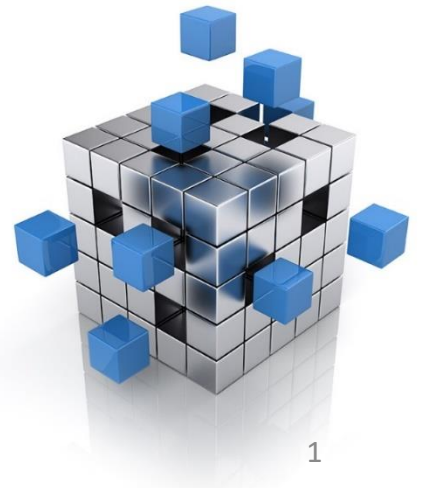


Premier pas

Introduction au traitement des données



Sommaire

- Vue d'ensemble des logiciel d'analyse de données
- Pourquoi Python ?
- Python comment ?
- Un serpent peut en cacher un autre
- Introduction à Jupyter Notebook
- Concrètement

Vue d'ensemble des logiciels

- Les logiciel Bureautique :
 - Excel
- Les logiciels généralistes à interface graphique :
 - SAS
 - SPSS
 - Tanagra
 - Statistica
 - RapidMiner

Vue d'ensemble des logiciels

- Les logiciels généraliste en ligne de commande :
 - R
 - Python
- Les logiciels spécialisés :
 - Requêtage SQL : Oracle SQL Developer, PL/SQL Developer
 - Requêtage Hadoop : Hive, Pig
 - Visualisation : D3JS

Pourquoi Python ?

Ses caractéristiques



- Langage de programmation généraliste avec un écosystème scientifique.
- Open source et donc en constante évolution.
- Forte communauté et donc facilité à trouver de l'aide sur les forums.
- Cadre unifié pour mettre en œuvre tout le processus de l'analyse données.
- Langage interprété et non compilable -> On peut traiter le programme ligne par ligne.

Python comment ?

- **Python en mode interactif**

Peut s'utiliser comme une calculatrice (très améliorée).

On lance python dans un shell.

Le prompteur `>>>` python apparaît.

Puis on "discute" avec python en tapant des commandes ou instructions.

Python comment ?

- **Les scripts**

Un programme est une séquence d'instructions.

Dans le cas d'un programme en langage Python, on parle souvent de **script Python**.

Un script se présente sous la forme d'un fichier texte avec l'extension **.py**

Les bibliothèques sous python

- Une des grandes forces du langage Python réside dans le nombre important de bibliothèques logicielles externes disponibles.
- Une bibliothèque est un ensemble de fonctions. Celles-ci sont regroupées et mises à disposition afin de pouvoir être utilisées sans avoir à les réécrire.
- Une bibliothèques doit être importé pour être utilisée

Les bibliothèques sous python

- NumPy

- Module pour la manipulation de matrices, tableaux multidimensionnels et fonctions mathématiques opérant sur ces tableaux.

- SciPy

- Module pour l'optimisation, l'algèbre linéaire, les statistiques, traitement d'image, ...

- Pandas

- Module permettant la manipulation et l'analyse de données, notamment des données numériques et des série temporelle -> DataFrame, Panels, ...

