

## **Les liens utilisés ou évoqués pendant la formation Python :**

Site Officiel python

<https://www.python.org/>

Dépôt officiel pour les modules python

<https://pypi.python.org/pypi>

La librairie pour faciliter le traitement et passage d'arguments à un script python (en remplacement de optparse (<http://docs.python.org/library/optparse.html>)):

<https://docs.python.org/dev/library/argparse.html>

Le module tkFileDialog pour naviguer dans le système de fichier avec Tkinter

<https://pythonspot.com/tk-file-dialogs/>

Tutoriel / Description API Tkinter

<https://insolor.github.io/effbot-tkinterbook-archive/>

API pour tests unitaires

<https://docs.python.org/library/unittest.html>

API BeautifulSoup pour parsing xml/HTML (V4)

<https://www.crummy.com/software/BeautifulSoup/bs4/doc/>

Lancer des commandes dans l'OS

<https://docs.python.org/library/subprocess.html>

Générer dynamiquement des classes

<https://code.activestate.com/recipes/201294-dynamic-classes>

Ajouter des méthodes/attributs à des classes/instances

<http://henry.precheur.org/python/Dynamically%20create%20a%20type%20with%20Python>

Librairie pour la gestion du module Zip

<https://docs.python.org/library/zipfile.html>

Une base SQL embarquée dans Python

<https://docs.python.org/library/sqlite3.html>

Librairie pour gestion MySQL sous python

<https://pypi.org/project/MySQL-python/>

SQLAlchemy : pont entre l'objet et le relationnel

<http://www.rmunn.com/sqlalchemy-tutorial/tutorial.html>

Librairie pour expression régulière

<https://docs.python.org/library/re.html>

Les décorateurs

<https://wiki.python.org/moin/PythonDecoratorLibrary>

Tutoriel pour les décorateurs

<https://www.artima.com/weblogs/viewpost.jsp?thread=240808>

<https://openclassrooms.com/courses/apprenez-a-programmer-en-python/les-decorateurs>

<https://www.artima.com/weblogs/viewpost.jsp?thread=240845>

Librairie pour manipulation de mimetypes

<https://docs.python.org/library/mimetypes.html>

Documentation officielle python

<https://docs.python.org/library>

Un exemple de tutoriel pour démarrer avec PIL, librairie python pour le traitement d'images

<https://github.com/martinmcbride/python-imaging-book-examples>

La version française encore en ligne divintopython

<http://python.developpez.com/cours/DiveIntoPython/php/frdiveintopython>

Quelques exemples d'utilisateur de la librairie matplotlib

<https://matplotlib.org/examples>

Tutoriel pour bien comprendre les décorateurs en python (à base de kebab)

<http://fr.openclassrooms.com/informatique/cours/le-pattern-decorator-en-python>

Pour l'univers Windows, quelques package pré-compilés

<https://www.lfd.uci.edu/~gohlke/pythonlibs>

PIP : le remplaçant de easy\_install

<https://pip.pypa.io/en/latest/installation/>

Synthèse des outils pour la génération de docs sous python

<https://wiki.python.org/moin/DocumentationTools>

Bibliothèque et informations relatives aux tests joints au docstring

<https://docs.python.org/2/library/doctest.html>

Bibliothèque et informations relatives à la génération de documentation (PDF, HTML) pour des modules/packages python

<https://docs.python.org/2/library/pydoc.html>

Bibliothèque et informations relatives aux techniques pour distribuer votre egg à des tiers

<https://docs.python.org/2/distutils/builtdist.html>

Bibliothèque et informations relatives aux techniques pour distribuer les sources de votre egg à des tiers

<https://docs.python.org/2/distutils/sourcedist.html>

Pour la génération d'egg vierge

<http://pythonpaste.org/script>

Un tutoriel pour bien démarrer avec pyodbc (connecteur universel pour BDD)

<https://code.google.com/p/pyodbc/wiki/GettingStarted>

Un comparatif point à point entre Matlab et Numpy

<https://numpy.org/doc/stable/user/numpy-for-matlab-users.html>

Un tutoriel pour bien démarrer en Numpy

<https://docs.scipy.org/doc/numpy-dev/user/quickstart.html>

Bibliothèque et informations relatives aux tests de performances pour votre code

<https://docs.python.org/2/library/timeit.html>

A faire à tout moment pour se rappeler de la philosophie de votre code

import this !

Python en C, C en python... pour de meilleures performances :

<https://cython.org/>

Distribuer en .exe vos application :

<https://cx-freeze.readthedocs.io/en/latest/>

Utiliser vos librairies C en python :

<https://docs.python.org/2/library/ctypes.html>

Interfacer avec du C, les meilleures approches :

[http://www.scipy-lectures.org/advanced/interfacing\\_with\\_c/interfacing\\_with\\_c.html](http://www.scipy-lectures.org/advanced/interfacing_with_c/interfacing_with_c.html)

Cours pour la création de script shell

<https://openclassrooms.com/courses/reprenez-le-controle-a-l-aide-de-linux/introduction-aux-scripts-shell>

Cours pour l'apprentissage du XML et des DTD

<https://openclassrooms.com/courses/structurez-vos-donnees-avec-xml/qu-est-ce-que-le-xml>

Un tutoriel pour bien démarrer avec numpy

<http://cs231n.github.io/python-numpy-tutorial/>

Une librairie python complémentaire pour le tri de gros fichiers CSV

<https://pypi.python.org/pypi/csvsort>