



# Une introduction à Python

Christophe Jorssen

*Professeur de Physique-Chimie en PSI (et d'Informatique)*

[christophe.jorssen@gmail.com](mailto:christophe.jorssen@gmail.com)

<https://github.com/cjorssen/presentation-python-JD>

Lycée Jacques Decour – Paris

Jeudi 4 avril 2019



Ce document est mis à disposition selon les termes de la licence Creative Commons “Attribution - Partage dans les mêmes conditions 3.0 non transposé”.





## Script Python (hello.py)

```
1 print('Hello world!')
```



# Plan

- 1 Un peu de « vécu »**
- 2 Installation et mise à jour de l'écosystème Python**
- 3 Quelques façons d'utiliser Python**
- 4 Introduction au langage Python**
- 5 Quelques outils pour l'enseignement de Python**



## 1 Un peu de « vécu »

## 2 Installation et mise à jour de l'écosystème Python

## 3 Quelques façons d'utiliser Python

## 4 Introduction au langage Python

## 5 Quelques outils pour l'enseignement de Python

Pour commencer, quelques exemples utilisés :

- dans ma classe;
- dans le cadre du MOOC « Physique : préparation à l'entrée dans l'enseignement supérieur » de l'École Polytechnique dont je suis un des coauteurs.



## Oscillateur quasi sinusoïdal

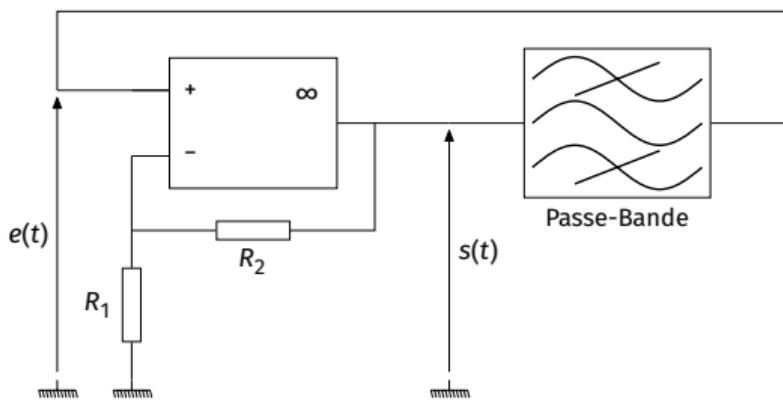
### 1 Un peu de « vécu »

#### ■ Oscillateur quasi sinusoïdal

- Trajectoire d'un projectile
- Propagation d'un paquet d'onde
- Analyse spectrale du son d'une flûte, mode propres d'une corde pincée
- Propagation d'une onde le long d'une « échelle de perroquet »



## Oscillateur quasi sinusoïdal



Système non linéaire :

➤ si  $|Ge| < V_{sat}$ , alors

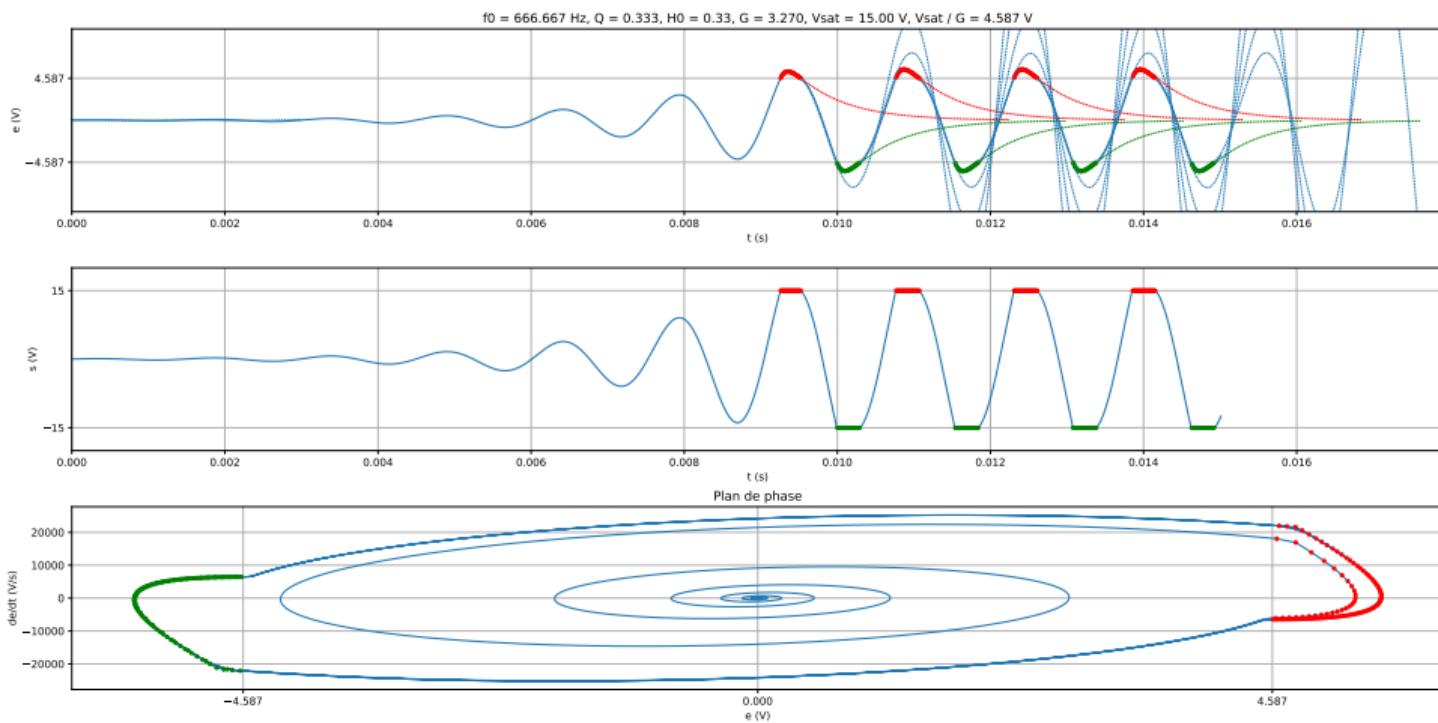
$$\frac{d^2s}{dt^2} + (1 - GH_0) \frac{\omega_0}{Q} \frac{ds}{dt} + \omega_0^2 s = 0;$$

