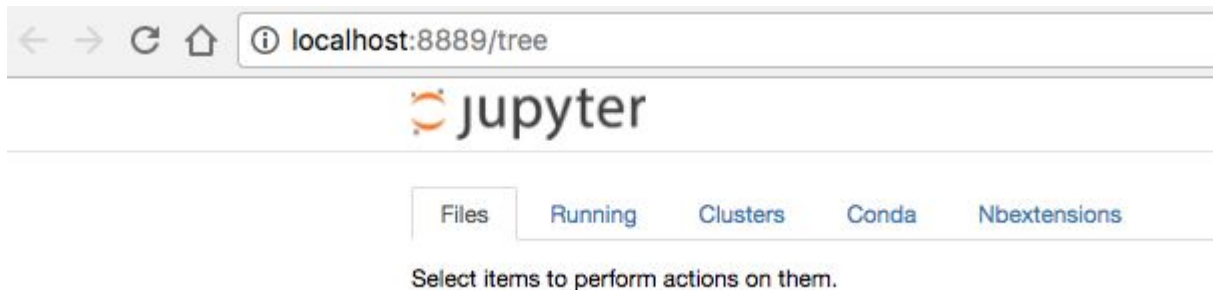


Jupyter Extensions PyDataATX Meetup

Jaya Zenchenko 2017-04-25

Install Extensions

- Have anaconda installed
- I have py35
- Follow: https://github.com/Jupyter-contrib/jupyter_nbextensions_configurator
 - `conda install -c conda-forge jupyter_nbextensions_configurator`
- Start jupyter notebook



My favorites!

- Highlights of my favorite extensions

Exercise Extension

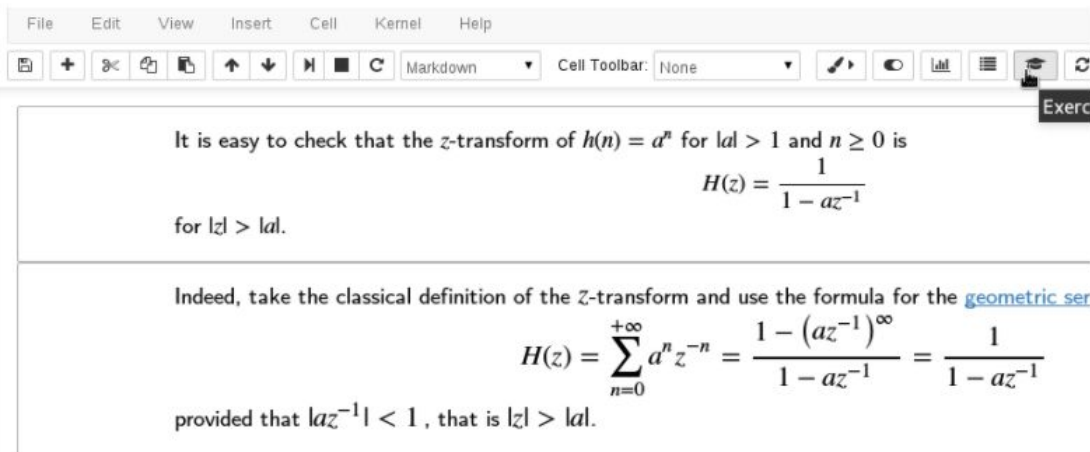
Copyright (c) IPython-Contrib Team. Distributed under the terms of the Modified BSD License.

Exercises extensions -- exercise and exercise2

These are two extensions for Jupyter, for hiding/showing solutions cells. They use the same approach and codebase and differ only by the type of `cell widget` used to show/hide the solutions. The two extensions can be used simultaneously. They require the `rubberband` extension to be installed and activated.

The example below demonstrates some of the features of the exercise extensions.

- First, an solution or "details" cell is created by (a) selecting two cells with the rubberband and (b) clicking on the menu-button [exercise extension]
- Second, the two next cells are selected using the keyboard shortcut Shift-J and a solution is created using the shortcut Alt-D [exercise2 extension]
- Third, the two solutions are expanded by clicking on the corresponding widgets
- Fourth, the solutions are removed by selecting them and clicking on the buttons in the toolbar.



File Edit View Insert Cell Kernel Help

Cell Toolbar: None

It is easy to check that the z -transform of $h(n) = a^n$ for $|a| > 1$ and $n \geq 0$ is

$$H(z) = \frac{1}{1 - az^{-1}}$$

for $|z| > |a|$.

