

B

Bibliothèques tierces

La philosophie Batteries Included de Python rencontre ses limites lorsque des fonctionnalités très spécifiques sont recherchées. Cette limitation n'est cependant pas bloquante grâce à la facilité d'extension du langage : il est possible de trouver des bibliothèques tierces pour la quasi-totalité des besoins.

D'autre part, certains modules initialement présents dans la bibliothèque standard ont été volontairement délaissés au fur et à mesure des versions du langage, pour préférer des solutions externes.

Cette annexe liste un certain nombre de bibliothèques externes, les plus fréquemment utilisées, organisées par thèmes :

- bases de données ;
- traitement de texte ;
- packaging, distribution ;
- tests fonctionnels et contrôle qualité ;
- MS-Windows ;
- interfaces graphiques ;
- reporting et conversion ;
- jeux et 3D ;
- audio et vidéo ;
- bibliothèques scientifiques ;
- Web.

Chacune des bibliothèques est présentée par un court texte et une URL suit le même schéma d'installation, présenté ci-après.

Installer une bibliothèque externe

L'ensemble des bibliothèques externes présentées dans cette annexe sont très simples à installer car basées sur le module `distutils`, présenté au chapitre 13. Ces bibliothèques externes sont souvent livrées dans un fichier compressé sous la forme `NomDuPaquet-version.zip` ou `NomDuPaquet-version.tar.gz`.

Installer une extension se fait en trois étapes :

- 1 décompression du paquet, par l'outil `tar` ou équivalent ;
- 2 construction du paquet dans le répertoire de décompression, par l'option `build` du script `setup.py` ;
- 3 installation du paquet dans Python, par l'option `install` du script `setup.py`.

Lorsque la première étape est effectuée, on retrouve dans le répertoire décompressé une structure commune à toutes ces bibliothèques, à savoir :

- Un fichier `setup.py`, qui contient la configuration et l'appel au framework `distutils`.
- Un fichier `setup.cfg`, optionnellement présent, qui contient des informations supplémentaires, lorsqu'une compilation est nécessaire.
- Des informations sur l'extension, contenues dans les fichiers `INSTALL` et `README`.
- Une certain nombre de fichiers source.

La construction du paquet prépare un sous-répertoire `build` qui contient les éléments à fournir à Python.

Enfin, la dernière étape recopie ces fichiers dans le répertoire `site-packages` de l'installation de Python. Elle peut donc nécessiter les droits d'administrateur.

Installation de `lxml`

```
$ tar -xzf lxml-0.7.tgz
$ cd lxml
$ python setup.py build
running build
running build_py
creating build
creating build/lib.linux-i686-2.4
creating build/lib.linux-i686-2.4/lxml
creating build/lib.linux-i686-2.4/lxml/tests
[...]
```

```
$ sudo python setup.py install
running install
running build
running build_py
running build_ext
running install_lib
creating /usr/lib/python2.4/site-packages/lxml
creating /usr/lib/python2.4/site-packages/lxml/tests
copying build/lib.linux-i686-2.4/lxml/tests/test_etree.py -> /usr/lib/
python2.4/site-packages/lxml/tests
[...]
```

Une fois l'installation effectuée, le nouveau module doit être disponible dans le prompt.

Vérification de l'installation

```
tziade@Tarek:/home/packages/lxml$ python
Python 2.4.1 (#2, Mar 30 2005, 21:51:10)
[GCC 3.3.5 (Debian 1:3.3.5-8ubuntu2)] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> import lxml
>>>
```

À SAVOIR **Contrôler la bonne installation d'une bibliothèque**

Certaines bibliothèques fournissent des tests (souvent basés sur le framework pyUnit) qui peuvent être lancés pour vérifier que l'installation est correcte et que tout fonctionne comme prévu.

Lorsque le paquet est disponible sur PyPI, il est possible d'utiliser `setuptools` ou `pip` pour procéder à une installation automatique.