Les bibliothèques sous python

- Matplotlib

- Permet de tracer et visualiser des données sous forme de graph

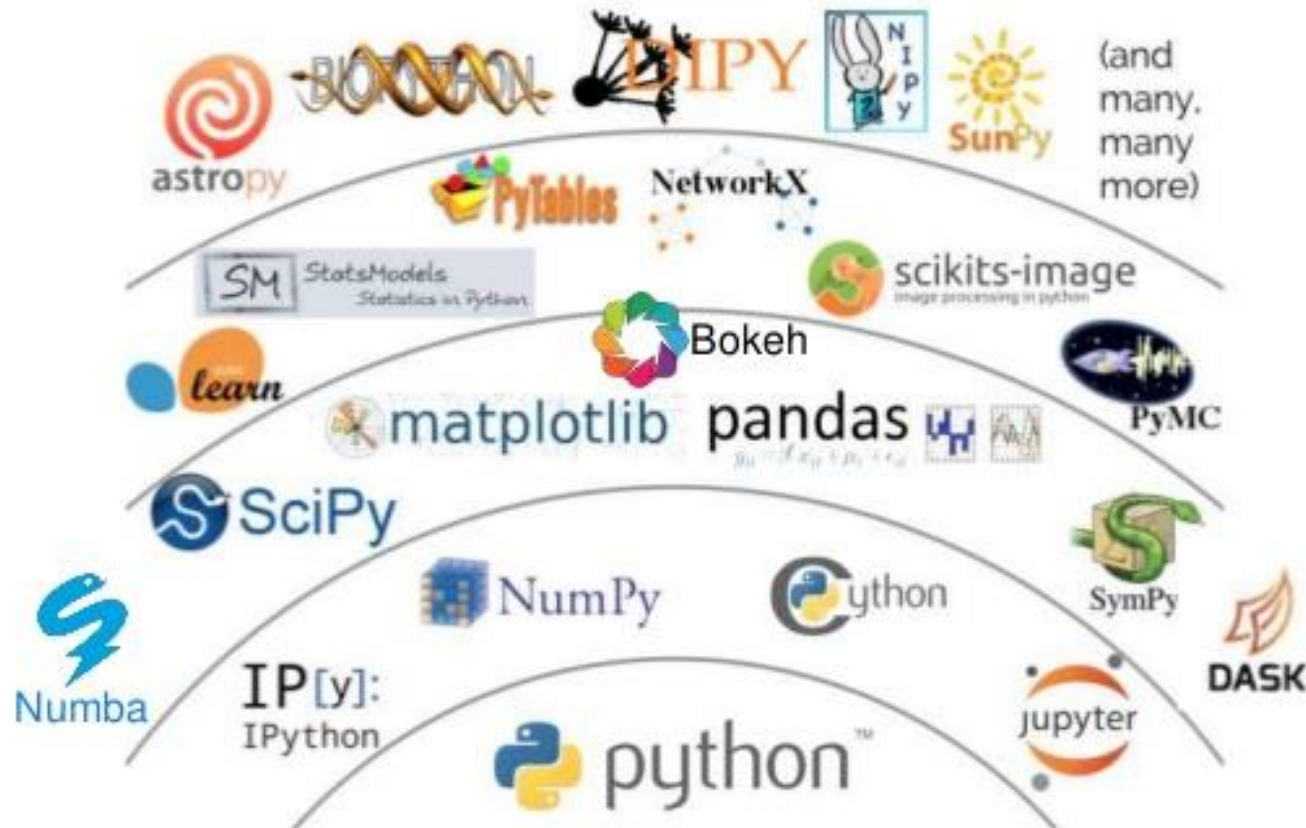
- Scikit-learn

- Destiné à l'apprentissage automatique -> Forêts aléatoires, régression logistiques, classement, ...

- Tensorflow

- Apprentissage automatique pour intelligence Artificielle

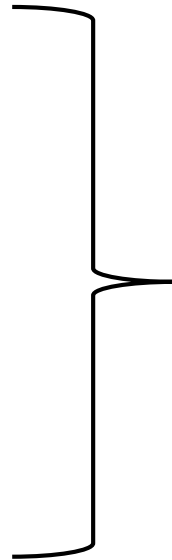
Un serpent peut en cacher un autre



Pourquoi python ?

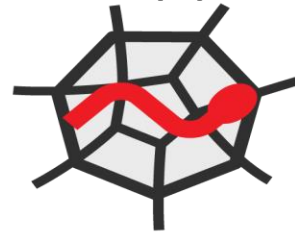
Les librairie sous python

- SciPy
- NumPy
- Pandas
- Scikit-learn
- Tensorflow



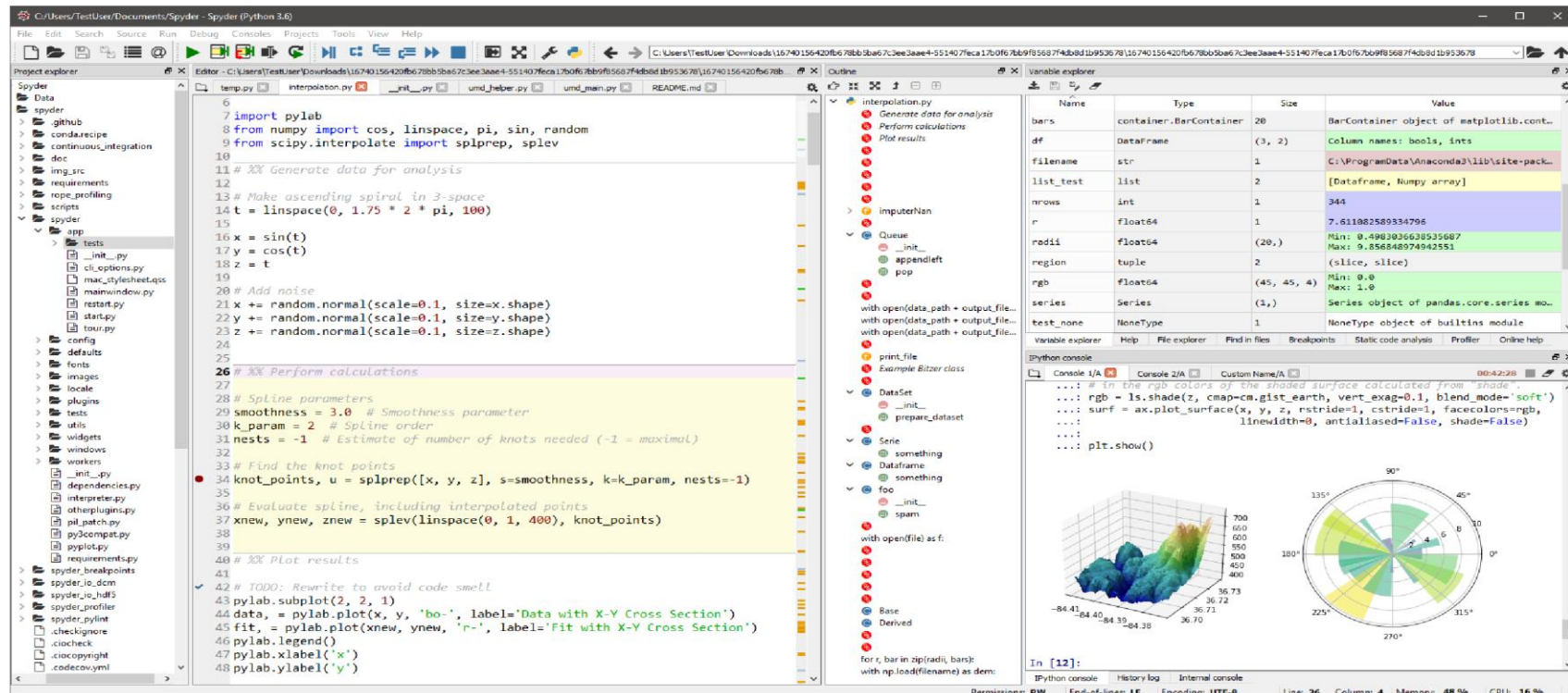
Pourquoi Python ?

