A Brief History of Jupyter Notebooks

William Horton
Two different worlds of Python
1. print("hello, world!")
```python
In [1]: print('hello, world!')
hello, world!
```
What is a Jupyter Notebook?
Why?
My personal journey
I do think Mathematica doesn't get the credit it deserves for pioneering notebooks

news.ycombinator.com/item?id=222788...

Mathematica's notebook definitely strongly inspired Colab's notebook. Colab is an implementation of the Jupyter notebook format and UI. Jupyter, which launched around 2011, itself was strongly inspired by (1) the IPython console from around 2003, and (2) the Sage Notebook which I launched around 2006.

I can tell you definitively that Sage Notebook is very Mathematica inspired. The IPython console looked a lot like Mathematica, mainly because Fernando Perez (who was a physicist) had used Mathematica a lot and wanted something similar but (much) better. In 2005 there was a project to make an IPython notebook interface as an OS X graphical application, which got demoed at Sage Day 1 (in Feb 2006). That motivated me to get interested in doing something similar, but using Javascript and HTML instead. I hired Alex Clemeshu, who just finished his physics undergrad and was a heavy Mathematica user to work on Sage fulltime. He did a lot of work with me during 2006 to create a web-based notebook interface (and also to provide a mathematica-like graphics compatibility layer for Python, which is in Sage). The Sage notebook felt pretty similar in 2007 to what Jupyter notebook feels like, and it definitely inspired the UI. We developed Sage notebook heavily and then all sort of lost interest and moved on to other things (e.g., Jason Grout, who was involved a lot with the Sage notebook went to work at Bloomberg, where he did a massive amount of work on JupyterLab). Fortunately, Fernando Perez and others got incredible grant support and many fantastic engineers together built the Jupyter notebook. Jupyter notebook provided the same sort of cell/output UI as we had with the Sage notebook, but was much more general purpose (many kernels) and used more “modern” implementation techniques, by 2011 standards at least.

There's a lot of amazing things about the Mathematica notebook that we never even tried to implement. For example, Mathematica has a much more sophisticated nested structure. Also, by default Mathematica shares one kernel across multiple notebooks (or at least it did last time I tried it).
Jupyter Notebooks go mainstream

Why Jupyter is data scientists’ computational notebook of choice

An improved architecture and enthusiastic user base are driving uptake of the open-source web tool.

Jeffrey M. Perkel

The Scientific Paper is Obsolete

Here's what's next.

Story by James Somers
This talk:
I. Past
II. Present
III. Future
What this talk is not about
How to use Jupyter Notebooks
Introductory material for Jupyter notebooks

https://realpython.com/jupyter-notebook-introduction/
Loving or Hating Jupyter Notebooks
The First Notebook War

https://www.youtube.com/watch?v=QR7gR3niNWw
Past
(Apologies to all the former science Ph.D’s out there)
factor \( x^3 + 2x^2 + x + 2 \)

Result:

\((x + 2)(x^2 + 1)\)
Mathematica (1988)

Created by Stephen Wolfram

Notebook interface designed by Theodore Gray

An all-in-one system: language (Wolfram), execution environment, scientific libraries
Mathematica
Mathematica is another breakthrough Macintosh application. It does for students of calculus, symbolic algebra, and some discrete mathematics what calculators did for those learning arithmetic. Confirmed mathphobic students may very well be drawn into Mathematica’s impressive displays and its ability to solve equations quickly and display graphical results. It could enable you to absorb the algebra and calculus that seemed impossible to comprehend from a textbook.
EXCELLENCE

NeXT Computer

The NeXT Computer shows what can be done when a personal computer is designed as a system, and not a collection of hardware elements. It features the latest-generation high-speed components: Motorola’s 25-MHz 68030 CPU and 68882 floating-point unit, and a 4-megabyte-per-second SCSI (small computer system
HP DeskJet Printer

If you’re looking for a truly silent printer, this HP ink-jet will suit you. The DeskJet’s output is close to laser-printer quality at a much more tolerable price of $995. It comes
We couldn’t have said it better.

“Advanced MS-DOS PROGRAMMING exemplifies how a highly technical book can be both informative and readable.... Duncan's strengths include a style that is at once easily read, a thorough coverage of the subject matter heretofore unknown, and the frequent use of examples in the form of assembly language programs and code fragments.”

BYTE magazine
John Unger, IBM Issue 1986

“Makes good reading out of even the most elaborate technical descriptions.”

