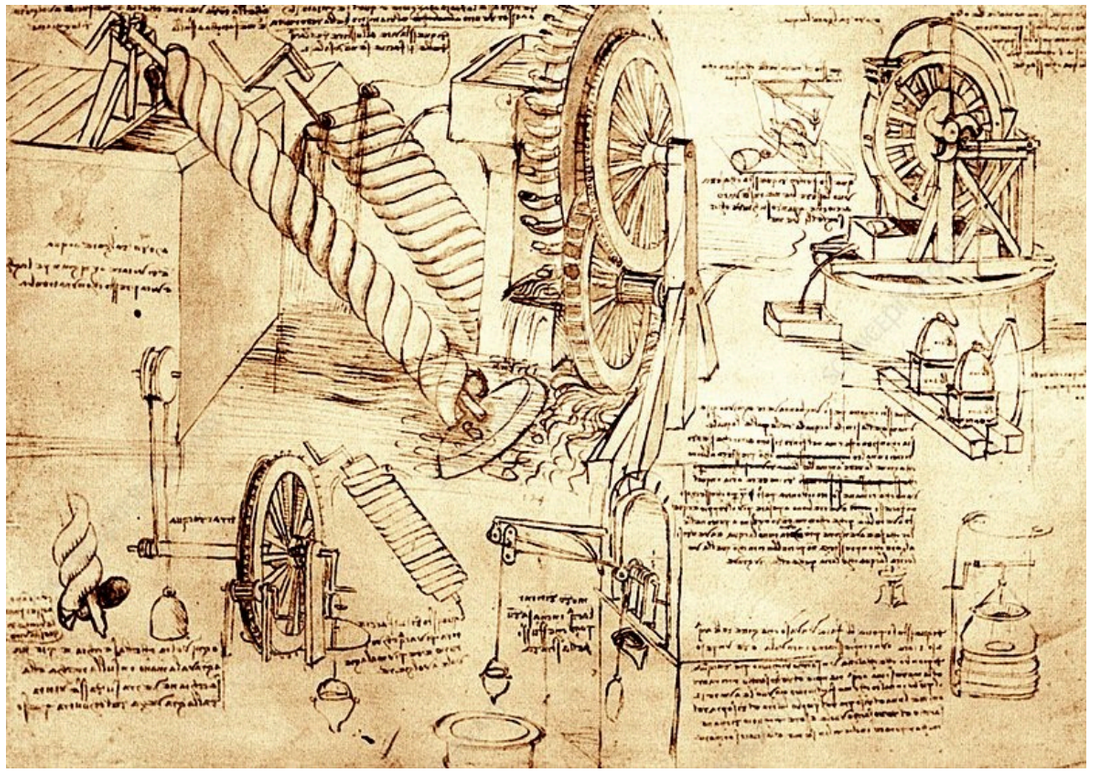
(Computational) Notebooks

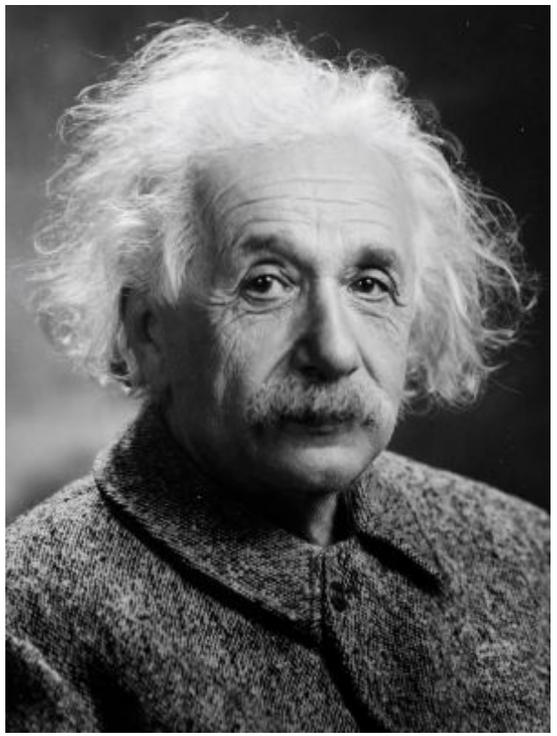
Salvo Nicotra

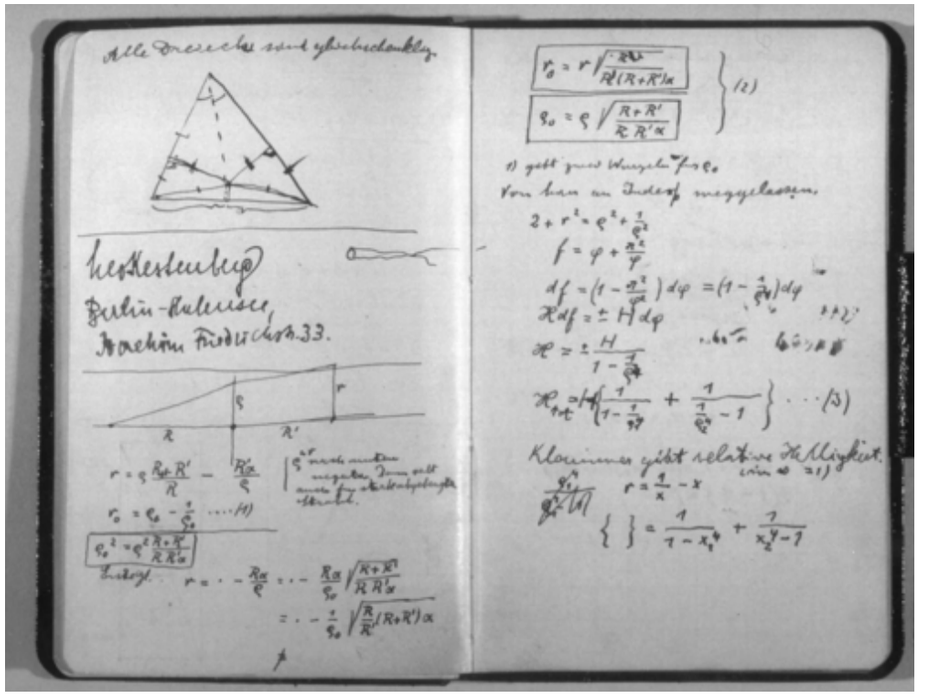
Are we going to make computations using notebooks ?

There are some great examples



Codex Atlanticus





Einstain’s scratch book - calculations on gravity

Use a real notebook!

To get value you need lot of time

White Rabbit

We need some magic tools

