**Introduction to Jupyter Notebooks**

Jupyter Notebooks originated as “iPython,” originally developed for Python programming. Later, when it started supporting additional languages, it was renamed Jupyter, which stands for Julia, Python, and R. However, now, it supports many other languages. A Jupyter Notebook is a browser-based application that allows you to create and share documents containing code, equations, visualizations, narrative text links, and more. It is like a scientist’s lab notebook, where a scientist records all steps to perform their experiments and the results they can reproduce. In the same way, a Jupyter Notebook allows a Data Scientist to record their data experiments and results that others can reuse. Now a Jupyter Notebook file allows you to combine descriptive text, code blocks, and code output in a single file. When you run the code, it generates the output, including plots and tables, within the notebook file. And you can then export the notebook to a PDF or HTML file format that can then be shared with anyone. Next, let’s learn about Jupyter Lab. Jupyter Lab is a browser-based application that allows you to access multiple Jupyter Notebook files, other code, and data files. In addition, it extends the functionalities of Jupyter Notebooks by enabling you to work with multiple notebooks, text editors, terminals, and custom components in a flexible, integrated, and extensible manner. It is compatible with several file formats like CSV, JSON, PDF, Vega, and so on. And it is also an open source. Jupyter Notebooks can be used with cloud-based services like IBM and Google Colab. They don't require any installation on your local machine. They give you access to the Jupyter Notebook environment and allow you to import and export notebooks using the standard IPython Notebook file format. Also, these services support the Python language and other languages as well. Jupyter Notebooks can be installed via the command line using the pip install function. It can also be downloaded locally on your laptop through the Anaconda Platform from Anaconda dot com. Anaconda is one of the popular distributions which includes Jupyter and Jupyterlab. So, for this course, you have access to a hosted version of JupyterLab in Skills Network Labs, so you do not require any installations on your own device to complete the hands-on labs. As shown here, you will see a screen that will launch the Jupyter Lab in the virtual environment. Simply click the Open tool tab. In this video, you learned that Jupyter Notebooks are used in Data Science for recording experiments and projects. Jupyter Lab is compatible with many files and Data Science languages. And there are different ways to install and use Jupyter Notebooks.

**Getting started with Jupyter**

In the lab session of this module, you can launch a notebook using the Skills Network virtual environment. After selecting the check box, click the Open tool tab, and the environment will open the Jupyter Lab. Here you see the open notebook. On opening the notebook, you can change the name of the notebook. Click File. Then click Rename Notebook to give the required name. And you can now start working on your new notebook. In the new notebook, print “hello world”.

Then click the Run button to show that the environment is giving the correct output. On the main menu bar at the top, click Run. In the drop-down menu, click Run Selected Cells to run the current highlighted cells. Alternatively, you can use a shortcut, press Shift + Enter. In case you have multiple code cells, click Run All cells to run the code in all the cells. You can add code by inserting a new cell. To add a new cell, click the plus symbol in the toolbar. In addition, you can delete a cell. Highlight the cell and on the main menu bar, click Edit, and then click Delete Cells. Alternatively, you can use a shortcut by pressing D twice on the highlighted cell. Also, you can move the cells up or down as required. So, now you have learned to work with a single notebook. Next, let’s learn to work with multiple notebooks. Click the plus button on the toolbar and select the file you want to open. Another notebook will open. Alternatively, you can click File on the menu bar and click Open a new launcher or Open a new notebook. And when you open the new file, you can move them around. For example, as shown, you can place the notebooks side by side. On one notebook, you can assign variable one to the number 1, and variable two to the number 2 and then you can print the result of adding the numbers one and two.

As a data scientist, it is important to communicate your results. Jupyter supports presenting results directly from the notebooks. You can create a Markdown to add titles and text descriptions to help with the flow of the presentation. To add markdown, click Code and select Markdown. You can create line plots and convert each cell and output into a slide or sub-slide in the form of a presentation.

The slides functionality in Jupyter allows you to deliver code, visualization, text, and outputs of the executed code as part of a project.

Now, when you have completed working with your notebook or notebooks, you can shut them down. Shutting down notebooks release their memory. Click the stop icon on the sidebar, it is the second icon from the top. You can terminate all sessions at once or shut them down individually. And after you shut down the notebook session, you will see “no kernel” at the top right. This confirms it is no longer active, you can now close the tabs.

