



# Python - Programmation avancée

[Mise à jour le : 17/12/2022]

- [Python 3.11: Cool New Features for You to Try](#)

## Sommaire

1. AI
  1. [Build an Alexa- or Siri-Equivalent Bot in Python Using OpenAI](#)
2. Algorithmes, maths et science des données
  1. [Sorting Algorithms in Python](#)
  2. [Recursion in Python: An Introduction](#)
  3. [The k-Nearest Neighbors \(kNN\) Algorithm in Python](#)
  4. [Simplify Complex Numbers With Python](#)
  5. [Math for Data Science](#)
  6. [Python Data Science Tutorials](#)
  7. [Logistic Regression in Python](#)
  8. [NumPy, SciPy, and Pandas: Correlation With Python](#)
  9. [Fourier Transforms With scipy.fft: Python Signal Processing](#)
  10. [Draw the Mandelbrot Set in Python](#)
  11. [How to Do a Binary Search in Python](#)
  12. [Data Visualization With Python](#)
  13. [Linear Regression in Python](#)
3. API
  1. [How to Do a Binary Search in Python](#)
4. Asynchrone
  1. [Getting Started With Async Features in Python](#)
5. Bases de données et SGBD
  1. [Généralités](#)
  2. [Python Database Tutorials](#)
  3. [Des bases de données en Python avec sqlite3](#)
  4. [Build a Contact Book With Python, PyQt, and SQLite](#)
  5. [Introduction to Python SQL Libraries](#)
  6. [Recipes from Python SQLite docs](#)
6. Bonnes pratiques
  1. [Technical Writing for Developers](#)
  2. [Best Practices for More Pythonic Code](#)
  3. [Python vs C++: Selecting the Right Tool for the Job](#)
  4. [Write More Pythonic Code](#)
  5. [LBYL vs EAFP: Preventing or Handling Errors in Python](#)
  6. [When Do You Use an Ellipsis in Python?](#)
  7. [Should You Be Using Python's Walrus Operator?](#)
7. CLI Python
  1. [Build Command Line Interface with Python's argparse](#)
  2. [Python Command Line Arguments](#)

8. CPython
  1. [Your Guide to the CPython Source Code](#)
9. Documentation
  1. [Build Your Python Project Documentation With MkDocs](#)
10. Embarqué
  1. [PikaScript](#)
11. Excel
  1. [A Guide to Excel Spreadsheets in Python With openpyxl](#)
12. Exceptions
  1. [Python Exceptions: An Introduction](#)
  2. [Python KeyError Exceptions and How to Handle Them](#)
  3. [LBYL vs EAFP: Preventing or Handling Errors in Python](#)
13. Exécution de code
  1. [Python's exec\(\): Execute Dynamically Generated Code](#)
14. Fonctionnel
  1. [Python's filter\(\): Extract Values From Iterables](#)
15. Générateurs
  1. [Python's filter\(\): Extract Values From Iterables](#)
16. Interfaces graphiques
  1. [Python GUI Programming](#)
  2. [GUI Programming With PyQt](#)
17. Journalisation
  1. [Logging in Python](#)
  2. [The Ins and Outs of Logging in Python, Part 1](#)
18. Jupyter
  1. [How to create a dashboard in Python with Jupyter Notebook?](#)
19. Langages
  1. [Natural Language Processing With Python's NLTK Package](#)
  2. [Java vs Python: Basic Python for Java Developers](#)
20. Linter
  1. [Learn Python ASTs, by building your own linter](#)
21. Machine learning
  1. [Machine Learning With Python](#)
  2. [Python AI: How to Build a Neural Network & Make Predictions](#)
22. Mémoire
  1. [Memory Management in Python \(RealPython\)](#)
  2. [Memory Management in Python \(Honeybadger\)](#)
  3. [Memory Management in Python - Part 1: What Are Pointers?](#)
23. Mobile
  1. [Build a Mobile Application With the Kivy Python Framework](#)
24. Modules et packages
  1. [Python Zip Imports: Distribute Modules and Packages Quickly](#)
  2. [Dependency Management With Python Poetry](#)
  3. [How to Publish an Open-Source Python Package to PyPI](#)
  4. [How to create a Python package in 2022](#)
  5. [Using Python's pip to Manage Your Projects' Dependencies](#)
25. Multitâche
  1. [Multiprocessing Race Conditions in Python](#)
  2. [Python Concurrency & Parallel Programming](#)
  3. [Threading in Python: The Complete Guide](#)
  4. [Speed Up Your Python Program With Concurrency](#)

