



**Rencontre CDO Alliance**

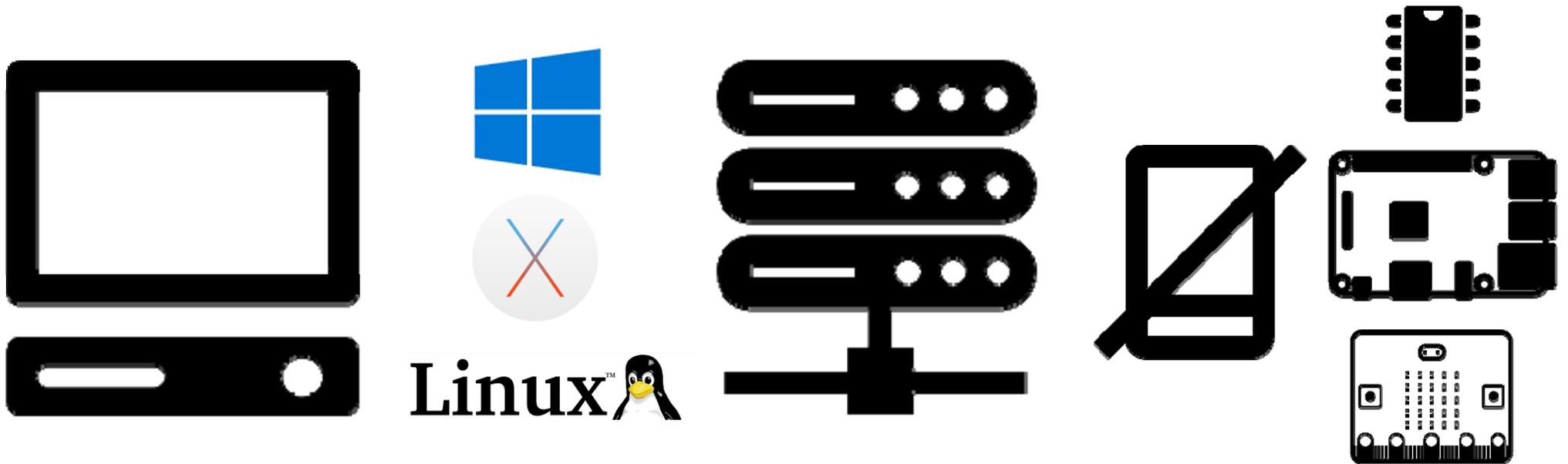
**Sébastien FREAL-SAISON**

**04/07/2018 – EDF SmartLab La Défense**



**python**™

## Sur quelles plateformes trouver python ?



Aujourd'hui, principale distribution python : Anaconda Python (plusieurs millions d'utilisateurs)

# Quelle est la popularité de Python ?

## Google PYPL index, mars 2018

Worldwide, Mars 2018 comparé à mars 2017:

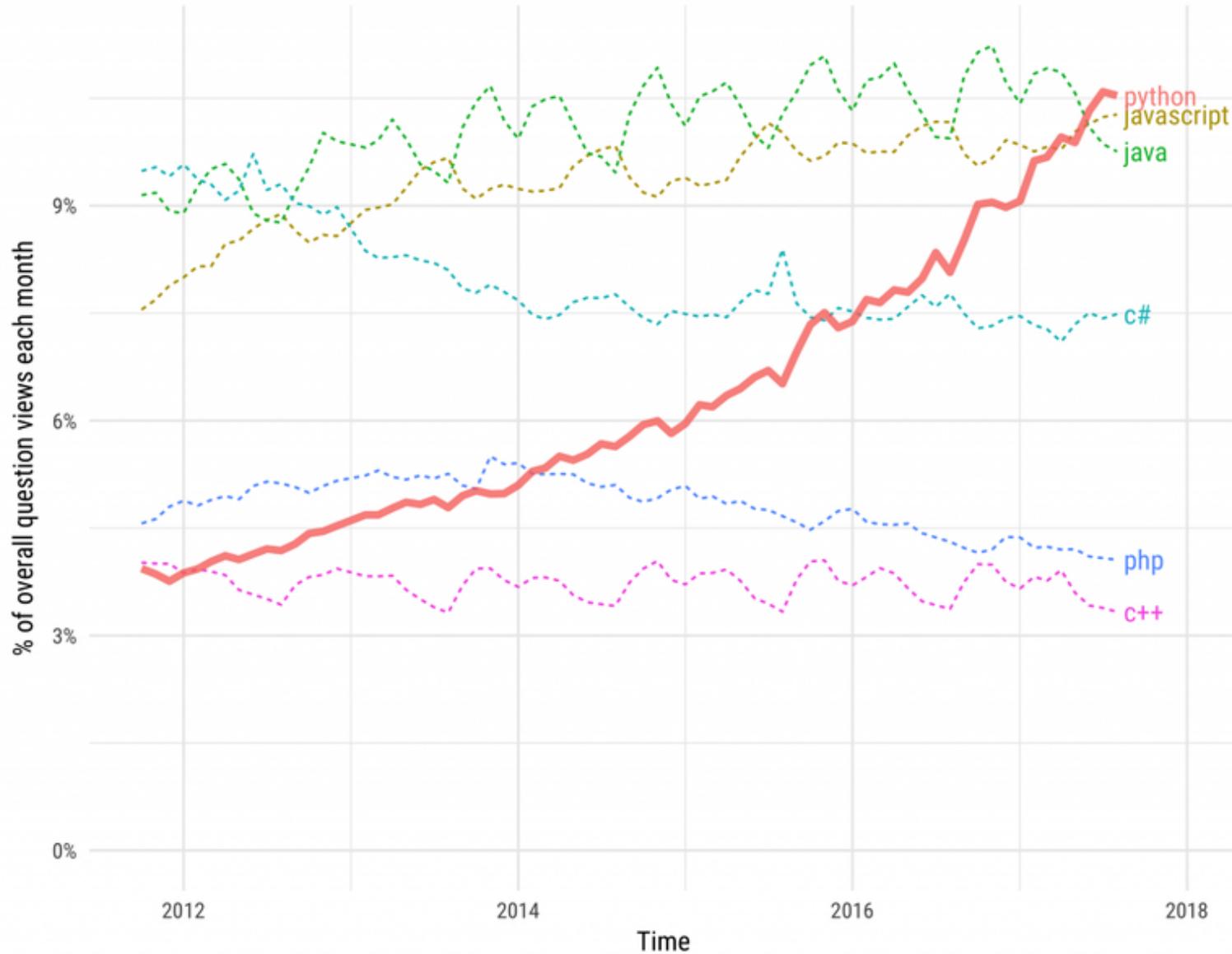
Popularity of Programming Language :