Online Today

“One of the most authoritative in its field.... The book deserves a place on the shelf of everyone who has ever given a fleeting thought to programming the IBM PC and compatibles.”

PC Magazine
Key architectural details of Mathematica Notebooks

Two parts to the system: kernel and front-end

The front-end sends the input to the kernel, which returns the result, which is displayed

Mathematica notebooks are objects that can be manipulated by Mathematica programs
The Mathematica Notebook Interface
\[ \text{In[1]} := \] 
\[ \text{Series[Exp[f[x + h] - f[x - h]], \{h, 0, 6\}] } \]

\[ \text{Out[1]} := \] 
\[ 1 + 2 f'[x] h + 2 f''[x] h^2 + \]
\[ \frac{4 f'[x]}{3} \frac{f''[x]}{3} h^3 + \]
\[ \frac{4}{3} f''[x] h^3 + \frac{2 f''[x] f'''[x]}{3} h^4 + \]
\[ \frac{4 f''[x]}{15} + \frac{2 f'''[x]}{3} \frac{f''[x]}{3} h^5 + \]
\[ \frac{4 f''[x]}{15} + \frac{2 f'''[x]}{9} f''[x] h^6 + \]
\[ \frac{4 f'''[x]}{9} h + 0[h] \]
In[1]:=  
Plot3D[BesselJ[0, Sqrt[x^2 + y^2]], {x, -10, 10}, {y, -10, 10}, PlotPoints->100]

Out[1]=

A three-dimensional plot of a Bessel function.
Going further back...
Literate Programming

Donald Knuth

 Implemented the “WEB” system

 Concepts of “tangled” and “woven”
Maple

Scientific computing environment + programming language

First GUI for Maple released in 1989

1992 -- release of the “worksheet” interface:

“Beginning with the Macintosh user interface for Maple V, the new user interfaces will all support the concept of a "worksheet" which integrates text, Maple input commands, Maple output, and graphics into one document.”
\begin{verbatim}
> 2 + 2
> 2 + 2;
> 2 + 2:
> ifactor(203490);
> \text{(2)} (3)^2 (5) (7) (17) (19)
> ?factor
> help(factor);
> ifactor(203490);
> ifactor(203490)
\end{verbatim}
with(plots);

Warning, the name changecoords has been redefined

[Interactive, animate, animate3d, animatecurve, arrow, changecoords, complexplot, complexplot3d, conformal, conformal3d, contourplot, contourplot3d, coordplot, coordplot3d, cylinderplot, densityplot, display, display3d, fieldplot, fieldplot3d, gradplot, gradplot3d, graphplot3d, implicitplot, implicitplot3d, inequal, interactive, interactiveparams, listcontourplot, listcontourplot3d, listdensityplot, listlistplot, listplot, listplot3d, loglogplot, logplot, matrixplot, multiple, odeplot, pareto, plotcompare, pointplot, pointplot3d, polarplot, polygonplot, polygonplot3d, polyhedra, supported, polyhedraphot, replplot, rootlocus, semilogplot, setoptions, setoptions3d, spacecurve, sparsematrixplot, sphereplot, surfdata, textplot, textplot3d, tubeplot]

> A := x^2 + y^2 = z^2;

> implicitplot3d(A, x = -2 .. 2, y = -2 .. 2, z = -2 .. 2);
Maple vs Mathematica
# Standard Math Notation

Entering mathematical expressions that look like mathematical expressions is very easy in Maple. The equation editor automatically formats fractions and exponents as you type. You can enter the expression the same way you would write it down, and it appears in Maple as it would when written in your textbook. This makes the mathematics easy to enter and easy to read. Mathematica, however, uses some non-standard notation which requires the user to translate back and forth between standard mathematics and Mathematica syntax.

Here are examples of expressions entered using the default settings in both systems.

<table>
<thead>
<tr>
<th>Maple</th>
<th>VS</th>
<th>Mathematica</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \sin(2x) )</td>
<td></td>
<td>( \text{Sin}[2 \times] )</td>
</tr>
<tr>
<td>( 5x - 7 = 3x + 2 )</td>
<td></td>
<td>( 5 \times - 7 == 3 \times + 2 )</td>
</tr>
<tr>
<td>( 2x^2 + \cos\left(\frac{x}{2}\right) )</td>
<td></td>
<td>( 2 \times^2 + \text{Cos}\left[\frac{x}{2}\right] )</td>
</tr>
<tr>
<td>( \lim_{x \to 0} \frac{\sin(x)}{x} )</td>
<td></td>
<td>( \text{Limit}[\text{Sin}[x] / x, x \to 0] )</td>
</tr>
</tbody>
</table>

See notes 1 and 2.

