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.

LaTeX TeX

This is a test list:

1. item 1

- subitem 1
- subitem 2

2. item 2
3. item 3

Printed Using Python

In [1]:

```
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.
```

Pyout (and Text Wrapping)

In [2]:

```
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
```

Out [2]:

```
'\nAliquam blandit aliquet enim, eget scelerisque eros adipiscing quis. Nunc sed metus \nut lorem co
```

In [3]:

```
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.
```

In [4]:

```
import numpy as np

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

```
[[0.28568166 0.8747998  0.87645362 0.51011938 0.06167899 0.6253242
  0.21695898 0.35406203 0.76399062 0.38721428]
 [0.59226394 0.23033422 0.11576507 0.0131951  0.34366223 0.96629731
  0.2867491  0.95194302 0.60324146 0.55986092]
 [0.36955543 0.78864789 0.73933855 0.39474922 0.74616752 0.9144543
  0.88600249 0.42611302 0.49375306 0.4260594 ]
 [0.40550295 0.85035162 0.5525894  0.21827199 0.67949174 0.93909704
  0.0331135  0.27240638 0.39332899 0.19852766]
 [0.32876315 0.97305405 0.11060386 0.20685979 0.3897287  0.01538051
  0.44747911 0.99865014 0.89374066 0.5141975 ]
 [0.10450336 0.42284722 0.95628045 0.32792639 0.11370905 0.32150692
  0.28631773 0.58203321 0.21240863 0.87954985]
 [0.62257223 0.79092658 0.72718477 0.0039627  0.61581427 0.28007586
  0.4653752  0.24737437 0.97801711 0.31160009]
 [0.03592867 0.56885907 0.05229575 0.12322391 0.45236765 0.98892923
  0.15013782 0.81404334 0.71795481 0.60145161]
 [0.01582381 0.23420526 0.18574213 0.6497537  0.71730148 0.0068443
  0.32733317 0.81837686 0.58895758 0.37633478]
 [0.64226276 0.77550803 0.23729951 0.9287232  0.14250076 0.23955818
  0.70490581 0.84959453 0.46939408 0.01230405]]
```

Out [4]:

```
array([[0.28568166, 0.8747998 , 0.87645362, 0.51011938, 0.06167899,
        0.6253242 , 0.21695898, 0.35406203, 0.76399062, 0.38721428],
       [0.59226394, 0.23033422, 0.11576507, 0.0131951 , 0.34366223,
        0.96629731, 0.2867491 , 0.95194302, 0.60324146, 0.55986092],
       [0.36955543, 0.78864789, 0.73933855, 0.39474922, 0.74616752,
        0.9144543 , 0.88600249, 0.42611302, 0.49375306, 0.4260594 ],
       [0.40550295, 0.85035162, 0.5525894 , 0.21827199, 0.67949174,
        0.93909704, 0.0331135 , 0.27240638, 0.39332899, 0.19852766],
       [0.32876315, 0.97305405, 0.11060386, 0.20685979, 0.3897287 ,
        0.01538051, 0.44747911, 0.99865014, 0.89374066, 0.5141975 ],
       [0.10450336, 0.42284722, 0.95628045, 0.32792639, 0.11370905,
        0.32150692, 0.28631773, 0.58203321, 0.21240863, 0.87954985],
       [0.62257223, 0.79092658, 0.72718477, 0.0039627 , 0.61581427,
        0.28007586, 0.4653752 , 0.24737437, 0.97801711, 0.31160009],
       [0.03592867, 0.56885907, 0.05229575, 0.12322391, 0.45236765,
        0.98892923, 0.15013782, 0.81404334, 0.71795481, 0.60145161],
       [0.01582381, 0.23420526, 0.18574213, 0.6497537 , 0.71730148,
        0.0068443 , 0.32733317, 0.81837686, 0.58895758, 0.37633478],
       [0.64226276, 0.77550803, 0.23729951, 0.9287232 , 0.14250076,
        0.23955818, 0.70490581, 0.84959453, 0.46939408, 0.01230405]])
```

Image

In [5]:

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

Out [5]:

In [1231]:

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

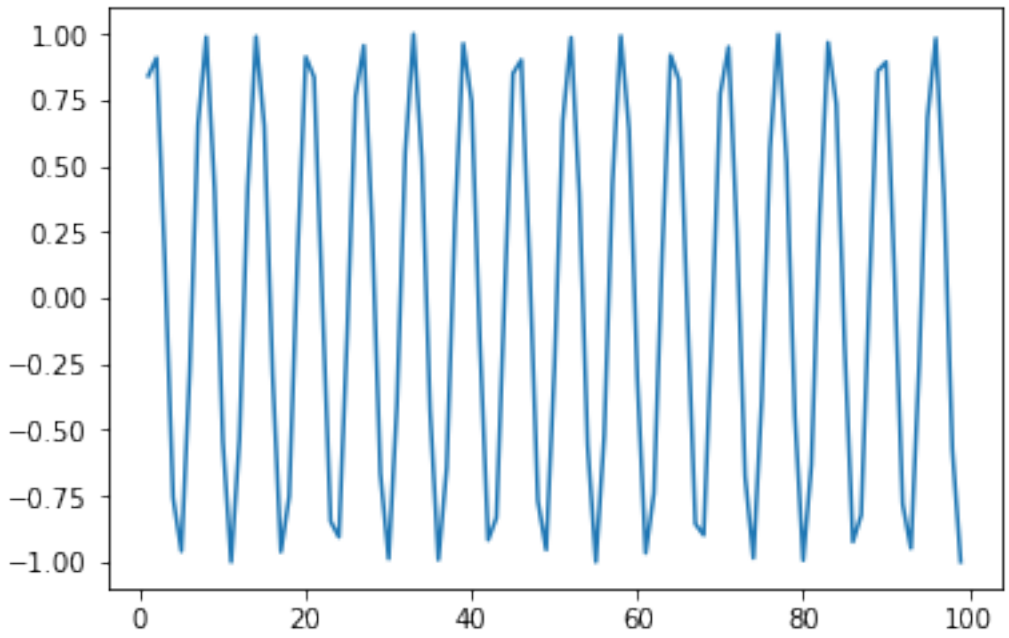
```
text
```

In [7]:

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

In [8]:

```
x = np.arange(1,100)
y = np.sin(x)
plt.plot(x,y)
plt.show()
```



Operator Highlighing Check

In [9]:

```
#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
```

```
1953188.1556827284
```

Out [9]:

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

Pandas Output

Here we test the output of **Pandas**

First a *markdown* table:

Column 1	Column 2
1	3

Column 1	Column 2
~~a~~	b
4	&

Pandas

In [10]:
```
import pandas as pd
pd.DataFrame(np.random.randn(10,3))
```

Out [10]:

	0	1	2
0	-1.565342	-0.260043	-1.427162
1	-0.267812	1.022688	-0.268030
2	0.104852	0.415075	0.958796
3	0.210758	-0.500437	-1.584460
4	-0.754263	-2.317940	-0.384726
5	-0.062044	-0.804551	0.914101
6	-2.193517	2.356933	0.542824
7	-1.246683	0.981807	-0.216905
8	-0.784741	-0.647911	0.134776
9	0.008086	1.652312	-0.468785

Sympy output

In [11]:
```
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))
```

Out [11]:

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

x + y as plain text.

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

Line Length

In []:
```
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
```