Notebook Interface

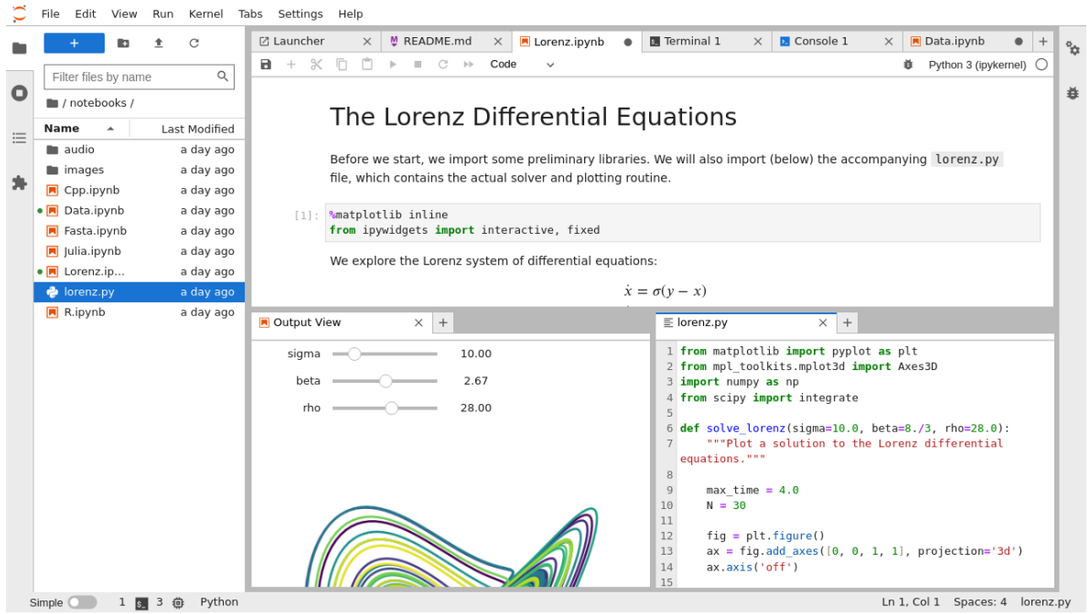
A Wikipedia definition

> A notebook interface (also called a computational notebook) is a virtual notebook environment It pairs the functionality of word processing software with both the shell and kernel of that notebook's programming language. Millions of people use notebooks interface to analyze data for science, journalism, and education. Wikipedia