Indeed, take the classical definition of the z -transform and use the formula for the [geometric series](#)

$$H(z) = \sum_{n=0}^{+\infty} a^n z^{-n} = \frac{1 - (az^{-1})^{\infty}}{1 - az^{-1}} = \frac{1}{1 - az^{-1}}$$

provided that $|az^{-1}| < 1$, that is $|z| > |a|$.

Code Folding

Codefolding

This extension enables the CodeMirror feature to allow codefolding in code cells

section: notebook

require path: usability/codefolding/main

compatibility: 4.x

Enable

Disable

```
In [2]: class MyClass(object):  
        """  
        This is a test class  
        """  
        def afun(param1):
```

Parameters

reset

Hotkey to fold/unfold code

⌘F

Change

▼ /nbextensions/usability/codefolding/readme.md

This extension adds codefolding functionality from CodeMirror to a codecell.

After clicking on the gutter (left margin of codecell) or typing `Alt+F`, the code gets folded. See the examples below. The folding status is saved in the cell metadata of the notebook, so reloading of a notebook will restore the folding view.

Supported modes

Three different folding modes are supported:

Indent Folding

Python-style code folding, detects indented code.

```
In [2]: class MyClass(object):  
        """  
        This is a test class
```

Drag And Drop

Drag and Drop

This extension allows dragging&dropping images from the desktop or other programs into a notebook.

section: notebook

require path: usability/dragdrop/main

compatibility: 4.x

Enable

Disable



▼ /nbextensions/usability/dragdrop/readme.md

This IPython notebook extension allows dragging&dropping images from the desktop or other programs into a notebook. A new markdown cell is created below the currently selected cell and the image is embedded. The notebook has been tested with Firefox and Chrome.

A demo video showing drag&drop of images is here: <http://youtu.be/buAL1bTZ73c>

Installation

Copy the contents of the `dragdrop` directory to a new `/nbextensions/usability/dragdrop` directory of your user's IPython directory, or from IPython simply call

```
import IPython
IPython.html.nbextensions.install_nbextension('https://raw.githubusercontent.com/ipython-contrib/IPython-notebook-extensions/master/nbextensions/usability/dragdrop/main.js')
```

Then load the extension from within the IPython notebook:

```
%%javascript
IPython.load_extensions('usability/dragdrop/main');
```

Or, for permanent installation instructions, please see the [readme](#), or the [wiki](#).

Internals

The image will be uploaded to the server into the directory where your notebook resides. This means, the image is not copied into the notebook itself, it will only be linked to. The markdown

Python Markdown

Python Markdown

Allows embedding Python-code in markdown cells

section: notebook

require path: usability/python-markdown/main

compatibility: 4.x

Enable

Disable

▼ /nbextensions/usability/python-markdown/readme.md

Description

The **Python Markdown** extension allows displaying Python output in markdown cells. For example: If you set variable `a` in Python

```
a = 1.23
```

and write the following line in a markdown cell:

```
a = {{a}}
```

It will be displayed as:

```
a = 1.23
```

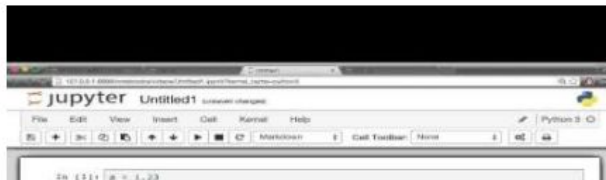
Python Code in Markdown Cells

```
In [15]: a = 2.123
```

The variable `a` is 2.123.

```
In [16]: b=Latex(r'$b = \frac{\epsilon}{2}$')
```

You can also embed Latex: $b = \frac{\epsilon}{2}$ in here!



Scratchpad

Scratchpad

Adds a scratchpad cell to Jupyter notebook.

