SlidesAPartirDUnNotebook

May 17, 2018

1 Générer des transparents à partir d'un notebook jupyter

Sage propose l'utilisation du notebook jupyter par la commande

sage -n jupyter

(cf l' introduction à Sage de Vincent Delecroix).

La session créée avec le notebook est sauvegardée dans un fichier json d'extension .ipynb (pour IPYthon NoteNook).

On présente ici la convertion d'un tel fichier en une présentation au format html (comme celle-ci) en utilisant jupyter enrichi de nbconvert.

1.1 Exemple d'utilisation

On peut activer une vue du notebook permettant de spécifier un comportement des cellules lors d'une présentation.

Cette activation se fait par le menu "View > Cell Toolbar > Slideshow".

On peut attribuer le "Slide Type" d'une cellule parmi Rien (par défaut), Slide, Sub-Slide, Fragment, Skip, Notes.

A partir de ces informations, il est possible via jupyter enrichi de l'utilitaire nbconvert de génerer une présentation au format html par la commande suivante dans un terminal:

jupyter nbconvert SlidesAPartirDUnNotebook.ipynb --to slides

ou

jupyter-nbconvert SlidesAPartirDUnNotebook.ipynb --to slides

Cela produit le fichier SlidesAPartirDUnNotebook.slides.html dont vous voyez normalement un affichage en plein écran.

1.2 Slide

Une cellule de type slide "Slide".

Une cellule de type slide "Rien" ("-") se rajoute au slide précédent.

Une cellule de type slide "Sub-Slide" se rajouter en dessous du slide précédent dans une colonne de slides.

(Changer de colonne de slide envoie sur le haut de la colonne de destination.)

Une cellule de type "Fragment" apparait dans le (sub-)slide courant.

Une cellule de type "Notes" ci-dessous dans le notebook n'apparait pas dans la présentation sauf si c'est la version pour 'orateur (ou peut-être d'autres sorties comme latex).

Note: Penser à acheter deux bagettes.

Une cellule de type "Skip" comme ci-dessous n'apparait jamais en dehors du notebook.

Note: Penser à acheter de la confiture.

1.3 Installation

1.3.1 Jupyter

A partir de Ubuntu 17.04 il existe un package pour jupyter:

```
apt-get install jupyter-notebook jupyter-core python-ipykernel
```

Personnellement (Ubuntu 16.04), je suis passé par pip (conda est possible):

```
sudo pip install jupyter
```

1.3.2 nbconvert

Je résume http://nbconvert.readthedocs.io/en/latest/install.html .

```
pip install nbconvert
```

Pour avoir plus de conversions:

```
apt-get install pandoc
```

```
sudo apt-get install texlive-xetex
```

1.4 Possibilitées d'utilisation

1.4.1 Latex dans les cellules MarkDown

On peut utiliser latex dans les cellules: $e^{i\pi} = -1$.

$$\begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

1.4.2 Du code dans les cellules MarkDown

On peut afficher du code dans les cellules pour différents langages.
Exemple: Python

```
print('Hello World')
```

Exemple: C

```
int i = 0;
printf("Hello World\n");
```

1.4.3 Du texte HTML généré par du code.

Un code peut générer du html.

```
In [2]: from IPython.display import HTML

        HTML('Hello World')
```

```
Out[2]: <IPython.core.display.HTML object>
```

On peut afficher le résultat d'une execution de code au format html dans le résultat:

```
In [3]: def say_hello_world():
            return 'Hello World'
        HTML(say_hello_world())
```

```
Out[3]: <IPython.core.display.HTML object>
```

1.4.4 Du texte Latex généré par du code

```
In [4]: from IPython.display import display,Math

        Math(latex(Matrix(4)))
```

