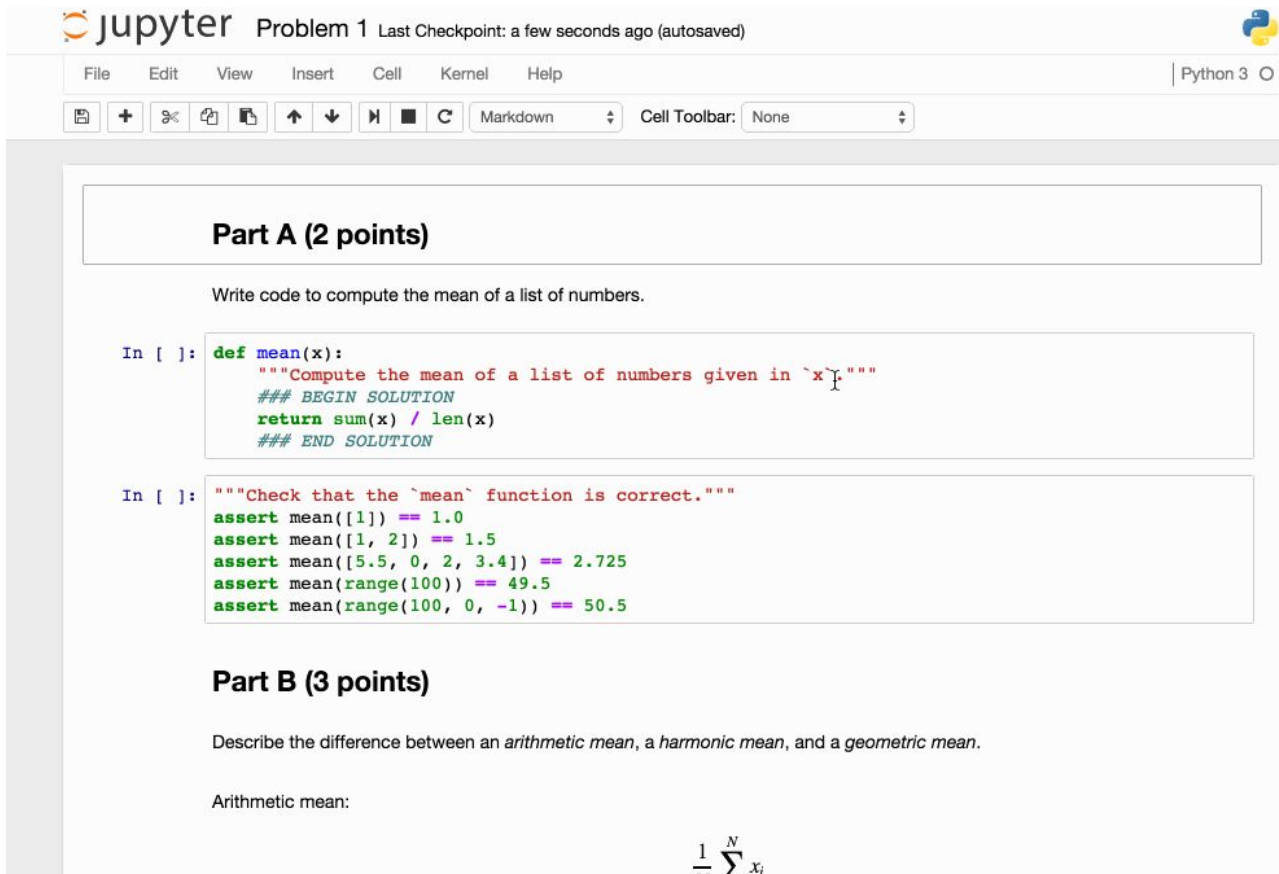


Other Moons

And final thoughts

nbgrader

A system for assigning and grading Jupyter notebooks.



jupyter Problem 1 Last Checkpoint: a few seconds ago (autosaved) Python 3

File Edit View Insert Cell Kernel Help

Cell Toolbar: None

Part A (2 points)

Write code to compute the mean of a list of numbers.

```
In [ ]: def mean(x):  
        """Compute the mean of a list of numbers given in `x`"""  
        ### BEGIN SOLUTION  
        return sum(x) / len(x)  
        ### END SOLUTION
```

```
In [ ]: """Check that the `mean` function is correct."""  
assert mean([1]) == 1.0  
assert mean([1, 2]) == 1.5  
assert mean([5.5, 0, 2, 3.4]) == 2.725  
assert mean(range(100)) == 49.5  
assert mean(range(100, 0, -1)) == 50.5
```

Part B (3 points)

Describe the difference between an *arithmetic mean*, a *harmonic mean*, and a *geometric mean*.

