

presentation_carnet_jupyter

February 14, 2020

1 Introduction aux Carnets Jupyter

Journées académiques de l'IREM de Lille 13-14 février 2020

Éric Wegrzynowski (eric.wegrzynowski@univ-lille.fr) département informatique, FST, Univ. Lille membre du groupe Informatique de l'IREM

De Wikipédia :

[Jupyter](#) est une application web utilisée pour programmer dans plus de 40 langages de programmation, dont: [Python](#), Julia, Ruby, R, ou encore Scala. Jupyter est une évolution du projet [IPython](#). Jupyter permet de réaliser des calepins ou [notebooks](#), c'est-à-dire des programmes contenant à la fois du texte en [markdown](#) et du code en Julia, Python, R... Ces notebooks sont utilisés en science des données pour explorer et analyser des données.

1.1 Structure d'un carnet

Un carnet Jupyter est constitué d'une succession de cellules.

Ces cellules peuvent être de trois natures :

- markdown : des cellules de texte au format markdown
- code : des cellules de code exécutable dans le langage choisi pour le carnet (ici c'est Python3)
- raw : des cellules brutes dans lesquelles ce qui est rédigé n'est pas interprété.

Format des fichiers : JSON (JavaScript Object Notation). Extension ipynb.

1.2 Du texte et du code

La fonction ci-dessous est une implantation en Python de la fonction qui à un entier n associe $\lfloor \sqrt{n} \rfloor$ sans utiliser la fonction `sqrt` du module `math`.

```
[1]: def racine_carree(n):  
    a, b = 1, n  
    rac = (a + b) // 2  
    carre_rac = rac * rac  
    while (carre_rac > n) or (carre_rac + 2*rac + 1 <= n):  
        if carre_rac + 2*rac + 1 <= n:  
            a = rac
```

```
    else:
        b = rac
        rac = (a + b) // 2
        carre_rac = rac * rac
    return rac
```

```
[2]: racine_carree(10**6)
```

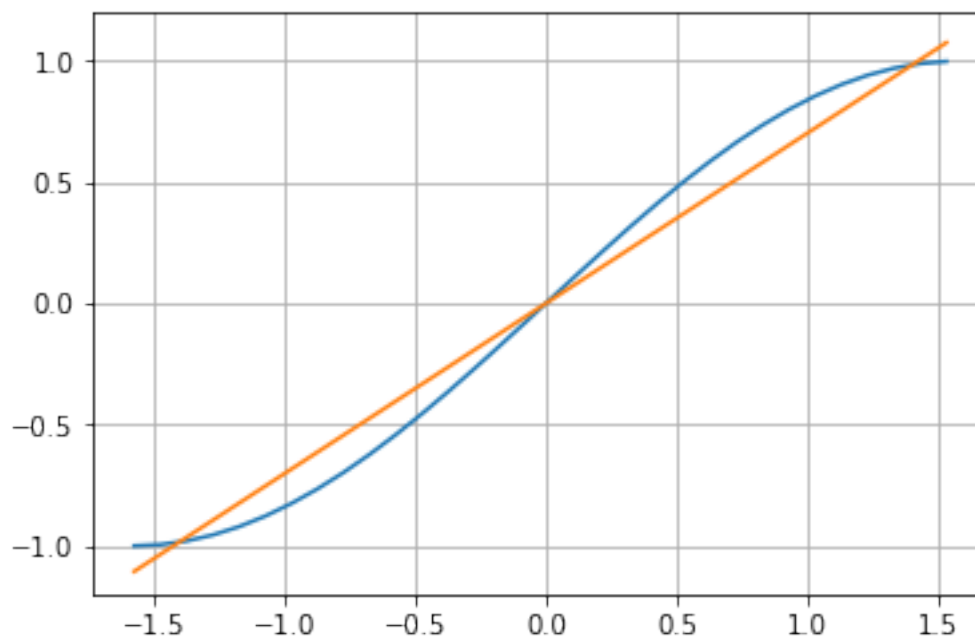
```
[2]: 1000
```

1.3 Des graphiques dans un carnet

```
[3]: %matplotlib inline
import pylab
from math import sin, pi
```

```
[4]: def dessiner_secante(h):
    absc = pylab.arange(-pi/2, pi/2, 0.1)
    pylab.plot(absc, [sin(x) for x in absc],
               absc, [sin(h)*x/h if h!=0 else x for x in absc])
    pylab.ylim(-1.2, 1.2)
    pylab.grid()
    pylab.show()
```

```
[5]: dessiner_secante(1.4)
```



