

IDE

Integrated Development Environment

- RStudio
- Visual Studio Code
- Pycharm
- JupyterLab
- ...

What are components of an IDE?

- source code editor
 - syntax highlighting, auto-completion, bug checking
- build automation
 - compilation, testing, ...
- debugger
 - breakpoints etc.
 - (global environment, e.g. RStudio)
- version control, e.g. git

Why use an IDE?

2 layers:

- **personal (scientific) use**
 - simplification (all-in-one) → useful for beginners
 - efficiency (auto-completion etc.)
 - error prevention
- **teams / corporate**
 - standardization
 - quick onboarding with set configuration

Notes on IDEs

- IDEs are only helper tools that bundle components
 - i.e. Python \neq PyCharm, R \neq RStudio
 - also, components might be hard to distinguish: project \neq script \neq console (RStudio)

Other thoughts?

Jupyter

What is Jupyter? You mean Notebooks?

Tutorial: <https://coderefinery.github.io/jupyter/>

What are notebooks used for?

- linear workflows in data science / research
- pitfalls:
<https://scicomp.aalto.fi/scicomp/jupyter-pitfalls/>

An example: visualizing data in the notebook ✨

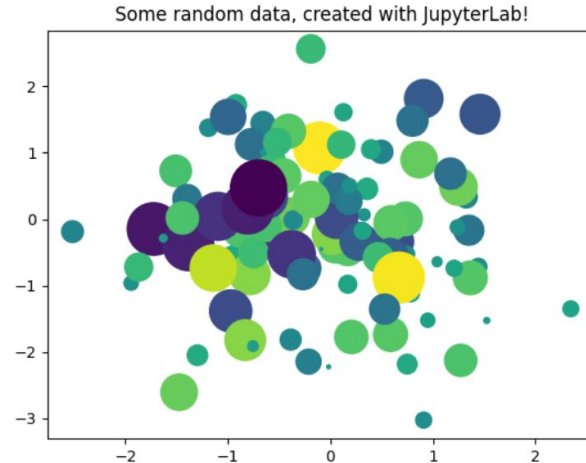
Below is an example of a code cell. We'll visualize some simple data using two popular packages in Python, visualize it.

Note how the code and the results of running the code are bundled together.

```
[1]: from matplotlib import pyplot as plt
import numpy as np

# Generate 100 random data points along 3 dimensions
x, y, scale = np.random.randn(3, 100)
fig, ax = plt.subplots()

# Map each onto a scatterplot we'll create with Matplotlib
ax.scatter(x=x, y=y, c=scale, s=np.abs(scale)*500)
ax.set(title="Some random data, created with JupyterLab!")
plt.show()
```



