Jupyter Notebook Cheatsheet

Keyboard Shortcuts

The following two lists are the keyboard shortcuts that are enabled by default in a Jupyter notebook. If you have edited the configuration of the keyboard shortcuts in your Jupyter Notebook environment, you may view a list of the currently available shortcuts from the Jupyter Help menu.

Command Mode	***(press `Esc` to enable command mode)***
`Alt-R`	Enter/Exit RISE Slideshow
`F`	find and replace
`Shift-C`	open the nbconfigurator page for RISE
`Shift-F`	(un)set current cell as a Fragment cell
`Shift-I`	(un)set current cell as a Slide cell
`Shift-U`	(un)set current cell as a Sub-slide cell
`Ctrl-Shift-F`	open the command palette
`Ctrl-Shift-P`	open the command palette
`Enter`	enter edit mode
`P`	open the command palette
`Shift-Enter`	run cell, select below
`Ctrl-Enter`	run selected cells
`Alt-Enter`	run cell and insert below
`Y`	change cell to code
`M`	change cell to Markdown
`R`	change cell to raw
`1`	change cell to heading 1
`2`	change cell to heading 2
`3`	change cell to heading 3
`4`	change cell to heading 4
`5`	change cell to heading 5
`6`	change cell to heading 6
`K`	select cell above
`Up`	select cell above
`Down`	select cell below
`J`	select cell below
`Shift-K`	extend selected cells above
`Shift-Up`	extend selected cells above
`Shift-Down`	extend selected cells below
`Shift-J`	extend selected cells below
`A`	insert cell above
`B`	insert cell below
`X`	cut selected cells
`C`	copy selected cells
`Shift-V`	paste cells above
`V`	paste cells below

Z	undo cell deletion
D,D	delete selected cells
Shift-M	merge selected cells, or current cell with cell below if only one cell is selected
Ctrl-S	Save and Checkpoint
S	Save and Checkpoint
L	toggle line numbers
O	toggle output of selected cells
Shift-O	toggle output scrolling of selected cells
H	show keyboard shortcuts
I,I	interrupt the kernel
0,0	restart the kernel (with dialog)
Esc	close the pager
Q	close the pager
Shift-L	toggles line numbers in all cells, and persist the setting
Shift-Space	scroll notebook up
Space	scroll notebook down

Edit Mode	***(press `Enter` to enable to enter edit mode in a cell)***
Tab	code completion or indent
Shift-Tab	tooltip
Ctrl-]	indent
Ctrl-[	dedent
Ctrl-A	select all
Ctrl-Z	undo
Ctrl-/	comment
Ctrl-D	delete whole line
Ctrl-U	undo selection
Insert	toggle overwrite flag
Ctrl-Home	go to cell start
Ctrl-Up	go to cell start
Ctrl-End	go to cell end
Ctrl-Down	go to cell end
Ctrl-Left	go one word left
Ctrl-Right	go one word right
Ctrl-Backspace	delete word before
Ctrl-Delete	delete word after
Ctrl-Y	redo
Alt-U	redo selection
Ctrl-M	enter command mode
Ctrl-Shift-F	open the command palette
Ctrl-Shift-P	open the command palette
Esc	enter command mode
Shift-Enter	run cell, select below
Ctrl-Enter	run selected cells
Alt-Enter	run cell and insert below
Ctrl-Shift-Minus	split cell at cursor
Ctrl-S	Save and Checkpoint
Down	move cursor down
Up	move cursor up

Markdown

This is a list of common syntax used in Markdown cells in a Jupyter notebook to produce rich formatted text and embedded multimedia. To see the rendered output for these examples, you can open any Jupyter notebook, start editing a Markdown cell, insert the sample text, and execute the cell to view the formatted output (e.g,. by pressing CTRL+Enter). A copy of these samples can be found on GitHub (some github-specific features are not supported): https://guides.github.com/pdfs/markdown-cheatsheet-online.pdf

Headings

```
# This is an <h1> tag
## This is an <h2> tag
###### This is an <h6> tag
```

Emphasis

```
*This text will be italic*
_This will also be italic_
**This text will be bold**
__This will also be bold__
*You **can** combine them*
```

Blockquotes

```
As Nicholas Chrisman said:
> Geographers never get lost. They
> just do accidental field work.
```

Unordered Lists

```
* Unordered Item 1
* Unordered Item 2
 * Unordered Item 2a
 * Unordered Item 2b
```

Ordered Lists

```
1. Ordered Item 1
2. Ordered Item 2
3. Ordered Item 3
 1. Ordered Item 3a
 2. Ordered Item 3b
```

Images (linked)

```
![Logo](images/logo.png)
Format: ![Alt Text](url)
```

The URL portion for an image may be a full path to an Internet-accessible location (e.g., `https://esri.ca/images/logo.png`), or a path to a location accessible relative to your local notebook server (e.g., a path that points to `images/logo.png` will work if there an `images` folder within the same path as your notebook, with a `logo.png` file inside it).

