Published in Analytics Vidhya

Hannan Satopay Follow
Nov 18, 2019 · 10 min read · Listen

The Ultimate Markdown Guide (for Jupyter Notebook)

> *All code and no heading makes Jupyter a dull read.*

You may have heard the above proverb, but if not, then well, I have just made it up! But it

notebook to offer the reader a sufficient explanation of both the code and the concept.

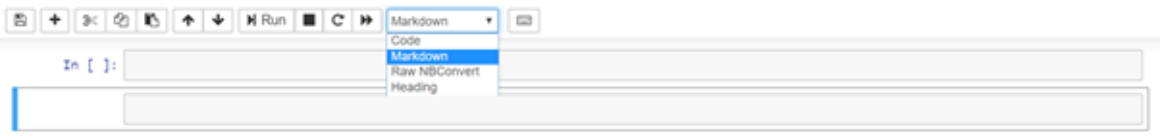
But first…

What's Markdown?

Markdown is a lightweight Markup language with a plain text syntax. John Gruber developed the Markdown language in 2004 in a collaborative effort with Aaron Swartz, intending to enable people to "write with easy-to-read and easy-to-write plain text format and potentially convert it to structurally correct XHTML (or HTML)." Markdown is designed to be as easy-to-read and easy-to-write as possible. Readability, however, is emphasized above all else.

Nevertheless, Markdown is not a substitute for, or even close to, HTML. Its syntax is minimal, correlating only to a tiny proportion of HTML tags. Markdown's idea is to make reading, writing, and editing prose easy without the intention to create a syntax that's just for quickly adding HTML tags. HTML is a format for publishing, while Markdown is a format for reading. Therefore, the formatting syntax of Markdown tackles just issues that can be expressed in plain text. You simply use HTML for any Markup that is not covered by the Markdown syntax. You don't need to preface it or delimit it to indicate that you are switching from Markdown to HTML — you just use the tags.

Among the data enthusiasts, Jupyter notebook is in trend. It is simple to use and helps you to create and share documents that include code, visualizations, and narration. Recall that a Jupyter Notebook is a series of cells that can store text or code. Cells shape a notebook's core. Markdown Cells allows you to write and render Markdown syntax. Here's where you can explain and document the processes. On the other hand, code cells allow you to write and run program code like Python.



Jupyter Notebook — Types of Cells

Now that we have some background knowledge about Markdown, without any further ado, let's dive into the syntax!

Markdown Syntax Collections

HEADINGS

Markdown Syntax

```
# Header 1

## Header 2

### Header 3

#### Header 4

##### Header 5

###### Header 6
```

Equivalent HTML Syntax

```
<h1>Header 1</h1>

<h2>Header 2</h2>

<h3>Header 3</h3>

<h4>Header 4</h4>

<h5>Header 5</h5>

<h6>Header 6</h6>
```



```
### Header 3

#### Header 4

##### Header 5

###### Header 6
```

TEXT EMPHASIS

BOLD

Markdown Syntax

```
**This is bold text**

__This is bold text__
```

Equivalent HTML Syntax

```
<strong>This is bold text</strong>
```

Rendered Output

```
**This is bold text**
__This is bold text__
```

ITALIC

Markdown Syntax

```
*This is italic text*
```

```
<em>This is italic text</em>
```

Rendered Output

```
*This is italic text*
_This is italic text_
```

BOLD & ITALIC

Markdown Syntax

```
***Bold and Italic***

___Bold and Italic___
```

Equivalent HTML Syntax

```
<strong><em> Bold and Italic </em></strong>
```

Rendered Output

```
***Bold and Italic***
___Bold and Italic___
```

STRIKETHROUGH

Markdown Syntax

```
<del>Scratch this</del>
```

Rendered Output

~~Scratch this~~

BACKSLASH ESCAPE

Backslash Escape prevents Markdown from interpreting a character as an instruction, rather than as the character itself.

Markdown Syntax

```
\# Not a header
```

Equivalent HTML Syntax

```
# Not a header
```

Rendered Output

\# Not a header

PARAGRAPHS

A paragraph is simply one or more consecutive lines of text, separated by one or more blank lines.