section: notebook

require path: usability/scratchpad/main

compatibility: 4.x

Enable

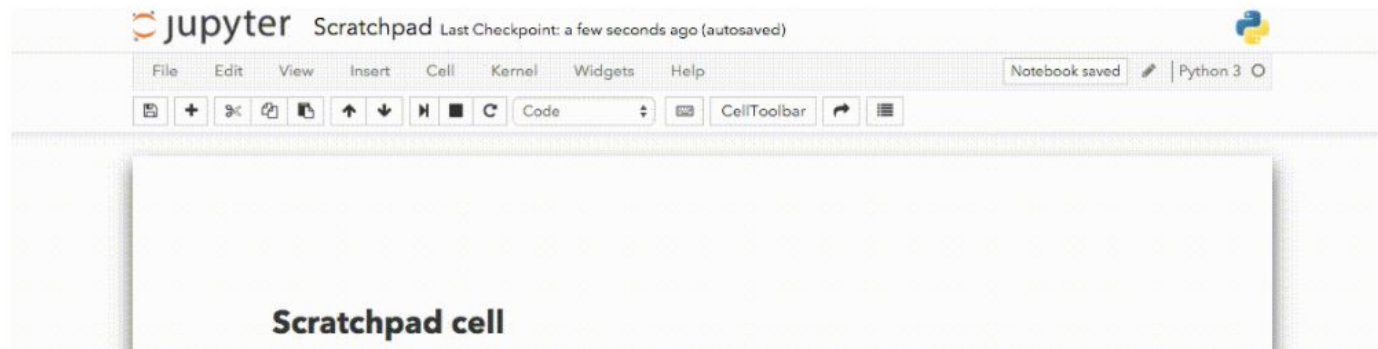
Disable

▼ /nbextensions/usability/scratchpad/README.md

Scratchpad notebook extension

Adds a scratchpad cell to Jupyter notebook. This is a cell in which you can execute code against the current kernel without modifying the notebook document.

The scratchpad can be toggled by clicking the icon in the bottom-right, or via the keyboard shortcut `Ctrl-B`.



Spellchecker

spellchecker

Adds a CodeMirror overlay mode for Typo.js spellchecking

section: notebook

require path: usability/spellchecker/main

compatibility: 4.x

Enable

Disable

▼ Parameters

 reset

☒ enable spellchecker for all Markdown/Raw cells on notebook load

☒ add a toolbar button to toggle spellchecker on and off for all Markdown/Raw cells

language code to use with typo.js

en_US

url for the dictionary .dic file to use

https://cdn.jsdelivr.net/codemirror.spell-checker/latest/en_US.dic

url for the dictionary .aff file to use

https://cdn.jsdelivr.net/codemirror.spell-checker/latest/en_US.aff

▼ /nbextensions/usability/spellchecker/README.md

This extension provides a CodeMirror overlay mode to highlight incorrectly spelled words in Markdown and Raw cells:

Rubberband

Rubberband

The rubberband extension allows selecting multiple cells

section: notebook

require path: usability/rubberband/main

compatibility: 4.x

Enable

Disable



▼ /nbextensions/usability/rubberband/readme.md

Multi-Cell selection using a rubberband. This extension is only available for IPython version 3.x.

Description

The *rubberband* extension allows selecting multiple cells. Cells are selected by pressing `shift` or `ctrl + shift` + left mouse button click and dragging the rubber band over the cells.

- `shift` + left mouse button : select cells that are currently touched by the rubberband
- `ctrl + shift` + left mouse button : select cells that were touched by the rubberband

The `ctrl + shift` action is useful when scrolling inside the notebook. Scrolling is activated when the mouse reaches the upper or lower boundary of the notebook area. For now, the mouse has to be moved to achieve continuous scrolling.

A short video demonstrating the rubberband extension can be found here:



Search and Replace

Search-Replace

Search and replace

section: notebook

require path: usability/search-replace/main

compatibility: 4.x

Enable

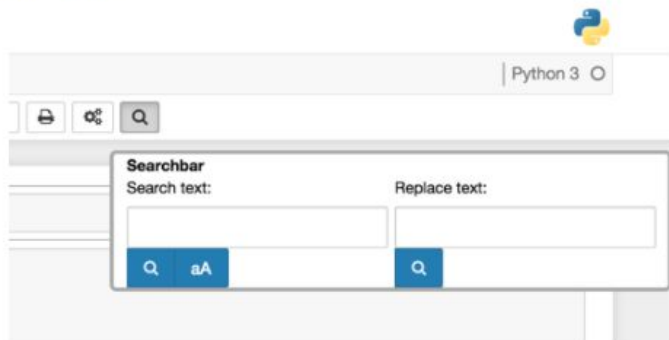
Disable



▼ /nbextensions/usability/search-replace/readme.md

Description

This extension provides a notebook-wide search & replace toolbar.



