



Une introduction à Python

Christophe Jorssen

christophe.jorssen@gmail.com

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

Lycée Jacques Decour – Paris

Jeudi 4 avril 2019



❖ 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



Oscillateur quasi sinusoïdal

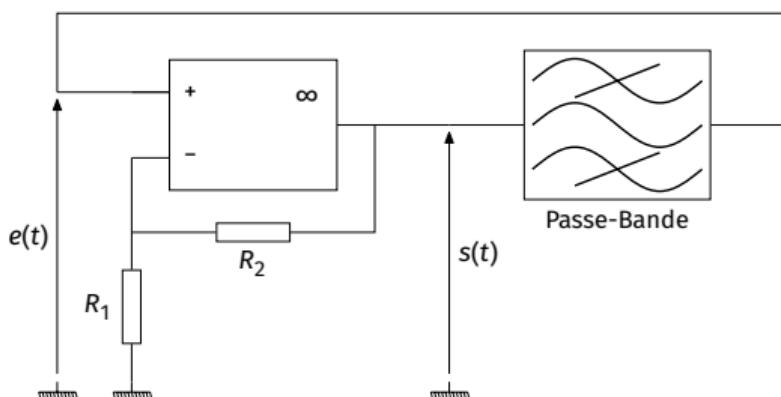
1 Un peu de « vécu »

■ 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

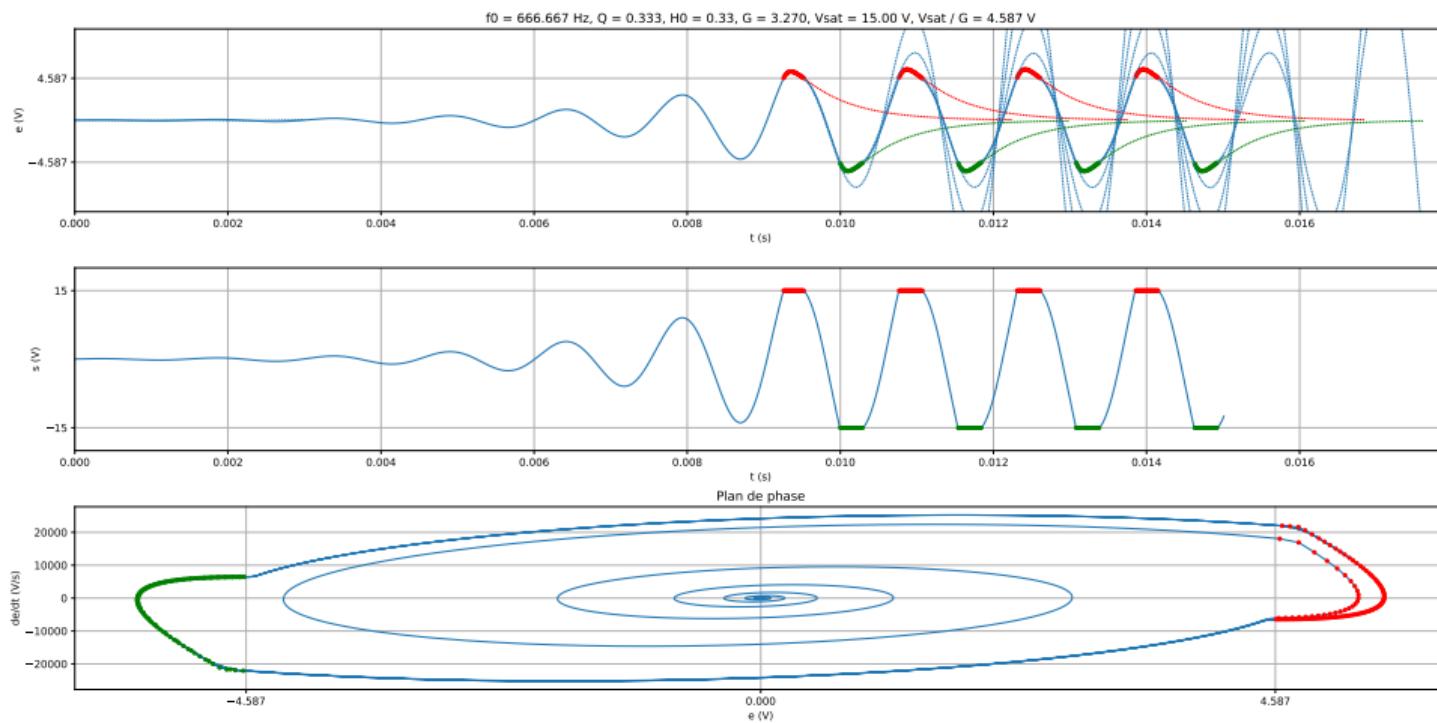
$$\frac{d^2s}{dt^2} + (1 - GH_0) \frac{\omega_0}{Q} \frac{ds}{dt} + \omega_0^2 s = 0;$$