Markdown Syntax

```
We are learning Markdown.
```

Equivalent HTML Syntax

```
<p>Hello World!</p>

<p>We are learning Markdown.</p>
```

LINE BREAK

A line break is the termination of the previous line and the beginning of a new line.

Markdown Syntax

To force a line return, place two or more empty spaces at the end of a line and press the Enter key.

```
Some text

Some more text
```

Equivalent HTML Syntax

The line break tag starts with
 tag with no closing tag which breaks the line, and the remaining contents begin with a new line.

```
Some text <br>

Some more text
```

Rendered Output



BLOCKQUOTES

Blockquotes can hold the large chunk of text and are generally indented.

Markdown Syntax

```
> This is a blockquote
```

Equivalent HTML Syntax

```
<blockquote>This is a blockquote</blockquote>
```

Rendered Output

NESTED BLOCK QUOTING

Markdown Syntax

```
> some text

>> and then some more text

>>> and then some more
```

Equivalent HTML Syntax

```
<p>and then some more</p>
</blockquote>
</blockquote>
</blockquote>
```

Rendered Output

HORIZONTAL LINE

Markdown Syntax

```
---
___
***
```

Equivalent HTML Syntax

```
<hr>
```

Rendered Output

Markdown Syntax

```
1. Cheese

2. Carrot

3. Coconut
```

Note: Numbering is irrelevant

Equivalent HTML Syntax

```
<ol>
 <li>Cheese</li>
 <li>Carrot</li>
 <li>Coconut</li>
</ol>
```

Rendered Output

1. Cheese
2. Carrot
3. Coconut

UNORDERED LIST

The Unordered list is a bullet list.

Markdown Syntax

```
- Cheese

- Carrot
```



```
<ul>
 <li>Cheese</li>
 <li>Carrot</li>
 <li>Coconut</li>
 </ul>
```

Rendered Output

GRAPHICS

You can attach graphics (such as images) to a notebook in Markdown cells.

Note1: You can also Drag and Drop your images to the Markdown cell to attach it to the notebook.

Note2: Below I have used links to images on the web but you can very well use an offline image by adding the complete filename (plus the file path if it is in a different directory other then the Jupyter Notebook).

Markdown Syntax

One simple way of adding an image to a Markdown cell is through the following syntax:

```
![](https://www.python.org/static/community_logos/python-logo-master-
v3-TM.png)
```

If you want to add a hover title to the image then you can simply modify the syntax like below:

You can also use the reference-style format for the images:

```
![][some-id]

[some-id]: https://www.python.org/static/community_logos/python-logo-master-v3-TM.png "Python Logo"
```

Equivalent HTML Syntax

```
<img src="https://www.python.org/static/community_logos/python-logo-master-v3-TM.png" title="Python Logo"/>
```

Rendered Output

HYPERLINKS

AUTOMATIC LINKS

Markdown Syntax

```
https://en.wikipedia.org
```

```
<a href="https://en.wikipedia.org">https://en.wikipedia.org</a>
```

STANDARD LINKS

Markdown Syntax

```
[click me](https://en.wikipedia.org)
```

Equivalent HTML Syntax

```
<a href="https://en.wikipedia.org">click me</a>
```

STANDARD LINKS (WITH MOUSE-OVER TITLES)

Markdown Syntax

```
[click me](https://en.wikipedia.org "Wikipedia")
```

Equivalent HTML Syntax

```
<a href="https://en.wikipedia.org" title=”Wikipedia”>click me</a>
```

REFERENCE-STYLE LINKS

Markdown Syntax



Note: Link IDs are not case-sensitive.

You don't really have to give your link an ID. If you use the words in the first set of brackets to later define the link, Markdown will understand it. This works as follows:

```
This is [a reference][]

[a reference]: https://en.wikipedia.org "Wikipedia"
```

Equivalent HTML Syntax

```
This is <a href="https://en.wikipedia.org" title="Wikipedia">a
reference</a>
```

Rendered Output

ANCHOR LINKS

An anchor link is a link on a page that brings you to a specific place on that page. In a Jupyter Notebook, it can be used to link to any section of the notebook for easy navigation.