➤ si  $|Ge| \geq V_{sat}$ , alors

$$\frac{d^2s}{dt^2} + \frac{\omega_0}{Q} \frac{ds}{dt} + \omega_0^2 s = 0.$$



## Oscillateur quasi sinusoïdal



Un peu de « vécu »



Installation et mise à jour de l'écosystème Python



Quelques façons d'utiliser Python



Introduction au langage Python



Quelques outils pour l'enseignement de Python



Oscillateur quasi sinusoïdal

<https://github.com/cjorssen/presentation-python-JD/blob/master/python/oscillateurQS-2.py>



## Trajectoire d'un projectile

### 1 Un peu de « vécu »

- Oscillateur quasi sinusoïdal
- **Trajectoire d'un projectile**
- Propagation d'un paquet d'onde
- Analyse spectrale du son d'une flûte, mode propres d'une corde pincée
- Propagation d'une onde le long d'une « échelle de perroquet »

Un peu de « vécu »

○○  
○○○○  
●●○  
○○  
○○○  
○○

Installation et mise à jour de l'écosystème Python

○  
○○○○  
○○○○○  
○○○○○○

Quelques façons d'utiliser Python

○  
○○  
○○○  
○○○○○

Introduction au langage Python

○○

Quelques outils pour l'enseignement de Python

○  
○○  
○○

Trajectoire d'un projectile

<https://github.com/cjorssen/presentation-python-JD/blob/master/python/chute-libre.ipynb>

Un peu de « vécu »



Installation et mise à jour de l'écosystème Python



Quelques façons d'utiliser Python



Introduction au langage Python



Quelques outils pour l'enseignement de Python



Trajectoire d'un projectile

<https://github.com/cjorssen/presentation-python-JD/blob/master/python/volant.ipynb>



## Propagation d'un paquet d'onde

## 1 Un peu de « vécu »

- Oscillateur quasi sinusoïdal
- Trajectoire d'un projectile
- Propagation d'un paquet d'onde**
- Analyse spectrale du son d'une flûte, mode propres d'une corde pincée
- Propagation d'une onde le long d'une « échelle de perroquet »

Un peu de « vécu »



Installation et mise à jour de l'écosystème Python



Quelques façons d'utiliser Python



Introduction au langage Python



Quelques outils pour l'enseignement de Python



Propagation d'un paquet d'onde

<https://github.com/cjorssen/presentation-python-JD/blob/master/python/plot3D-paquet-onde.ipynb>



Analyse spectrale du son d'une flûte, mode propres d'une corde pincée

## 1 Un peu de « vécu »

- Oscillateur quasi sinusoïdal
- Trajectoire d'un projectile
- Propagation d'un paquet d'onde
- Analyse spectrale du son d'une flûte, mode propres d'une corde pincée**
- Propagation d'une onde le long d'une « échelle de perroquet »

Un peu de « vécu »



Installation et mise à jour de l'écosystème Python



Quelques façons d'utiliser Python



Introduction au langage Python



Quelques outils pour l'enseignement de Python



Analyse spectrale du son d'une flûte, mode propres d'une corde pincée

<https://github.com/cjorssen/presentation-python-JD/blob/master/python/FFT-flute.ipynb>

Un peu de « vécu »



Installation et mise à jour de l'écosystème Python



Quelques façons d'utiliser Python



Introduction au langage Python



Quelques outils pour l'enseignement de Python



Analyse spectrale du son d'une flûte, mode propres d'une corde pincée

<https://github.com/cjorssen/presentation-python-JD/blob/master/python/onde-stationnaire.ipynb>



Propagation d'une onde le long d'une « échelle de perroquet »

## 1 Un peu de « vécu »

- Oscillateur quasi sinusoïdal
- Trajectoire d'un projectile
- Propagation d'un paquet d'onde
- Analyse spectrale du son d'une flûte, mode propres d'une corde pincée
- **Propagation d'une onde le long d'une « échelle de perroquet »**

Un peu de « vécu »



Installation et mise à jour de l'écosystème Python



Quelques façons d'utiliser Python



Introduction au langage Python



Quelques outils pour l'enseignement de Python



Propagation d'une onde le long d'une « échelle de perroquet »

<https://github.com/cjorssen/presentation-python-JD/blob/master/python/echelle-de-perroquet.ipynb>



## 1 Un peu de « vécu »

## 2 Installation et mise à jour de l'écosystème Python

## 3 Quelques façons d'utiliser Python

## 4 Introduction au langage Python

## 5 Quelques outils pour l'enseignement de Python

Un peu de « vécu »

○○  
○○○○  
○○○  
○○○○  
○○○  
○○

Installation et mise à jour de l'écosystème Python

○  
●○○○  
○○○○○  
○○○○○○

Quelques façons d'utiliser Python

○  
○○  
○○○○  
○○○○○○

Introduction au langage Python

○○

Quelques outils pour l'enseignement de Python

○  
○○  
○○

Nécessité d'une distribution

## 2 Installation et mise à jour de l'écosystème Python

### ■ Nécessité d'une distribution

- Installation d'Anaconda
- Mise à jour et installation de paquets complémentaires



Nécessité d'une distribution

Ne jamais installer « à la main » les différents constituants de l'écosystème Python.



Installer une **distribution** proposant, entre autres :

- un **IDE** (*integrated development environment*);
- un **gestionnaire de paquets**.

Un peu de « vécu »



Installation et mise à jour de l'écosystème Python



Quelques façons d'utiliser Python



Introduction au langage Python



Quelques outils pour l'enseignement de Python



Nécessité d'une distribution

# Les distributions

**Pyzo** <https://pyzo.org/>

**Anaconda**  <https://www.anaconda.com/distribution/> 



Nécessité d'une distribution

## La distribution Anaconda

- Installe les outils nécessaires à une **utilisation scientifique** (au sens large) de Python.
- Installe un IDE dédié à Python : **spyder**.
- Installe un gestionnaire de paquet avancé : **conda**.
- Installe une **interface graphique** permettant une utilisation « à la souris » de la distribution et de ses outils.