- 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



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 »



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 »



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 »

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 »**



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



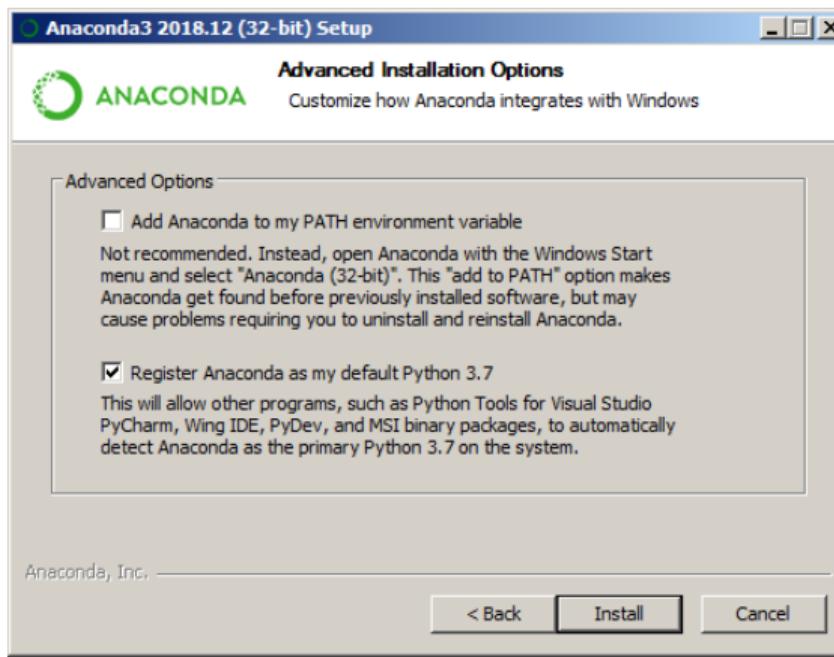
Lors de l'installation de Python, il faut :

- 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

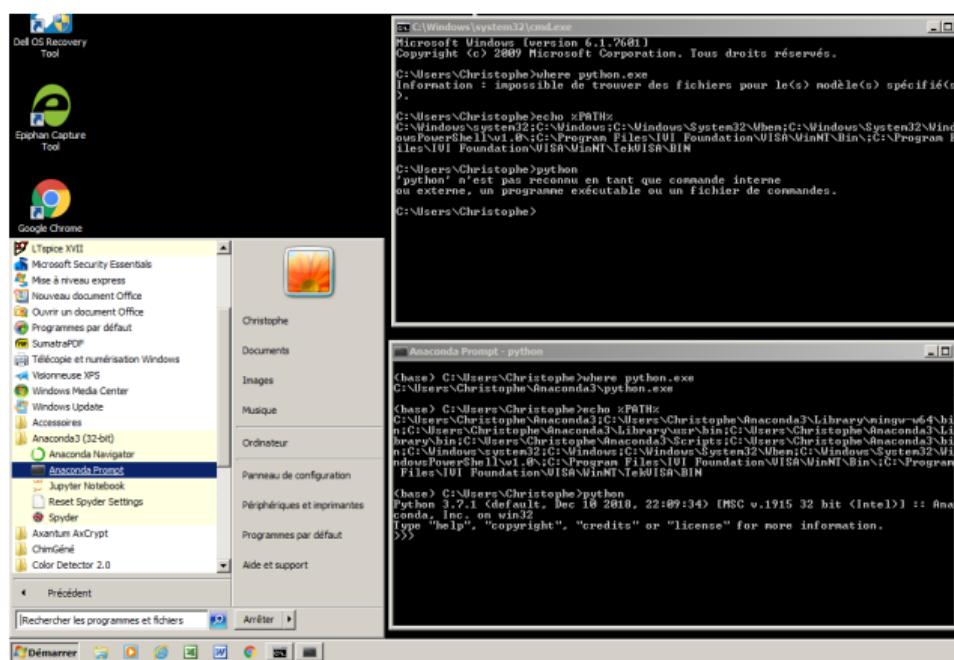
Installation d'Anaconda (MS-Windows)





