

### The Jupyter talk talk

Jim Pivarski

Princeton University - IRIS-HEP

June 2, 2021

## Three fundamental modes of computation



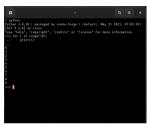








Notebook

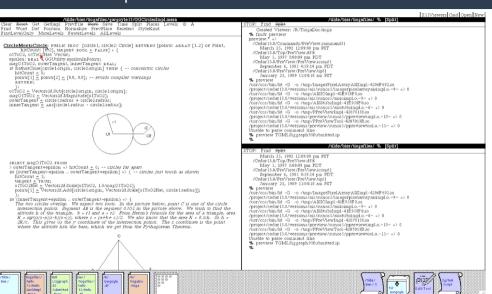




-- script-file.py All (6.0)

**Ephemeral** 



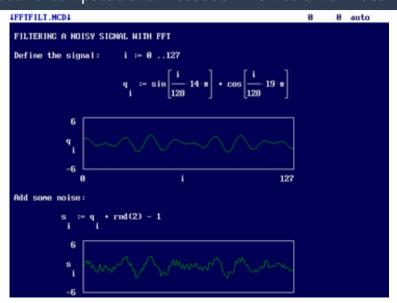


1982

Xerox's
Tioga editor
for the Cedar
language

(Xerox-PARC invented everything GUI-related)

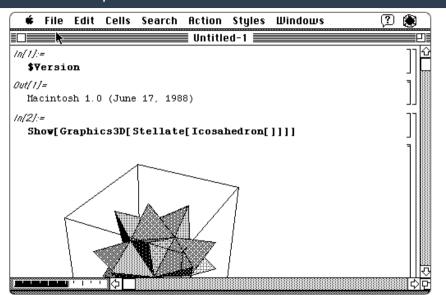




## 1986

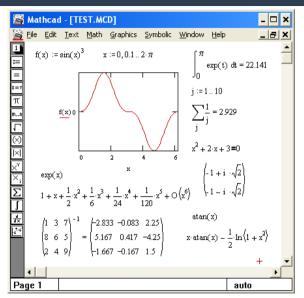
Mathcad, first computational notebook to run on PCs





1987
Mathematica

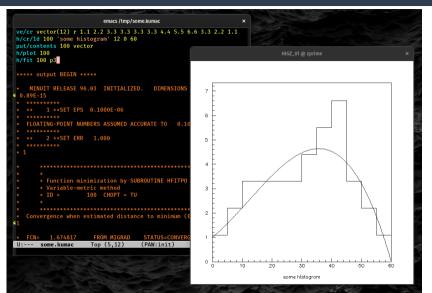




1992

Mathcad again (on Windows)





#### 2000

PAW-mode for Emacs (number of users = 1)

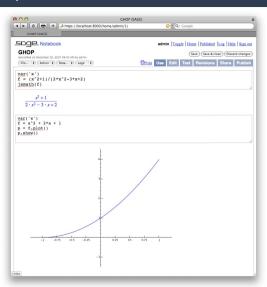


```
Shell - Konsole <2>
[~/test]> ipython -profile tutorial
Python 2.2 (#1, Feb 24 2002, 16:21:58)
Type "copyright", "credits" or "license" for more information
IPython 0.2.10pre18 -- An enhanced Interactive Python.
       -> Introduction to IPvthon's features.
object -> Details about 'object'; object? also works, ?? prints more.
helm -> Python's own helm system
@magic -> Information about TPython's 'magic' @ functions.
IPython profile: tutorial
*** Pasting of code with ">>>" or "..." has been enabled.
 n [1]: # This illustrates IPython's extensible syntax processing and profiles.
 n [2]: # In this mode, you can paste directly code examples:
 n [3]: >>> # Fibonacci series:
 n [4]: ... # the sum of two elements defines the next
 n [5]: ... a, b = 0, 1
 161: >>> while h < 10:
                 print b
                 a \cdot b = b \cdot a + b
 n [7]: # You can create a log of all input for later reuse as code or in IPython
 n [8]: logstart mylogfile.txt
Activating auto-logging.
Current session state plus future input saved to: mylogfile.txt
Logging mode: backup
 n [9]: # And you have direct access to the system shell
 n [10]: | le
class.py error.py qplot.py mylogfile.txt
```