## Installation d'Anaconda

### 2 Installation et mise à jour de l'écosystème Python

- Nécessité d'une distribution
- **Installation d'Anaconda**
- Mise à jour et installation de paquets complémentaires



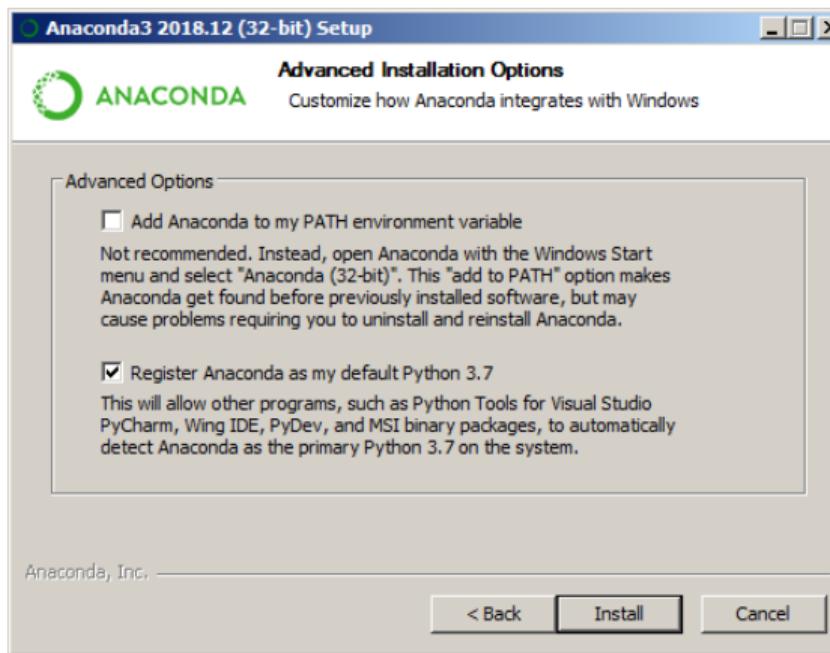
Lors de l'installation de Python, il faut :

- faire attention à ce que le Python nouvellement installé **ne perturbe pas** un Python déjà installé;
- faire en sorte que le Python que l'on utilise soit **trouvé**;
- faire en sorte que le Python que l'on utilise soit **le bon**.



Installation d'Anaconda

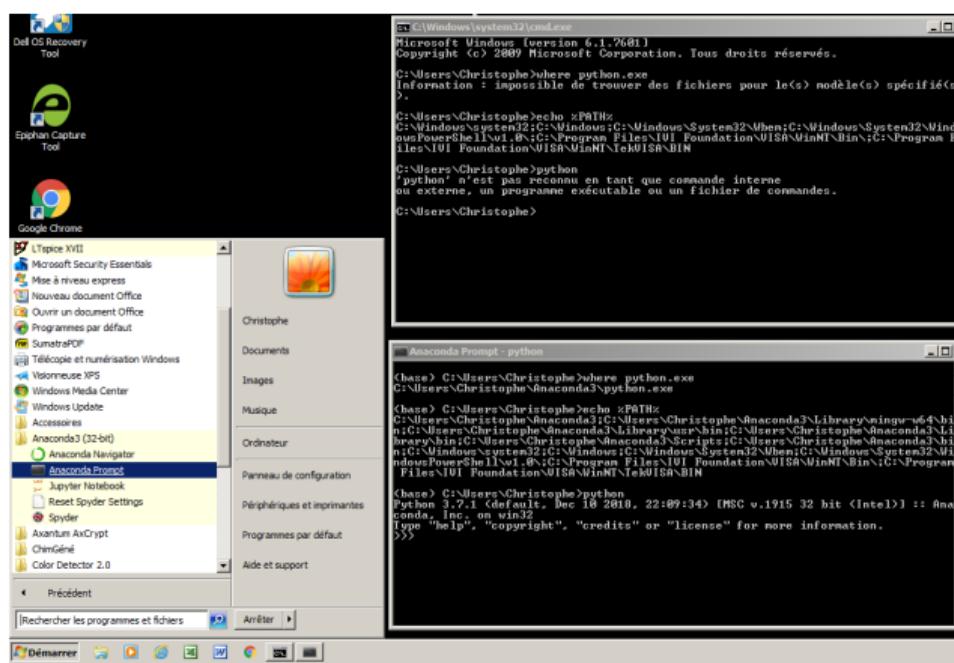
# Installation d'Anaconda (MS-Windows)





## Installation d'Anaconda

## Installation d'Anaconda (MS-Windows)





## Installation d'Anaconda

# Installation d'Anaconda (MacOS)

```
christophe — -bash — 107x27
(base) MBPdeChristophe:~ christophe$ which python
/usr/local/anaconda3/bin/python
(base) MBPdeChristophe:~ christophe$ head .bash_profile
export PATH="/usr/local/anaconda3/bin:/usr/local/bin:$PATH"
# added by Anaconda3 2018.12 installer
# >>> conda init >>>
# !! Contents within this block are managed by 'conda init' !!
__conda_setup="${CONDA_REPORT_ERRORS=false '/anaconda3/bin/conda' shell.bash hook 2> /dev/null}"
if [ $? -eq 0 ]; then
    \eval "$__conda_setup"
else
    if [ -f '/anaconda3/etc/profile.d/conda.sh' ]; then
        . '/anaconda3/etc/profile.d/conda.sh'
(base) MBPdeChristophe:~ christophe$
```