Utilisation de `setuptools`

`setuptools` est une bibliothèque tierce qui fournit des fonctionnalités au-dessus de `distutils`, dont un script d'installation de paquets disponibles sur PyPI.

► <http://peak.telecommunity.com/DevCenter/setuptools>

Son installation est simplifiée par un script appelé `ez_setup.py`, disponible sur le site du projet `setuptools`, à l'adresse : http://peak.telecommunity.com/dist/ez_setup.py.

Installation de setuptools

```
$ wget http://peak.telecommunity.com/dist/ez_setup.py
--2009-03-05 23:12:12-- http://peak.telecommunity.com/dist/ez_setup.py
Résolution de peak.telecommunity.com... 209.190.5.234
Connexion vers peak.telecommunity.com|209.190.5.234|:80...connecté.
requête HTTP transmise, en attente de la réponse...200 OK
Longueur: 9716 (9,5K) [text/plain]
Saving to: `ez_setup.py'

100%[=====>] 9.716      35,0K/s  in 0,3s

2009-03-05 23:12:12 (35,0 KB/s) - « ez_setup.py » sauvegardé [9716/9716]
$ python ez_setup.py -U setuptools
Searching for setuptools
Reading http://pypi.python.org/simple/setuptools/
Best match: setuptools 0.6c9
Processing setuptools-0.6c9-py2.6.egg
setuptools 0.6c9 is already the active version in easy-install.pth
Installing easy_install script to /Library/Frameworks/Python.framework/
Versions/2.6/bin
Installing easy_install-2.6 script to /Library/Frameworks/
Python.framework/Versions/2.6/bin

Using /Library/Frameworks/Python.framework/Versions/2.6/lib/python2.6/
site-packages/setuptools-0.6c9-py2.6.egg
Processing dependencies for setuptools
Finished processing dependencies for setuptools
```

Une fois `setuptools` installé, une nouvelle commande appelée `easy_install` est disponible. Elle installe tout paquet disponible sur PyPi, grâce à son nom.

Installation de BeautifulSoup avec `easy_install`

```
$ easy_install BeautifulSoup
Searching for BeautifulSoup
Reading http://pypi.python.org/simple/BeautifulSoup/
Reading http://www.crummy.com/software/BeautifulSoup/
Reading http://www.crummy.com/software/BeautifulSoup/download/
Best match: BeautifulSoup 3.1.0.1
Downloading http://www.crummy.com/software/BeautifulSoup/download/
BeautifulSoup-3.1.0.1.tar.gz
Processing BeautifulSoup-3.1.0.1.tar.gz
...
```

Bases de données

Python fournit les briques de base (DBAPI) à tout type de connecteur de base de données et propose quelques modules d'accès à des formats ultra standards comme BerkeleyDB. Cependant, aucun connecteur aux SGBD courants n'est intégré dans la bibliothèque standard.

Toutes les bases de données du marché peuvent être bien évidemment attaquées depuis Python, et cette section présente les connecteurs les plus courants. Elle inclut également un connecteur LDAP et un ORM.

Gadfly

Codé en Python, Gadfly est un mini-système SGBD complet. L'installation de cette extension permet de créer des fichiers de stockage qui peuvent être manipulés via le langage SQL, en mode direct ou en mode client-serveur.

Gadfly supporte une charge relativement limitée et est en général utilisé pour le prototypage d'applications client-serveur : la norme DBAPI étant respectée, ce connecteur peut être facilement interchangé sans modification de code.

► <http://gadfly.sourceforge.net/>

pysqlite

pysqlite est un connecteur compatible DBAPI vers le système SQLite. Ce système léger de SGBD (non client-serveur) est de plus en plus prisé dans les applications qui ont des besoins de stockage simples et un accès unique aux données, comme les applications web. sqlite est parfois plus rapide que les SGBD client-serveur classiques.

► <http://initd.org/tracker/pysqlite>

mysql-python

mysql-python est un connecteur vers le célèbre SGBD MySQL.

