



Bienvenue sur Python, MicroPython et CircuitPython

Rédacteur(s) : Philippe Mariano

[Mise à jour le 1/6/2023]



Ce Wiki, consacré aux bases de Python et aux cartes à microcontrôleur programmables en "MicroPython", est destiné à des élèves de lycée inscrits dans la spécialité NSI.

Présentation

- **Python** est un langage de programmation interprété, multiparadigme et multiplateformes. Il favorise la programmation impérative structurée, fonctionnelle et orientée objet. Python est distribué sous licence libre et présente une syntaxe épurée et simplifiée, ce qui en fait un outil adapté à l'apprentissage de la programmation. Ce wiki traite uniquement la version 3.
- **MicroPython** est une implémentation simple et efficace du langage de programmation Python 3, qui inclut un petit sous-ensemble de la bibliothèque standard Python et qui est optimisée pour fonctionner sur des microcontrôleurs. Il est suffisamment compact pour s'adapter à 256 ko d'espace de code et à 16 ko de RAM.
- **CircuitPython** est un dérivé open source du langage de programmation MicroPython destiné aux étudiants et aux débutants. Le développement de CircuitPython est soutenu par Adafruit Industries. Il s'agit d'une implémentation logicielle du langage de programmation Python 3, écrit en C.

Python News

Sommaire

1.  **MICROPYTHON, CIRCUITPYTHON**
 1. [Embedded Python: Build a Game on the BBC micro:bit](#)
 2. [BrainPad - Python lessons](#)
 3. [Wio - Premiers pas avec ArduPy](#)
 4. [Build Physical Projects With Python on the Raspberry Pi](#)
2.  **PYTHON**
 1. **INSTALLATION - DEMARRAGE**
 1. REPL : l'interpréteur de commandes

2. IDLE : l'éditeur fourni avec Python

3.  **VSCode - Premier programme en Python "étape par étape"**

4. plus loin - [Managing Multiple Python Versions With pyenv](#)

2. BASES DU LANGAGE

1. [Variables, constantes, types numériques et E/S dans la console](#)

2. [Numbers in Python](#)

3. [Les fonctions](#)

4. Les structures de données

1. [Common Python Data Structures \(Guide\)](#)

2. Opérateurs

1. [Python's "in" and "not in" Operators: Check for Membership](#)

3. Les séquences

1. [Généralités](#)

2. [Les chaînes de caractères](#)

3. [Listes](#)

4. [Tuples](#)

4. [Les dictionnaires](#)

5. [Les ensembles \(set\)](#)

5. [Les modules et packages](#)

6. [Les fichiers](#)

7. [Tests, exceptions, assertions et module doctest](#)

8. Documentation

1. [Documenting Python Code: A Complete Guide](#)

9. Script

1. [How to Run Your Python Scripts. \[Quiz\]](#)

10. PEP 8

1. [Bonnes pratiques de codage](#)

2. [How to Write Beautiful Python Code With PEP 8](#)

3. **[3 Tips For Writing Pythonic Code](#)**

11. **[Quiz](#)**

3. PROGRAMMATION INTERMEDIAIRE

1. [Intermediate Python Tutorials](#)

2. [Namespaces and Scope in Python](#)

3. [Python Inner Functions: What Are They Good For?](#)

4. [How to Get and Use the Current Time in Python](#)

5. [Working With JSON Data in Python](#)

4.  **PROGRAMMATION AVANCEE ...**

3. JEUX

1. [Top Python Game Engines](#)

2. [Set Up the Diagram of Dice Faces](#)

3. [Make Your First Python Game: Rock, Paper, Scissors!](#)

4. [PyGame: A Primer on Game Programming in Python](#)

5. [Arcade: A Primer on the Python Game Framework](#)

6. [Build an Asteroids Game With Python and Pygame](#)

7. [Build a Platform Game in Python With Arcade](#)

8. [Build a Tic-Tac-Toe Game Engine With an AI Player in Python](#)

4. PROJETS

1. [Learn by Doing](#)

2. [190 Python Projects with Source Code](#)

3. [70+ Python Projects for Beginners, Intermediate and Experienced Developers](#)
4. [Il pleut dehors? Créer une application CLI météo avec Python](#)
5. [Advent of Code: Solving Your Puzzles With Python](#)
6. [Build a Command-Line To-Do App With Python and Typer](#)
7. [Draw the Mandelbrot Set in Python](#)
8. [Build a Bulk File Rename Tool With Python and PyQt](#)
9. [Build a Site Connectivity Checker in Python](#)
10. [Build a URL Shortener With FastAPI and Python](#)
11. [Build a Quiz Application With Python](#)
12. [Python and PyQt: Building a GUI Desktop Calculator](#)
13. [ChatterBot: Build a Chatbot With Python](#)

5. Outils

1. Création

1. [Thonny: The Beginner-Friendly Python Editor](#)
2. **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](#)
3. [Notebook Jupyter, binder et Google Colab](#)
4. [Installer et utiliser PIP](#)

2. Documentation

1. [Mémento de Markdown](#)

3. Entraînement en ligne

1. [INGInious](#)
2. [Cercles informatiques](#)
3. [France lol](#)
4. [Learn by Doing](#)

4. 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](#)

5. Simulation

1. **En ligne** : [Python Tutor](#)

6. CONCOURS

1. [Castor](#) (du CM1 à la terminale)
2. [Algoréa](#) (collège, lycée)
3. [Alkindi](#) (cryptanalyse, secondes)
4. [Prologin](#)
5. [Les trophées NSI](#)

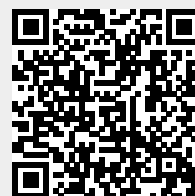
- [Bibliographie](#)
- [Webographie](#)
- [Glossaire](#)

From:

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

Permanent link:

<https://webge.fr/dokuwiki/doku.php?id=python:accueilpython&rev=1685714755>



Last update: **2023/06/02 16:05**