## Mise à jour et installation de paquets complémentaires

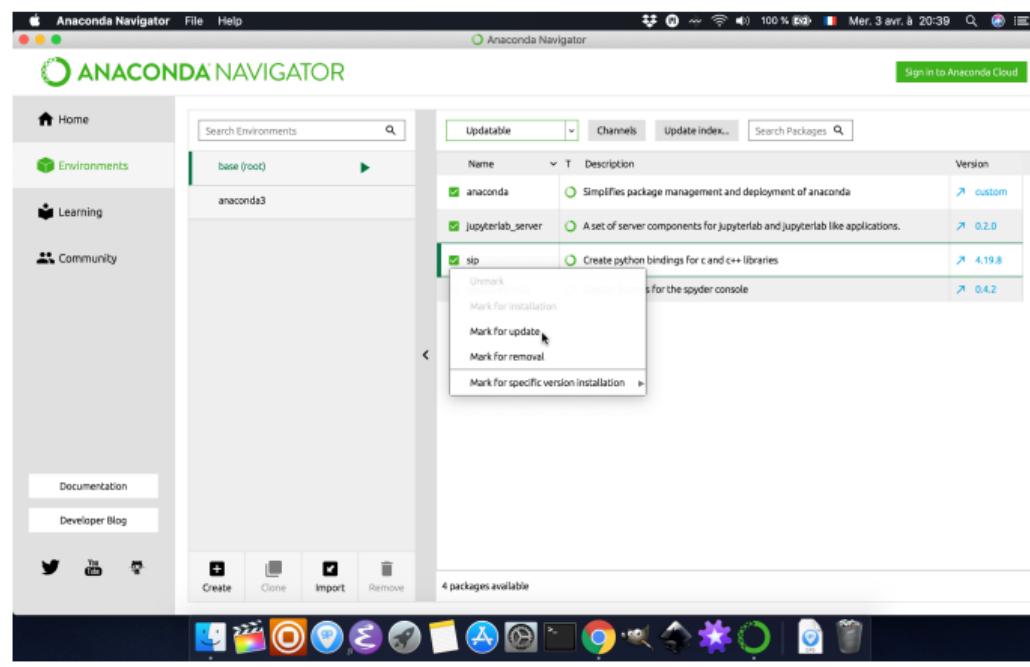
### 2 Installation et mise à jour de l'écosystème Python

- Nécessité d'une distribution
- Installation d'Anaconda
- Mise à jour et installation de paquets complémentaires



Mise à jour et installation de paquets complémentaires

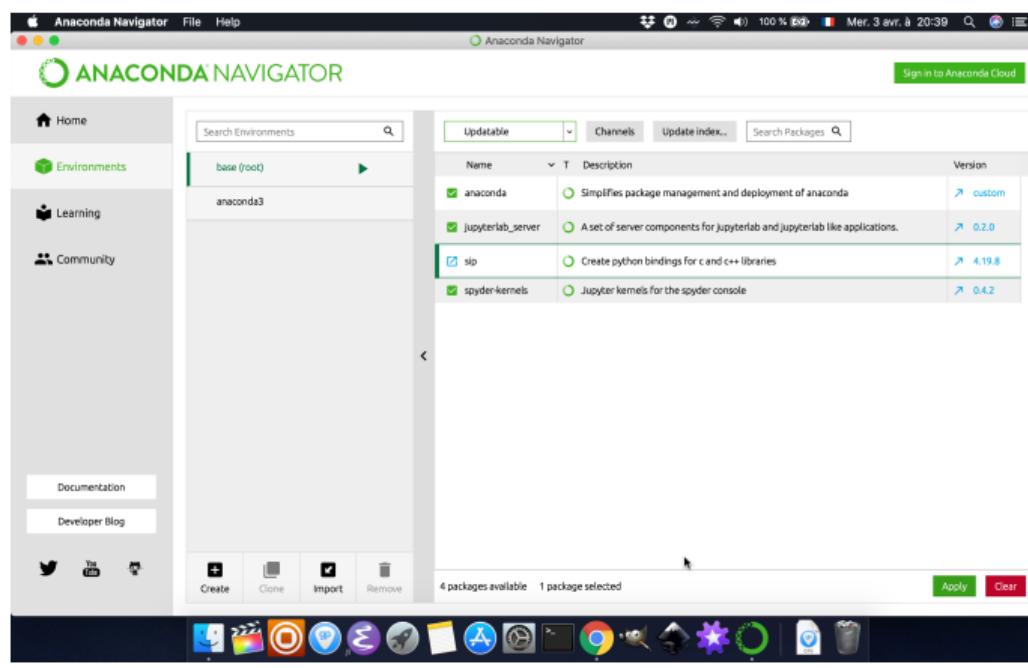
# À l'aide de l'interface graphique : mise à jour





Mise à jour et installation de paquets complémentaires

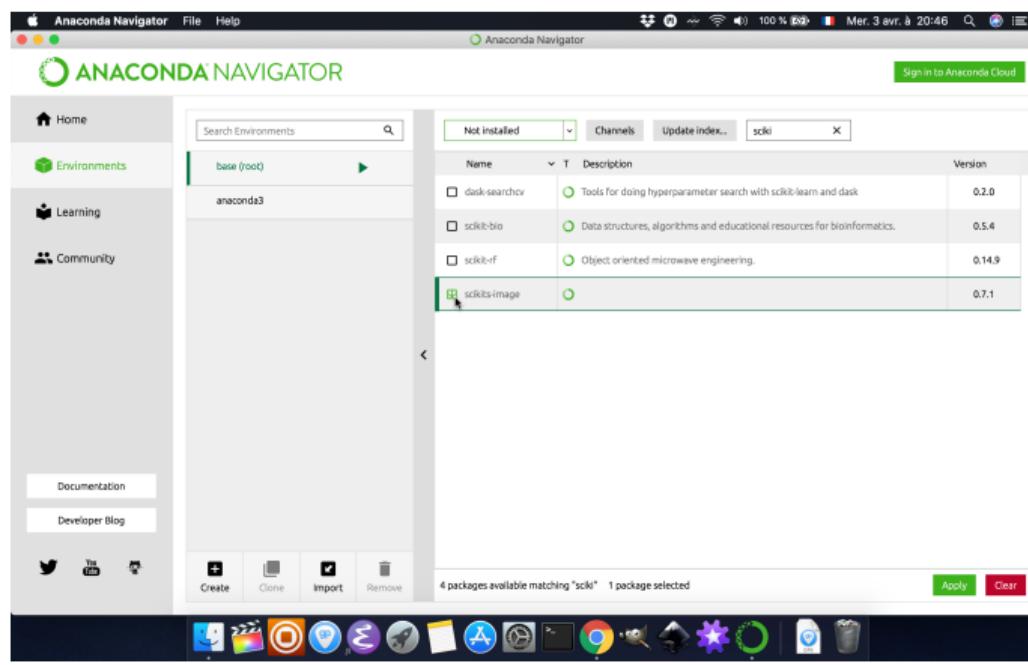
# À l'aide de l'interface graphique : mise à jour