- Interface de développement :



SPYDER

The Scientific Python Development Environment



Introduction Jupyter Notebook



Application web qui permet aux utilisateurs d'interagir avec le code et de transformer le navigateur web en terminal interactif et en logiciel de traitement de texte en même temps.

- Programmer dans le navigateur
- Le code, les instructions et la sortie son afficher en ligne
- Utile pour écrire un code qui raconte une histoire
- Utile pour réaliser des compte rendue
- Utiliser par les scientifiques et les chercheurs

Introduction Jupyter Notebook



```
[I 00:32:22.199 NotebookApp] JupyterLab extension loaded from C:\ProgramData\Anaconda3\lib\site-packages\jupyterlab
[I 00:32:22.199 NotebookApp] JupyterLab application directory is C:\ProgramData\Anaconda3\share\jupyter\lab
[I 00:32:22.246 NotebookApp] Serving notebooks from local directory: C:\Users\rafik
[I 00:32:22.246 NotebookApp] The Jupyter Notebook is running at:
[I 00:32:22.246 NotebookApp] http://localhost:8888/?token=ab548b603e138ecfe973f70ab3ed99e7613e929b9049954c
[I 00:32:22.246 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).
[C 00:32:22.315 NotebookApp]

Copy/paste this URL into your browser when you connect for the first time,
to login with a token:
http://localhost:8888/?token=ab548b603e138ecfe973f70ab3ed99e7613e929b9049954c
[I 00:32:22.562 NotebookApp] Accepting one-time-token-authenticated connection from ::1
```