See notes 3.

See notes 4.

See notes 5.
Combining Text and Results

In Maple, it is very easy to combine text and mathematics in the same sentence. You can even have calculated results appear in the middle of a sentence, so that the sentence changes automatically if the results are updated.

\[
\text{The function } \frac{1}{(x-1)^2} \text{ has an essential discontinuity at } x = 1. 
\]

By changing the definition of the function and re-executing the document, the new discontinuity is found and the statement is updated appropriately:

\[
\text{The function } \frac{1}{(x-3)^2} \text{ has an essential discontinuity at } x = 3. 
\]

In Mathematica, it is not possible to combine text and mathematics results in this way. You can combine text and static math in the same cell, but you cannot display calculated results. If your results change, you must edit your statement by hand.
Enter vs. Shift Enter

In Maple, once you have entered your problem, you press the <Enter> key to tell Maple to perform the computation and give you the result. Typing “2+2 <Enter>” results in 4.

In Mathematica, typing “2+2 <Enter>” moves the cursor to the next line, without calculating anything. To ask Mathematica to perform the computation, you must press <Shift>+<Enter>. This non-standard interaction requires users to adapt their normal behavior.
Maple Personal Edition. Because it's fun.

Maple Personal Edition for people who use Maple to experiment, explore and play.

Buy Maple Personal Edition for $239*

<table>
<thead>
<tr>
<th></th>
<th>Home Cloud</th>
<th>Home Desktop</th>
<th>Home Desktop + Cloud</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Mathematica Online</strong></td>
<td>$172/year</td>
<td><strong>Mathematica Desktop</strong> MOST POPULAR</td>
<td>$344</td>
</tr>
<tr>
<td></td>
<td>$18/month</td>
<td></td>
<td>Buy Now</td>
</tr>
<tr>
<td><strong>Mathematica Desktop</strong> with 1 year of Personal License Service</td>
<td>$516</td>
<td></td>
<td>Buy Now</td>
</tr>
<tr>
<td></td>
<td>$258/year</td>
<td></td>
<td>Buy Now</td>
</tr>
<tr>
<td><strong>Mathematica Desktop</strong> with 1 year of Personal License Service Plus</td>
<td>$602</td>
<td></td>
<td>Buy Now</td>
</tr>
<tr>
<td></td>
<td>$301/year</td>
<td></td>
<td>Buy Now</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Rise of Open Source
SciPy + IPython + Matplotlib

SciPy (2001): Created by Travis Oliphant, Eric Jones, and Pearu Peterson, it was a combination of scientific libraries for Python on top of the Numeric array type (later replaced by NumPy)

IPython (2001): “The IPython project provides on enhanced interactive environment that includes, among other features, support for data visualization and facilities for distributed and parallel computation” (from “IPython: A System for Interactive Scientific Computing”, 2007)
SciPy + IPython + Matplotlib (cont.)

Matplotlib (2003): “It was conceived by John Hunter in 2002, originally as a patch to IPython to enable interactive MatLab-style plotting via gnuplot from the IPython command-line. Fernando Perez was, at the time, scrambling to finish his PhD, and let John know he wouldn’t have time to review the patch for several months. John took this as a cue to set out on his own, and the matplotlib package was born, with version 0.1 released in 2003.”

Fernando Perez

Created IPython in 2001 as a graduate student

Currently: Associate Professor, Statistics, UC Berkeley and Berkeley Institute for Data Science Senior Fellow

Recipient of the 2012 Award for the Advancement of Free Software from the Free Software Foundation, and the 2017 ACM Software System Award
SageMath is a free open-source mathematics software system licensed under the GPL. It builds on top of many existing open-source packages: NumPy, SciPy, matplotlib, SymPy, Maxima, GAP, FLINT, R and many more. Access their combined power through a common, Python-based language or directly via interfaces or wrappers.

Mission: Creating a viable free open source alternative to Magma, Maple, Mathematica and Matlab.
Sage cont.

Created by William Stein

Implemented in Python and Cython

Open Source license (GPLv3)
Sage Notebook/Worksheets

2 + 2

\[ 4 \]

Why is $2 + 3 = 5$ and not $13$? I thought Sage did everything modulo 4...
You can also evaluate a cell using a keyboard shortcut.