Mise à jour et installation de paquets complémentaires

# À l'aide de l'interface graphique : installation d'un paquet





Mise à jour et installation de paquets complémentaires

## En ligne de commande

```
christophe — bash — 80x19
Last login: Wed Apr  3 20:26:27 on ttys000
[(base) MBPdeChristophe:~ christophe$ conda update --all
Collecting package metadata: done
Solving environment: done

# All requested packages already installed.

[(base) MBPdeChristophe:~ christophe$ conda install scikit-image
Collecting package metadata: done
Solving environment: done

# All requested packages already installed.

(base) MBPdeChristophe:~ christophe$ ]
```



## 1 Un peu de « vécu »

## 2 Installation et mise à jour de l'écosystème Python

## 3 Quelques façons d'utiliser Python

## 4 Introduction au langage Python

## 5 Quelques outils pour l'enseignement de Python



## Éditeur et ligne de commande

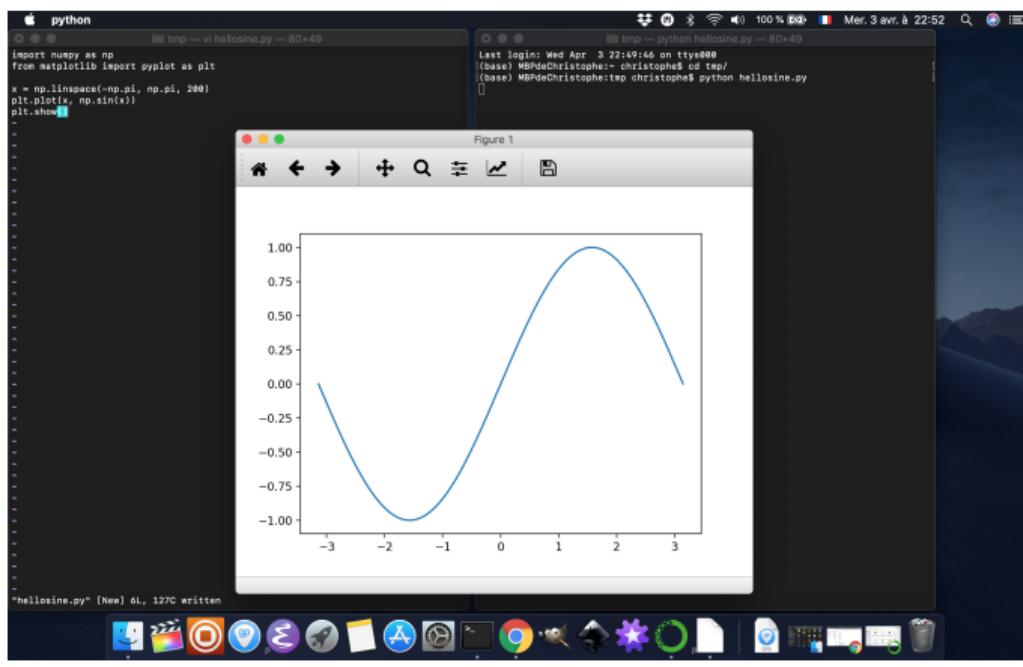
### 3 Quelques façons d'utiliser Python

- Éditeur et ligne de commande
- IDE
- Dans le *cloud* (dans le navigateur)
- Avec des *notebooks*



Éditeur et ligne de commande

## Un exemple *hardcore* : éditeur et exécution dans la console





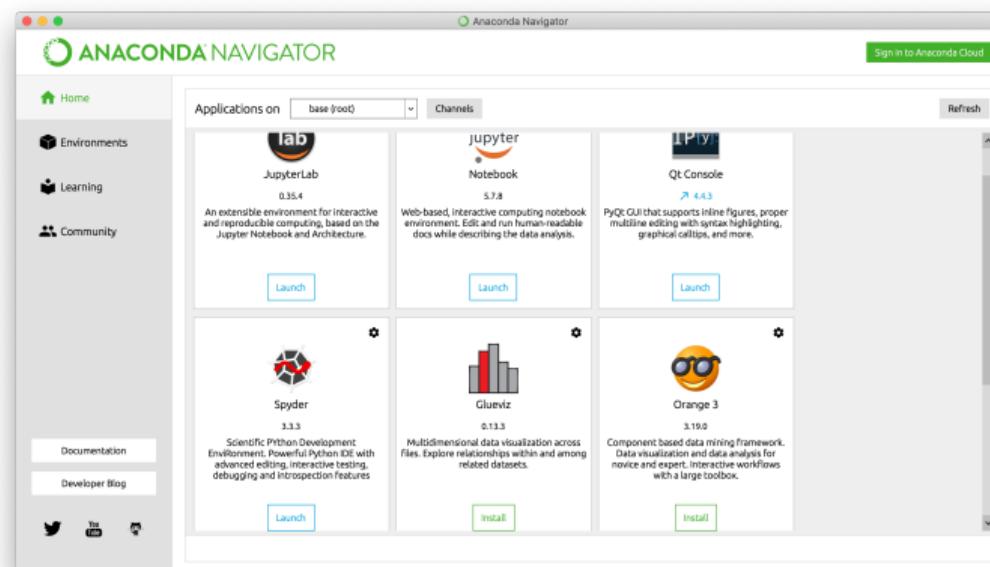
### 3 Quelques façons d'utiliser Python

- Éditeur et ligne de commande
- **IDE**
- Dans le *cloud* (dans le navigateur)
- Avec des *notebooks*