Installation d'Anaconda

Installation d'Anaconda (MS-Windows)





Installation d'Anaconda

Installation d'Anaconda (MacOS)

A screenshot of a macOS terminal window titled "christophe — -bash — 107x27". The window contains the following text:

```
(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$
```

The terminal window is set against a dark background. Below the terminal is the macOS Dock with various application icons. At the bottom of the screen are standard Mac OS X navigation icons.

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



Mise à jour et installation de paquets complémentaires

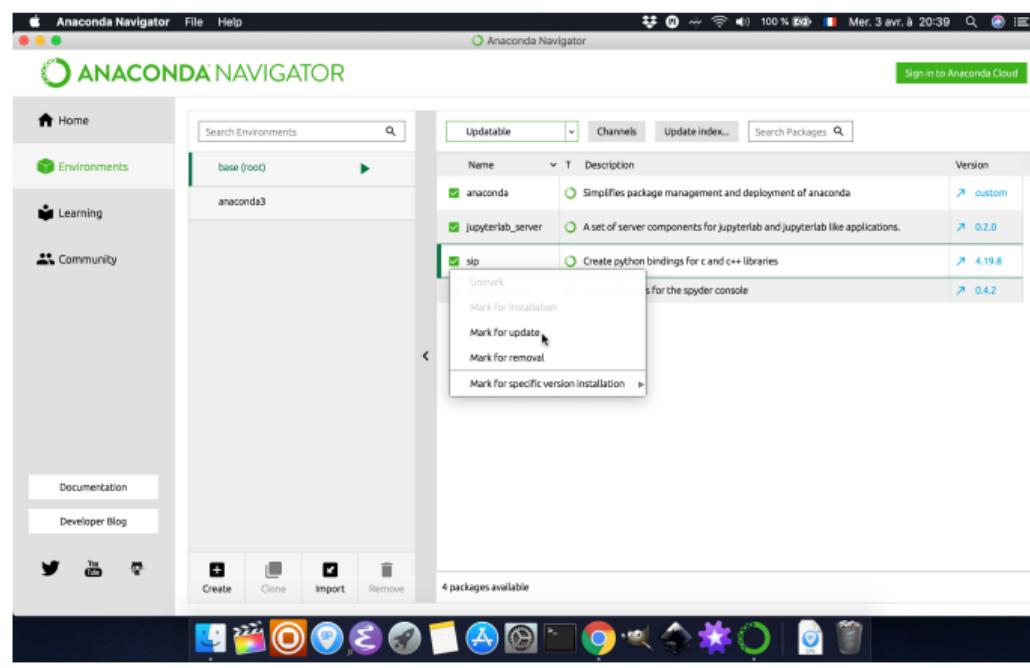
2 Installation et mise à jour de l'écosystème Python

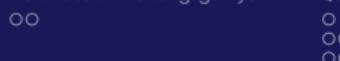
- Nécessité d'une distribution
- Installation d'Anaconda
- Mise à jour et installation de paquets complémentaires