Rank	Change	Langage	Part	Trend
1		Java	22.7 %	-0.8 %
2		Python	21.69 %	+5.4 %
3	↑↑	Javascript	8.53 %	+0.3 %
4	↓	PHP	8.33 %	-1.7 %
5	↓	C#	7.99 %	-0.7 %
6		C	6.42 %	-1.3 %
7	↑	R	4.23 %	+0.4 %
8	↓	Objective-C	3.81 %	-1.1 %
9		Swift	3.0 %	-0.6 %
10		Matlab	2.39 %	-0.4 %

Quelles sont les fréquences comparées des requêtes sur le moteur de recherche pour des tutoriaux de langages de programmation ?

## Growth of major programming languages

Based on Stack Overflow question views in World Bank high-income countries



Le plus important site pour développeurs, référence de partage de connaissances entre développeurs.

16 000 000 questions.

## Quels sont les avantages de python selon les utilisateurs ?

D'après un sondage d'utilisateurs, les **particularités techniques** ne sont pas mis au premier plan.

- ① Impacts des choix de design du langage sur les développeurs
- ② Le rôle de la communauté de développeurs

- 1 Great libraries
- 2 Readable code
- 3 Beautiful code
- 4 Rapid development
- 5 Large community
- 6 Open source
- 7 Elegant
- 8 Great community
- 9 Object oriented
- 10 Dynamic typing

```

177     default='Y',
178 )
179 )
180 global_scale_setting = FloatProperty(
181     name="Scale",
182     min=0.01, max=1000.0,
183     default=1.0,
184 )
185
186 def execute(self, context):
187
188     # get the folder
189     folder_path = (os.path.dirname(self.filepath))
190
191     # get objects selected in the viewport
192     viewport_selection = bpy.context.selected_objects
193
194     # get export objects
195     obj_export_list = viewport_selection
196     if self.use_selection_setting == False:
197         obj_export_list = [i for i in bpy.context.scene.objects]
198
199     # deselect all objects
200     bpy.ops.object.select_all(action='DESELECT')
201
202     for item in obj_export_list:
203         item.select = True
204         if item.type == 'MESH':
205             file_path = os.path.join(folder_path, "{}.obj".format(item.name))
206             bpy.ops.export_scene.obj(filepath=file_path, use_selection=True,
207                 axis_forward=self.axis_forward_setting,
208                 axis_up=self.axis_up_setting,
209                 use_animation=self.use_animation_setting,
210                 use_mesh_modifiers=self.use_mesh_modifiers_setting,
211                 use_edges=self.use_edges_setting,
212                 use_smooth_groups=self.use_smooth_groups_setting,
213                 use_smooth_groups_bitflags=self.use_smooth_groups_bitflags_setting,
214                 use_normals=self.use_normals_setting,
215                 use_uv=self.use_uv_setting,
216                 use_materials=self.use_materials_setting,

```

## Python, le langage de programmation agile

La communauté Python

Python, le langage de la donnée

Conclusion

# Syntaxe précise, concise :

## « Less Code is Better »

- Une tâche qui demande environ 20 lignes de code C ou 7 en Java peut souvent être codée en 1 ligne en Python.
  - Gain de temps sur le développement,
  - Gain de temps sur la maintenance,
  - Gain sur la qualité
  - Gain sur la sécurité

# Exemple de code

## Caractéristiques techniques

- langage multi paradigmes
- Typage dynamique
- Pas de sucre syntaxique.
- Blocs définis par des espaces/Tabs  
Conséquence : code aéré, lisible
- Extensible, Flexible
- agile plutôt que rapide (vs C/Java)  
Met la vitesse de développement en avant de la rapidité d'exécution

### Fonction factorielle en Python

```
def factorielle(n):  
    if n < 2:  
        return 1  
    else:  
        return n * factorielle(n - 1)
```

### Fonction factorielle en C

```
int factorielle(int n) {  
    if (n < 2) {  
        return 1;  
    } else {  
        return n * factorielle(n - 1);  
    }  
}
```



*« Python is fast enough for our site and allows us to produce maintainable features in record times, with a minimum of developers. »*