A technical definition

> A notebook is a shareable document that combines computer code, plain language descriptions, data, rich visualizations like 3D models, charts, graphs and figures, and interactive controls. A notebook, along with an editor (like JupyterLab), provides a fast interactive environment for prototyping and explaining code, exploring and visualizing data, and sharing ideas with others.

https://docs.jupyter.org/en/latest/



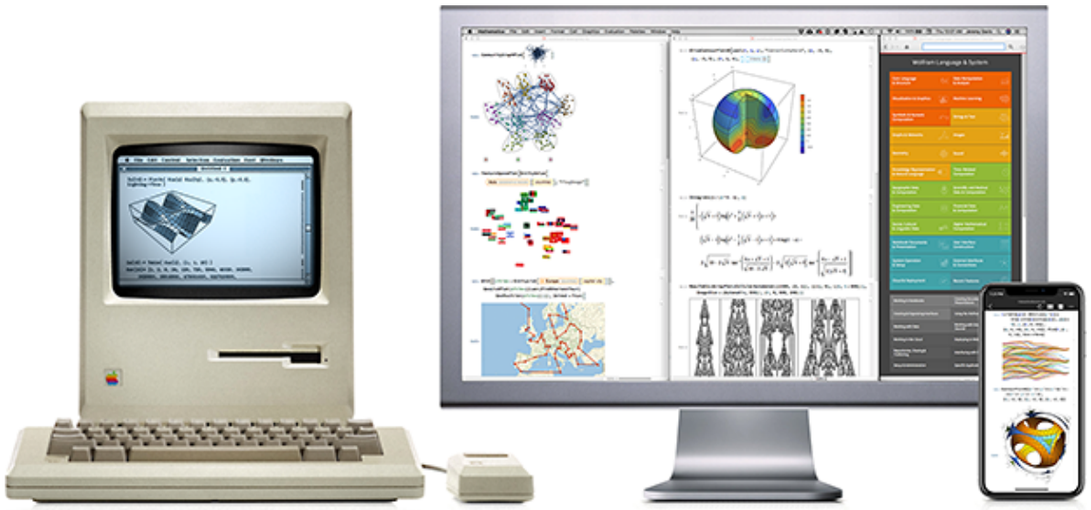
A computational notebook document, shown inside JupyterLab

Origins

In 1988 the notebook interface was used as Mathematica frontend

Source

> It’s fun today to launch Mathematica 1.0 on an old computer, and compare it with today:



Literate Programming

> Literate programming: Instead of imagining that our main task is to instruct a computer what to do, let us concentrate rather on explaining to human beings what we want a computer to do.

Donald Knuth (1984)

Literate computing

> A literate computing environment is one that allows users not only to execute commands interactively but also to store in a literate document format the results of these commands along with figures and free-form text that can include formatted mathematical expressions.

Millman KJ and Perez F (2014) Implementing Reproducible Research

Developing open source scientific practice

Chapter in Reproducible Research

Reproducibility in computational research requires:

1. sharing of scientific software, data, and knowledge necessary for reproducible research;
2. readable, tested, validated, and documented software as the basis for reliable scientific outcomes;
3. high standards of computational literacy in the education of mathematicians, scientists, and engineers;
4. open source software developed by collaborative, meritocratic communities of practice.

What is a notebook then ?

> A Notebook is an interactive, editable document defined by code. It's a computer program, but one that's designed to be easier to read and write by humans."

Mike Bostock (creator of D3.js, founder of https://observablehq.com/)