IDE

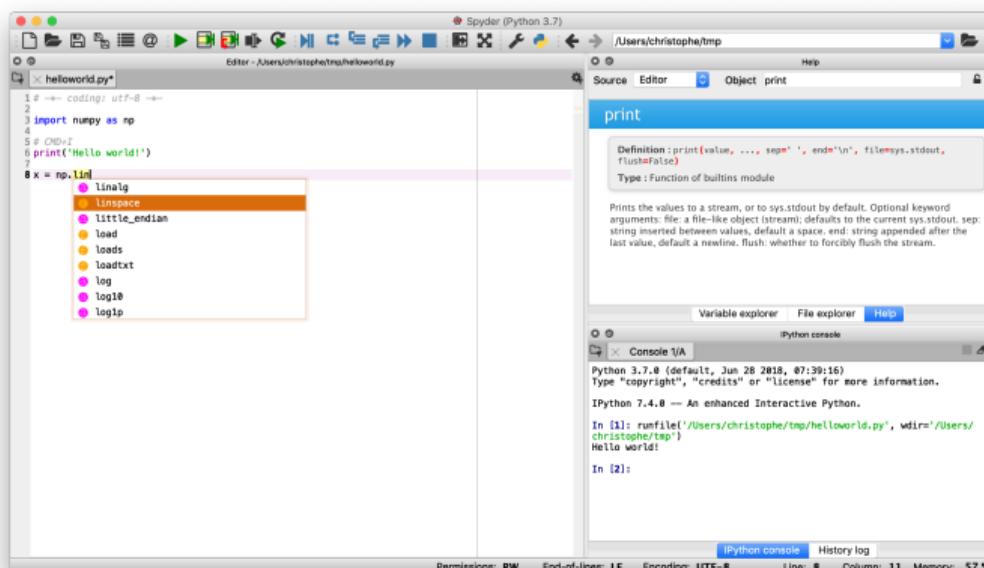
# L'IDE d'Anaconda : spyder





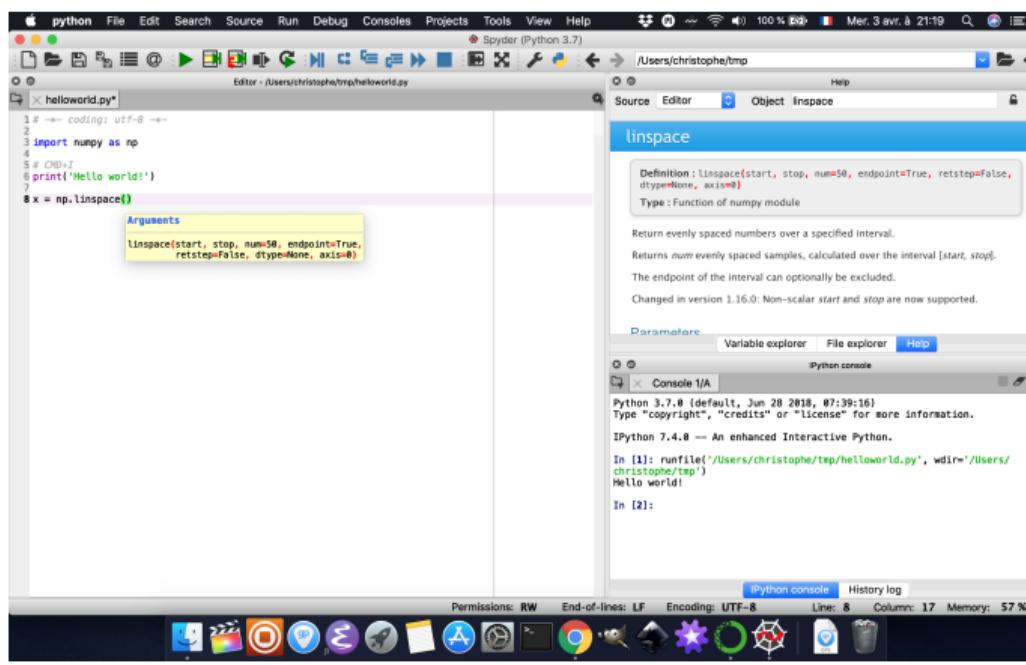
IDE

# L'IDE d'Anaconda : spyder





# L'IDE d'Anaconda : spyder





Dans le *cloud* (dans le navigateur)

### 3 Quelques façons d'utiliser Python

- Éditeur et ligne de commande
- IDE
- **Dans le *cloud* (dans le navigateur)**
- Avec des *notebooks*

○○  
○○○○  
○○○  
○○  
○○○  
○○  
○○

○  
○○○○  
○○○○○  
○○○○○○

○  
○○  
○○○○  
○●○○○○  
○○○○○

○○

○  
○○  
○○

Dans le *cloud* (dans le navigateur)



C'est la solution de dépannage qui peut fonctionner sur n'**importe quel ordinateur** ou **smartphone** disposant d'une connexion internet d'un navigateur récent.

Un peu de « vécu »



Installation et mise à jour de l'écosystème Python



Quelques façons d'utiliser Python



Introduction au langage Python



Quelques outils pour l'enseignement de Python



Dans le *cloud* (dans le navigateur)

# CS50 sandbox

The screenshot shows a web browser window titled "CS50 Sandbox". The URL in the address bar is "https://sandbox.cs50.io". The page content includes the title "CS50 Sandbox", a subtitle "temporary programming environments for students and teachers", and a note "with support for [these environments](#)". Below this, there are two buttons: "Create sandbox" (highlighted) and "Recent sandboxes". A list of options with checkboxes follows: "with embedded browser" (unchecked), "with terminal window" (checked), "with text editor, [with these tabs](#)" (checked), and "with X window" (checked). At the bottom are "Create" and "or configure" buttons.

Un peu de « vécu »



