

How Does Jupyter Display a Plot?

Making Passagemath Plots Work
in Any Notebook

Samuel Chen and Matthias Köppe

UC Davis
JupyterDay 2026

passagemath is a modular version of SageMath

SageMath

Open source software for mathematics

passagemath

A modular version of SageMath
with separately installable Python packages

I expected a plot. I got text.

```
from sage.plot.plot import plot
import math
```

```
p = plot(math.sin, 0, 6)
p
```

Graphics object consisting of
1 graphics primitive

Before: passagemath-plot 10.8.1, plain Python kernel

The same table can appear as text or HTML

```
import pandas as pd
df = pd.DataFrame({"x": [1, 2], "y": [3, 4]})
```

```
print(df)
```

	x	y
0	1	3
1	2	4

pandas provides `_repr_html_()`.

```
display(df)
```

	x	y
0	1	3
1	2	4

An object can provide its own HTML

```
class Blob:
    def _repr_html_(self):
        return "<h1>I am a blob</h1>"
```

```
Blob()
```

HTML output

I am a blob

DisplayFormatter collects the representations

```
fmt = get_ipython().display_formatter  
data, metadata = fmt.format(Blob())
```

```
{  
  "text/plain": "<Blob ...>",  
  "text/html": "<h1>I am a blob</h1>"  
}
```

A MIME bundle (plain-text value shortened).

The method runs during formatting

```
display(Blob())
```

Python kernel

Frontend



For a plot, step 2 creates image data.

Plain IPython collected text, but no image from our plot

```
data, metadata = fmt.format(p)
sorted(data)
```

```
['text/plain']
```

Sage's display integration

```
__rich_repr__()
```

IPython's standard SVG hook

```
__repr_svg__()
```

The new method returns SVG to IPython

Simplified success path

```
def _repr_svg_(self):  
    return self._render_svg().decode("utf-8")
```

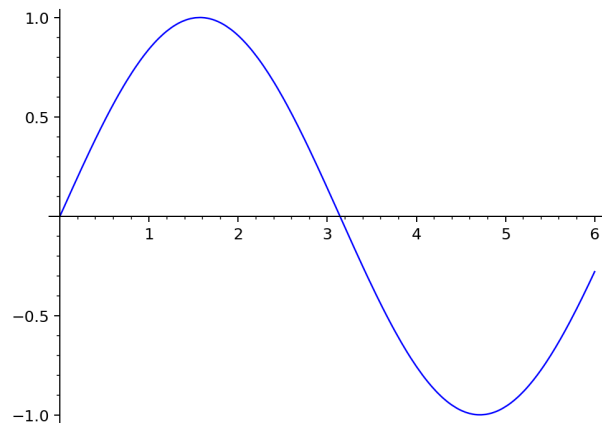
`_render_svg()` reuses Sage's drawing code
to create SVG image data.

With the SVG hook, IPython receives image data

```
plot(math.sin, 0, 6)
```

Formats collected

```
image/svg+xml  
text/plain
```



After: same baseline with the source SVG methods attached

We connected Sage's drawing code to IPython's display interface

Existing: Sage's drawing code

Added: standard display methods

Result: an image in the notebook

github.com/passagemath/passagemath

IPython rich display documentation

Questions and feedback