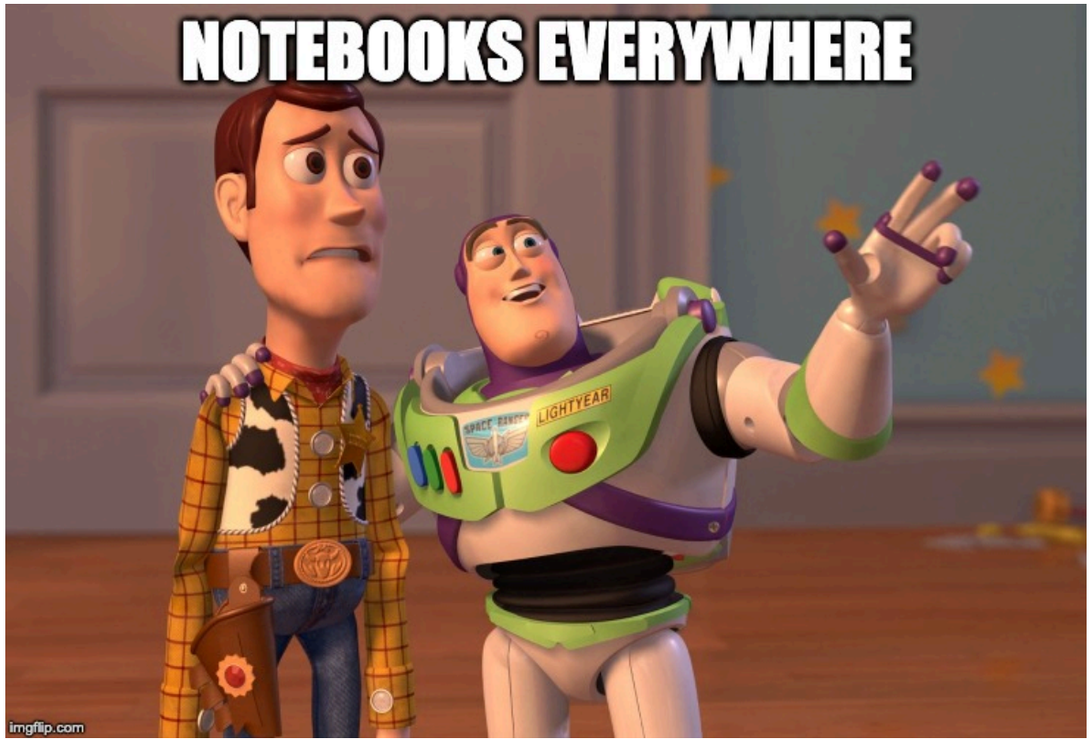
> When we look at the future and we try to think what does it look like to work with data, we think it looks an awful lot like a notebook experience.

Michelle Ufford (http://noteable.io/, formely Head of Data Netflix)

Edit: Notable.io shutdown

Use Cases

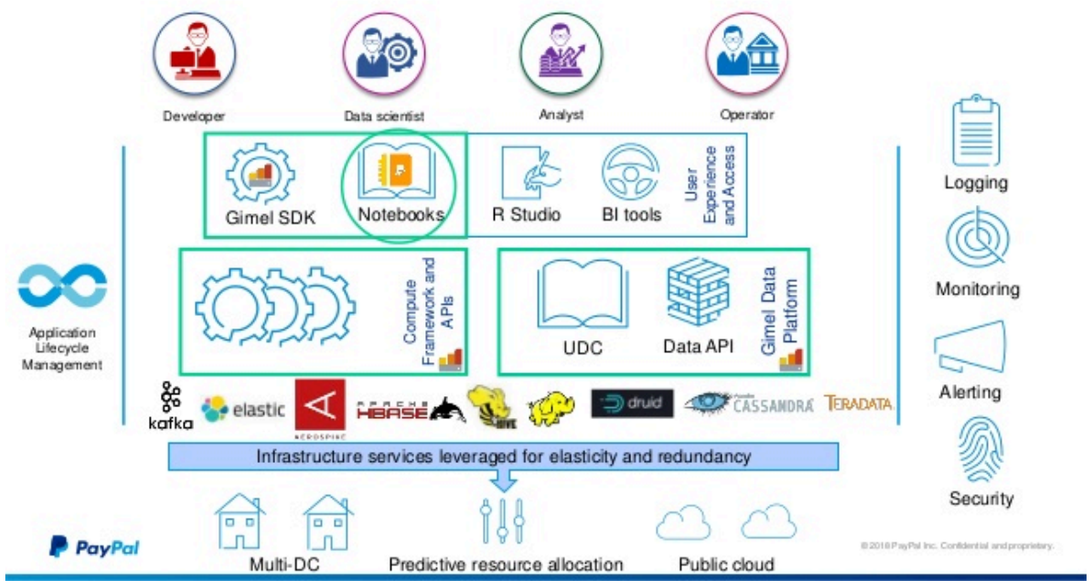
Netflix: Data Role



https://softwareengineeringdaily.com/2019/01/15/not
at-netflix-with-matthew-seal/

- Over the last 10 years, data engineering has become a key component of what makes Netflix successful. There are many different engineering roles who interact with the data infrastructure at Netflix. These include data analysts, machine learning scientists, analytics engineers and software engineers.
- As data engineering and data science has grown, the tooling has expanded. The people in different data roles at Netflix might use Apache Spark, or Presto, or Python, or Scala, or SQL, or many other applications to study data. But in recent years, there is one tool that has stood out for its ability to be distinctly useful - Jupyter Notebooks.