- If a cell isn’t active (such as below, in the live documentation), click in it.
- Then hold down the Shift key while you press the Enter key.

We call this “Shift-Enter”. Try doing Shift-Enter with this cell.
Starting the notebook server

You can start running a notebook server from the command line using the following command:

```bash
ipython notebook
```
The IPython Notebook

Introduction

The notebook extends the console-based approach to interactive computing in a qualitatively new direction, providing a web-based application suitable for capturing the whole computation process: developing, documenting, and executing code, as well as communicating the results. The IPython notebook combines two components:

A web application: a browser-based tool for interactive authoring of documents which combine explanatory text, mathematics, computations and their rich media output.

Notebook documents: a representation of all content visible in the web application, including inputs and outputs of the computations, explanatory text, mathematics, images, and rich media representations of objects.
Project Jupyter (2014)
Jupyter

Ju(lia) + Pyt(hon) + R
Project Jupyter (2014)

Spun off of IPython by Fernando Perez

Includes the Notebook interface and other language-agnostic parts of IPython
Present
In [1]: print("hello, world!")

hello, world!

In [ ]:
JupyterLab 1.0: Jupyter's Next-Generation Notebook Interface

JupyterLab is a web-based interactive development environment for Jupyter notebooks, code, and data. JupyterLab is flexible: configure and arrange the user interface to support a wide range of workflows in data science, scientific computing, and machine learning. JupyterLab is extensible and modular: write plugins that add new components and integrate with existing ones.
The Lorenz Differential Equations

Before we start, we import some preliminary packages. We will also import (below) the accompanying `lorenz.py` file, which contains the actual solver and plotting routines.

```python
import matplotlib.pyplot as plt
import numpy as np
import scipy as sp

from scipy import integrate
```

We explore the Lorenz system of differential equations:

\[
\begin{align*}
\frac{dx}{dt} &= \sigma(y - x) \\
\frac{dy}{dt} &= x - y - xz \\
\frac{dz}{dt} &= xy - \beta z
\end{align*}
\]

Let's change \(\sigma, \beta, \gamma\) with ipywidgets and examine the trajectories.

```python
from lorenz import solve_lorenz
w = integrate.odeint(solve_lorenz, (1.0, 0.0, 0.0), t)
```

For the default set of parameters, we see the trajectories existing around two points, called attractors.

The object returned by `solve_lorenz` is a `numpy` array and it has attributes that contain the current result and arguments.

```python
t, x, y, z = w
```

After interacting with the system, we can take the result and perform further computations. In this case, we compute the average positions in \(x, y\) and \(z\).

```python
x_avg = x.T.mean(axis=1)
y_avg = y.T.mean(axis=1)
z_avg = z
```

Creating histograms of the average positions (across different trajectories) show that, on average, the trajectories seem about the attractors.
Communicate your results with Voilà.

Voilà helps you communicate insights, by transforming a Jupyter Notebook into a stand-alone web application you can share. It gives you control over what your readers experience in a secure and customizable interactive dashboard.

So easy, voilà!

In this example notebook, we demonstrate how voila can render notebooks making use of ipywidget's `@interact`.

59 * 21 = 1239

59

21
Binder

Turn a Git repo into a collection of interactive notebooks

Have a repository full of Jupyter notebooks? With Binder, open those notebooks in an executable environment, making your code immediately reproducible by anyone, anywhere.

Build and launch a repository

GitHub repository name or URL

Git branch, tag, or commit

Path to a notebook file (optional)

File
launch

Copy the URL below and share your Binder with others:

Fill in the fields to see a URL for sharing your Binder.

Copy the text below, then paste into your README to show a badge:
Broad language support
**Jupyter kernels**

Kernel Zero is **IPython**, which you can get through **ipykernel**, and is still a dependency of **jupyter**. The IPython kernel can be thought of as a reference implementation, as CPython is for Python.

Here is a list of available kernels. If you are writing your own kernel, feel free to add it to the table!

<table>
<thead>
<tr>
<th>SageMath</th>
<th>Jupyter 4</th>
<th>Any</th>
<th>many</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wolfram Engine, i.e., a Wolfram Desktop or Mathematica installation; <strong>wolframscript</strong> is optional but recommended</td>
<td>Wolfram Language for Jupyter</td>
<td>Wolfram Engine, i.e., a Wolfram Desktop or Mathematica installation; <strong>wolframscript</strong> is optional but recommended</td>
<td>Wolfram Engine, i.e., a Wolfram Desktop or Mathematica installation; <strong>wolframscript</strong> is optional but recommended</td>
</tr>
</tbody>
</table>
What is Colaboratory?