**Cuong Do** – *Software Architect, YouTube.com*

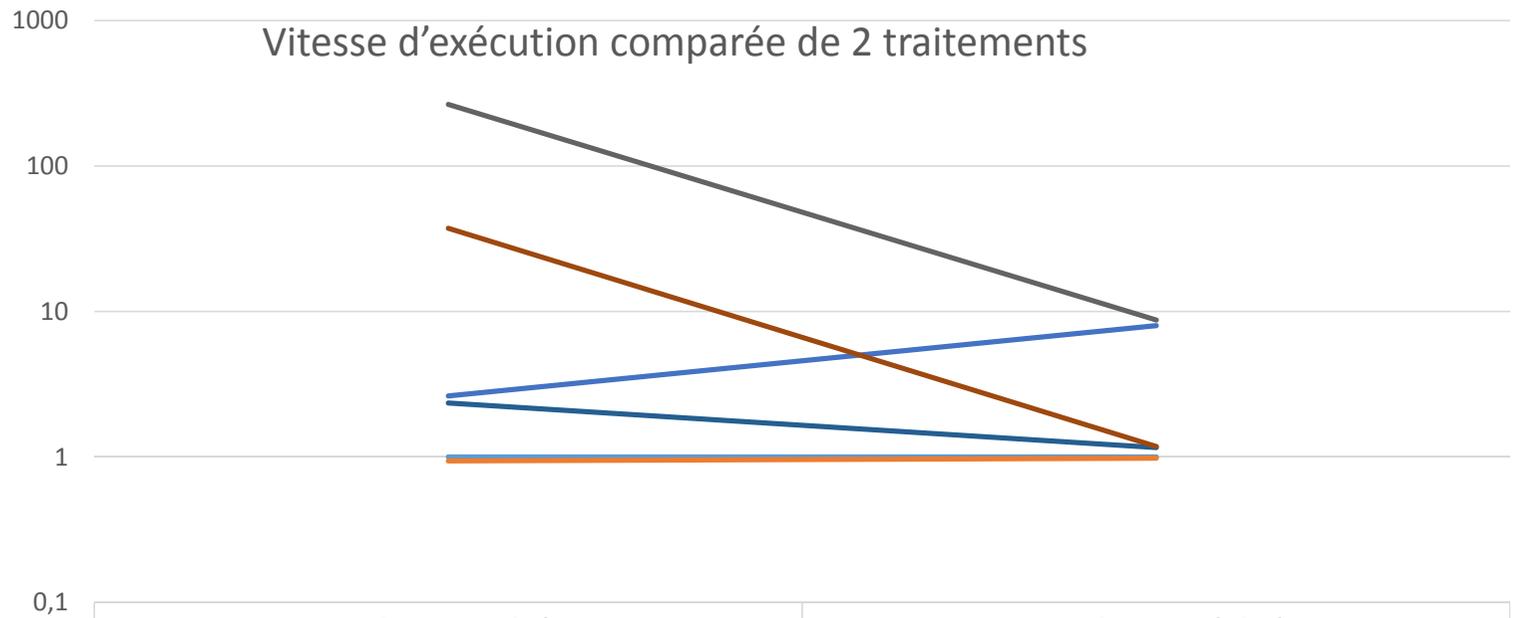


**Alex Martelli, Google since 2005, 'uber tech Lead'**

It all got started, I believe, because the very earliest Googlers (Sergey, Larry, Craig, ...) made a good engineering decision: "Python where we can, C++ where we must" -- they used (a subset of) C++ for the parts of the software stack where very low latency and/or tight control of memory were crucial, and Python, allowing more rapid delivery and maintenance of programs, for other parts.

These days hardware is cheap, meataware is expensive. I know that I can write a lot more, faster, better AND clearer in Python than I can in other languages, simply because the batteries tend to be included. Sure, it may be slower but are you really going to notice the difference between .05 seconds and .0005 seconds?

# Benchmark



	recursion_quicksort	matrix_multiply
C gcc 4.8.5	1	1
Java 1.8.0_15	2,63	7,99
Matlab R 2018a	2,35	1,16
R 3.3.1	264,98	8,74
Python 3.6.3	37,32	1,18
Julia 0.6.2	0,94	0,98

# Python, un langage 'ciment'

Pour améliorer la performance, python intègre dans des extensions des fonctionnalités développées en C.

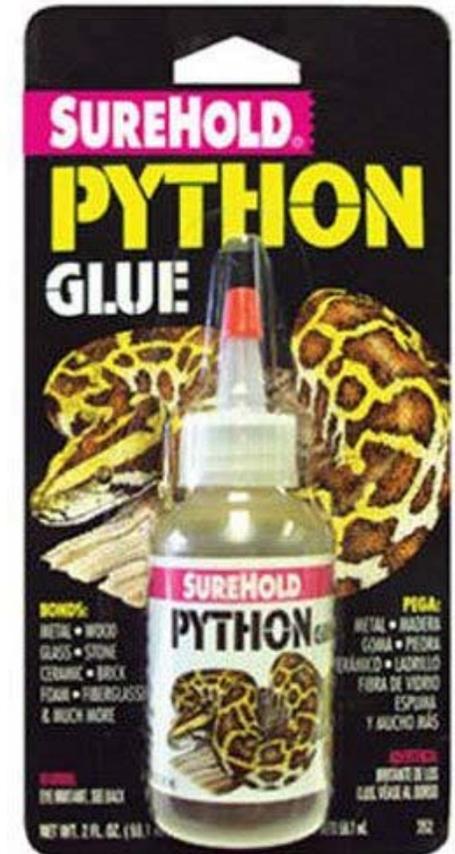
Il permet d'assembler des briques logicielles dans un processus global de production ...

“Around 13 000 freely accessible libraries offer a ready-made implemented solution for virtually every conceivable task.”



*«Python plays a key role in our production pipeline. Without it a project the size of The Avengers would have been very difficult to pull off. From crowd rendering to batch processing to compositing, Python binds all things together.»*

**Tommy Burnette**  
Head of Global Pipeline, Lucasfilm



Autres implementations de python (principale en C) : Jython (java), IronPython (C#/.Net)

## Succès de python : frameworks web

- Après un succès comme langage de scripting serveurs, python connait un succes dans les outils web :
  - Zope, Plone CMS
  - Django / Pyramid/Turbogears Tornado/Werkzeug, ...
  - Cherrypy, Bottle, Flask



Instagram



The Washington Post



Python, le langage de programmation agile

## **La communauté Python**

Python, le langage de la donnée

Conclusion

## Communauté ouverte et vivante

- Opensource : le code est partagé auditable, et réutilisé
- Qualité des librairies, évolution et adaptation aux nouveaux besoins
- Large Communauté : beaucoup de projets existent, pour tous les besoins

Solutions to many problems have already been devised by other developers. Extensive online documentation, numerous books, forums, and mailing lists, and also congresses and conferences across the globe make it easy to learn this fascinating programming language.

The combined creativity and productivity of the community and the intensive transfer of knowledge and expertise within the community makes it an extended workbench of a company's own IT department.