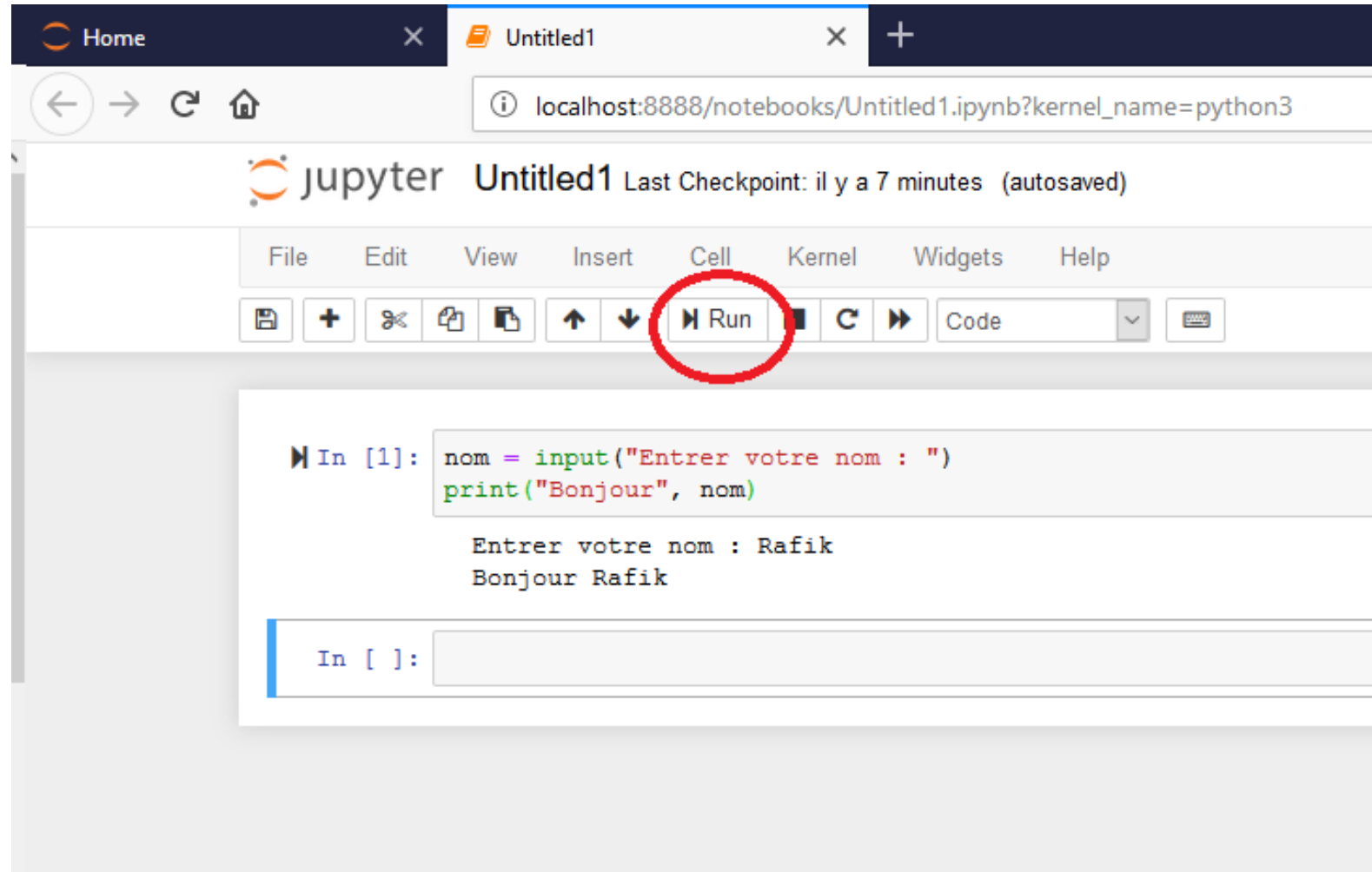
Introduction Jupyter Notebook



The screenshot shows the Jupyter Notebook web interface in a browser window. The address bar shows 'localhost:8888/tree'. The interface has tabs for 'Files', 'Running', and 'Clusters'. Below the tabs, there's a message 'Select items to perform actions on them.' and buttons for 'Upload', 'New', and a refresh icon. The 'New' button is circled in red. Below this is a table listing files and folders in the current directory.

	Name	Last Modified	File size
<input type="checkbox"/>	/		
<input type="checkbox"/>	3D Objects	il y a 2 ans	
<input type="checkbox"/>	Contacts	il y a 2 heures	
<input type="checkbox"/>	Desktop	il y a 2 heures	
<input type="checkbox"/>	Documents	il y a 2 heures	
<input type="checkbox"/>	Downloads	il y a 2 heures	
<input type="checkbox"/>	Favorites	il y a 2 heures	
<input type="checkbox"/>	Links	il y a 2 heures	
<input type="checkbox"/>	Music	il y a 2 heures	
<input type="checkbox"/>	OneDrive	il y a 2 heures	
<input type="checkbox"/>	Pictures	il y a 2 heures	
<input type="checkbox"/>	Roaming	il y a 2 ans	
<input type="checkbox"/>	Saved Games	il y a 2 heures	
<input type="checkbox"/>	Searches	il y a 2 heures	
<input type="checkbox"/>	Videos	il y a 2 heures	
<input type="checkbox"/>	Untitled.ipynb	il y a 5 heures	555 B

Introduction Jupyter Notebook



Raccourci clavier pour
Run

Shift



Entrée

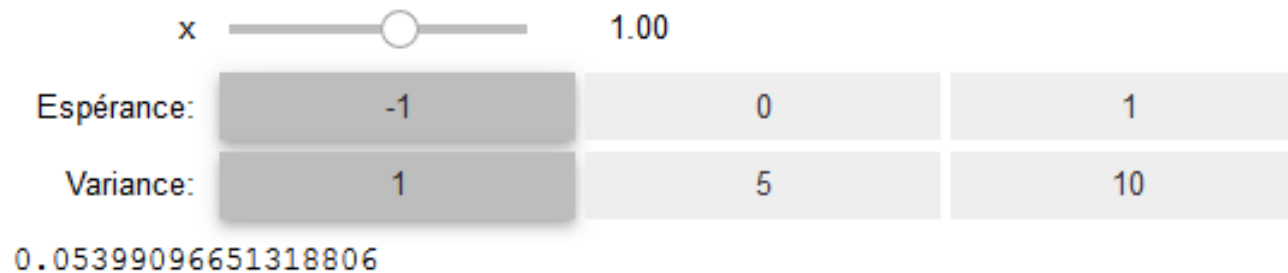
Introduction Jupyter Notebook

Générer un curseur

In [36]:

```
from ipywidgets import ToggleButtons, interact
import scipy.stats

res=interact(scipy.stats.norm.pdf, x=FloatSlider(min=-8, max=8, step=1, value=1),
             loc=ToggleButtons(description='Espérance:', options=[-1, 0, 1]),
             scale=ToggleButtons(description='Variance:', options=[1, 5, 10]));
```



Introduction Jupyter Notebook

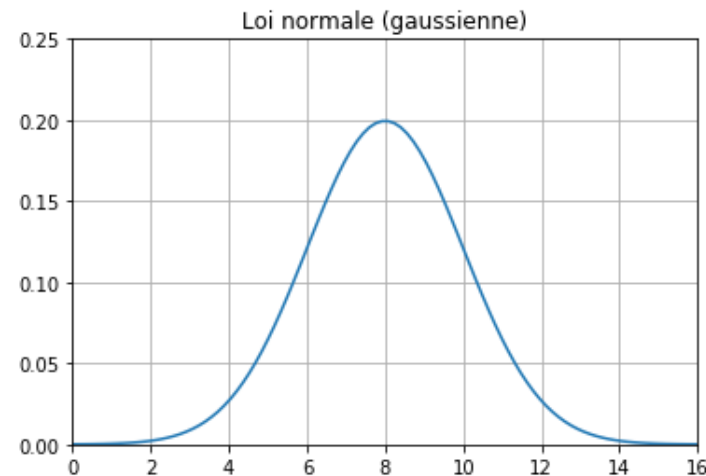
Tracer un graph

```
In [28]: import matplotlib.pyplot as plt
import scipy.stats
import numpy as np

x = np.linspace(0, 16.0, 100)

plt.plot(x, scipy.stats.norm.pdf(x, 8, 2))

plt.grid()
plt.xlim(0, 16)
plt.ylim(0, 0.25)
plt.title('Loi normale (gaussienne)')
plt.savefig("normal_distribution.png")
plt.show()
```



