# Jupyter notebook custom conversion

Romain Madar

August 2018

## Contents

# 1 nbconvert latex test

**Lorem ipsum** dolor sit amet, consectetur adipiscing elit. Nunc luctus bibendum felis dictum sodales. Ut suscipit, orci ut interdum imperdiet, purus ligula mollis *justo*, non malesuada nisl augue eget lorem. Donec bibendum, erat sit amet porttitor aliquam, urna lorem ornare libero, in vehicula diam diam ut ante. Nam non urna rhoncus, accumsan elit sit amet, mollis tellus. Vestibulum nec tellus metus. Vestibulum tempor, ligula et vehicula rhoncus, sapien turpis faucibus lorem, id dapibus turpis mauris ac orci. Sed volutpat vestibulum venenatis.

This is a test list:

1. item 1
    - subitem 1
    - subitem 2
2. item 2
3. item 3

## 1.1 Printed Using Python

```
next_paragraph = """
Aenean vitae diam consectetur, tempus arcu quis, ultricies urna
    . Vivamus venenatis sem
quis orci condimentum, sed feugiat dui porta.
"""

def identity_dec(ob):
    return ob

@identity_dec
def nifty_print(text):
    """Used to test syntax highlighting"""

    print(text * 2)

nifty_print(next_paragraph)
```

```
Aenean vitae diam consectetur, tempus arcu quis, ultricies urna
    . Vivamus venenatis sem
quis orci condimentum, sed feugiat dui porta.

Aenean vitae diam consectetur, tempus arcu quis, ultricies urna
    . Vivamus venenatis sem
quis orci condimentum, sed feugiat dui porta.
```