Out[4]:

$$\begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

Attention apparement seul le dernier affichage apparait sauf si on utilise la commande display.

```
In [5]: from IPython.display import display
        m2 = Math(latex(Matrix(2)))
        m3 = Math(latex(Matrix(3)))
        display(m2)
        display(m3)
```

$$\begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}$$

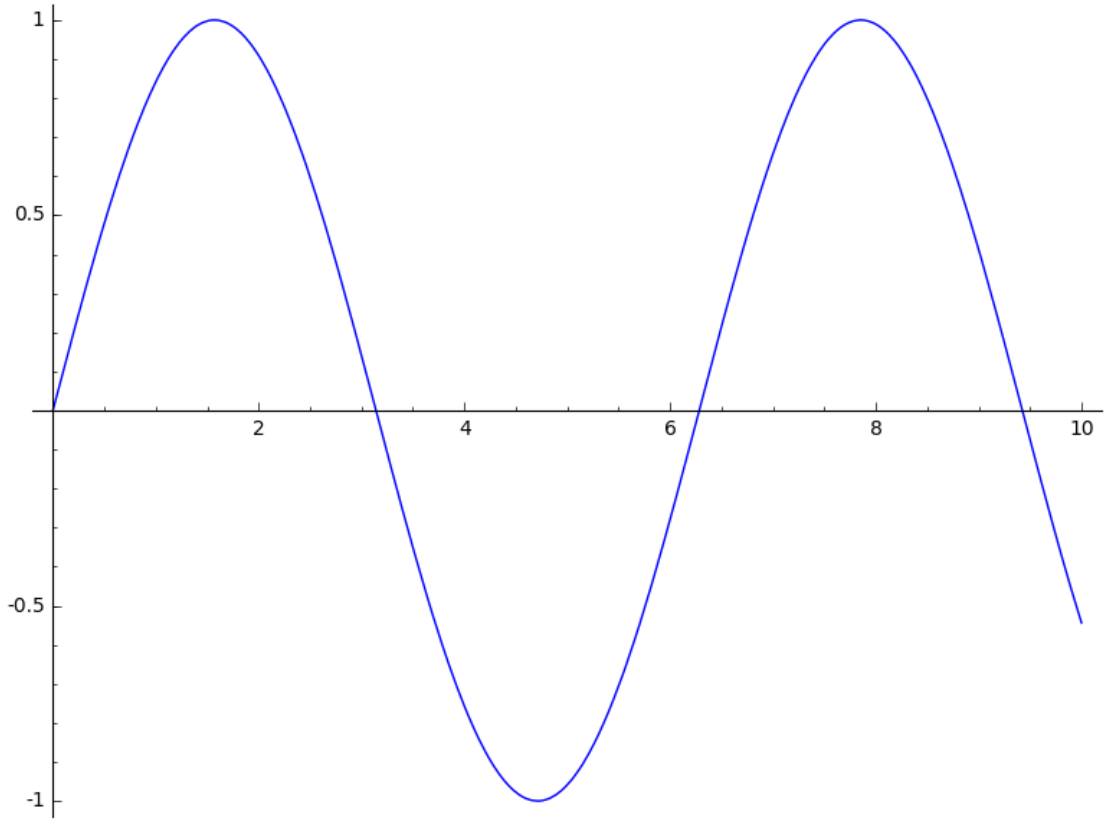
$$\begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

```
In [6]: from IPython.display import Latex

        display(Latex('$1+1=2$ du texte ${4 \choose 2}$'))
```

$1+1=2$ du texte $\binom{4}{2}$

```
In [14]: display(plot(sin,0,10))
```



Affichage de trace en latex

```
In [15]: for i in range(6):
             display(Math(latex(i)))
```

$$0$$

$$1$$

$$2$$

$$3$$

$$4$$

$$5$$

$$4$$

In [18]:
```
var('x')

for i in range(6):
    display(Math(latex((1-x^i)/(1-x))))
```

$$0$$

$$1$$

$$\frac{x^2 - 1}{x - 1}$$

$$\frac{x^3 - 1}{x - 1}$$

$$\frac{x^4 - 1}{x - 1}$$

$$\frac{x^5 - 1}{x - 1}$$

Il y a des possibilités de ne conserver que l'affichage de la sortie d'un calcul (mais mon utilisation n'est pas très stable).

In [ ]:
```
## Autres formats de sortie

```` sh
jupyter nbconvert SlidesAPartirDUnNotebook.ipynb --to latex
````


```` sh
jupyter nbconvert SlidesAPartirDUnNotebook.ipynb --to pdf
````

La liste des formats envisagés par la commande est

asciidoc, custom, html, latex, markdown, notebook, pdf, python, rst, script, slides.
```