## Documentation / support

- Qualité du développement va de pair avec la qualité de la documentation
- Qualité du support
  - Support 'public' : [stackoverflow.com](https://stackoverflow.com) : si l'on sait décrire son problème sur un moteur de recherche, on trouvera la solution
- Il est facile d'apprendre python : via des sites internet, via des MOOC  
(cf [www.fun-mooc.fr](http://www.fun-mooc.fr))

## Environnement de datascience ouvert :



Project Jupyter exists to develop open-source software, open-standards, and services for interactive computing across dozens of programming languages.

- Notebooks : Documents
- Noyaux : python, R, Julia, ... +40 !
- Widgets : interface interactive
- JupyterHub : partage



**jupyter** Welcome to the Jupyter Notebook Server

**WARNING**  
Don't rely on this server

Your server is hosted that

**Run some Python code**

To run the code below:

1. Click on the cell to select it
2. Press **SHIFT+ENTER**

A full tutorial for using the Jupyter Notebook is available at [http://jupyter.org](#)

```
In [ ]: %matplotlib inline
import pandas as pd
import numpy as np
import matplotlib
```

---

**jupyter** Lorenz Differential Equations (autosaved)

File Edit View Insert Cell Kernel Help Python 3

Cell Toolbar: None

## Exploring the Lorenz System

In this Notebook we explore the [Lorenz system](#) of differential equations:

$$\begin{aligned}\dot{x} &= \sigma(y - x) \\ \dot{y} &= \rho x - y - xz \\ \dot{z} &= -\beta z + xy\end{aligned}$$

This is one of the classic systems in non-linear differential equations. It exhibits a range of complex behaviors as the parameters  $(\sigma, \beta, \rho)$  are varied, including what are known as *chaotic solutions*. The system was originally developed as a simplified mathematical model for atmospheric convection in 1963.

```
In [7]: interact(Lorenz, N=fixed(10), angle=(0.,360.),
                sigma=(0.0,50.0), beta=(0.,5), rho=(0.0,50.0))
```

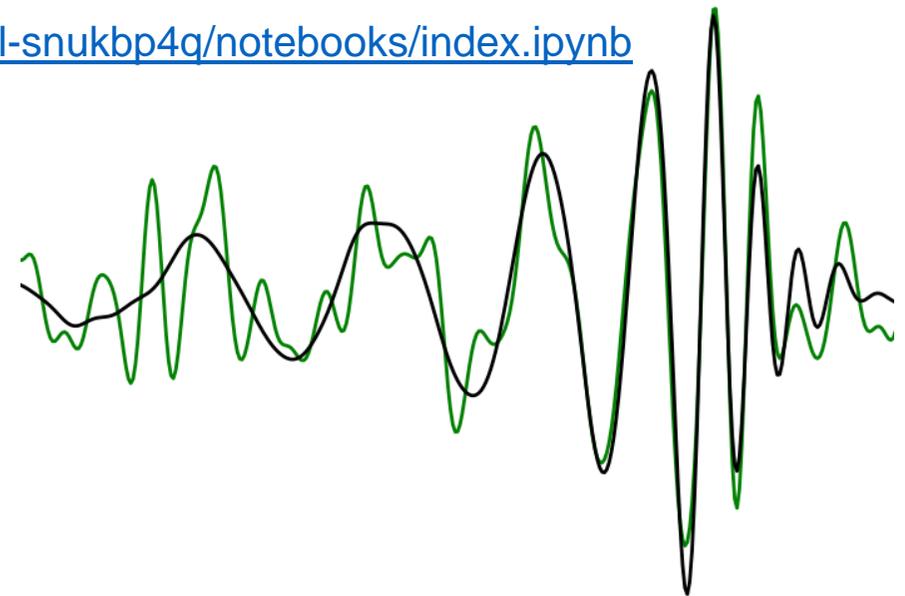
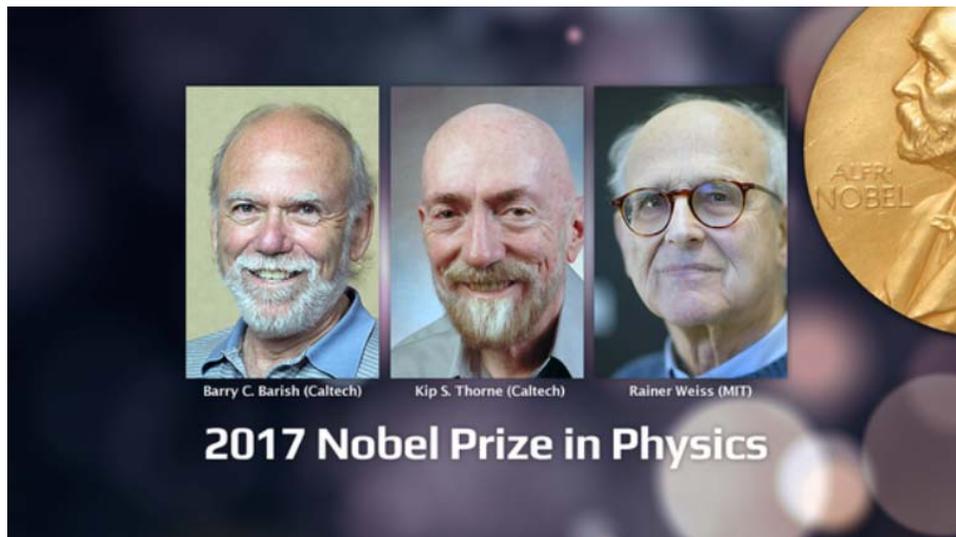
angle 308.2  
max\_time 12  
 $\sigma$  10  
 $\beta$  2.6  
 $\rho$  28

# Succès de python : Recherche scientifique : Outil de la démarche 'Science reproductible'

Découverte des ondes gravitationnelles [https://losc.ligo.org/s/events/GW150914/GW150914\\_tutorial.html](https://losc.ligo.org/s/events/GW150914/GW150914_tutorial.html)