## 1.2 Pyout (and Text Wrapping)

```
Text = """
Aliquam blandit aliquet enim, eget scelerisque eros adipiscing
    quis. Nunc sed metus
ut lorem condimentum condimentum nec id enim. Sed malesuada
    cursus hendrerit. Praesent
et commodo justo. Interdum et malesuada fames ac ante ipsum
    primis in faucibus.
Curabitur et magna ante. Proin luctus tellus sit amet egestas
    laoreet. Sed dapibus
neque ac nulla mollis cursus. Fusce mollis egestas libero
    mattis facilisis.
"""
Text #Use print(Text) instead to get text wrapping in pdf
```

```
'\textbackslash{}nAliquam blandit aliquet enim, eget
    scelerisque eros adipiscing quis. Nunc sed metus \
    textbackslash{}nut lorem condimentum condimentum nec id
    enim. Sed malesuada cursus hendrerit. Praesent \
    textbackslash{}net commodo justo. Interdum et malesuada
    fames ac ante ipsum primis in faucibus. \textbackslash{}
    nCurabitur et magna ante. Proin luctus tellus sit amet
    egestas laoreet. Sed dapibus \textbackslash{}nneque ac
    nulla mollis cursus. Fusce mollis egestas libero mattis
    facilisis.\textbackslash{}n'
```

```
print(Text)
```

```
Aliquam blandit aliquet enim, eget scelerisque eros adipiscing
    quis. Nunc sed metus
ut lorem condimentum condimentum nec id enim. Sed malesuada
    cursus hendrerit. Praesent
et commodo justo. Interdum et malesuada fames ac ante ipsum
    primis in faucibus.
Curabitur et magna ante. Proin luctus tellus sit amet egestas
    laoreet. Sed dapibus
neque ac nulla mollis cursus. Fusce mollis egestas libero
    mattis facilisis.
```

```
import numpy as np

a = np.random.rand(10,10)
print(a)
a
```

```
[[0.81632631 0.89816275 0.12130491 0.4171127  0.81342409
    0.97502141
  0.98536459 0.98511233 0.68830589 0.58976414]
 [0.02713838 0.39218547 0.25160237 0.08229395 0.30770868
    0.85632565
  0.44919185 0.16286998 0.33010748 0.24163147]
```

```
 [0.81929727 0.05795412 0.8925537  0.85529752 0.20782781
    0.68019862
  0.17732165 0.69354023 0.66100412 0.94989246]
 [0.74413864 0.57979159 0.61355802 0.52906865 0.56496098
    0.27584301
  0.78289566 0.56039591 0.44880647 0.58372359]
 [0.56363311 0.83184182 0.10245143 0.52748516 0.71059322
    0.19123904
  0.51533385 0.34409446 0.3596236  0.62867894]
 [0.51052702 0.40212988 0.0693162  0.1101496  0.06689001
    0.66558757
  0.48448209 0.95160391 0.42766452 0.45951457]
 [0.90618675 0.4713798  0.04319986 0.58465643 0.42153666
    0.84645518
  0.60432124 0.33069549 0.26819627 0.127129  ]
 [0.78395823 0.24334728 0.41075852 0.17349541 0.09710229
    0.24446243
  0.35735395 0.14182983 0.39341188 0.91346758]
 [0.14628826 0.26338406 0.92524764 0.73148944 0.88581738
    0.47768863
  0.27251026 0.19600422 0.97583938 0.50112446]
 [0.62766423 0.25144896 0.69460853 0.68137878 0.02884555
    0.60276952
  0.78112115 0.63788289 0.64406786 0.45617689]]
```

```
array([[0.81632631, 0.89816275, 0.12130491, 0.4171127 ,
    0.81342409,
     0.97502141, 0.98536459, 0.98511233, 0.68830589,
         0.58976414],
    [0.02713838, 0.39218547, 0.25160237, 0.08229395,
        0.30770868,
     0.85632565, 0.44919185, 0.16286998, 0.33010748,
         0.24163147],
    [0.81929727, 0.05795412, 0.8925537 , 0.85529752,
        0.20782781,
     0.68019862, 0.17732165, 0.69354023, 0.66100412,
         0.94989246],
    [0.74413864, 0.57979159, 0.61355802, 0.52906865,
        0.56496098,
     0.27584301, 0.78289566, 0.56039591, 0.44880647,
         0.58372359],
    [0.56363311, 0.83184182, 0.10245143, 0.52748516,
        0.71059322,
     0.19123904, 0.51533385, 0.34409446, 0.3596236 ,
         0.62867894],
    [0.51052702, 0.40212988, 0.0693162 , 0.1101496 ,
        0.06689001,
     0.66558757, 0.48448209, 0.95160391, 0.42766452,
         0.45951457],
    [0.90618675, 0.4713798 , 0.04319986, 0.58465643,
        0.42153666,
     0.84645518, 0.60432124, 0.33069549, 0.26819627, 0.127129
         ],
    [0.78395823, 0.24334728, 0.41075852, 0.17349541,
        0.09710229,
     0.24446243, 0.35735395, 0.14182983, 0.39341188,
         0.91346758],
```


Figure 1: png

```
[0.14628826, 0.26338406, 0.92524764, 0.73148944,
    0.88581738,
 0.47768863, 0.27251026, 0.19600422, 0.97583938,
     0.50112446],
[0.62766423, 0.25144896, 0.69460853, 0.68137878,
    0.02884555,
 0.60276952, 0.78112115, 0.63788289, 0.64406786,
     0.45617689]])
```

### 1.2.1 Image

```
from IPython.core.display import Image
Image(data="http://ipython.org/_static/IPy_header.png")
```

```
print('text')
```

```
text
```

```
%matplotlib inline
import matplotlib.pyplot as plt
import numpy as np
```

```
x = np.linspace(-10,10,100)
y = np.sin(x)
plt.plot(x,y)
plt.show()
```

### 1.2.2 Operator Highlighing Check

```
#This is a comment with an operation x @ y in it.
test = 5**9 + 2 - x@ y / (7 % 2) + True * 7
print(test)

a = set([1,2,3,4,5,6,7,8,9,0])
b = set([2,4,6,8,0])
a & b
```

