Enter) to run the cell!

```
In [2]: import time
import sys

sys.path.insert(1, "..")

from MLUMLA_EN_M1_Lab1_quiz_questions import *
```

After you open a notebook, you can start to build! In the next two labs for this module, you will learn how a Jupyter notebook works.



Using SageMaker to host and manage Jupyter notebooks

Introduction to SageMaker

Amazon SageMaker is a fully managed service for Jupyter notebooks:

- Handles provisioning and hosting of notebooks and their required compute resources (instances) for you
- Uses instances that are optimized for ML
- Replaces failed hardware automatically



Amazon SageMaker is a fully managed service that is based on Jupyter notebooks. For more information, see the Amazon SageMaker page at <https://aws.amazon.com/pm/sagemaker>. With the service, you can prepare data and build, train, and deploy ML models for any use case with fully managed infrastructure, tools, and workflows.

SageMaker is serverless

- Data scientists don't need to handle low-level environment setups.
- You can focus on the fun part: developing, training, and deploying ML models.
- In the labs of this course, you will learn how to use this great tool to manage your ML projects.



Because SageMaker is serverless, you can focus on developing, training, and deploying ML models.