In this video, you learned how to: Run, delete, and insert a code cell. Run multiple notebooks at the same time. Present a notebook using a combination of Markdown and code cells. And shut down your notebook sessions after you have completed your work.

**Jupyter Kernels**

A notebook kernel is a computational engine that executes the code contained in a Notebook file. Jupyter Kernels for many languages exist, and we will explore some that are relevant in Data Science. When a Notebook document opens, the related kernel launches automatically. When the Notebook is executed, the kernel performs the computation and produces the results. Depending on your settings, you may need to install other notebook languages in your Jupyter environment. In the Skills Network lab environment, a few languages have been pre-installed for you. The first one is the Python kernel. When you launch a notebook, pick the language you are interested in for your Data Science project. The Python kernel allows you to run python cells. You can run the Python script in the cells to produce an output. The top right corner of the Notebook shows the name of the kernel. Here it shows the Python kernel. You have the option to run other kernels. The Skills Network virtual Jupyter environment has Apache, Julia, R, and Swift. You can use any language to execute your code, either by selecting the kernel on the launch page or clicking the top right icon and selecting the kernel from the drop-down menu. If running the kernel on your local machine, you will need to manually install the languages through your command line interface (CLI). In this video, you learned that The kernel acts like a computational engine and executes the code in a Notebook file. Jupyter Notebook supports different languages, and you can switch to a different kernel as per your requirement.

**Jupyter Architecture**

Jupyter architecture implements a two-process model with a kernel and a client. The client is the interface offering the user the ability to send code to the kernel. It is the browser in a Jupyter Notebook. The kernel executes the code and returns the result to the client for display. Jupyter Notebooks represent your code, metadata, contents, and outputs. When you save the Notebook, it is sent from your browser to the Notebook server. It saves the notebook file on a disk as a JSON file with a .ipynb (pronounced as dot i PI NB) extension. The Notebook server is responsible for saving and loading the notebooks. And the kernel executes the cells of code contained in the Notebook when the user runs them. The Jupyter architecture uses the NB convert tool to convert files to other formats. For example, if we want to convert a notebook file into an HTML file, the notebook is first modified by a preprocessor, then an exporter converts the notebook to the new file format. Finally, a postprocessor will work on the exported file to give the final output. After conversion, on giving the url of the file, the HTML file displays.

In this video, you learned that: Jupyter implements a two-process model with a kernel and a client. The Notebook server is responsible for saving and loading the notebooks. The kernel executes the cells of code contained in the Notebook. And the Jupyter architecture uses the NB convert tool to convert files to other formats.

**Additional Anaconda Jupyter Environments**

Computational notebooks combine code, computational output, explanatory text, and multimedia resources into a single document. Jupyter Notebook is a popular type of computational notebook because it supports dozens of programming languages. JupyterLab and VS Code are popular environments for creating and modifying Jupyter Notebooks on a local device. JupyterLab is an open-source, web-based application based on Jupyter Notebook. You can create code, interactive visualizations, text, and equations, just like with Jupyter Notebook. JupyterLab includes expanded features with some of Anaconda's most extensive pre-installed Python libraries, including NumPy, Pandas, and Matplotlib. Anaconda is a free and open-source distributor for Python and R, the top languages used in data science and machine learning. Anaconda has fifteen hundred plus libraries. It is free to install and has free community support for any users who need help with Python. The downloadable Anaconda Navigator graphical user interface allows users to install new packages on their local device without using a command line interface or ‘CLI.’ You can download Anaconda Navigator from the given URL. Here is the home page of the Anaconda Navigator. To launch JupyterLab, click Launch in the JupyterLab box. If the Launch button is missing, click Install first, and then click Launch. To start with the Jupyter Notebook, type Jupyter Notebook(anaconda3) in the search bar and press enter. The JupyterLab dashboard opens in the n on the localhost. It is specifically designed to manage Jupyter Notebooks. To create a new Jupyter Notebook, click New and select Python 3. This opens a new notebook in a new tab. You will see the URL, which shows the filename and the kernel. It also shows the Last Checkpoint. Let’s rename your notebook by clicking Untitled. Type a name for the notebook and click Rename. Next, you will review two main cell types: Code and Markdown. In the dropdown menu, select Code. A code cell contains code to be executed in the kernel and displays its output. To execute the cell, click Run. Alternatively, in the dropdown menu, select Markdown. A Markdown cell contains rich text and displays its output in place when it executes. To download a notebook, go to File and click Download as. You will see several download options. You can select the option you want. VS Code is a free, open-source code editor for debugging and task-running operations. VS code works on Linux, Windows, and macOS. It supports multiple languages, syntax highlighting, auto-indentation, and more. VS Code is one of the most popular development environment tools. If you prefer to install VS code separately, without using Anaconda Navigator, you can go to code.visualstudio.com, click the download option that applies to your device, then follow the install instructions. A separate installation of VS Code will work the same as in Anaconda Navigator, but it will not configure for Anaconda, Python, or Jupyter Notebooks. To open VS Code using Anaconda Navigator, open Anaconda Navigator, find the VS Code application, and click Launch. Once installed, you will see the Get Started screen. You need to install a few extensions to execute Python code in VS Code. First, click Extensions or use Ctrl + Shift + X keys to open Extensions. Then search for “Python”; all the extensions related to Python will appear. Once you install the extensions, click File. Then select New File. In New File, select Jupyter Notebook. The notebook will look like this. Notice that the kernel is Python. Write your code and then execute it using the RUN icon. You will get a confirmation that your code has been executed successfully. And finally, navigate to File and select Save.

