Goal: demystify customization

Our explicit objective is to make it less overwhelming for people to get started customizing JupyterLab

Outline

• Living with “stock” JupyterLab
  — settings
  — themes
• Installing 3rd Party extensions
• Examples of our own custom extensions
JupyterLab Architecture

A Collection of Extensions
[or ‘Extensions All the Way Down’]


TechAtBloomberg.com

© 2018 Bloomberg Finance L.P. All rights reserved.
Default JupyterLab Extensions

What you get out of the box
Installing Extensions

...from the command line

```
jupyter labextension install <npm-package-name>[@<version>]
```

*examples:*

```
jupyter labextension install @jupyterlab/xkcd-extension
jupyter labextension install @jupyterlab/xkcd-extension@0.4.0
```

...from the user interface

if you enable the experimental Extension Manager

* “permission denied” error addressed in a few slides.*
Community-Developed Extensions

https://github.com/topics/jupyterlab-extension
Extension Shout Outs

- plotly
- jupyterlab-drawio
- jupyterlab-github
Removing extension and extension lifecycle

...from the command line

jupyter labextension list

jupyter labextension disable @jupyterlab/xkcd-extension

jupyter labextension enable @jupyterlab/xkcd-extension

jupyter labextension uninstall @jupyterlab/xkcd-extension

...from the user interface
if you enable the experimental Extension Manager
Local Flavors

Using \texttt{--app-dir} flag / environment variable to create and move between different sets of extensions

Any information that JupyterLab persists is stored in its application directory, including settings and built assets. This is separate from where the Python package is installed (like in \texttt{site-packages}) so that the install directory is immutable.

The application directory can be overridden using \texttt{--app-dir} in any of the JupyterLab commands, or by setting the \texttt{JUPYTERLAB_DIR} environment variable.

\url{https://jupyterlab.readthedocs.io/en/stable/user/extensions.html#advanced-usage}
Extensions in the wild

Walking through different extension types

demo
HelloExtension

Walking through a basic extension

some definitions...

plugin - code that adds functionality to the jupyterlab application

extension - npm package that exports one or more plugins
Extending JupyterLab

- Application Plugins
Extending JupyterLab

- Application Plugins
- Mime Renderer Extensions
Extending JupyterLab

- Application Plugins
- Mime Renderer Extensions
- Theme Extensions
“I have opinions”

*Variety of opinion is necessary for objective knowledge.*
*And a method that encourages variety is also*
*the only method that is compatible with a humanitarian outlook*

- Paul Feyerabend, *Against Method*

- What if you don’t like the notion of a file browser, or terminal, inside your JupyterLab environment?

  You can take these components, and use them standalone, or mix and match

  See [examples/app](https://github.com/jupyterlab/jupyterlab/tree/master/examples/app) in JupyterLab git repository

- Examples:
  - [ScriptedForms](https://github.com/jupyterlab/jupyterlab/tree/master/examples/app)
  - [ThebeLab](https://github.com/jupyterlab/jupyterlab/tree/master/examples/app)
We are hiring!
https://www.bloomberg.com/careers

Questions?

Our sample extensions are available at
github.com/ivanov/potatoes
from sidecar import Sidecar
from ipyleaflet import Map, basemaps, basemap_to_tiles

m = Map(center=(52, 10), zoom=8, basemap=basemaps.CartoDB.DarkMatter)
strata_all = basemap_to_tiles(basemaps.Strava.All)
m.add_layer(strata_all)

sc = Sidecar