1.4 Pour quelques widgets de plus

```
[6]: from ipywidgets import interact, FloatSlider
```

```
[7]: _ = interact(dessiner_secante,  
              h=FloatSlider(min=0, max=1.5, step=0.1, value=1.3))
```

```
interactive(children=(FloatSlider(value=1.3, description='h', max=1.5), Output()), _dom_classes=
```

1.5 Python Tutor dans un carnet

Trouvé sur https://www.codekodo.net/documentation/guides/Comment_ex%C3%A9cuter_un_programme_P

```
[8]: from metakernel import register_ipython_magics  
register_ipython_magics()
```

```
[9]: %%tutor  
liste = list(range(4))  
for i in range(4):  
    liste.pop()
```

```
<IPython.lib.display.IFrame at 0x7f7dac03b588>
```

1.6 Cartes géographiques

Nécessite la bibliothèque [folium](#).

```
[10]: import folium
```

```
[11]: center = (50.609, 3.139)  
carte = folium.Map(location=center, zoom_start=16)  
folium.Marker(location=center, popup="Le sanctuaire").add_to(carte)  
carte
```

```
[11]: <folium.folium.Map at 0x7f7da4c348d0>
```

1.7 Pas que du Python dans les cellules codes d'un carnet

1.7.1 Commandes systèmes

```
[12]: pwd
```

```
[12]: '/home/eric/Enseignement/IREM/AtelierJupyter2020'
```

```
[13]: ls
```

```
800px-Jupyter_logo.svg.png
```

```
Python@
```

```
SysComRF@
```

```
carnets_choisis.ipynb
```

```
resume.txt
```

```
Tris@
```

```

carnets.zip          resume.txt~  Untitled.ipynb
presentation_carnet_jupyter.ipynb  script.py
presentation_carnet_jupyter.md      Stegano@

```

1.7.2 Commandes magiques

Elles débutent par un %, comme dans %matplotlib ou %%tutor.

Il en existe plusieurs dizaines et on peut en créer de nouvelles. Cf <https://ipython.readthedocs.io/en/stable/interactive/magics.html#>

En voici quelques unes.

```
[14]: %%javascript
      alert("coucou")
```

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

```
[15]: %whos
```

Variable	Type	Data/Info
FloatSlider	MetaHasTraits	<class
'ipywidgets.widget<...>idget_float.FloatSlider'>		
NamespaceMagics	MetaHasTraits	<class
'IPython.core.magi<...>mespace.NamespaceMagics'>		
carte	Map	<folium.folium.Map
object at 0x7f7da4c348d0>		
center	tuple	n=2
dessiner_secante	function	<function
dessiner_secante at 0x7f7dae3bc158>		
folium	module	<module 'folium' from
'/u<...>ages/folium/__init__.py'>		
get_ipython	function	<function get_ipython at
0x7f7dde58c9d8>		
getsizeof	builtin_function_or_method	<built-in function
getsizeof>		
interact	_InteractFactory	
<ipywidgets.widgets.inter<...>object at 0x7f7dd43da978>		
json	module	<module 'json' from
'/usr<...>hon3.5/json/__init__.py'>		
np	module	<module 'numpy' from
'/us<...>kages/numpy/__init__.py'>		
pi	float	3.141592653589793
pylab	module	<module 'pylab' from
'/us<...>/dist-packages/pylab.py'>		
racine_carree	function	<function racine_carree
at 0x7f7db0e7cc80>		

```
register_ipython_magics    function                <function
register_ipytho<...>magics at 0x7f7dac06c620>
sin                       builtin_function_or_method    <built-in function sin>
var_dic_list              function                <function var_dic_list
at 0x7f7dd4372ea0>
```

```
[16]: %%writefile script.py
from math import sqrt
a = 2
print(sqrt(a))
```

Overwriting script.py

```
[17]: %run script.py
```

1.4142135623730951

<Figure size 432x288 with 0 Axes>

```
[18]: %quickref
```

1.8 Exportation

Les carnets peuvent être exportés dans différents formats

- HTML
- Markdown
- LaTeX
- PDF (nécessite LaTeX)
- scripts Python
- et bien d'autres

1.9 Extensions

- [toc](#)
- [jupyter](#)

1.10 Liens

- [site de Jupyter](#)
- [essayer les carnets en ligne depuis le site de Jupyter](#)
- [Ce carnet](#)