In this video, you learned that Jupyter is a popular computational notebook tool because it supports dozens of programming languages. The Anaconda Navigator GUI can launch multiple applications on a local device. Jupyter environments in the Anaconda Navigator include JupyterLab and VS Code. And you can download Jupyter environments separately from the Anaconda Navigator, but they may not be configured properly.

**Additional Cloud-Based Jupyter Environments**

Computational notebooks combine code, computational output, explanatory text, and multimedia resources in a single document. ​ Jupyter notebook is a popular type of computational notebook because it supports dozens of programming languages. ​ Popular cloud-based environments used to create and modify Jupyter notebooks include:​ JupyterLite, and​ GoogleColaboratory​.​ JupyterLite is a lightweight tool built from JupyterLab components ​that executes entirely in the browser.​ JupyterLite does not require a dedicated Jupyter server. ​ Only a web server is required, which means ​we can deploy JupyerLite as a static website.​ We can also use it to create interactive graphics and visualizations because it supports many visualization libraries like Altair, Plotly, and ipywidgets​. Since JupyterLite is a distribution of JupyterLab, ​it includes the latest improvements and features.​ To launch JupyterLite, open a browser and type jupyter.org/try-jupyter/lab in the URL field. Then press Enter.​ JupyterLite will appear.​ Next, click Python(Pyodide). ​​ Here is a view of a JupyterLite notebook. We know this is a JupyterLite notebook because ​we see the kernel is Python Pyodide. ​ This kernel allows installing and running Python packages in a browser. You will notice different kernels depending on the type of Jupyter environment you use. For cloud-based Jupyter environments, Python Pyodide and Python Pyolite are common kernels. ​ The default kernel for JupyterLite is​ Pyolite. ​ Pyolite is a Python kernel based on Pyodide. Pyolite runs in the background, so that intensive computations can execute quickly. ​ Other kernels can also be used with JupyterLite.​ Google Colaboratory (or 'GoogleColab') is a free Jupyter notebook environment that runs entirely in the cloud.​ GoogleColab Jupyter notebooks execute on a browser, and GoogleColab projects are stored on Google Drive and GitHub. ​ You can upload and share notebooks without setup and installation.​ You can also clone projects from GitHub and execute them in GoogleColab.​ Most machine learning and visualization libraries are pre-installed, like scikit-learn and matplotlib.​ With GoogleCollab, you can develop many trending data science applications “on the fly”, which is to say, quickly without a lot of setup or preparation.​

To open the Colab notebook, open Google Drive, and​ click New​​ To explore GoogleColab,​ from the Google Drive menu, select More. ​Then select Google Colaboratory.​​ The GoogleColab notebook will appear.​ In the notebook, write the code ​in the code section, and then to execute the code,​ click the Run icon.​ To add more Code or Text cells, you need to click ​+Code and​ +Text. ​ Here, text cells are used to write rich text, or you can set these cells as Markdown cells.​

In this video, you learned that:​ Jupyter is a popular computational notebook tool because it supports dozens of programming languages.​ The Anaconda Navigator GUI can launch multiple applications.​ Additional open-source Jupyter environments include the following: JupyterLab, JupyterLite, VS Code, and Google Colaboratory. ​ JupyterLite is a browser-based tool.​