Arithmetic mean:

$$\frac{1}{N} \sum_{i=1}^N x_i$$

nbdime

nbdime provides tools for diffing and merging of Jupyter Notebooks.

- nbdiff compare notebooks in a terminal-friendly way
- nbmerge three-way merge of notebooks with automatic conflict resolution
- nbdiff-web shows you a rich rendered diff of notebooks
- nbmerge-web gives you a web-based three-way merge tool for notebooks
- nbshow present a single notebook in a terminal-friendly way

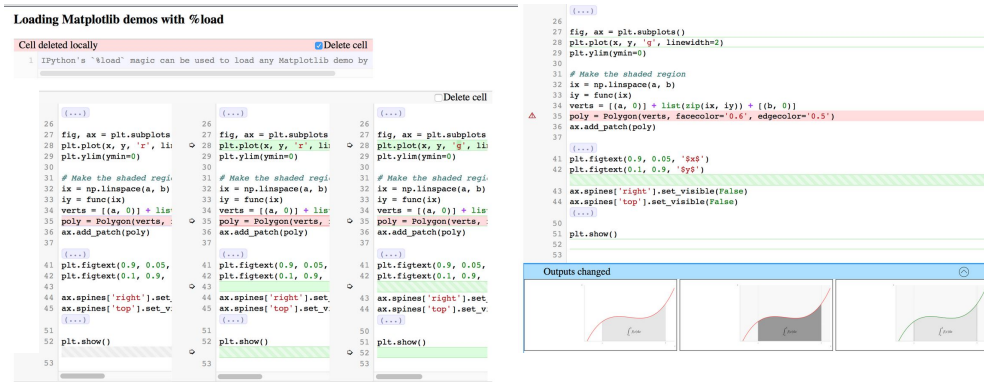
```
!!! D:\ipython 2018-11-30 13:12:30
```

```
## modified /cells/9/outputs/0/data/text/plain:
```

```
- <matplotlib.figure.Figure at 0x10ea05940>  
+ <matplotlib.figure.Figure at 0x10eb21860>
```

```
## replaced /cells/14/outputs/0/data/image/png:
```

```
- iVBORw0K...<snip base64, md5=3f7d4e61ee33aaae...>  
+ iVBORw0K...<snip base64, md5=1d6960ad89e9de61...>
```



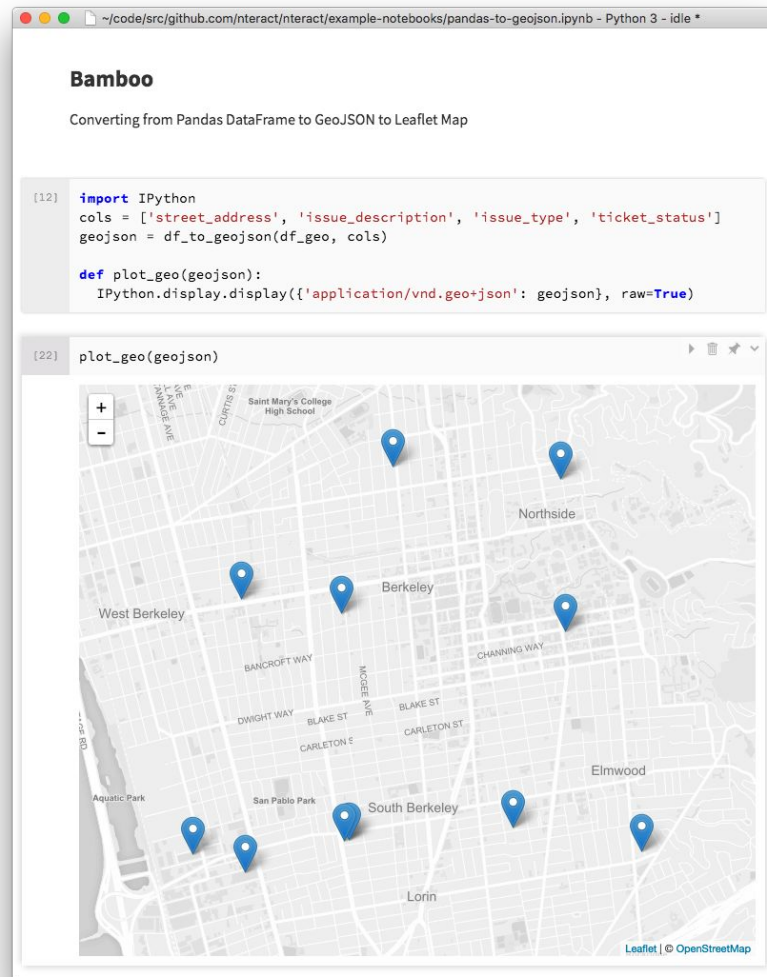
Thebe

Standalone Jupyter cells for static sites (Javascript Plugin)