5. [The subprocess Module: Wrapping Programs With Python](#)
6. [map\(\) vs submit\(\) With The ProcessPoolExecutor in Python](#)
7. [Python Threading Quiz](#)
26. pdf
  1. [pikepdf : a Python library for reading and writing PDF files](#)
27. **POO - Programmation Orientée Objet**
  1. [Classe et objet](#)
  2. [Héritage](#)
  3. [Polymorphisme](#)
28. Sécurité
  1. [Evolution of Access Control Explained Through Python](#)
  2. [Secure Password Handling in Python](#)
29. Structures de données
  1. Enumération
    1. [Build Enumerations of Constants With Python's Enum](#)
  2. Tuple
    1. [Write Pythonic and Clean Code With namedtuple](#)
  3. File et pile
    1. [Python Stacks, Queues, and Priority Queues in Practice](#)
    2. [Python's deque: Implement Efficient Queues and Stacks](#)
    3. [How to Implement a Python Stack](#)
    4. [Reverse Python Lists: Beyond .reverse\(\) and reversed\(\)](#)
    5. [Build a Hash Table in Python With TDD](#)
  4. Dictionnaires
    1. [Python's ChainMap: Manage Multiple Contexts Effectively](#)
30. Tests
  1. [Python's doctest: Document and Test Your Code at Once](#)
  2. [Test Your Python Apps](#)
  3. [Effective Python Testing With Pytest](#)
  4. [Documentation unit tests](#)
  5. [Using Mypy in production at Spring](#)
  6. [Invasive procedures: Python affordances for performance measurement](#)
31. Web
  1. [Python Web Development Tutorials](#)
  2. [HTML and CSS for Python Developers](#)
  3. [Model-View-Controller \(MVC\) Explained - With Legos](#)
  4. [Python's Requests Library \(Guide\)\[Quiz\]](#)
  5. [Python's urllib.request for HTTP Requests](#)
  6. [Exploring HTTPS With Python](#)
  7. [Bottle](#)
    1. [Un serveur web en Python avec \*\*Bottle\*\*.](#)
  8. [Django](#)
    1. [Django Tutorials](#)
    2. [Django vs. Flask in 2021: Which Framework to Choose](#)
  9. [Docker](#)
    1. [Python Docker Tutorials](#)
  10. [Flask](#)
    1. [Flask by Example](#)
    2. [Flask Tutorials](#)
    3. [Accepting Payments with Stripe, Vue.js, and Flask](#)
    4. [Primer on Jinja Templating](#)

5. [Sending Emails With Python](#)
  6. [Front-end Web Development Tutorials](#)
  11. [Brython: Python in Your Browser](#)
    1. [Python vs JavaScript for Pythonistas, etc](#)
  12. [MQTT](#)
    1. [Fundamentals of MQTT](#)
    2. [Beginner's Guide To Using Paho-MQTT](#)
  13. [PyScript](#)
    1. [A First Look at PyScript: Python in the Web Browser](#)
    2. [PyScript — unleash the power of Python in your browser](#)
  14. [REST](#)
    1. [Python's urllib.request for HTTP Requests](#)
    2. [Python and REST APIs: Interacting With Web Services](#)
  15. [Scraping](#)
    1. [A Practical Introduction to Web Scraping in Python](#)
    2. [Web Scraping Tutorials](#)
  16. [Socket](#)
    1. [Socket Programming in Python \(Guide\)](#)
  17. [Twitter](#)
    1. [How to Make a Twitter Bot in Python With Tweepy](#)
  32. [XML](#)
    1. [A Roadmap to XML Parsers in Python](#)
- 

## 1. Outils

### 1. **Création**

1. [Kivy: The Open Source Python App development Framework.](#) for Android, iOS, GNU/Linux, OS X et Windows. .
2. [Thonny: The Beginner-Friendly Python Editor](#)
3. **VSCode**
  1. [L'environnement de développement intégré \(IDE\) VSCode](#)
  2. [Python interactif \(IPython\) et Jupyter dans VSCode](#)
  3. [Installer et utiliser Anaconda, Anaconda avec VSCode](#)
4. [Notebook Jupyter, binder et Google Colab](#)
5. [Installer et utiliser PIP](#)
  1. [Using Python's pip to Manage Your Projects' Dependencies](#)
6. [Visual TK : Visual Python TKinter GUI creator](#)

### 2. **Docker**

1. [How to Deploy a Jupyter Notebook to Docker](#)

### 3. **Documentation**

1. [Mémento de Markdown](#)

### 4. **Entraînement en ligne**

1. [INGInious](#)
2. [Cercles informatiques](#)
3. [France lol](#)

### 5. **Environnement d'exécution**

1. [Environnements virtuels Python : introduction](#)
2. [pipx — Install and Run Python Applications in Isolated Environments](#)
3. [Managing Multiple Python Versions With pyenv](#)

### 6. **Sauvegarde et collaboration**

1. [Transférer des fichiers avec FileZilla client](#)
2. [Travail collaboratif dans VSCode](#)
3. **Git et Github**
  1. [Gestion de versions : démarrer avec git et Github](#)
  2. [Introduction to Git and GitHub for Python Developers](#)
  3. [Advanced Git Tips for Python Developers](#)
7. **Script**
  1. [File injector \(Stéganographie\)](#)
8. **Simulation**
  1. **En ligne** : [Python Tutor](#)

From:

<https://webge.fr/dokuwiki/> - **WEBGE Wikis**

Permanent link:

<https://webge.fr/dokuwiki/doku.php?id=python:progavancee&rev=1671273075>

Last update: **2022/12/17 11:31**