Note: The ID used for the linking should be unique to avoid misdirection.

Technique 1

Create a new cell above the section you want to link to and add the following line of code:

To create an anchor link that links to the above section, simply add the following line of code:

```
[Section title](#id)
```

Technique 2

If you have a section with the heading — My Great Heading then to add an anchor link to this section, simply add a hyphen in place of the blank space like below:

```
[Section title](#My-Great-Heading)
```

Rendered Output

TASK LISTS

Markdown Syntax

```
- [x] Some task
- [ ] Some more task
```

```
<input type="checkbox" disabled> Some more task
```

Rendered Output

- [x] Some task
- [ ] Some more task

CODE SECTION

Markdown Syntax

Inline Code:

````
`some piece of inline code`

```

some piece of block code

```
````

You can also perform syntax highlighting like below:

````
```javascript

var s = "JavaScript syntax highlighting";

alert(s);

```

```python

s = "Python syntax highlighting"
````




````

Equivalent HTML Syntax

You can use the `<code>` tags in HTML to get similar results. For syntax highlighting, appropriate classes have to be utilized for the different elements.

Rendered Output

````
`some piece of inline code`
```
some piece of block code
```
```javascript
var s = "JavaScript syntax highlighting";
alert(s);
```
```python
s = "Python syntax highlighting"
print s
````

TABLE

A table can be constructed using `|` (pipe symbol) and `-` (dash) to mark columns and rows.

Markdown Syntax

```
|Header|Header|Header|
|------|------|------|
|A     |B     |C     |
|X     |Y     |Z     |
```

Note: It is not important to add spaces after each text, roughly aligning with the columns will do just fine. Also, the number of dashes is irrelevant and is just cosmetical.

The text in each header and cell of a table will by default justify to the right.

For manually changing the justification, you can use:
```

Use `:-` for left centered

Equivalent HTML Syntax

```
<table>
<thead>
<tr><th>Header</th><th>Header</th><th>Header</th></tr>
</thead>
<tbody>
<tr><td>A</td><td>B</td><td>C</td></tr>
<tr><td>X</td><td>Y</td><td>Z</td></tr>
</tbody>
</table>
```

Rendered Output

```
|Header|Header|Header|
|------|------|------|
|A     |B     |C     |
|X     |Y     |Z     |
```

INLINE HTML

TEXT COLOR

```
<span style="color:color">Text</span>
```

Where color = `blue|red|green|pink|yellow`

Rendered Output

TEXT FONT FAMILY

```
<span style="font-family:Comic Sans MS">This is a text</span>
```

For a list of some commonly used fonts, checkout CSS Font Family List.

Rendered Output

COLORED NOTE BOXES

Use one of the following tags to display text in a colored box. The color of the box is determined by the alert type that is specified.

Blue boxes (alert-info)

```
<div class="alert alert-block alert-info">
<b>Tip:</b> Use blue boxes (alert-info) for tips and notes.</div>
```

Yellow boxes (alert-warning)

```
<div class="alert alert-block alert-warning">
<b>Example:</b> Use yellow boxes for examples that are not inside code
```

Green boxes (alert-success)

```
<div class="alert alert-block alert-success">

<b>Success:</b> This alert box indicates a successful or positive
action.

</div>
```