C++



How to use Jupyter with C++

Julia



How to use Jupyter with
Julia

GNU Octave



How to use Jupyter with
GNU Octave

R



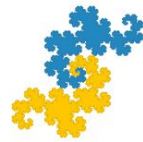
How to use Jupyter with R

Ruby



How to use Jupyter with
Ruby

Scheme



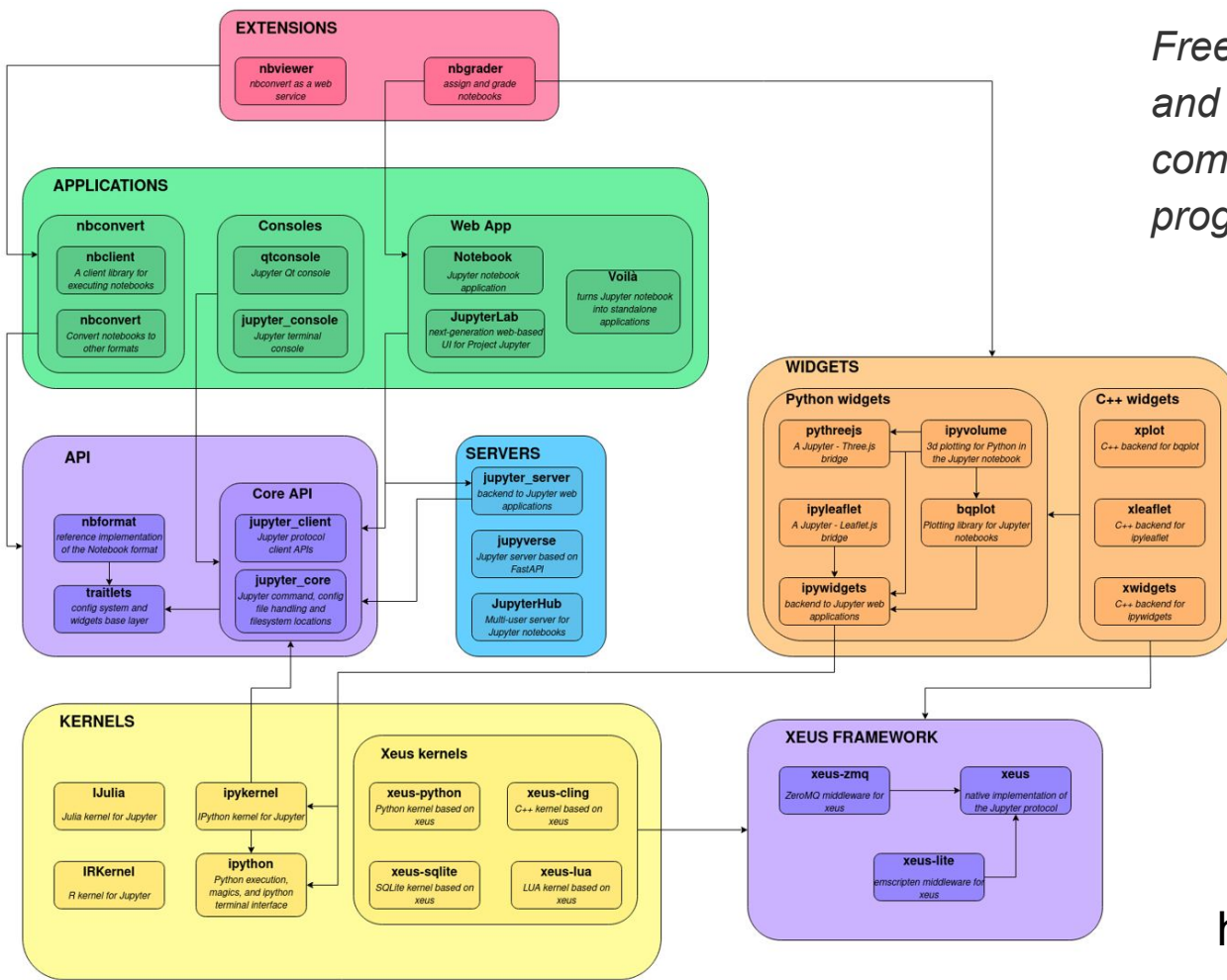
How to use Jupyter with
Calysto Scheme

Jupyter is more than notebooks!




*Free software, open standards,
and web services for interactive
computing across all
programming languages*



<https://jupyter.org/>

Free software, open standards,
and web services for interactive
computing across all
programming languages



<https://jupyter.org/>

<h3>JupyterLab</h3>  <p>The latest web-based interactive development environment</p>	<h3>Jupyter Notebook</h3>  <p>The original web application for creating and sharing computational documents</p>	<h3>JupyterLite</h3>  <p>JupyterLite (Wasm powered Jupyter) deployed as static GitHub Pages</p>
---	--	--

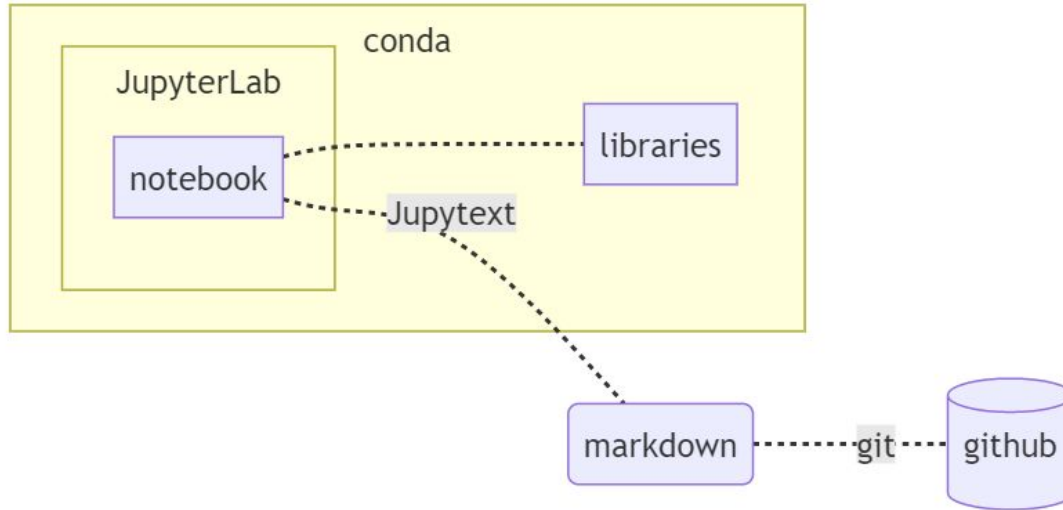
<h3>Jupyter Widgets</h3>  <p>HTML widgets in Jupyter notebooks for interactive exploration of input data</p>	<h3>Voilà</h3>  <p>Share insights by converting notebooks into interactive dashboards</p>
---	---

Some examples

- [JupyterLite](#)
- web application: [Clinical trial tool](#)
- data pipelines: [ploomber.io](#)
- JupyterLab (~IDE) → demo

Notebooks and version control

Workflow for sample project:



Jupyter

- .ipynb files are **json** files containing metadata (→ demo)
- problem: git looks for changes in the plain text (→ demo)
 - commit .ipynb file
 - don't make any content changes, simply execute the cells
 - observation: git detects changes in the file
- solution: strip execution and output metadata (→ demo)
- convert to markdown file (rendered on github to look like a notebook)

<https://jupyter.readthedocs.io/en/latest/>

also: <https://nbdime.readthedocs.io/en/latest/>