Traitement des données expérimentales et analyse des résultats en python

[https://hub.mybinder.org/user/losc-tutorial-l-event\\_tutorial-snukbp4q/notebooks/index.ipynb](https://hub.mybinder.org/user/losc-tutorial-l-event_tutorial-snukbp4q/notebooks/index.ipynb)





Python, le langage de programmation agile

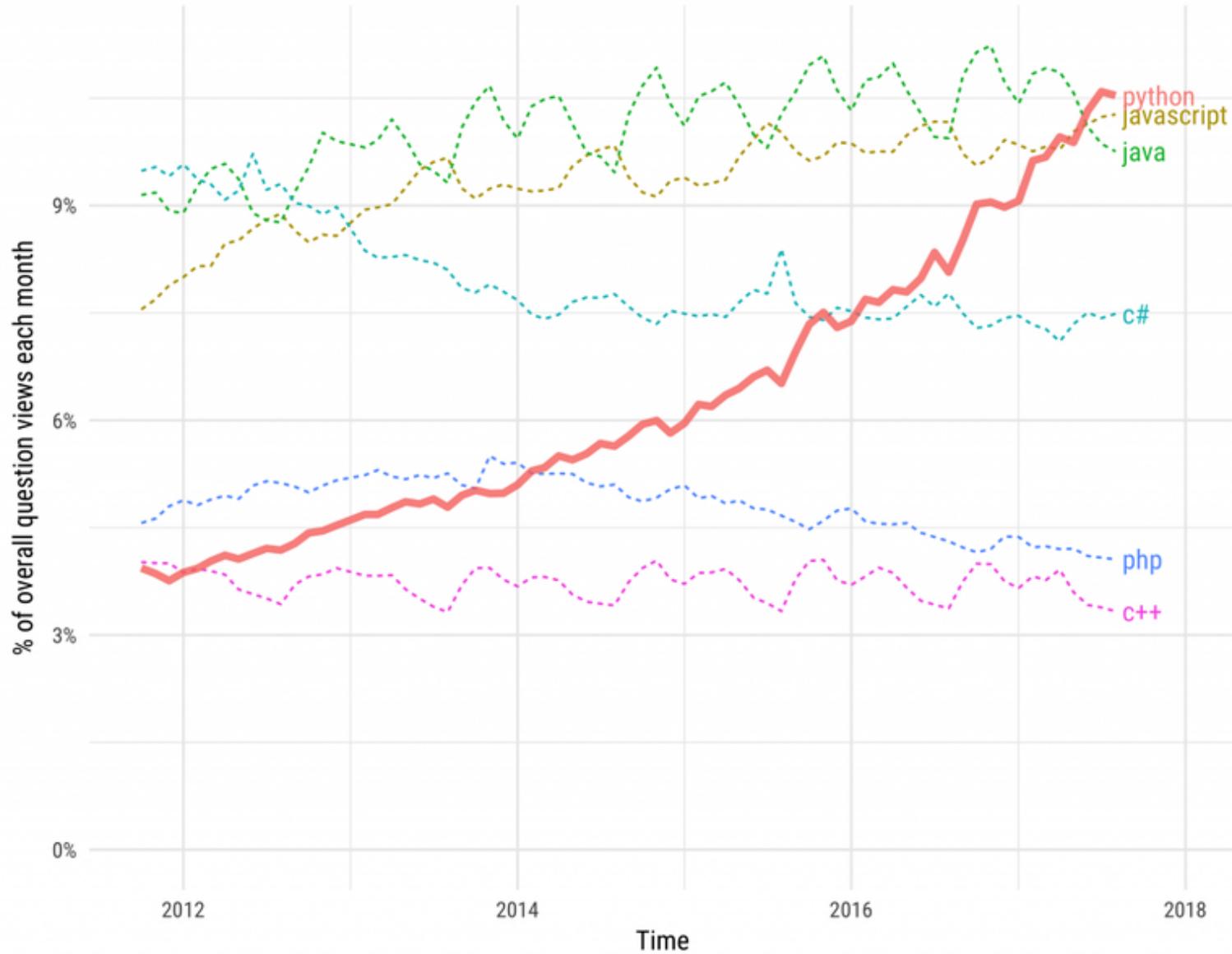
La communauté Python

**Python, le langage de la donnée**

Conclusion

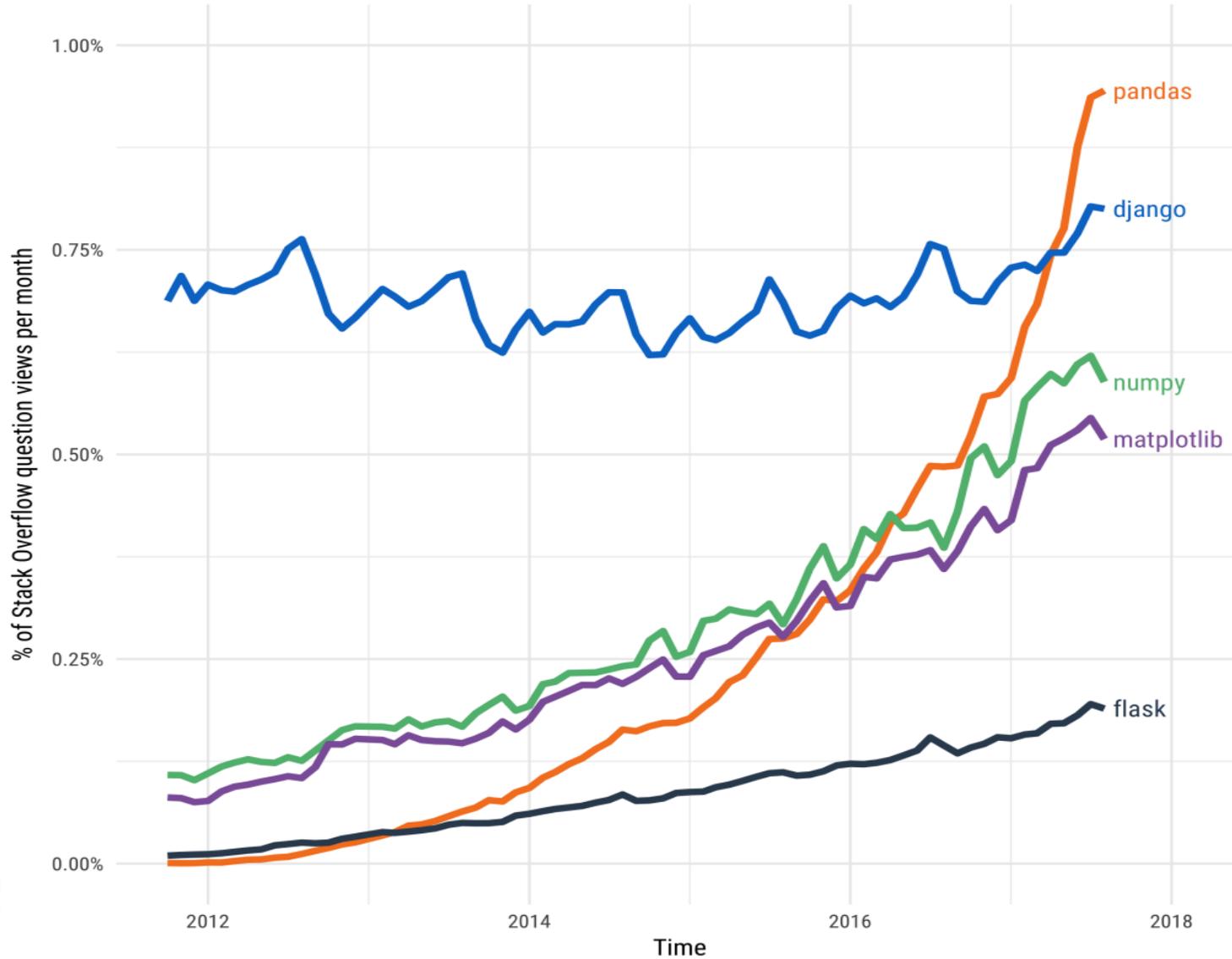
## Growth of major programming languages

Based on Stack Overflow question views in World Bank high-income countries