Mise à jour et installation de paquets complémentaires

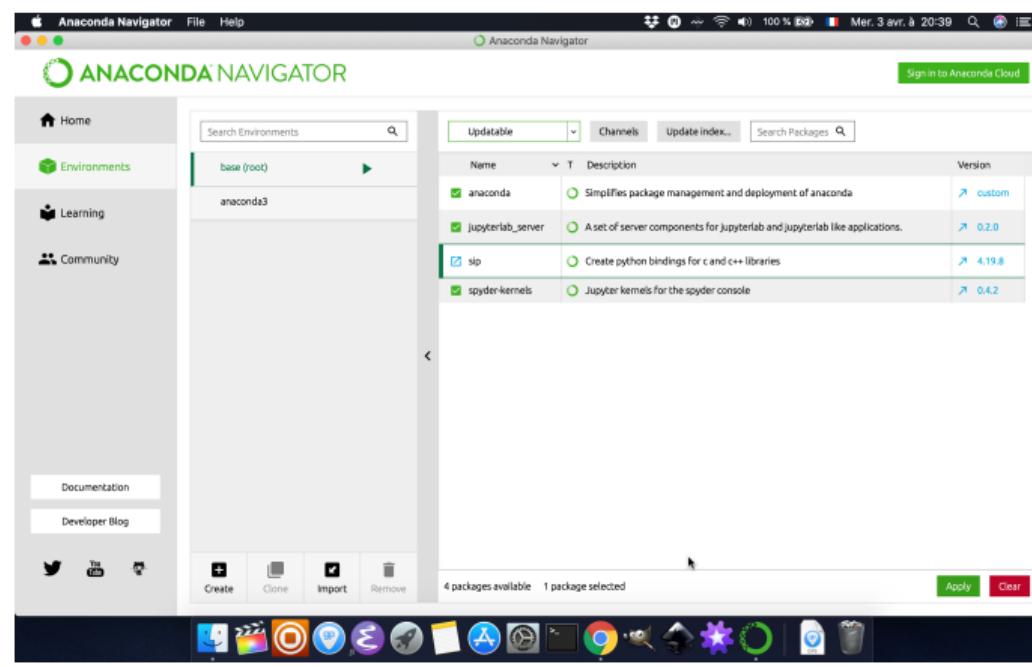
À l'aide de l'interface graphique : mise à jour





Mise à jour et installation de paquets complémentaires

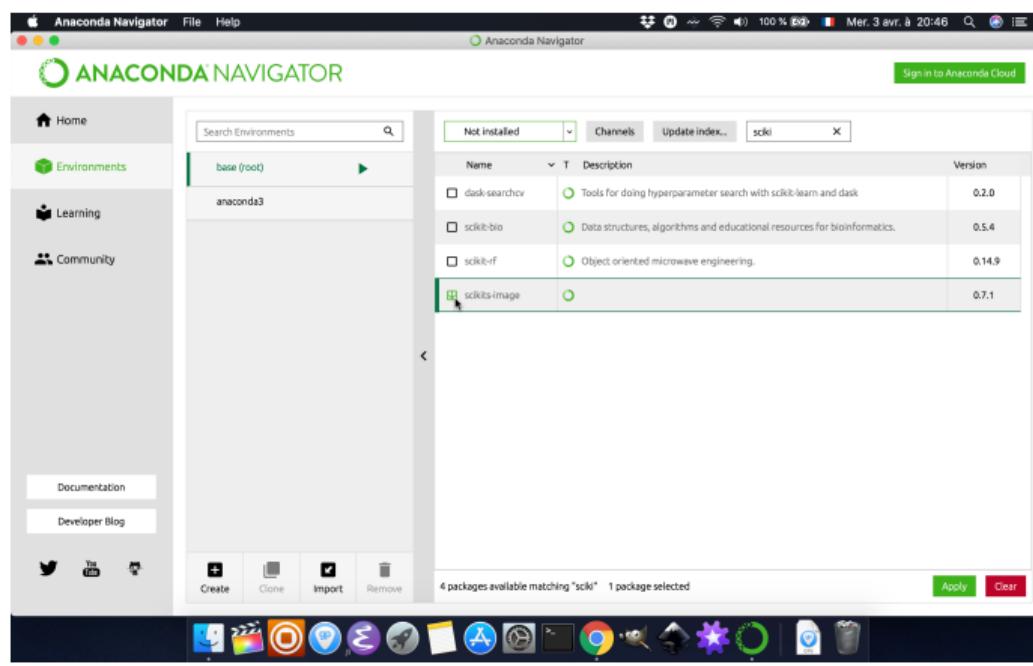
À l'aide de l'interface graphique : mise à jour