Installation et mise à jour de l'écosystème Python



Quelques façons d'utiliser Python



Introduction au langage Python

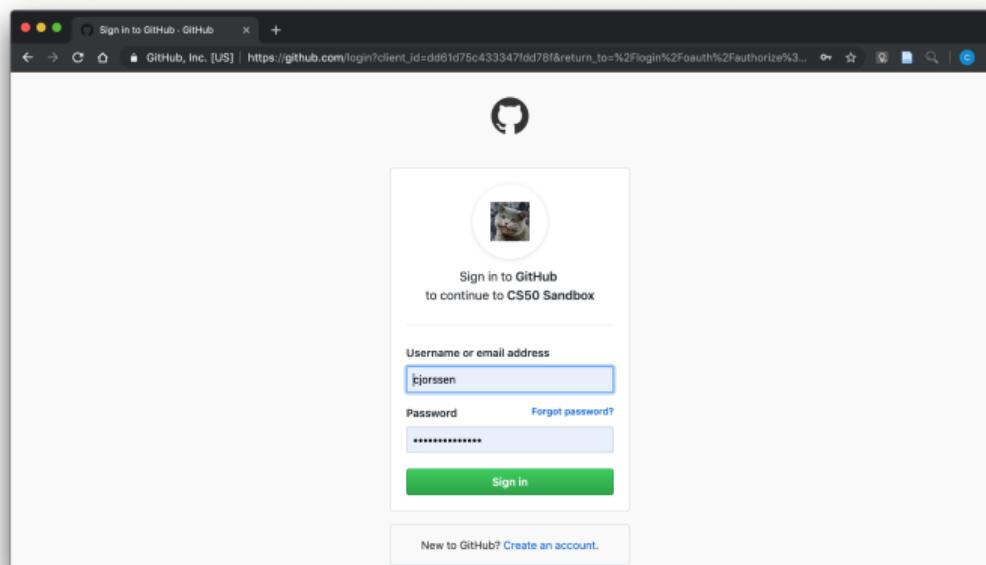


Quelques outils pour l'enseignement de Python



Dans le *cloud* (dans le navigateur)

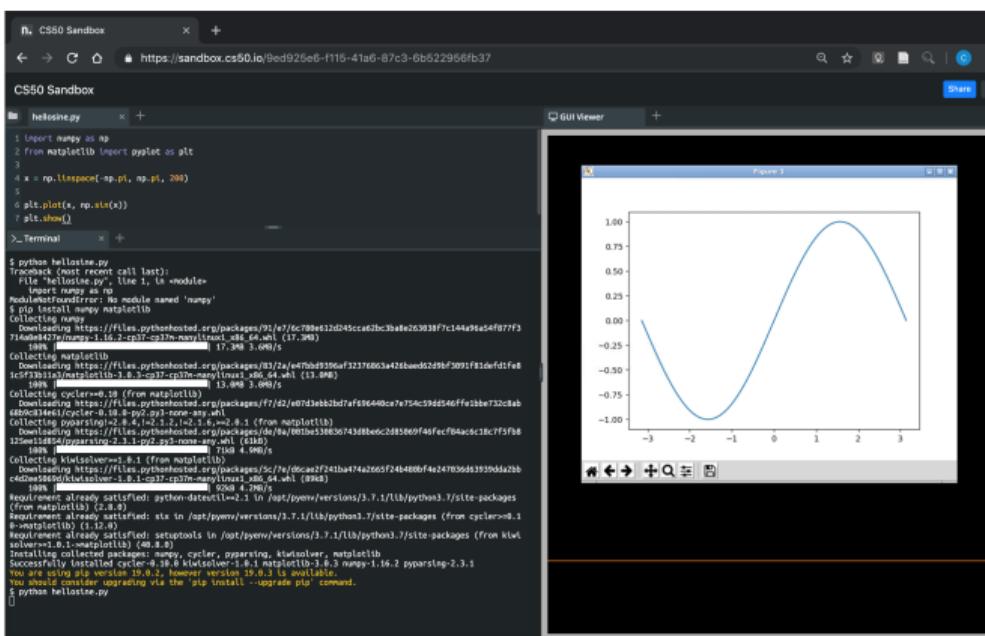
# CS50 sandbox





Dans le *cloud* (dans le navigateur)

# CS50 sandbox



The screenshot shows the CS50 Sandbox interface. On the left, there's a code editor window titled "hellosine.py" containing the following Python code:

```

1 import numpy as np
2 from matplotlib import pyplot as plt
3
4 x = np.linspace(-np.pi, np.pi, 200)
5
6 plt.plot(x, np.sin(x))
7 plt.show()

```

Below the code editor is a terminal window showing the execution of the script:

```

$ python hellosine.py
Traceback (most recent call last):
  File "hellosine.py", line 1, in <module>
    import numpy as np
ModuleNotFoundError: No module named 'numpy'
$ pip install numpy
Collecting numpy
  Downloading https://files.pythonhosted.org/packages/92/e/c/79b612d245ccac1bc3948e263918f7c144a94d4f877ff3
    100% |████████████████████████████████| 17.3MB 3.0KB/s
Collecting matplotlib
  Downloading https://files.pythonhosted.org/packages/02/1/a/e7bdad9296ef32370863a427bwe0d2dfbf3095fb12defdf1e8
    100% |████████████████████████████████| 17.3MB 3.0KB/s
Collecting cycler==0.10.0 (from matplotlib)
  Downloading https://files.pythonhosted.org/packages/f7/d2/6fd1eb3b2d7af7696400cae7e754c59dd546ffeb3be732c8ab
    100% |████████████████████████████████| 13.0MB 3.0KB/s
Collecting pyparsing==2.3.1 (from cycler==0.10.0->matplotlib)
  Downloading https://files.pythonhosted.org/packages/0e/5a/001be538836743d8beec2d85869fa8fcf704a8c8cfc5f5fb8
    100% |████████████████████████████████| 13.0MB 3.0KB/s
Collecting kiwisolver==1.1.0 (from pyparsing==2.3.1->matplotlib)
  Downloading https://files.pythonhosted.org/packages/5c/74/244bfb4e2470366d3939d42bb
    100% |████████████████████████████████| 9.9MB 3.0KB/s
Requirement already satisfied: python-dateutil==2.1 in /opt/pyenv/versions/3.7.1/lib/python3.7/site-packages
  (from matplotlib==2.2.2->kiwisolver==1.1.0->pyparsing==2.3.1->cycler==0.10.0->matplotlib)
Requirement already satisfied: six in /opt/pyenv/versions/3.7.1/lib/python3.7/site-packages (from cycler>=0.10.0->matplotlib)
Requirement already satisfied: python-dateutil==2.1 in /opt/pyenv/versions/3.7.1/lib/python3.7/site-packages (from kiwisolver==1.1.0->pyparsing==2.3.1->cycler==0.10.0->matplotlib)
Installing collected packages: numpy, cycler, pyparsing, kiwisolver, matplotlib
Successfully installed cycler==0.10.0 kiwisolver==1.1.0 matplotlib==2.2.2 numpy==1.16.2 pyparsing==2.3.1
You are using Python 3.7.1, however version 3.7.2 is available.
You should consider upgrading via the pip install --upgrade pip command.
$ python hellosine.py

```

To the right of the terminal is a "GUI Viewer" window titled "Figure 3" showing a plot of the sine function from -π to π.