## Stack Overflow Traffic to Questions About Selected Python Packages

Based on visits to Stack Overflow questions from World Bank high-income countries



Accroissement de la popularité de python depuis 2014 corrélé à un développement de l'utilisation du module pandas (Python Analysis of Data)



Question :

Python est-il autant employé comme langage de programmation que comme langage de données ?

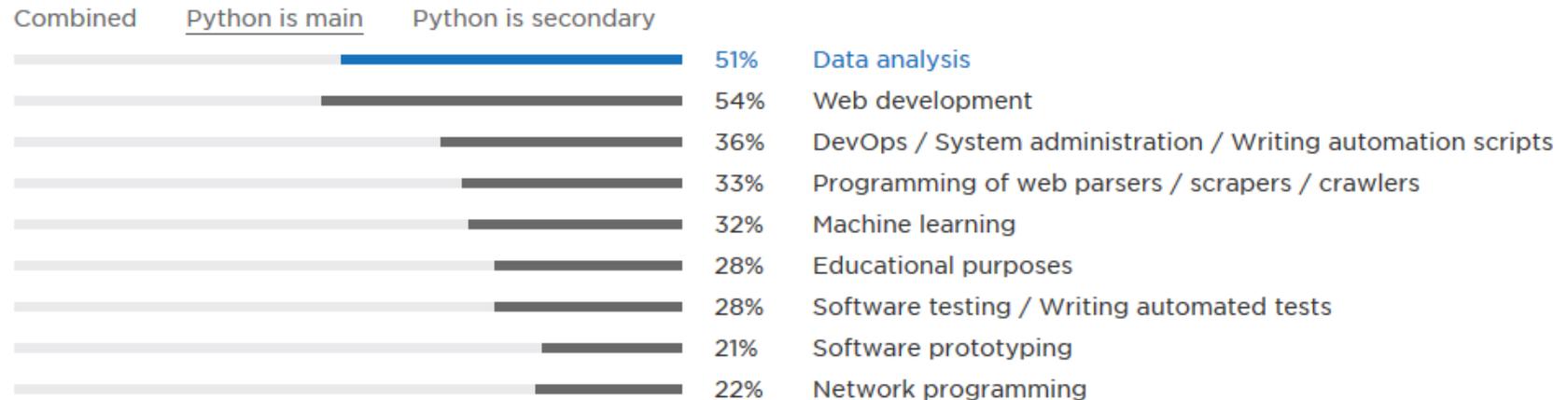


Suivre

Only one in nine respondents guessed the web developers:data scientists ratio correctly. It's 1:1 [bit.ly/2I7Zomr](https://bit.ly/2I7Zomr) #pythondevsurvey2017

Python Developers Survey 2017 Results

### What do you use Python for? (multiple answers)



[See all the results](#)