Colaboratory, or "Colab" for short, allows you to write and execute Python in your browser, with

- Zero configuration required
- Free access to GPUs
- Easy sharing

Whether you're a student, a data scientist or an AI researcher, Colab can make your work easier. Watch Introduction to Colab to learn more, or just get started below!

Getting started

The document you are reading is not a static web page, but an interactive environment called a Colab notebook that lets you write and execute code.

For example, here is a code cell with a short Python script that computes a value, stores it in a variable, and prints the result:

```python
[ ] seconds_in_a_day = 24 * 60 * 60

86400
```

To execute the code in the above cell, select it with a click and then either press the play button to the left of the code, or use the keyboard shortcut "Command/Control+Enter". To edit the code, just click the cell and start editing.

Variables that you define in one cell can later be used in other cells:

```python
[ ] seconds_in_a_week = 7 * seconds_in_a_day

604800
```

Colab notebooks allow you to combine executable code and rich text in a single document, along with images, HTML, LaTeX and more. When you create your own Colab notebooks, they are stored in your Google Drive account. You can easily share your Colab notebooks with co-workers or friends, allowing them to comment on your notebooks or even edit them. To learn more, see Overview of Colab. To create a new Colab notebook you can use the File menu above, or use the following link: create a new Colab notebook.
Layers
Custom fastai layers and basic functions to grab them.

Basic manipulations and resize

```python
def module(*args, **kwargs):
    """Decorator to create an `nn.Module` using `f` as `forward` method"
    pa = [inspect.Parameter(o, inspect.Parameter.POSITIONAL_OR_KEYWORD) for o in args]
    pb = [inspect.Parameter(k, inspect.Parameter.POSITIONAL_OR_KEYWORD, default=v)
        ```
```json
{
  "cells": [
    {
      "cell_type": "code",
      "execution_count": null,
      "metadata": {},
      "outputs": [],
      "source": [
        "\# default_exp layers\n",
        "\# default_cls \_cat 3"
      ]
    },
    {
      "cell_type": "code",
      "execution_count": null,
      "metadata": {},
      "outputs": [],
      "source": [
        "# import\n",
        "from fastai2.imports import *\n",
        "from fastai2.torch_imports import *\n",
        "from fastai2.torch_core import *\n",
        "from torch.nn.utils import weight_norm, spectral_norm"
      ]
    }
  ]
}
```
Future(?)
More IDE-like Jupyter and more Jupyter in IDEs

JupyterLab 2.0 -- released April 2020

Added JupyterLab Language Server Protocol + Debugger

VSCode shipping improved native support for Jupyter Notebooks
Hosted solutions/notebook as an interface to compute
Real-time collaboration

The notebook you’ll love to use

Deepnote is a new kind of data science notebook. Jupyter-compatible with real-time collaboration and easy deployment. Oh, and it’s free.

Live collaborative editing

Multiple users can collaborate on a project. As soon as a collaborator is added to a project [see Notes on Managing Courses] they share both the project and the associated files.

Live collaborative editing is possible in CoCalc. If one of your collaborators updates a notebook, the rest can see the changes as they are being made (similar to Google Docs).

@Mention collaborators in chat

CoCalc chats support an @mentions feature, where you type @ and a list appears of collaborators, which you can select from. Anybody mentioned there will get alerted (unless they are mentioned again in the next few hours, since we don’t want to spam people). This helps ensure people know about chats. Any chat will cause the notification count to go up in the bell in the upper right, whether or not you are mentioned.
How to review & work with git?
So excited about @github Codespaces. You can edit Jupyter Notebooks directly on GitHub, and serve arbitrary web applications on various ports.

Sign up for the beta if you have not already
github.com/features/codes...

Here is a demo with fastpages:

JupyterOnGitHub fastpages
Demo of GitHub Codespaces w/fastpages. Shows you how you can edit a notebook with VSCode, and see updates on...
youtube.com
† Automatically drop links to a Jupyter notebook with **the right dependencies** in your PRs with this simple Actions workflow ↓. Powered by @mybinderteam. It's also free 💰 to use.

Instructions: gist.github.com/hamelsmu/f8f98...
Conclusion

@hortonhearsafoo