## **Traditional IDEs**

Most IDEs include a source code editor, debugger, and compiler. They work perfectly for tech assessments for programmers and developers. Not for data science and machine learning assignments though.

In many data science problems, the solution can be a simple prediction or a ‘Yes/No’ answer. You may have to change the parameters quickly and want to change only one line or variable and not re-run all code.

Arriving at any result may require hours of logical analysis.

## 

## **Hence, Jupyter Notebooks**

[A read–eval–print loop (REPL) is a simple interactive computer programming environment that takes single user inputs, executes them, and returns the result to the user; a program written in a REPL environment is executed piecewise.](https://en.wikipedia.org/wiki/Read%E2%80%93eval%E2%80%93print_loop)

The [Jupyter Notebook](https://jupyter.org/) is an open-source web application that allows users to create and share documents containing live code, equations, visualizations, and narrative text. The easy-to-use, interactive data science environment provided by Jupyter works across several programming languages such as Python, R, Julia, and also many others. Jupyter Notebooks **not only work like an IDE**, but also as a **presentation or education tool**, and are great for data science assessments where the candidate is required to answer questions in a visual format.

**Individual cells for better analysis**Jupyter Notebooks allow to code using separate units or ‘cells’ that can be used independently of each other. This makes it easier to compute how various data parameters work with each other and to add notes or to partially write and test code.

**Interactive elements for better data visualization**The Notebook offers an interactive shell with embeddable graphics and tables, reusable cells, and some other presentation features which are relevant to the job at hand. This enables to present their output in a graphical format if needed; something that a traditional IDE does not support.

**Bash Commands: Right into your code**

**Tab Completion**

Why Should I Use Google Colab?  
  
There are several reasons to opt to use Google Colab instead of a plain Jupyter Notebook instance:  
 Pre-Installed Libraries  
 Saved on the Cloud  
 Collaboration  
 Free GPU and TPU Use  
 Let’s see these advantages in more detail:

Pre-Installed Libraries  
Anaconda distribution of Jupyter Notebook shipped with several pre-installed data libraries, such as Pandas, NumPy, Matplotlib, which is awesome. Google Colab, on the other hand, provides even more pre-installed machine learning libraries such as Keras, TensorFlow, and PyTorch.

Saved on the Cloud  
When you opt to use a plain Jupyter notebook as your development environment, everything is saved in your local machine. If you are cautious about privacy, this may be a preferred feature for you. However, if you want your notebooks to be accessible to you from any device with a simple Google log-in, then Google Colab is the way to go. All of your Google Colab notebooks are saved under your Google Drive account, just like your Google Docs and Google Sheets files.

Collaboration  
Another great feature that Google Colab offers is the collaboration feature. If you are working with multiple developers on a project, it is great to use Google Colab notebook. Just like collaborating on a Google Docs document, you can co-code with multiple developers using a Google Colab notebook. Besides, you can also share your completed work with other developers.

FAQ:  
<https://research.google.com/colaboratory/faq.html#:~:text=Colab%20allows%20anybody%20to%20write,learning%2C%20data%20analysis%20and%20education>.

Blog on Benefits of Colab and Ipython Notebooks and Jupyter:  
https://towardsdatascience.com/4-reasons-why-you-should-use-google-colab-for-your-next-project-b0c4aaad39ed