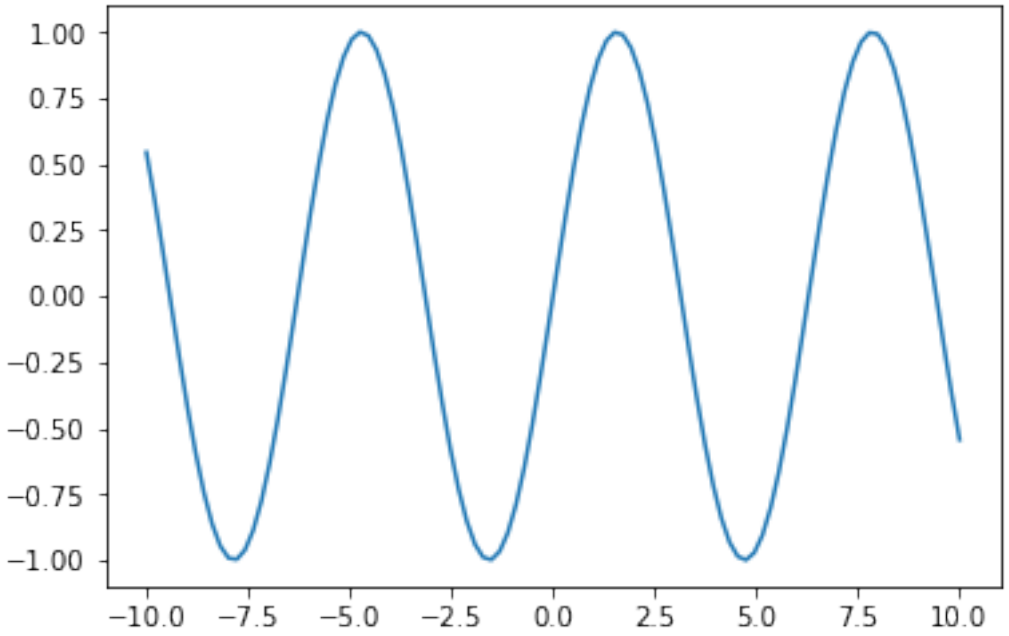


Figure 2: png

```
1953062.0590013973
```

```
\{0, 2, 4, 6, 8\}
```

#### 1.2.2.1 Pandas Output

Here we test the output of **Pandas**

First a *markdown* table:

| Column 1 | Column 2 |
|---|---|
| 1 | 3 |
| ~~a~~ | b |
| 4 | & |

#### 1.2.2.2 Pandas

```
1  import pandas as pd
2  pd.DataFrame(np.random.randn(10,3))
```

0

1

2

0

0.297290

1.539159

1.114557

1

0.694196

0.065893

1.569488

4

0.996257

-0.496768

0.403920

5

-1.175934

-1.196331

1.638312

6

1.207399

0.499761

1.664465

7

-0.974228

0.858901

0.773683

8

-0.094266

3.127974

-0.958878

9

1.068012

0.750389

1.840306

#### 1.2.2.2.1 Sympy output

```
import sympy
from sympy.abc import x, n, m
sympy.init_printing()
theta = sympy.Symbol('theta')
phi = sympy.Symbol('phi')

sympy.simplify(sympy.Ynm(n,m,theta,phi).expand(func=True))
```

$$\frac{\sqrt{\frac{(2n+1)\Gamma(-m+n+1)}{\Gamma(m+n+1)}}e^{im\phi}P_n^{(m)}(\cos(\theta))}{2\sqrt{\pi}}$$

x + y as plain text.

$\frac{P_n^{(m)}(\cos(\theta))}{2\sqrt{\pi}}\sqrt{\frac{(-m+n)!}{(m+n)!}(2n+1)}e^{im\phi}$

#### 1.2.2.2.2 Line Length

```
1 3 5 7 9 12 15 18 21 24 27 30 33 36 39 42 45 48 51 54 57 60 63
    66 69 72 75 78 81 84 87 90 93 96 99 103
```