► <http://sourceforge.net/projects/mysql-python>

psycopg

Connecteur pour PostgreSQL.

► <http://initd.org/projects/psycopg1>

ODBC

Certaines bases de données sous MS-Windows peuvent être accédées par le biais de l'ODBC (Open DataBase Connectivity). La bibliothèque Python Win32 Extensions fournit un certain nombre de modules dédiés à MS-Windows, dont le module ODBC.

► <http://www.python.org/windows/win32/>

python-ldap

python-ldap expose les API de OpenLDAP 2.x et quelques utilitaires annexes (lectures LDIF). Cette bibliothèque permet d'utiliser tout type de serveur compatible avec le standard LDAP (OpenLDAP, ActiveDirectory, etc.).

► <http://python-ldap.sourceforge.net/>

SQLAlchemy

SQLAlchemy est un ORM (*Object-Relational Mapper*) très utilisé dans la communauté. Un ORM permet de manipuler une base de données à travers des objets Python.

► <http://www.sqlalchemy.org>

Traitement de texte

Le besoin le plus fréquent en traitement de texte est la gestion du format XML. La bibliothèque standard propose des modules dédiés mais qui sont de plus en plus délaissés par les développeurs, en raison de problèmes de performances et surtout par un manque cruel de simplicité : manipuler un fichier XML avec ces modules néces-

site un effort relativement important pour un développeur Python, habitué à plus de concision et de simplicité ou ne fournit pas de performances correctes.

Cette section présente une extension dédiée au traitement du XML, plus performante et naturelle à utiliser car basée sur le principe des curseurs : `lxml`.

Un autre besoin récurrent est le traitement de fichiers HTML non stricts : ce type de format n'est pas lisible par des bibliothèques XML et doit être traité spécifiquement. L'extension `BeautifulSoup` propose un outil spécialisé.

lxml

`lxml` est un bind Python codé en Pyrex de `libxml` et `libxslt` qui fournit les mêmes API qu'`ElementTree`. Très rapide, pythonique et puissante, probablement la meilleure bibliothèque XML actuelle.

▸ <http://codespeak.net/lxml/>

Beautiful Soup

Lorsqu'il s'agit de lire du contenu HTML non strict, le développeur utilise en général les modules `HTMLParser` ou `SGMLParser` de la bibliothèque standard, ou dans certains cas, scrute le contenu avec une expression régulière.

`Beautiful Soup` propose une alternative intéressante en scrutant le texte à la recherche de balises, paramètres ou contenu.

▸ <http://www.crummy.com/software/BeautifulSoup/>

Packaging, distribution

Outre l'outil standard `distutils`, il existe une extension de plus en plus utilisée pour la distribution de programmes Python, à savoir `zc.buildout`.

`zc.buildout` est un outil qui installe un environnement de bibliothèques tierces en se basant sur un fichier de configuration et `setuptools`.

▸ <http://pypi.python.org/pypi/zc.buildout>

Tests fonctionnels et contrôle qualité

En matière de tests, les modules `unittest` et `doctest` couvrent tous les besoins basiques mais ne permettent pas de mettre en œuvre facilement des tests fonctionnels, qui restent spécifiques au type d'interface de l'applicatif.

Les extensions qui offrent un environnement de développement de tests fonctionnels, que ce soit pour des applications web ou desktop, fonctionnent toutes sur le même principe : elles mettent en œuvre un pont entre les tests et l'interface utilisateur. `Twill` et `Funkload` permettent de tester des applications web et `guitest` des applications GTK. Enfin, `PyLint` et `PyFlakes` offrent de bons garde-fous, complémentaires aux tests, pour s'assurer de la qualité du code écrit.

Twill

`Twill` fournit un langage de script simple qui teste une application web via des scripts Python. L'outil effectue des requêtes vers le serveur web et analyse les résultats.