Un peu de « vécu »



Installation et mise à jour de l'écosystème Python



Quelques façons d'utiliser Python



Introduction au langage Python



Quelques outils pour l'enseignement de Python



Dans le cloud (dans le navigateur)

# repl.it

The screenshot shows the repl.it interface. On the left, the file structure shows a folder named 'hellosine' containing 'main.py' and 'hellosine.png'. The code in 'main.py' is:

```
1 import numpy as np
2 from matplotlib import pyplot as plt
3
4 x = np.linspace(-np.pi, np.pi, 200)
5
6 plt.plot(x, np.sin(x))
7 plt.savefig('hellosine.png')
```

In the center, a terminal window shows the command run and the resulting output:

```
Python 3.6.1 (default, Dec 2015, 13:05:11)
[GCC 4.8.2] on linux
> [REDACTED]
```

At the bottom, a plot of the sine function is displayed, showing a wave from -pi to pi.



Avec des *notebooks*

### 3 Quelques façons d'utiliser Python

- Éditeur et ligne de commande
- IDE
- Dans le *cloud* (dans le navigateur)
- **Avec des *notebooks***



Avec des *notebooks*



Un *notebook* est une page html :

- qui peut mélanger :
  - du texte (mis en forme);
  - du  $\text{\LaTeX}$  (rendu);
  - du code Python et son exécution (y compris des représentations graphiques);
  - des vidéos ou de l'audio;
  - etc;
- totalement éditable dans le navigateur et dynamique (à condition de disposer d'un serveur jupyter).



Avec des *notebooks*



C'est une méthode qui fonctionne :

- en **local**;
- dans le **cloud** (à condition de disposer d'un serveur jupyter).



Avec des *notebooks*



D'après *Nature*, c'est l'avenir du chercheur!

- *Interactive notebooks : Sharing the code*, Nature (515), pp. 151–152 (6 novembre 2014), doi:10.1038/515151a.
- *Why Jupyter is data scientists' computational notebook of choice*, Nature (563), pp. 145-146 (2018), doi : 10.1038/d41586-018-07196-1



Avec des *notebooks*

- Les exemples du début de la présentation sont des *notebooks* utilisés dans le cadre d'un MOOC de la plate-forme FUN.
- *Nature* fournit régulièrement des exemples, comme celui-ci :  
[https://github.com/jperkel/example\\_notebook](https://github.com/jperkel/example_notebook).



## 1 Un peu de « vécu »

## 2 Installation et mise à jour de l'écosystème Python

## 3 Quelques façons d'utiliser Python

## 4 Introduction au langage Python

## 5 Quelques outils pour l'enseignement de Python



On continue dans les *notebooks* : d'abord

<https://github.com/cjorssen/presentation-python-JD/blob/master/python/utiliser-les-notebooks.ipynb> puis

[https://github.com/cjorssen/presentation-python-JD/blob/master/python/%C3%80%C2%0la%C2%0d%C3%A9couverte%C2%0de%C2%0Python.ipynb.](https://github.com/cjorssen/presentation-python-JD/blob/master/python/%C3%80%C2%0la%C2%0d%C3%A9couverte%C2%0de%C2%0Python.ipynb)



- 1 Un peu de « vécu »**
- 2 Installation et mise à jour de l'écosystème Python**
- 3 Quelques façons d'utiliser Python**
- 4 Introduction au langage Python**
- 5 Quelques outils pour l'enseignement de Python**



Blockly

## 5 Quelques outils pour l'enseignement de Python

- Blockly
- Python tutor



Blockly

# De Scratch vers Python

The screenshot shows the Blockly website interface. At the top, there's a navigation bar with links for Google for Education, ACCUEIL, GUIDES, REFERENCE, SAMPLES, SUPPORT, and PUBLICATIONS. Below the navigation is a search bar labeled "Rechercher". The main area is divided into two sections: a Scratch script editor on the left and a Python code editor on the right. The Scratch script consists of a "repeat" loop with a counter variable "Count" set to 1. Inside the loop, it prints "Hello World" and increments the counter by 1. The Python code generated is:

```
Count = None  
  
Count = 1  
while Count <= 3:  
    print('Hello World!' + str(Count))  
    Count = Count + 1
```

<https://developers.google.com/blockly/>



Python tutor

## 5 Quelques outils pour l'enseignement de Python

■ Blockly

■ Python tutor



## Python tutor

The screenshot shows a browser window with tabs for 'nature - christophe.jorssen@...', 'Cheatsheet | Font Awesome', and 'Visualize Python, Java, JavaS...'. The main content is titled 'pythonutor.com/visualize.html#mode=display'. It displays a Python 3.6 code editor with the following code:

```
1 def fact(n):
2     if n == 1:
3         return 1
4     return fact(n - 1) * n
5
6 fact(3)
```

Below the code, there are status indicators: a green arrow pointing right labeled 'line that has just executed' and a red arrow pointing right labeled 'next line to execute'. A note says 'Click a line of code to set a breakpoint; use the Back and Forward buttons to jump there.' At the bottom are navigation buttons: '<< First', '< Back', 'Step 12 of 14', 'Forward >', and 'Last >>'. To the right, there are two columns: 'Frames' and 'Objects'. The 'Frames' column shows three frames: 'Global frame' (fact), 'fact' (n=3), and 'fact' (n=2). The 'Objects' column shows a 'function fact(n)' and a 'Return value' of 1.

<http://pythontutor.com/visualize.html#mode=edit>



# Bibliographie

- *Informatique pour tous en CPGE*, Benjamin Wack *et al.*, Eyrolles, 2013
- *Data Structures and Algorithms in Python*, Michael T. Goodrich *et al.*, Wiley, 2013
- *Python Algorithms : Mastering Basic Algorithms in the Python Language*, Magnus Lie Hetland, APress, 2014
- PEP-8, <https://www.python.org/dev/peps/pep-0008/>
- <https://docs.scipy.org/doc/numpy/user/quickstart.html#the-basics>
- <https://www.youtube.com/user/cs50tv>