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

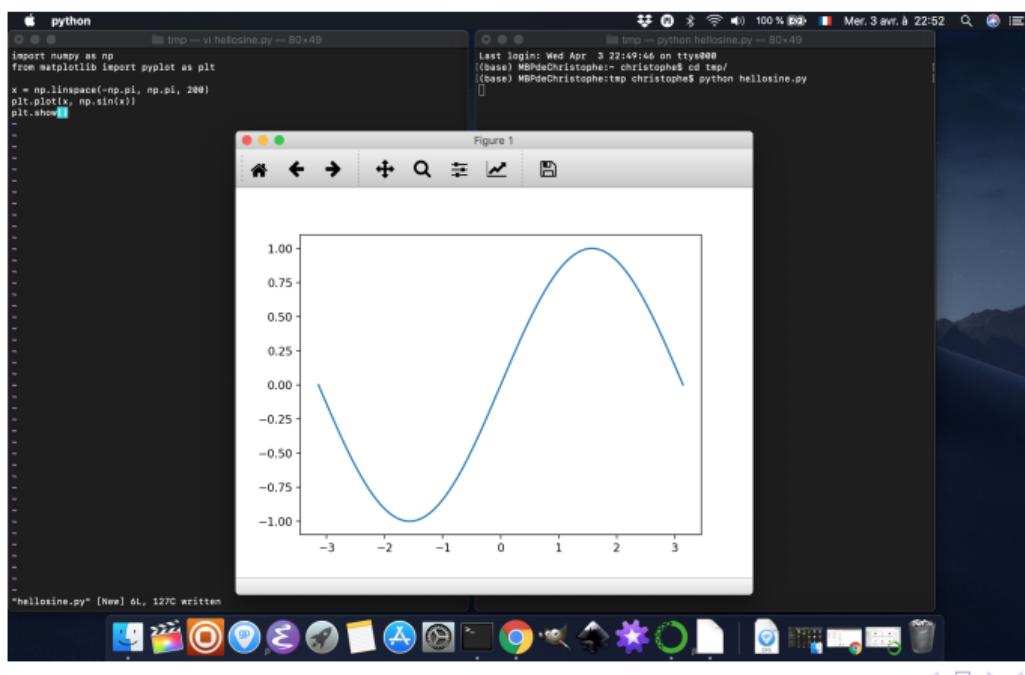
3 Quelques façons d'utiliser Python

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



Éditeur et ligne de commande

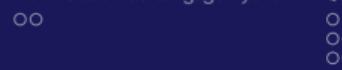
Un exemple *hardcore* : éditeur et exécution dans la console



