JuPyTer-Notebook1

March 27, 2018

1 This is markdown

Paragraphs are separated by a blank line.

2nd paragraph. *Italic*, **bold**, and `monospace`. Itemized lists look like:

- this one
- that one
- the other one

Note that --- not considering the asterisk --- the actual text content starts at 4-columns in.

> Block quotes are written like so.
>
> They can span multiple paragraphs, if you like.

Use 3 dashes for an em-dash. Use 2 dashes for ranges (ex., "it's all in chapters 12--14"). Three dots ... will be converted to an ellipsis. Unicode is supported.

```
In [1]: # this i my first jupyter note-book

        print('hello world')
```

```
hello world
```

```
In [2]: 'hello world'
```

```
Out[2]: 'hello world'
```

```
In [3]: var = 'raghav'
```

```
In [15]: var
```

```
Out[15]: 'gupta'
```

```
In [14]: var = 'gupta'
```

```
In [6]: !ls -l
        !echo '\nthis is a bash command'
        #!pip list
```

```
total 60
-rw-r--r-- 1 raghav raghav 16663 Mar 26 23:36 JuPyTer-Notebook1.ipynb
-rw-r--r-- 1 raghav raghav 38800 Mar 26 23:35 test.png

this is a bash command
```

```
In [7]: %lsmagic
```

```
Out[7]: Available line magics:
        %alias  %alias_magic  %autocall  %automagic  %autosave  %bookmark  %cat  %cd  %clear

        Available cell magics:
        %%!  %%HTML  %%SVG  %%bash  %%capture  %%debug  %%file  %%html  %%javascript  %%js

        Automagic is ON, % prefix IS NOT needed for line magics.
```

```
In [8]: !ls
        %ls
```

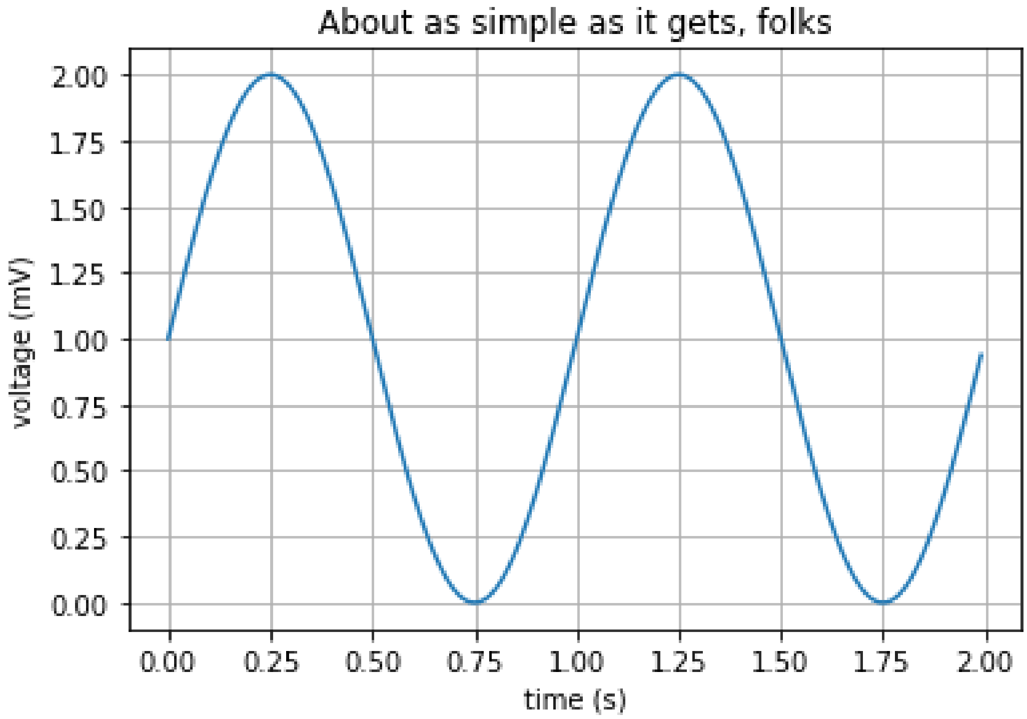
```
JuPyTer-Notebook1.ipynb  test.png
JuPyTer-Notebook1.ipynb  test.png
```

```
In [9]: %matplotlib inline
```

```
In [10]: import matplotlib.pyplot as plt
         import numpy as np

         t = np.arange(0.0, 2.0, 0.01)
         s = 1 + np.sin(2*np.pi*t)
         plt.plot(t, s)

         plt.xlabel('time (s)')
         plt.ylabel('voltage (mV)')
         plt.title('About as simple as it gets, folks')
         plt.grid(True)
         plt.savefig("test.png")
         plt.show()
```



```
In [11]: %%HTML
         <iframe width="450" height="260" style="border: 1px solid #cccccc;" src="https://thin

<IPython.core.display.HTML object>
```

```
In [12]: %%timeit
         for i in range(10**1):
             pass

608 ns ś 45.1 ns per loop (mean ś std. dev. of 7 runs, 1000000 loops each)
```

```
In [13]: import pandas as pd
         import numpy as np

         df = pd.DataFrame(np.random.randn(5, 10))
         df.head()

Out[13]:          0         1         2         3         4         5         6  \
         0  0.453589 -0.951406  0.770447 -1.067140  0.391311 -1.032733  0.285852
         1  0.173287  2.425505  1.008431  1.359177 -1.124596  0.537604 -0.309138
         2  0.825731  0.993222 -1.562482 -0.685979 -0.192471 -0.321000  0.225396
         3  0.080185 -0.985257 -2.329566  0.724593 -0.371769  1.347537  0.227808
```

```
4  0.320130  0.353393  0.433806  1.626688 -0.469187 -0.894356  1.185378

          7         8         9
0 -0.715373  0.073918  0.911274
1 -0.333655  0.398353 -1.158985
2 -1.745733  0.233456 -0.159295
3  0.808963 -0.798088  1.424620
4 -0.265192 -1.668665 -0.392350
```