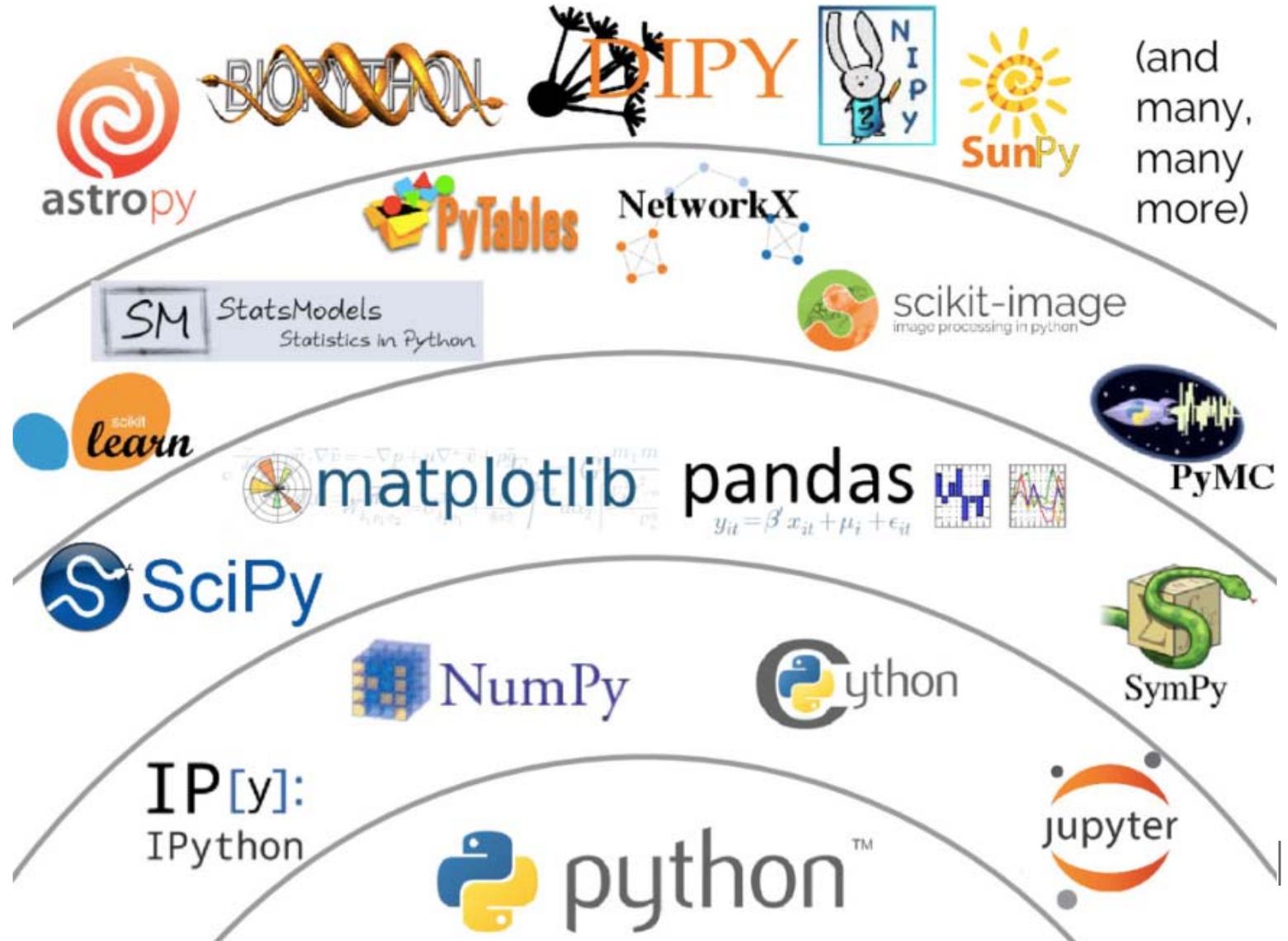
8 Retweets 23 J'aime



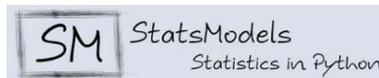
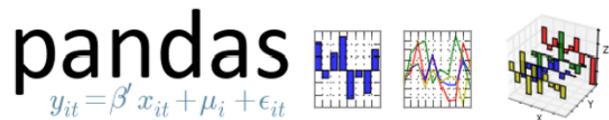
8 23

<https://www.jetbrains.com/research/python-developers-survey-2017/>

# Un écosystème datascience mûr



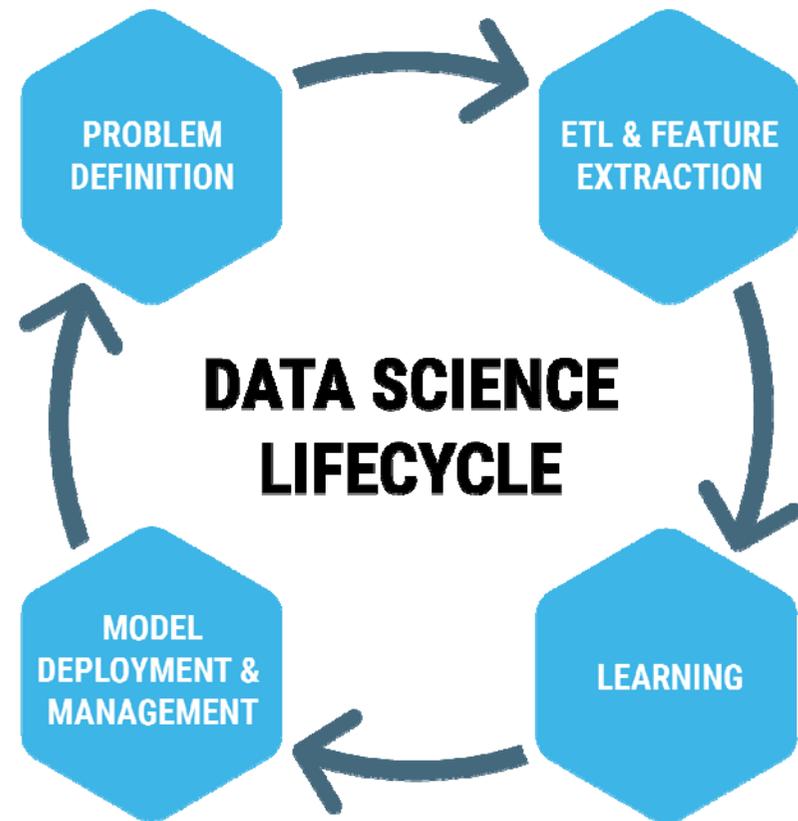
# Les principales bibliothèques de python utiles au Datascientist :