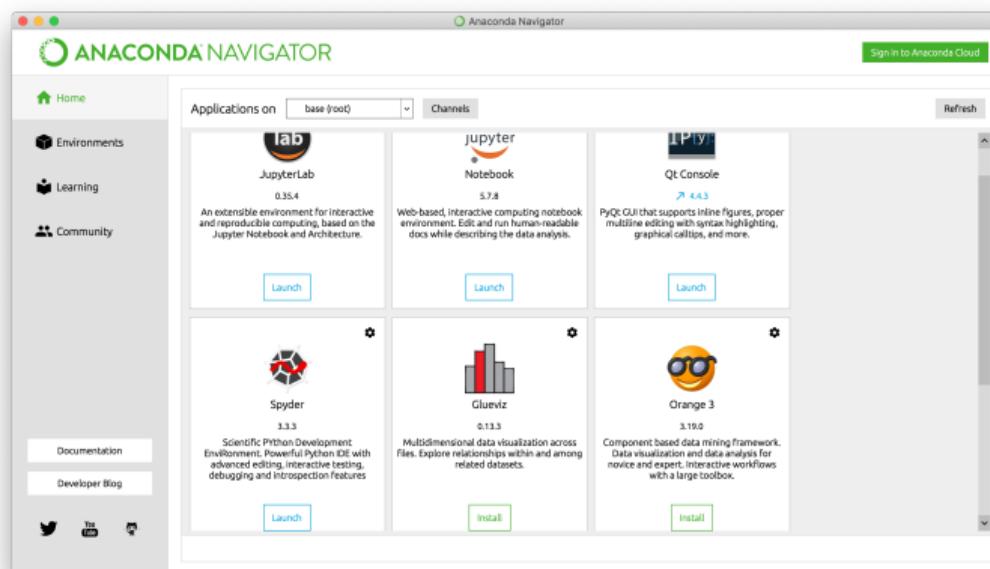


3 Quelques façons d'utiliser Python

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

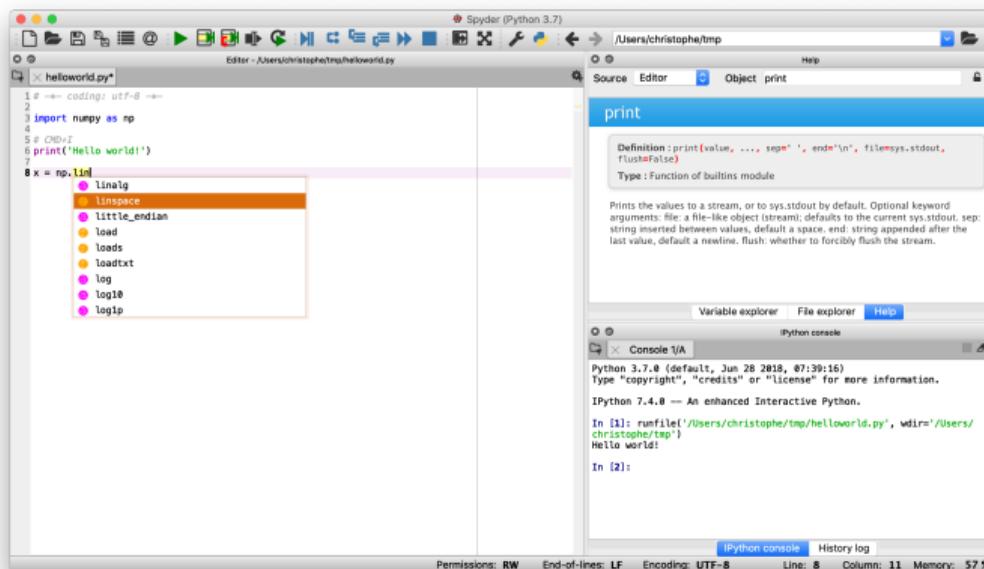


L'IDE d'Anaconda : spyder



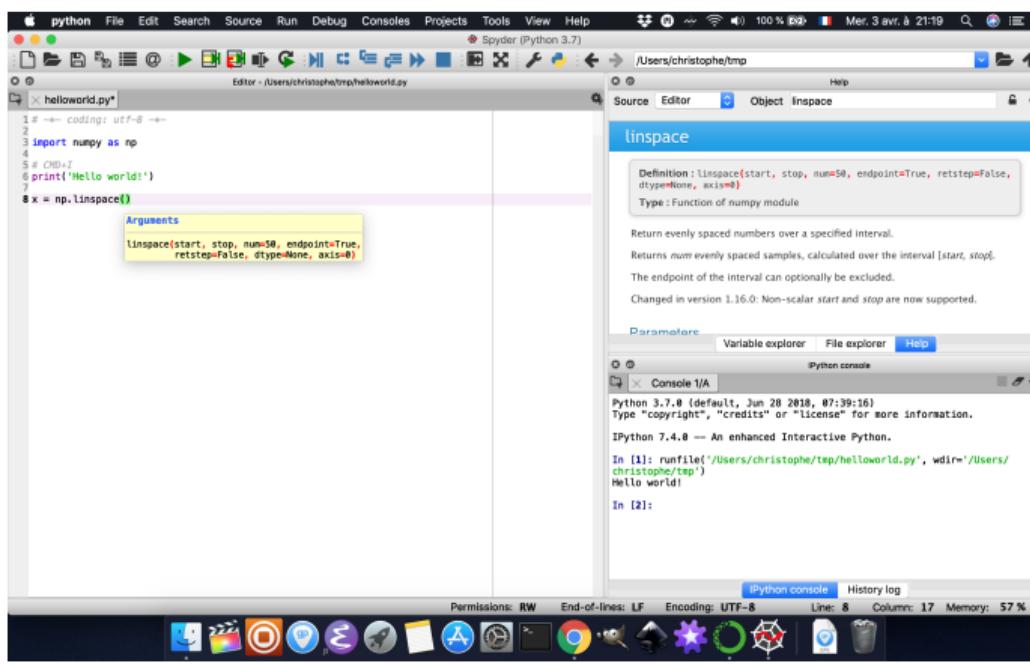
IDE

L'IDE d'Anaconda : spyder





L'IDE d'Anaconda : spyder

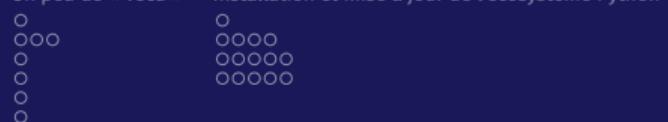




Dans le *cloud* (dans le navigateur)

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 »