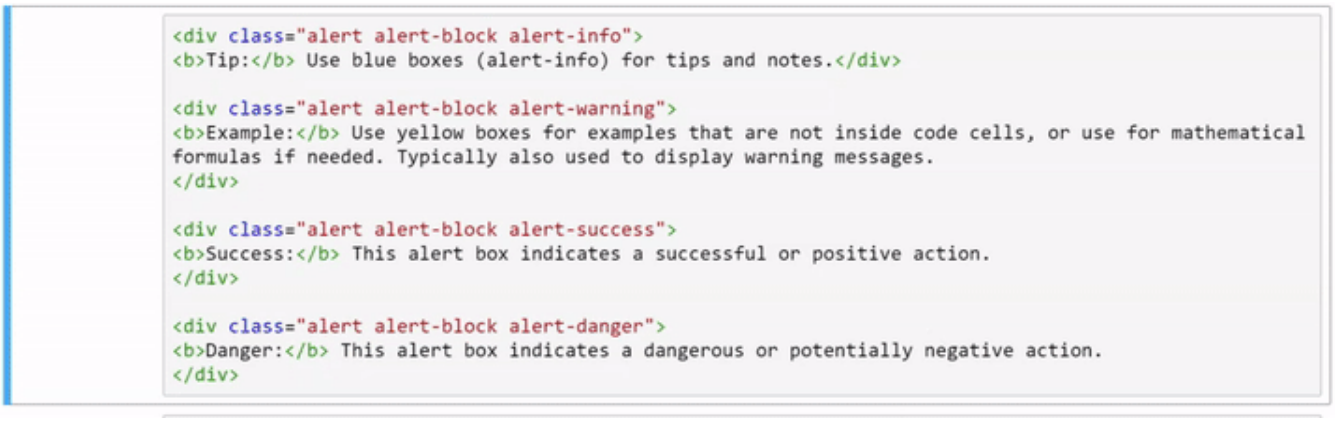
Red boxes (alert-danger)

```
<div class="alert alert-block alert-danger">

<b>Danger:</b> This alert box indicates a dangerous or potentially
negative action.

</div>
```

Rendered Output



```
<div class="alert alert-block alert-info">
<b>Tip:</b> Use blue boxes (alert-info) for tips and notes.</div>

<div class="alert alert-block alert-warning">
<b>Example:</b> Use yellow boxes for examples that are not inside code cells, or use for mathematical
formulas if needed. Typically also used to display warning messages.
</div>

<div class="alert alert-block alert-success">
<b>Success:</b> This alert box indicates a successful or positive action.
</div>

<div class="alert alert-block alert-danger">
<b>Danger:</b> This alert box indicates a dangerous or potentially negative action.
</div>
```

CELL BACKGROUND COLOR

Rendered Output

```
<code style="background:yellow;color:black">Useful for highlighting to grab the attention of the
reader towards certain points.</code>
```

I also tend to use the following color style when adding a piece of terminal code to a Markdown cell:

```
<p style="background:black">
<code style="background:black;color:white">C:\Users\YOUR_USERNAME> pip3
install roughviz
</code>
</p>
```

By the way, roughviz is a Python visualization library that I have created for creating sketchy/hand-drawn styled charts. Do check it out on Github and PyPI.

Rendered Output

```
<p style="background:black">
<code style="background:black;color:white">C:\Users\YOUR_USERNAME> pip3 install roughviz
</code>
</p>
```

HTML MARK TAG

Highlight parts of a text:

```
Do not forget to buy <mark>milk</mark> today.
```

Rendered Output

DEFINITION LISTS

```
<dl>
<dt>First Term</dt>
<dd>This is the definition of the first term.</dd>
<dt>Second Term</dt>
<dd>This is one definition of the second term. </dd>
<dd>This is another definition of the second term.</dd>
</dl>
```

Rendered Output

<dl>
<dt>First Term</dt>
<dd>This is the definition of the first term.</dd>
<dt>Second Term</dt>
<dd>This is one definition of the second term. </dd>
<dd>This is another definition of the second term.</dd>
</dl>

NAVIGATION MENU

It defines a set of navigation links.

```
<nav>

<a href=”https://www.google.com">LinkedIn</a> |

<a href=”/css/”>Github</a> |

<a href=”/js/”>Medium</a> |

</nav>
```

Rendered Output



```
<a href="https://medium.com/@hannansatopay">Medium</a>
</nav>
```

LaTeX MATH

Jupyter Notebooks' Markdown cells support LateX for formatting mathematical equations. To tell Markdown to interpret your text as LaTex, surround your input with dollar signs like this:

```
$\sqrt{k}$
```

Rendered Output

```
$\sqrt{k}$
```

GEOMETRIC SHAPES

Use this code with a decimal or hex reference number from here: UTF-8 Geometric shapes

```
&#reference_number;
```

If you found this article useful, then make sure to hit that clap button like crazy! Also, follow me on medium to get more awesome articles directly in your feed.