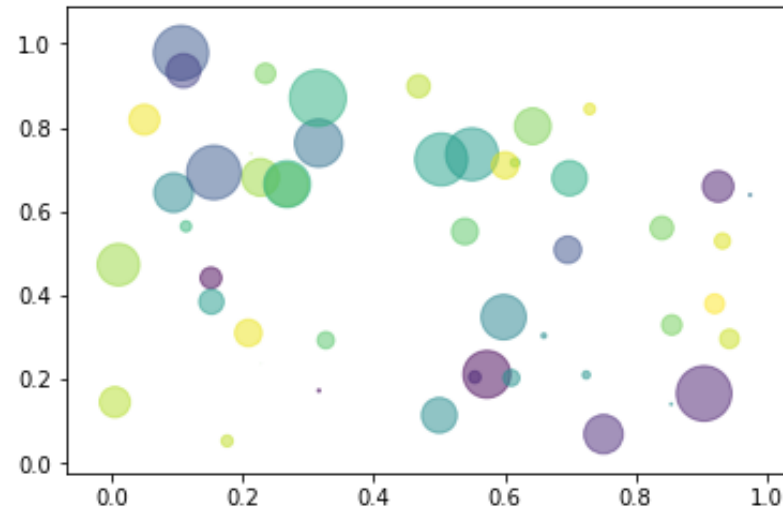
Introduction Jupyter Notebook

Tracer un graph

```
In [50]: import numpy as np
import matplotlib.pyplot as plt

N=50
x=np.random.rand(N)
y=np.random.rand(N)
colors=np.random.rand(N)
area=np.pi * (15 * np.random.rand(N))**2

plt.scatter(x, y, s=area, c=colors, alpha=0.5)
plt.show()
```



Introduction Jupyter Notebook

Créer un DataFrame

```
In [55]: import pandas as pd
import numpy as np
df = pd.DataFrame(np.random.randn(10, 5))
df.head()
```

Out[55]:

	0	1	2	3	4
0	0.101293	-1.158496	0.339858	-0.617471	-1.256263
1	-0.031837	0.210922	0.788922	1.415832	0.170070
2	1.025123	-1.475479	0.882929	0.030915	0.401047
3	0.311811	-0.376678	0.117401	-0.275404	0.660441
4	0.838328	-1.531435	-0.314659	0.910114	-1.912346

Introduction Jupyter Notebook

Les fonctions Magic

2 types de fonctions Magic :

- Les fonctions Magic, préfixées par un seul symbole %, agissent uniquement sur la ligne sur la quel elles se trouvent
- Les fonctions Magic, préfixées de 2 symboles %% agissent sur l'ensemble de la cellule.

Introduction Jupyter Notebook

Les fonctions Magic

```
In [41]: %pwd
```

```
Out[41]: 'C:\\Users\\rafik'
```

```
In [42]: %timeit range(1000)
```

```
233 ns ± 7.92 ns per loop (mean ± std. dev. of 7 runs, 1000000 loops each)
```

```
In [43]: %%timeit  
range(10)  
range(100)  
range(1000)
```

```
587 ns ± 16.9 ns per loop (mean ± std. dev. of 7 runs, 1000000 loops each)
```

Introduction Jupyter Notebook

Les fonctions Magic

In [24]: `%lsmagic`

Out[24]: Available line magics:

```
%alias %alias_magic %autocall %automagic %autosave %bookmark %cd %clear %cls %colors %config %connect_info %copy %ddir %debug %dhist %dirs %doctest_mode %echo %ed %edit %env %gui %hist %history %killbgscripts %ldir %less %load %load_ext %loadpy %logout %logon %logstart %logstate %logstop %ls %lsmagic %macro %magic %matplotlib %mkdir %more %notebook %page %pastebin %pdb %pdef %pdoc %pfile %pinfo %pinfo2 %popd %pprint %precision %profile %prun %psearch %psource %pushd %pwd %pycat %pylab %qtconsole %quickref %recall %rehashx %reload_ext %ren %rep %rerun %reset %reset_selective %rmdir %run %save %sc %set_env %store %sx %system %tb %time %timeit %unalias %unload_ext %who %who_ls %whos %xdel %xmode
```

Available cell magics:

```
%%! %%HTML %%SVG %%bash %%capture %%cmd %%debug %%file %%html %%javascript %%js %%latex %%markdown %%perl %%python %%pypy %%python %%python2 %%python3 %%ruby %%script %%sh %%svg %%sx %%system %%time %%timeit %%writefile
```