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:

CS50 Sandbox

temporary programming environments for students and teachers
with support for [these environments](#)

[Create sandbox](#) [Recent sandboxes](#)

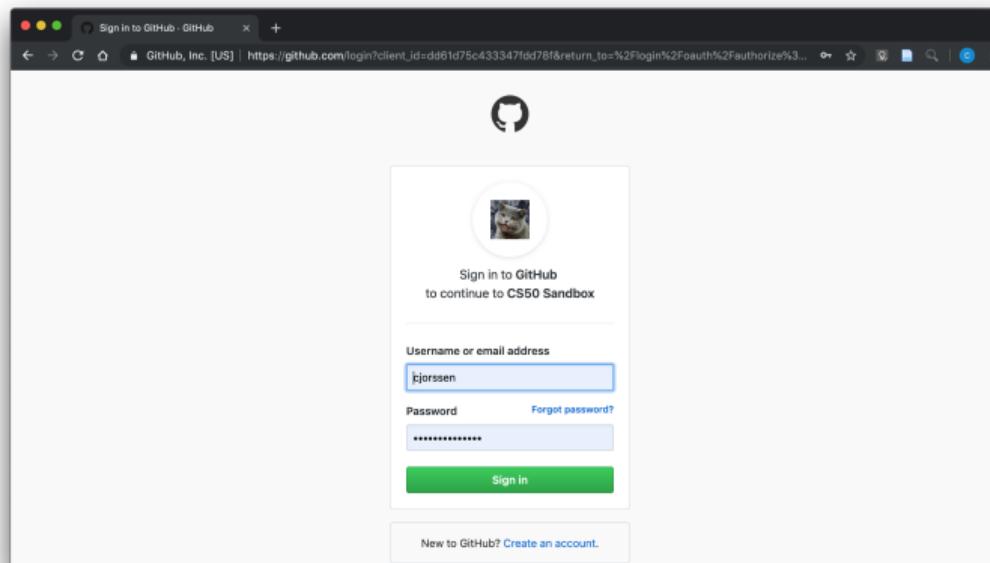
with embedded browser
 with terminal window
 with text editor, [with these tabs](#)
 with X window

[Create](#) [or configure](#)



Dans le *cloud* (dans le navigateur)

CS50 sandbox





Dans le *cloud* (dans le navigateur)