► <http://www.idyll.org/%7Et/www-tools/twill.html>

Funkload

`Funkload` est un outil basé sur `webunit`, qui écrit des tests fonctionnels en Python. Cet outil permet également de tester la montée en charge et génère des rapports complets. Les tests peuvent être conçus facilement via le navigateur grâce à `TCPWatch`.

► <http://funkload.nuxeo.org/>

guitest

Cet outil fournit des classes de base pour effectuer des tests unitaires sur des applications `PyGtk`.

► <http://gintas.pov.lt/guitest/>

PyLint

PyLint est un outil qui teste le code à la recherche d'erreurs ou de signes de mauvaise qualité. Ce programme est facilement configurable et extensible. Il est comparable à PyChecker mais propose plus de tests.

▸ <http://www.logilab.org/projects/pylint>

Pyflakes

Cet outil contrôle le code à la recherche d'erreurs, de code mort (impossible à appeler) ou de directives d'importation inutiles. Contrairement à PyChecker, cet outil n'exécute pas le code testé, ce qui le rend plus rapide et plus sécurisé.

▸ <http://divmod.org/projects/pyflakes>

MS-Windows

Il existe des bibliothèques spécialisées dans la programmation sur plate-forme MS-Windows et la technologie COM/ActiveX, à savoir les bibliothèques Win32 Extensions et win32com.

Win32 Extensions

La bibliothèque win32 présentée dans la section base de données pour l'ODBC, contient également des modules pour :

- les API win32 (un fichier d'aide WinHelp avec la liste des méthodes est fourni) ;
- les services NT ;
- les Memory Mapped Files ;
- les API win32pipe et win32 ;
- Les timers win32, etc.

▸ <http://www.python.org/windows/win32/>

win32com

win32com sert à programmer des clients ou des serveurs COM/ActiveX.

► <http://www.python.org/windows/win32com/>

Interfaces graphiques

Il existe plusieurs toolkits graphiques qui peuvent être utilisés par le biais de bibliothèques Python, pour remplacer Tkinter. Les plus répandus sont wxPython, PyQt et PyGTK.

wxPython

wxPython est une bibliothèque d'accès au toolkit wxWidgets, qui est de loin le plus portable des systèmes d'interface. Il existe en outre des outils de conception d'interfaces qui génèrent du code Python compatible avec wxPython, comme wxGlade.

► <http://www.wxpython.org/>

PyQT

PyQT est un bind vers le toolkit graphique Qt de Trolltech. Il offre un accès à des widgets très avancés, comme le contrôle texte Qscintilla, utilisé par certains éditeurs comme Eric3. En outre, QT designer est l'un des plus puissants éditeurs pour la conception d'interfaces graphiques. Attention cependant aux licences en fonction des cas d'utilisation, et des plates-formes.

► <http://www.riverbankcomputing.co.uk/pyqt/>

PyGTK

PyGTK fournit un lien entre Python et le toolkit GTK+ (Gimp toolkit), utilisé par l'environnement Gnome. L'outil Glade peut être utilisé pour concevoir des interfaces GTK et présente la même interface que wxGlade (qui s'en inspire).

► <http://www.pygtk.org/>

Reporting et conversion

En termes de reporting, il existe une bibliothèque Open Source incontournable nommée ReportLab éditée par la société éponyme. Elle génère des documents PDF et possède des fonctionnalités très puissantes.

RML2PDF est un outil de conversion du format RML vers PDF. rest2web est une bibliothèque de création de sites web statiques générés à partir de fichiers écrits au format reStructuredText.

ReportLab

Le toolkit ReportLab sert à concevoir en Python des systèmes de génération de PDF et fournit :

- un moteur de mise en page, Platypus ;
- une librairie étendue de widgets et de formes ;
- des points d'entrée pour toutes sources de données, etc.

► http://www.reportlab.org/rl_toolkit.html

RML2PDF

Le format Report Markup Language (RML) créé par la société ReportLab définit simplement un document dans un fichier de description XML. Un outil de conversion, nommé RML2PDF se charge ensuite de le convertir en PDF. Cet outil est payant, mais il existe une variante Open Source éditée par Tiny ERP.