- an easy way to let users on a web page run code examples on a server.

nteract

Native desktop app for Jupyter notebooks



Jupyterhub Spawners and Authenticators

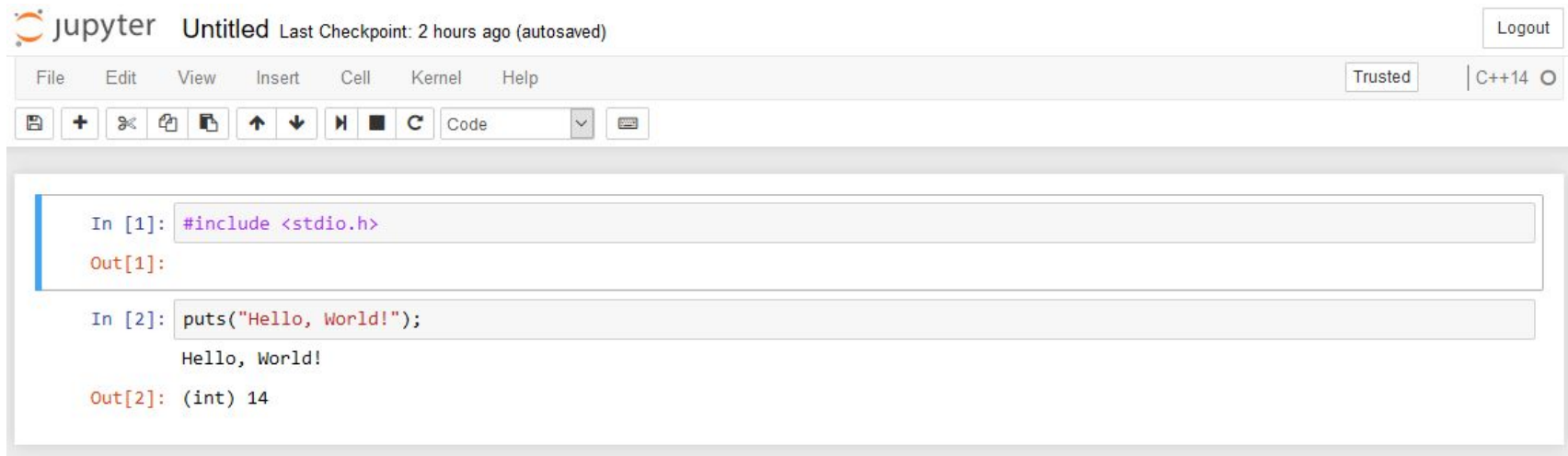
- Pluggable architecture allows for community tools
- Support for various forms of OAuth
- Dockerspawner
- Wrapspawner/Batchspawner for spinning up notebooks through batch queue

Jupyter Widget Server

Kernels ... lots of them

100+ language kernels

... even an Interactive C++ kernel (Cling Kernel) ???!!!!!!



The screenshot shows a Jupyter Notebook interface. At the top, the Jupyter logo is followed by the text "jupyter" and "Untitled". To the right of "Untitled" is the text "Last Checkpoint: 2 hours ago (autosaved)". In the top right corner, there is a "Logout" button. Below the title bar is a menu bar with the following items: File, Edit, View, Insert, Cell, Kernel, and Help. To the right of the menu bar is a "Trusted" button and a language selector showing "C++14" with a dropdown arrow. Below the menu bar is a toolbar with various icons for file operations, cell navigation, and execution. The main area of the notebook contains two code cells. The first cell has the input "In [1]: #include <stdio.h>" and the output "Out[1]:". The second cell has the input "In [2]: puts(\"Hello, World!\");" and the output "Hello, World!". Below the second cell is the output "Out[2]: (int) 14".

```
In [1]: #include <stdio.h>

Out[1]:

In [2]: puts("Hello, World!");
        Hello, World!

Out[2]: (int) 14
```


And much much more ...

This is a thriving ecosystem and there are new projects added every day!

How We Use Jupyter

Jupyter has changed how we program in our day-to-day lives

- Exploratory coding that requires you to weave in and out of code and output
- Recipes where you tweak options and parameters
- Workflows and Pipelines with distinct steps

Thanks

Contacts:

<https://github.com/Jupyter-Kale/Jupyter-and-its-moons>

Shreyas Cholia <scholia@lbl.gov>

Matt Henderson <mhenderson@lbl.gov>

Oliver Evans <oevans@lbl.gov>

Project Jupyter: <http://jupyter.org/community.html>