2002

iPython: cell-based Python in the terminal

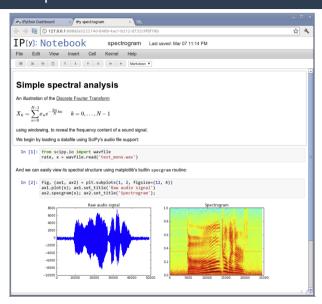




2005

SAGE: cell-based Python in a web browser



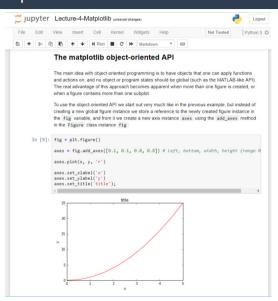


2012

iPythonnotebook: cell-based Python in a web browser

But popular this time!





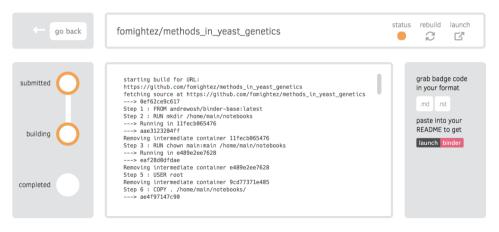
2015

Jupyter: rebranding to emphasize multilanguage support

 $\begin{array}{l} \textbf{Julia} \ + \\ \textbf{Python} \ + \\ \textbf{R} \end{array}$ 





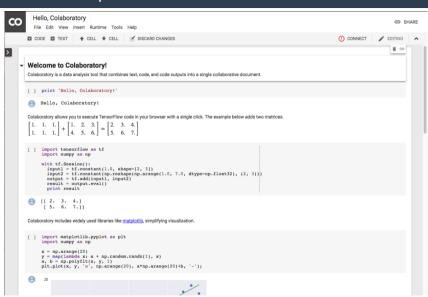


2016

#### Binder:

Jupyter notebooks on the cloud

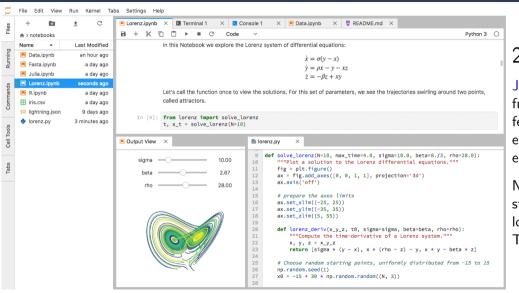




2018

Google
Colab:
Jupyterish
notebooks
on the cloud



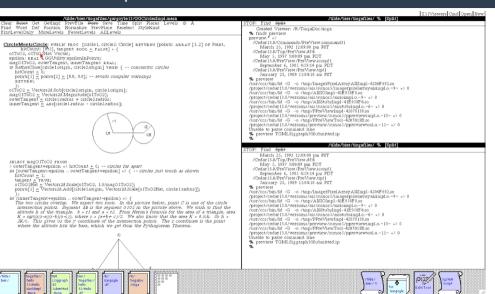


2018

JupyterLab: fully featured editor environment

Now it's starting to look like Tioga...

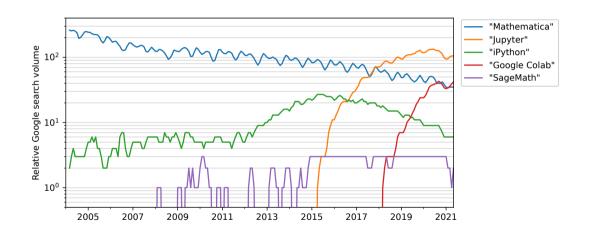




a hit

#### Jupyter is now mainstream





## Reasons for using notebooks



- Quick prototyping
- ▶ Iterative investigation, such as data analysis
- ► Literate programming: interweaving a program and text
- Publishing a widget that other people can run
- Giving a talk/teaching a class that others join in real time

# Reasons for using notebooks



- ► Quick prototyping
- ▶ Iterative investigation, such as data analysis
- ▶ Literate programming: interweaving a program and text
- Publishing a widget that other people can run
- ► Giving a talk/teaching a class that others join in real time



## The problems



1. "It's too small!"

2. "You're scrolling too fast!"

3. "Too much code in the cells!"

4. "It doesn't work for me!"