► <http://openreport.tiny.be/index.py/static/page/trml2pdf>

reStructuredText

reStructuredText est un format texte très utilisé pour la documentation de projets Python et pour l'écriture des docstrings des modules de code. Il introduit une syntaxe très simple qui permet la mise en page de texte.

Ce format est également très utilisé dans les systèmes wikiwikiweb, pour offrir aux utilisateurs un format simple à écrire et aussi riche que le HTML. Il est facilement convertible en rendu HTML par des outils comme rest2html.

► <http://docutils.sourceforge.net/rst.html>

rest2web

rest2web permet de générer des pages HTML statiques à partir de documents écrits au format reStructuredText.

▸ <http://www.voidspace.org.uk/python/rest2web/>

Jeux et 3D

En termes de programmation de jeux et plus généralement de scènes 3D, Python est un langage de script de choix. Les toolkits Pygame et Soya 3D permettent de bénéficier de la puissance de Python dans ce domaine.

VPython propose, quant à lui, un environnement de programmation 3D temps réel propice à l'étude de la modélisation.

Il est aussi possible de programmer en plus bas niveau en accédant directement aux bibliothèques 3D par le biais par exemple de PyOpenGL.

Pygame

Pygame fournit des modules d'extension pour la programmation de jeux 3D et de programmes multimédias, basés sur la bibliothèque SDL (Simple DirectMedia Layer).

▸ <http://www.pygame.org/>

Soya 3D

Soya 3D est un moteur 3D pour Python, écrit en Pyrex et doté de toutes les fonctionnalités d'un moteur professionnel.

▸ <http://home.gna.org/oomadness/fr/soya/>

vpython

vpython propose un environnement de programmation 3D complet, en fournissant sa propre version d'IDLE qui permet de programmer et d'animer interactivement des scènes 3D. Très pratique pour l'apprentissage de la mécanique.

▶ <http://vpython.org/>

PyOpenGL

Module d'extension offrant l'accès aux API d'OpenGL depuis Python.

▶ <http://pyopengl.sourceforge.net/>

Audio et Vidéo

Le domaine multimédia n'est pas en reste grâce à des bibliothèques très complètes comme PyMedia ou des modules spécifiques comme PyAlsa.

PyMedia

PyMedia propose un ensemble de modules pour manipuler tous les types de formats audio et vidéo (mp3, ogg, avi, mpeg, etc.), modifier les échantillons par quelques filtres et piloter le matériel.

▶ <http://pymedia.org/>

PyAlsa

PyAlsa est un wrapper pour le système ALSA (*Advanced Linux Sound Architecture*)

▶ <http://respyre.org/pyalsa.html>

Bibliothèques scientifiques

Cette section regroupe différentes bibliothèques scientifiques spécialisées dans les calculs numériques comme Numerical Python et SciPy, et dans les outils dédiés à des domaines particuliers comme Biopython.

Numerical Python

Numerical Python, qui se nomme maintenant SciPy, est une bibliothèque puissante de fonctions de manipulation de matrices, de transformées de Fourier, et autres utilitaires de calcul.

▸ <http://sourceforge.net/projects/numpy>

SciPy

SciPy complète Numerical Python en fournissant des fonctions de calculs statistiques, des modules de lecture et d'écriture de matrices au format Matrix Market, etc.

▸ <http://www.scipy.org/>

Biopython

Ce projet regroupe un ensemble de modules spécialisés dans la biologie moléculaire.

▸ <http://biopython.org>

Web

Pour terminer, voici une liste de frameworks de programmation web, qui proposent des fonctionnalités plus ou moins évoluées :

- Zope : <http://www.zope.org>
- Quixote : <http://www.mems-exchange.org/software/quixote/>
- CherryPy : <http://www.cherrypy.org/>
- Django : <http://www.djangoproject.com/>
- Turbogears : <http://www.turbogears.org/>
- Pylons : <http://pylonshq.com/>