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

Introduction Jupyter Notebook

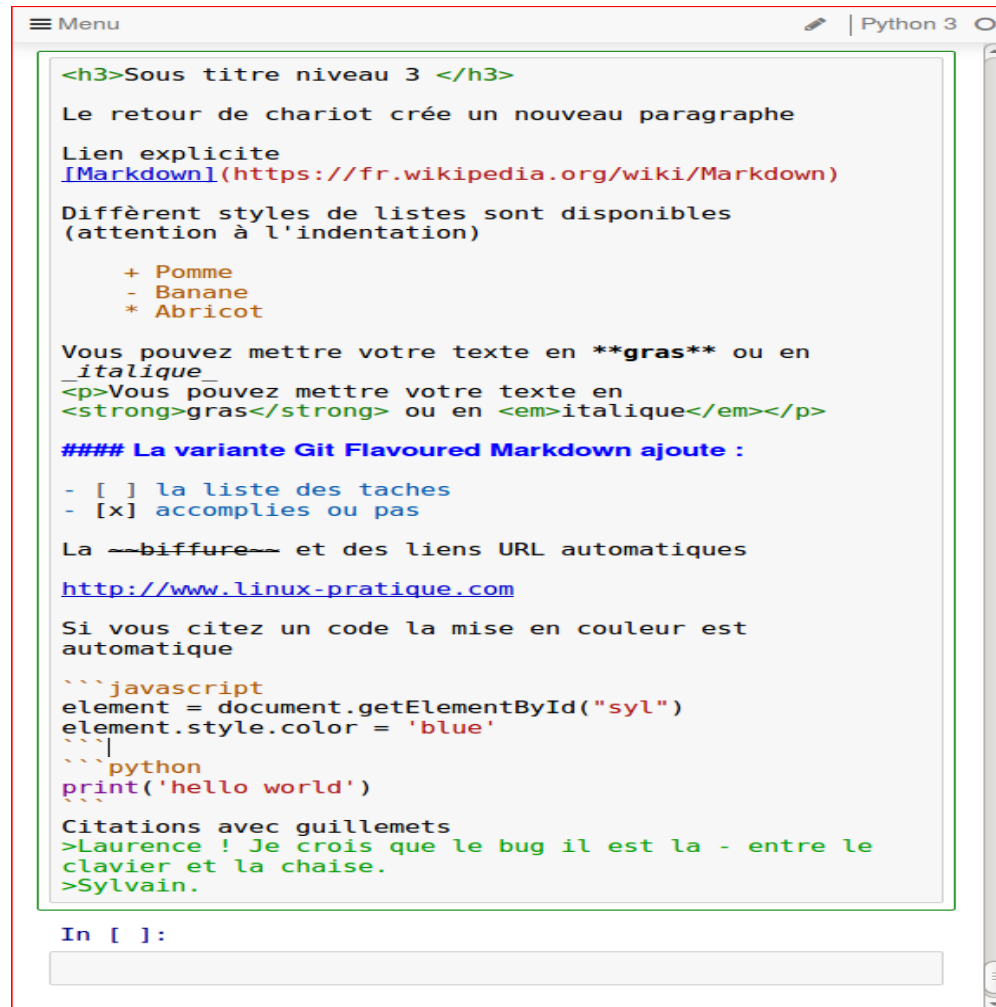
Programmation et rédaction

- Possibilité de structurer le texte en utilisant des balise HTML/CSS, Markdown, ou LaTeX



Introduction Jupyter Notebook

Programmation et rédaction



The screenshot shows a Jupyter Notebook editor window with a menu bar and a toolbar. The main area contains raw HTML and Markdown code. The code includes a heading, a paragraph, a link, a list, bold and italic text, a section header, a list, a strikethrough, a URL, a code block, and a citation.

```
<h3>Sous titre niveau 3 </h3>

Le retour de chariot crée un nouveau paragraphe

Lien explicite
[Markdown](https://fr.wikipedia.org/wiki/Markdown)

Différent styles de listes sont disponibles
(attention à l'indentation)

+ Pomme
- Banane
* Abricot

Vous pouvez mettre votre texte en gras ou en
italique
<p>Vous pouvez mettre votre texte en
<strong>gras</strong> ou en <em>italique</em></p>

#### La variante Git Flavoured Markdown ajoute :

- [ ] la liste des tâches
- [x] accomplies ou pas

La biffure et des liens URL automatiques

http://www.linux-pratique.com

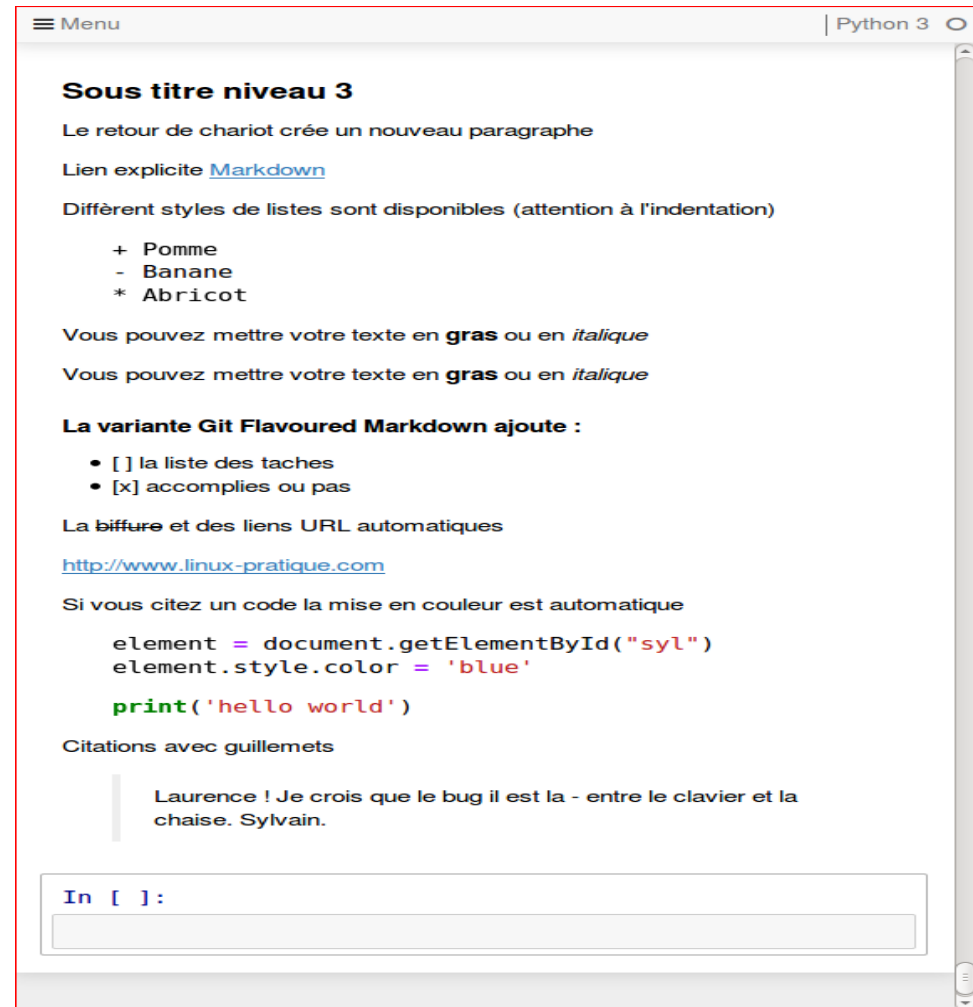
Si vous citez un code la mise en couleur est
automatique