Tree Filter

Tree Filter

An extension that allows you to filter by filename in the Jupyter notebook file tree (aka dashboard) page.

section: tree

require path: usability/tree-filter/index

compatibility: 4.x

Enable

Disable

Parameters

reset

Filter keyword

/nbextensions/usability/tree-filter/readme.md

Description

An extension that allows you to filter by filename in the Jupyter notebook file tree (aka dashboard) page. Based on <https://github.com/dfreder/jupyter-tree-filter.git>



Files Running Clusters

Select items to perform actions on them.

Upload New ↕

7518294

a5105c1855e8f0850a214

Chrome Clipboard

Chrome Clipboard

This extension allows using the system-wide clipboard to copy cells and images

section: notebook

require path: `usability/chrome-clipboard/main`

compatibility: 4.x

Enable

Disable

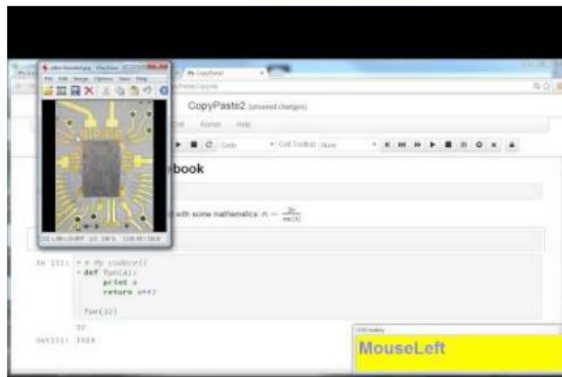


▼ /nbextensions/usability/chrome-clipboard/readme.md

Description

This IPython notebook extension adds system clipboard actions for single or multiple cells. It allows cut/copy/paste operation of notebook cells and images. Images will be saved to the directory where the current notebook sits. There is currently no way to embed images in markdown cells, due to the google-caja sanitizer used to prevent malicious code execution. Multi-cell operation is possible with the latest Jupyter version, or using the `rubberband` extension in this repository.

A demo showing single-cell copy & paste operating in Chrome is available on youtube: <http://youtu.be/iU9dNe4vMUY>



Execute Time

Description

This extension displays when the last execution of a code cell occurred, and how long it took.

Every executed code cell is extended with a new area, attached at the bottom of the input area, that displays the time at which the user sent the cell to the kernel for execution. When the kernel finishes executing the cell, the area is updated with the duration of the execution. The timing information is stored in the cell metadata, and restored on notebook load.

File

Edit

View

Insert

Cell

Kernel

Help

Python 2



```
In [1]: import time
import string
import random

def randword(delay=1, length=10):
    time.sleep(delay)
    return ''.join(
        random.choice(string.lowercase)
        for i in range(length)
    )
```

Last executed 2016-02-17 13:39:49 in 5ms

```
In [2]: randword(0.5)
```

Highlighter

highlighter

Enable to highlight select text in a markdown cell

section: notebook

require path: usability/highlighter/highlighter

compatibility: 3.x, Jupyter (4.x)

Enable

Disable



highlighter extension:

stable, the extension provides [several toolbar buttons](#) for highlighting a selected text *within a markdown cell*. Three different 'color schemes' are provided, which can be easily customized in the stylesheet `highlighter.css`. The last button enables to remove all highlightings in the current cell.

is works both [in rendered](#) and when the cell is in edit mode;

In both modes, it is possible to highlight formatted portions of text (In rendered mode, since the selected text loose its formatting, an heuristic is applied to find the best alignment with the actual text)

When no text is selected, the whole cell is highlighted;

The extension also provides two keyboard shortcuts (Alt-G and Alt-H) which fire the highlighting of the selected text.

Highlights can be preserved when exporting to html or to LaTeX -- details are provided in [export highlights](#)

▼ /nbextensions/usability/highlighter/readme.md

The highlighter extension:

- Firstable, the extension provides [several toolbar buttons](#) for highlighting a selected text *within a markdown cell*. Three different 'color schemes' are provided, which can be easily customized in the stylesheet `highlighter.css`. The last button enables to remove all highlightings in the current cell.
- This works both when the cell is *rendered* and when the cell is in edit mode;
- In both modes, it is possible to highlight formatted portions of text (In rendered mode, since the selected text loose its formatting, an heuristic is applied to find the best alignment with the actual text)
- When no text is selected, the whole cell is highlighted;
- The extension also provides two keyboard shortcuts (Alt-G and Alt-H) which fire the highlighting of the selected text.
- Highlights can be preserved when exporting to html or to LaTeX -- details are provided in [export highlights](#)

