Bienvenue sur Python, MicroPython et CircuitPython

Rédacteur(s) : Philippe Mariano

[Mise à jour le 2/8/2021]

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.
- **MicroPython vs CircuitPython** : contrairement à MicroPython, CircuitPython ne permet pas de faire du multithreading.
- **INFORMATIONS** : Python News: What's New From March 2021?

Sommaire

    2. Type d'une variable, copie, permutation, opérations
    3. Portée
    4. Entrées / sorties dans la console
  2. Les instructions de contrôle
    1. alternatives
    2. répétitives
  3. Les fonctions
    1. Création, appel, passage de paramètres, signature
    2. Fonctions lambda
    3. Fonctions natives (built-in)
  4. Les structures de données
    1. Common Python Data Structures (Guide)
    2. Les séquences
      1. Généralités
      2. Les chaînes de caractères
      3. Listes
      4. Tuples
    3. Les dictionnaires
    4. Les ensembles (set)
  5. Les modules et packages
    1. Les modules : utilisation et création
    2. Turtle, Numpy, Matplotlib, etc.
    3. Les packages
  6. Les fichiers
  7. Les exceptions, assertions et le module doctest
  8. Documentation
    1. Documenting Python Code: A Complete Guide
  9. Script
    1. How to Run Your Python Scripts. [Quiz]
  10. PEP 8 - Bonnes pratiques de codage
4. **Programmation avancée**
  1. 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
  2. API
    1. Using FastAPI to Build Python Web APIs
  3. Asynchrone
    1. Getting Started With Async Features in Python
  4. 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

5. Bonnes pratiques
    1. Best Practices for More Pythonic Code
    2. Python vs C++: Selecting the Right Tool for the Job
    3. Write More Pythonic Code
6. CLI Python
    1. Python Command Line Arguments
7. Fonctionnel
    1. Python's filter(): Extract Values From Iterables
8. Générateurs
    1. How to Use Generators and yield in Python
9. Interfaces graphiques
    1. Python GUI Programming
    2. GUI Programming With PyQt
10. Langage
    1. Natural Language Processing With Python's NLTK Package
11. Machine learning
    1. Python AI: How to Build a Neural Network & Make Predictions
12. Mobile
    1. Build a Mobile Application With the Kivy Python Framework
13. Multitâche
    1. Speed Up Your Python Program With Concurrency
    2. Python Concurrency Quiz
    3. Python Threading Quiz
14. Programmation Orientée Objet
    1. Classe et objet
    2. Héritage
    3. Polymorphisme
15. Structures de données
    1. How to Implement a Python Stack
16. Web
    1. Model-View-Controller (MVC) Explained – With Legos
    2. Python’s Requests Library (Guide)[Quiz]
    3. Exploring HTTPS With Python
    4. Bottle
        1. Un serveur web en Python avec **Bottle**.
    5. Django
        1. Django Tutorials
        2. Django vs. Flask in 2021: Which Framework to Choose
    6. Docker
        1. Python Docker Tutorials
    7. Flask
        1. Flask Tutorials
    8. Sending Emails With Python
    9. Front-end Web Development Tutorials
        1. Brython: Python in Your Browser
        2. Python vs JavaScript for Pythonistas, etc
    10. MQTT
        1. Fundamentals of MQTT
        2. Beginner’s Guide To Using Paho-MQTT
    11. REST
        1. Python and REST APIs: Interacting With Web Services

            12. Twitter
                1. How to Make a Twitter Bot in Python With Tweepy
            13. Web scraping
                1. A Practical Introduction to Web Scraping in Python
                2. Python Web Scraping
                3. Beautiful Soup: Build a Web Scraper With Python
        5. **Jeux**
            1. Make Your First Python Game: Rock, Paper, Scissors!
            2. Arcade: A Primer on the Python Game Framework
            3. Build an Asteroids Game With Python and Pygame
            4. Build a Platform Game in Python With Arcade
        6. **Projets**
            1. 70+ Python Projects for Beginners, Intermediate and Experienced Developers
    3. **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
            5. Managing Multiple Python Versions With pyenv
        2. **Documentation**
            1. Mémento de Markdown
        3. **Entraînement en ligne**
            1. INGInious
            2. France Ioi
        4. **Environnement d'exécution**
            1. pipx — Install and Run Python Applications in Isolated Environments
        5. **Sauvegarde et collaboration**
            1. Transférer des fichiers avec FileZilla client
            2. Travail collaboratif dans VSCode
            3. **Git etGithub**
                1. Gestion de versions : démarrer avec git et Github
                2. Introduction to Git and GitHub for Python Developers
    4. **Concours**
        1. Castor (du CM1 à la terminale)
        2. Algoréa (collège, lycée)
        3. Alkindi (cryptanalyse, secondes)
        4. Prologin

- **Ressources**
    - Memento Python 3.x
    - **Real Python**
        - Your Guide to the CPython Source Code
- Bibliographie
- Webographie
- Glossaire