Jupyter Notebook Intro

git clone https://github.com/Caltech-IPAC/grits2018-jupyter-tutorial/

git clone https://github.com/NASA-NAVO/aas_workshop_2018_summer

(much material drawn from SciPy2018 tutorial by J. Grout, M. Busonnier)

Evolution of the notebook

• 2001-2005: IPython interpreter

2011: IPython Notebook

• 2014: Jupyter Notebook

2018: Jupyter Lab

Interactive, Exploratory, Reproducible

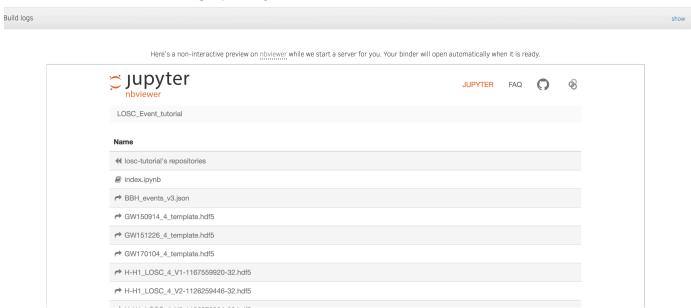
- Interactive, browser-based computing environment
- Exploratory data science, machine learning, visualization, analysis, statistics
- Reproducible document format:
 - Code
 - Narrative text (Markdown)
 - Equations (LaTeX)
 - Images, visualizations
- Over 100 programming languages
- Everything open-source (BSD license)

Live Code on Binder





Loading repository: losc-tutorial/LOSC_Event_tutorial/master



https://losc.ligo.org/tutorials

https://mybinder.org/v2/gh/losc-tutorial/LOSC Event tutorial/master

Jupyterlab Status

- https://github.com/jupyterlab
- ~4 years of development
- ~150 contributors
- ~60 components
- ~2600 releases (npm + python)
- ~13000 commits (more than classic NB)
- Currently Beta ("User 1.0", ready to use)

Jupyterlab Roadmap

- Beta releases from February 2018
 - For all users
 - For adventurous extension developers
- 1.0 this year
 - For all users and extension developers
- Eventually:
 - Classic notebook will be retired

Sharing notebooks

Notebook files are in JSON format

- Github is best for sharing
 - Renders notebooks in web interface
 - Store in standalone gist, or as part of repository
 - Dilemma: Clear all output? Save the executed version?
 - Can add "open in Binder" or Google Colab

"I Don't Like Notebooks"

- JupyterCon talk by Joel Grus https://t.co/30peBFwTbv
- Hidden state, out-of-order execution confuse beginners
- Developing code in notebooks discourages putting code in modules and writing tests
- Jupyter's text editing and autocomplete lag far behind professional editors and IDEs
- Too easy to make "shift-Enter" tutorials in which users don't learn much

Suggested best practices

Put code for reuse in modules

Develop code with editor or IDE

Commit notebooks with output cleared

Think "scratchpad" instead of "notebook"?