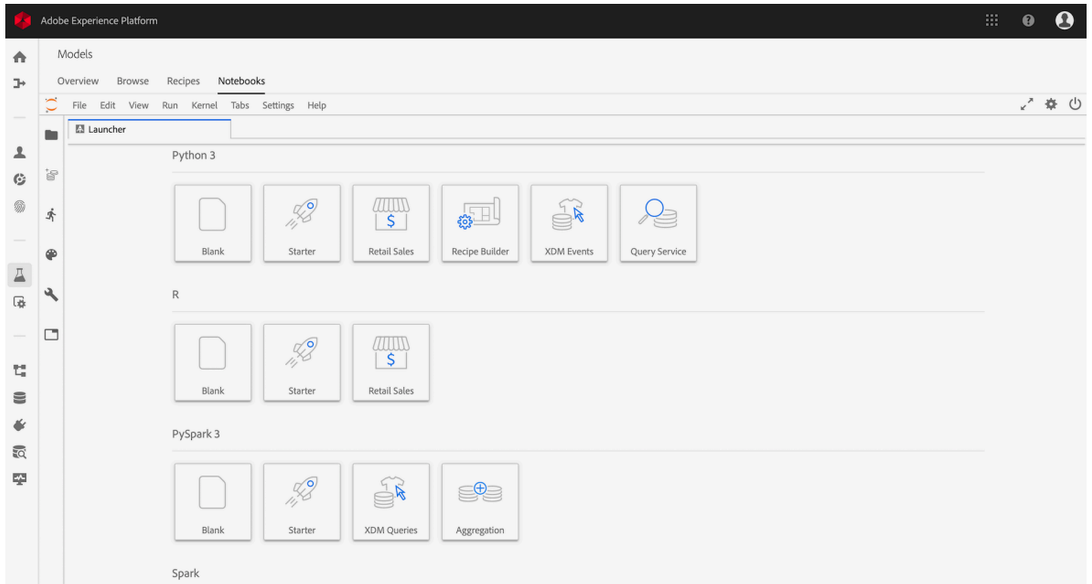
Pay Pal: need for a centralized (and explanatory) reference



- Unified Data Access: Simplifies querying across Hive, HBase, Kafka, etc. via Gimel Data API
- Interactive Analysis: Combines code, visuals, and documentation in one shareable tool.
- Enterprise-Ready: Secure, scalable, and audit-friendly for PayPal's internal us

https://medium.com/paypal-engineering/paypal-notebooks-powered-by-jupyter-fd0067bd00b0

Adobe: Data for enterprise



- Collaborative Workflows: Real-time sharing and teamwork on notebooks.
- Scalable Infra: Dynamically handles large-scale data and compute needs.
- Standardized Environment: Ensures consistency and reproducibility across teams.

https://medium.com/adobetech/reimagining-jupyter-notebooks-for-enterprise-scale-8bc6340d504a

Orelly: Bring Your Learning to Life

For a limited time only, you can preview one of our new premium content formats that will soon require an account upgrade to access.

Don’t just learn—do. Apply the skills you’re reading about in real time with interactive Jupyter Notebooks.

https://www.oreilly.com/online-learning/introducing-jupyter-notebooks.html

Interest over time

https://trends.google.com/trends/explore?date=2014-01-01%202025-03-27&q=jupyter,colab,Rstudio,kaggle,databricks&hl=en

List of Jupyter compatible Notebooks

- JupyterLab: Official next-gen interface for Jupyter; supports notebooks, terminals, code consoles, and more
- Jupyter Notebook (Classic): The original web-based interactive computing interface
- Anaconda: Local installation with Jupyter Notebook/Lab included via Navigator or CLI
- Binder: Free, temporary Jupyter environments from GitHub repos—ideal for demos and testing
- CoCalc: Offers standard Jupyter environments with collaboration features
- Kaggle Kernels: While lightly customized, still uses vanilla Jupyter backend and `.ipynb` compatibility
- Saturn Cloud: Supports launching standard Jupyter Notebooks in scalable cloud environments
- IBM Watson Studio (Lite): Offers standard JupyterLab environments in free tier
- JetBrains Datalore (Community Edition): Slightly modified UI but retains `.ipynb` compatibility and Jupyter-like workflows
- VS Code: Local IDE with native support for Jupyter Notebooks via the Python extension; runs vanilla `.ipynb` files in a Jupyter-compliant interface