Images (embedded)

To ensure your notebooks are easy to share, and are not subject to broken links or missing files, you may embed images directly into your notebook file. You can do this while editing a Markdown cell in your notebook by opening the ‘Edit’ menu and selecting ‘Insert image’. In the prompt that is displayed, choose an image from your files and click ‘OK’. The image will be stored internally in the notebook file within the current cell, and Markdown will be inserted into the text. When you execute the cell (e.g., by pressing CTRL+Enter), the image will be displayed in the rendered output.

Below is an example of the syntax that you will see inserted into your Markdown cell when embedding an image this way:

```
![example_image.jpg](attachment:example_image.jpg)
```

Links

```
http://github.com - automatic!
[GitHub](http://github.com)
```

Code (inline)

```
Snippets of code can be displayed inline, enclosed in single backticks – for
example, to emphasize references to things like `variable_names` within your
text.
```

Code (blocks)

Longer blocks of code can be enclosed by two lines with triple backticks (without any trailing spaces). If using a well-known language, the first set of backticks can be followed by the name of the language. For example, js or javascript following the backticks will render the following with code-style text that highlights elements of the code with different colours (e.g., core syntax words, variables, numbers, text strings):

~~~~
```js
function test() {
    console.log("sample javascript code");
}
```
~~~~

Task lists

```
- [x] this is a complete item
- [ ] this is an incomplete item
- [x] [links](), **formatting**, and basic HTML <del>tags</del> are supported
- [x] list syntax required (any unordered or ordered list supported)
```

Tables

```
First Header | Second Header
------------ | -------------
Content cell 1 | Content cell 2
Content column 1 | Content column 2
```

Simple HTML formatting

```
Most basic HTML tags will render as expected:
* <em>Italic</em>
* <strong>Bold</strong>
* <a href="https://esri.ca">Link</a>
```

Escape characters

Use backslashes to escape any of the following characters in your Markdown syntax:

```
\ backslash
` backtick
* asterisk
_ underscore
{} curly braces
[] square brackets
() parentheses
# hash mark
+ plus sign
- minus sign (hyphen)
. dot
! exclamation mark
```

Equations

```
Text enclosed by single `$` characters will be displayed as inline math
formulas, for example: $ArcGIS + Python + Jupyter = 💥$

When double `$$` characters enclose a formula, it will be displayed
as an equation on a separate line: $$ArcGIS + Python + Jupyter = 💥$$
```

Examples of equations using LaTeX syntax:

```
$$
F(k) = \int_{-\infty}^{\infty} f(x) e^{2\pi i k} dx
$$
```

```
$$
\begin{align}
\nabla \times \vec{\mathbf{B}} -\, \frac1c\, \frac{\partial\vec{\mathbf{E}}}{
\partial t} & = \frac{4\pi}{c}\vec{\mathbf{j}} \\
\nabla \cdot \vec{\mathbf{E}} & = 4 \pi \rho \\
\nabla \times \vec{\mathbf{E}}\, +\, \frac1c\, \frac{\partial\vec{\mathbf{B}}
}{\partial t} & = \vec{\mathbf{0}} \\
\nabla \cdot \vec{\mathbf{B}} & = 0
\end{align}
$$
```

IPython display Module

The IPython display module provides a variety of methods for inserting content into a notebook. Full documentation is online: https://ipython.readthedocs.io/en/stable/api/generated/IPython.display.html

Below are a few simple examples:

YouTube Video

```
from IPython.display import YouTubeVideo
YouTubeVideo(id="lXYFhgntmjg", width=800, height=450)
```

HTML

```
from IPython.display import HTML
HTML("<a class=\"sample\">Some HTML content!</a>"
     "<style>"
         ".sample:hover {color:magenta; cursor:pointer}"
     "</style>")
```

JavaScript

```
from IPython.display import Javascript
Javascript("alert(\"Hello World!\");")
```

The Education and Research Group at Esri Canada makes every effort to present accurate and reliable information. The Web sites and URLs used in this tutorial are from sources that were current at the time of production but are subject to change without notice to Esri Canada.