

De la bibliothèque standard à l'univers

Python » French » 3.10.0 » 3.10.0 Documentation » La bibliothèque standard

Sujet précédent

10. Spécification complète de la grammaire

Sujet suivant

Introduction

Cette page

Signalement de bogue
Montrer le code source

La bibliothèque standard

Alors que [La référence du langage Python](#) décrit exactement la syntaxe et la sémantique du langage Python, ce manuel de référence de la Bibliothèque décrit la bibliothèque standard distribuée avec Python. Il décrit aussi certains composants optionnels typiquement inclus dans les distributions de Python.

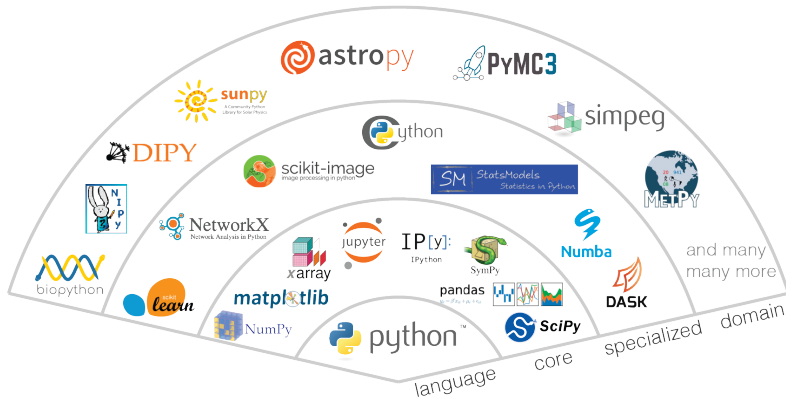
La bibliothèque standard de Python est très grande, elle offre un large éventail d'outils comme le montre la longueur de la table des matières ci-dessous. La bibliothèque contient des modules natifs (écrits en C) exposant les fonctionnalités du système telles que les interactions avec les fichiers qui autrement ne seraient pas accessibles aux développeurs Python, ainsi que des modules écrits en Python exposant des solutions standardisées à de nombreux problèmes du quotidien du développeur. Certains de ces modules sont définis explicitement pour encourager et améliorer la portabilité des programmes Python en abstrayant des spécificités sous-jacentes en API neutres.

Les installateurs de Python pour Windows incluent généralement la bibliothèque standard en entier, et y ajoutent souvent d'autres composants. Pour les systèmes d'exploitation Unix, Python est typiquement fourni sous forme d'une collection de paquets, il peut donc être nécessaire d'utiliser le gestionnaire de paquets fourni par le système d'exploitation pour obtenir certains composants optionnels.

Au delà de la bibliothèque standard, il existe une collection grandissante de plusieurs milliers de composants (des programmes, des modules, ou des *frameworks*), disponibles dans le [Python Package Index](#).


- [Introduction](#)
 - [Notes sur la disponibilité](#)
- [Fonctions natives](#)
- [Constantes natives](#)
 - [Constantes ajoutées par le module `site`](#)
- [Types natifs](#)
 - [Valeurs booléennes](#)
 - [Opérations booléennes — `and`, `or`, `not`](#)
 - [Comparaisons](#)
 - [Types numériques — `int`, `float`, `complex`](#)
 - [Les types itérateurs](#)
 - [Types séquentiels — `list`, `tuple`, `range`](#)

Un écosystème interdépendant pour le traitement de données




<https://jupyterearth.org/jupyter-resources/introduction/ecosystem.html>

Et au-delà...



Help Sponsors Log in Register

Find, install and publish Python packages with the Python Package Index





Or [browse projects](#)

332,134 projects 2,931,955 releases 4,975,830 files 542,291 users



The Python Package Index (PyPI) is a repository of software for the Python programming language.

PyPI helps you find and install software developed and shared by the Python community. [Learn about installing packages](#) .

Package authors use PyPI to distribute their software. [Learn how to package your Python code for PyPI](#) .

`https://pypi.org/`

Se repérer dans la jungle

- ▶ Des bibliothèques stables, bien intégrées et maintenues
- ▶ Des bibliothèques qui viennent, qui partent
- ▶ Et tous les autres codes disponibles...

Donc :

1. Développer des compétences sur des bibliothèques stables
2. Et son sens de l'orientation