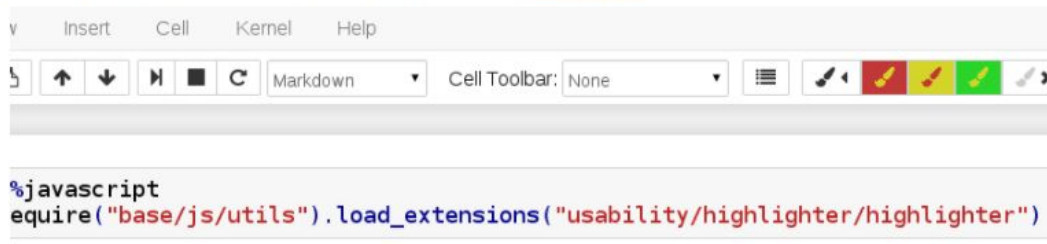


Table of Contents

Table of Contents (2)

The toc2 extension enables to collect all running headers and display them in a floating window, as a sidebar or with a navigation menu. The extension is also draggable, resizable, collapsable, dockable and features automatic numerotation with unique links ids, and an optional toc cell.

section: notebook

require path: `usability/toc2/main`

compatibility: `4.x`

Enable

Disable



Parameters

reset

☒ Automatically number notebook's sections

Maximum level of nested sections to display on the tables of contents

4

☐ Add a Table of Contents cell at the top of the notebook

☐ Display toc window at startup

☒ Display Table of Contents as a sidebar (otherwise as a floating window)

☐ Display Table of Contents as a navigation menu

/nbextensions/usability/toc2/README.md

Table of Contents (2)

But wait, there's more!

Lots more extensions - growing list!

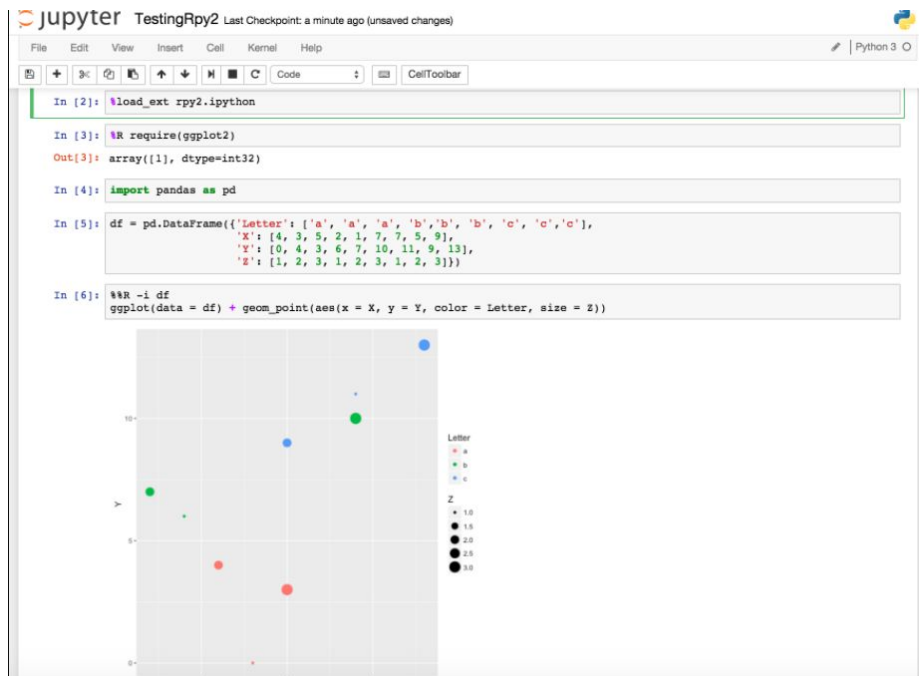
Beware and do not hoard extensions - try incrementally one at a time and make sure things are compatible before hosing your environment!

Other Jupyter Awesomeness

- Can write code in many languages - make sure to attach kernel - <https://github.com/jupyter/jupyter/wiki/Jupyter-kernels>
- Create dashboards: <https://github.com/jupyter/dashboards>
- Tim will present one in his talk

R?

Love R? Love Python but can't live without ggplot? Pass data between python and R in same notebook and plot away!



Pivot Tables

- <http://nicolas.kruchten.com/pivottable/examples/index.html>
- Love this! Will show a lightning talk on this later!