```javascript
element = document.getElementById("sylv")
element.style.color = 'blue'
```

```python
print('hello world')
```

Citations avec guillemets
>Laurence ! Je crois que le bug il est la - entre le
clavier et la chaise.
>Sylvain.

In [ ]:
```



The screenshot shows the same Jupyter Notebook editor window, but the code is rendered into a readable format. The heading is bold, the link is blue, the list is bulleted, the bold and italic text are formatted, the section header is bold, the list is bulleted, the strikethrough is shown, the URL is blue, the code block is preformatted, and the citation is shown with a vertical line.

Sous titre niveau 3

Le retour de chariot crée un nouveau paragraphe

Lien explicite [Markdown](https://fr.wikipedia.org/wiki/Markdown)

Différent styles de listes sont disponibles (attention à l'indentation)

- + Pomme
- Banane
- * Abricot

Vous pouvez mettre votre texte en **gras** ou en *italique*

Vous pouvez mettre votre texte en **gras** ou en *italique*

La variante Git Flavoured Markdown ajoute :

- [] la liste des tâches
- [x] accomplies ou pas

La ~~biffure~~ et des liens URL automatiques

<http://www.linux-pratique.com>

Si vous citez un code la mise en couleur est automatique

```
element = document.getElementById("sylv")
element.style.color = 'blue'

print('hello world')
```

Citations avec guillemets

Laurence ! Je crois que le bug il est la - entre le clavier et la chaise. Sylvain.

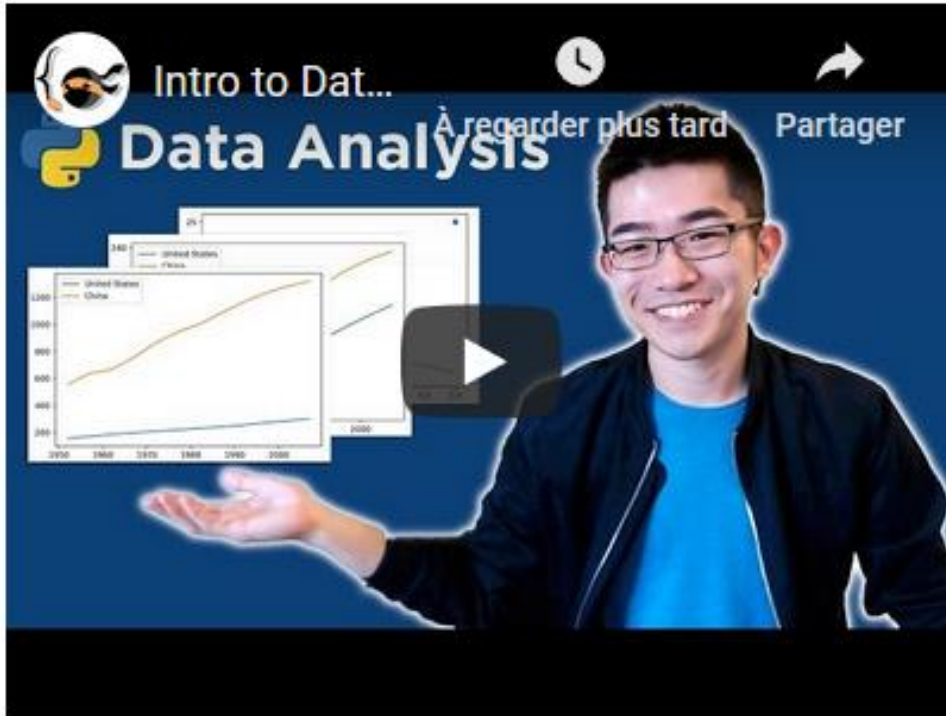
In []:

Introduction Jupyter Notebook

Programmation et rédaction

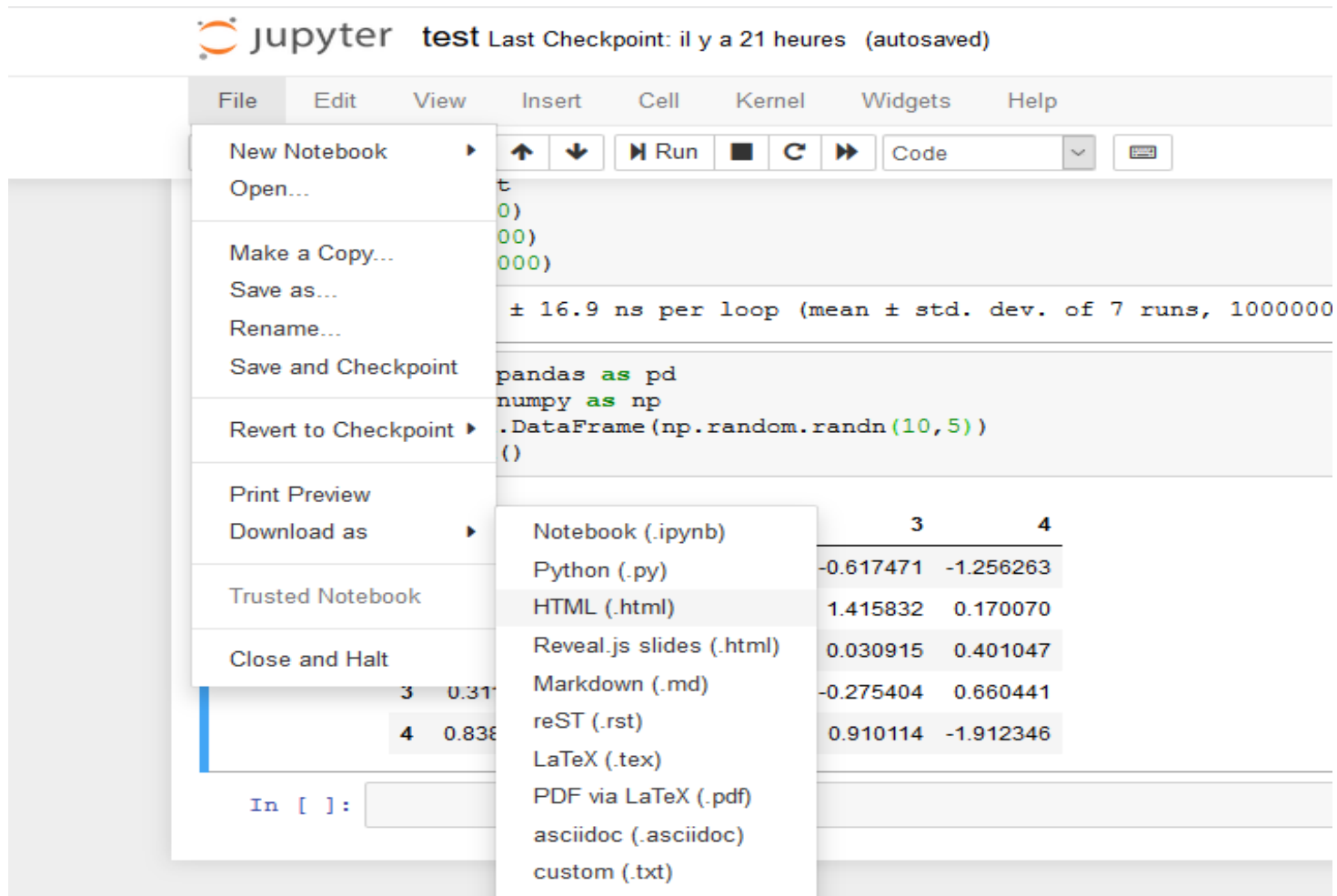
```
► In [26]: from IPython.display import YouTubeVideo  
YouTubeVideo('a9UrKTVeeZA')
```

Out[26]:



Introduction Jupyter Notebook

Programmation et rédaction -> Exporter son document



En pratique ça marche comment

- **Variable:** un emplacement mémoire que l'on peut référencer par un nom. Sert à stocker une information.
- Autrement dit : une boîte dans laquelle on peut mettre une valeur. ensuite, on peut réutiliser cette valeur (sans la connaître) en écrivant le nom de la variable.

$x = 3$

nom = 'Rafik'

En pratique ça marche comment

Python distingue différentes sortes de variables suivant le type de valeur qu'elles mémorisent :

- un nombre entier
- une chaîne de caractères
- un nombre à virgule
- des valeurs complexes de type liste

On passe à la pratique

Télécharger et installer Anaconda

On passe à la pratique

En utilisant la bibliothèque math :

1. Trouver les solutions de $3x^2 - 7x = 23$
2. Écrire un programme qui, à partir de la saisie d'un rayon et d'une hauteur, calcule le volume d'un cône droit.
*(Utiliser la fonction **input()** pour saisir les valeurs et **print()** pour afficher le résultat).*