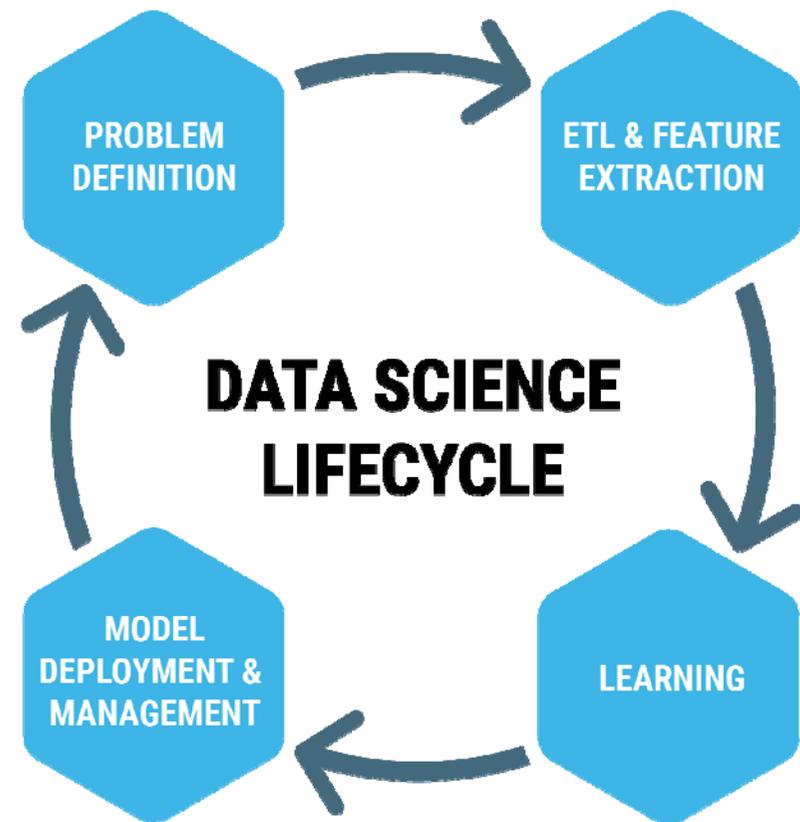
## Processus de datascience

- Objectif défini
- Import de données
- Exploration, nettoyage des données
- Modélisations / Machine learning
- Communication des résultats / application



## Autres dimensions à prendre en compte

- Collaboration
- Prototypage
- Changement d'échelle : application web
- Déploiement / automatisaion
- Machine Learning



**Python est un langage qui couvre tous ces aspects**

# Le reseau des Communautés pydata



Pro A community for developers and users of open source data tools

Connexion



**PyData**

Evènements, Conférences

Membres  
**79 174**

Groupes  
**106**

Pays  
**44**



## PyData Paris

<https://www.meetup.com/fr-FR/PyData-Paris/>

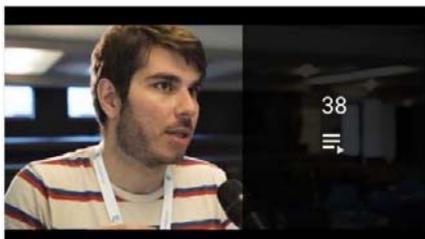
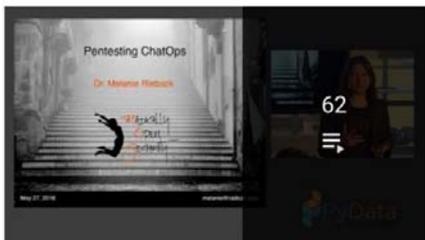
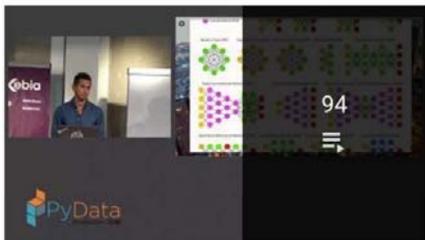
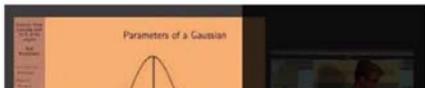
Paris, France · 1869 membres · Groupe public ?  
Réseau PyData - 106 groupes ?



Organisé par  
Sylvain Corlay et 6 autres personnes

-  PyData Amsterdam 2018  
PyData 2:56
-  Keynote: Are You Ready for GDPR? - Michele Appello  
PyData 46:55
-  Keynote: Pentesting ChatOps - Melanie Rieback  
PyData 48:39
-  Hands-on introduction to Deep Learning with Keras and Tensorflow - Rodrigo Agundez  
PyData 1:28:41
-  Structured Streaming with PySpark and Azure Databricks - Andrei Varanovich  
PyData 1:21:20
-  Conversational AI with Rasa Core & NLU - Tom Bocklisch  
PyData 1:37:55
-  Tricks, tips and topics in Text Analysis - Bhargav Srinivasa Desikan  
PyData 1:29:04
-  Creating correct and capable classifiers - Ian Ozsvald  
PyData 37:07
-  Forecasting airline passengers using designer machine learning - Alexander Beckus, Jan van der Vort



-  PyData Amsterdam 2018  
Mise à jour hier · 38 vidéos
-  PyData Keynotes  
Mise à jour hier · 62 vidéos
-  Machine Learning  
Mise à jour il y a 3 jours · 94 vidéos
-  Artificial Intelligence (AI)  
Mise à jour il y a 3 jours · 11 vidéos
-  Probabilistic Programming & Bayesian Methods  
Mise à jour il y a 3 jours · 36 vidéos



*Chaîne youtube, 29/06/2018*  
*50 000 abonnés*

# Python :

Un langage de programmation agile & un langage de données :

- Facile, concis, respectueux du programmeur et de son efficacité,
- Opensource, garantie de qualité ... et de sécurité
- Extensible, Boostable avec de très nombreuses librairies et langages
- Supporté par une large communauté
- Documentation fournie et de qualité, Partage des usages et des outils
- Armé pour les années qui viennent : pandas, scikit-learn, ... écosystème datascience à portée de tous, et intégré à Jupyterlab, l'environnement datascience du futur