CS50 sandbox

```

File: CS50 Sandbox
https://sandbox.cs50.io/bed925e6-f115-41a6-87c3-6b522956fb37

CS50 Sandbox

hellosine.py
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()

Terminal
$ 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 3.49kB/s
Collecting matplotlib
  Downloading https://files.pythonhosted.org/packages/02/1/a/e7bd996ef3237063a427bwead2d9bf3095fb12defdf1e8
    100% |████████████████████████████████| 17.3MB 3.49kB/s
Collecting cycler==0.10.0 (from matplotlib)
  Downloading https://files.pythonhosted.org/packages/f7/d/e/0fd1eb3b27af7696400ce7e754c59dd546ffeb3be732c8ab
    100% |████████████████████████████████| 13.4MB 3.49kB/s
Collecting pyparsing==2.3.1 (from matplotlib)
  Downloading https://files.pythonhosted.org/packages/da/9/001be538836743d8beec2d85869fa8fcf704a8c8cfc5f5fb8
    100% |████████████████████████████████| 13.4MB 3.49kB/s
Collecting kiwisolver==1.1.1 (from matplotlib)
  Downloading https://files.pythonhosted.org/packages/5c/74b40bbf4e2470366d3939dd42bb
    100% |████████████████████████████████| 9264 4.99kB
Requirement already satisfied: python-dateutil==2.1 in /opt/pyenv/versions/3.7.1/lib/python3.7/site-packages
  (from matplotlib==2.2.2)
Requirement already satisfied: six in /opt/pyenv/versions/3.7.1/lib/python3.7/site-packages (from cycler>=0.1
  0-matplotlib)
Requirement already satisfied: pyparsing==2.3.1 in /opt/pyenv/versions/3.7.1/lib/python3.7/site-packages (from kiwi
  solver==1.1.1)
Installing collected packages: numpy, cycler, pyparsing, kiwisolver, matplotlib
Successfully installed cycler==0.10.0 numpy==1.16.2 pyparsing==2.3.1
You are using Python 3.7.1, however version 3.6.5 is available.
You should consider upgrading via the pip install --upgrade pip command.
$ python hellosine.py

```



Dans le *cloud* (dans le navigateur)

repl.it

The screenshot shows the repl.it web-based development environment. On the left, there's a sidebar with file management icons and a list of files: 'main.py' and 'hellosine.png'. The main workspace has a 'run' button and a terminal window. The terminal window displays the following Python code and its execution:

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

Below the terminal, a message says: "Some files may fail to load on the browser. Click here to download". At the bottom, there's a preview of the generated 'hellosine.png' image, which is a plot of the sine function from -pi to 2*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 \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).



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.



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 un *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

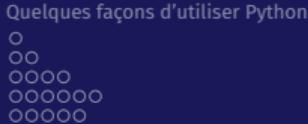
5 Quelques outils pour l'enseignement de Python



Blockly

5 Quelques outils pour l'enseignement de Python

- Blockly
- Python tutor



Blockly

De Scratch vers Python

The screenshot shows the Blockly website interface. On the left, a Scratch script is displayed, consisting of a 'repeat' loop with a counter variable 'Count'. Inside the loop, there is a 'print' block followed by a 'create text with' block containing the string 'Hello World'. A 'set Count to 10' block is also present. On the right, the generated Python code is shown:

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

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

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

○
○○
●○

Python tutor

5 Quelques outils pour l'enseignement de Python

■ Blockly

■ Python tutor



Python tutor

The screenshot shows the Python Tutor visualizer interface. On the left, a code editor displays Python 3.6 code for calculating the factorial of 3:1 def fact(n):
2 if n == 1:
3 return 1
4 return fact(n - 1) * n
5
6 fact(3)

Annotations below the code explain symbols: a green arrow points to the line 'line that has just executed', and a red arrow points to the line 'next line to execute'. A horizontal slider at the bottom allows jumping between steps. The right side of the interface shows the execution frames and objects. It starts with a 'Global frame' containing a 'fact' variable pointing to a 'function fact(n)' entry. Three subsequent frames show the state of variables 'fact' and 'n' as the function recurses down to n=1, where it returns the value 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>