

# 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`](#)

Des modules disponibles :

<https://docs.python.org/3/library/>

# Cheat sheet

Pour se rappeler des principales notions de Python :

<https://www.pythoncheatsheet.org/cheatsheet/basics>



Et toutes les autres bibliothèques existantes

De la supernovae aux petites comètes passagères...



# Philosophie générale

Résoudre ses problèmes nécessite :

- ▶ Identifier les bibliothèques avancées les plus pertinentes qui fournissent les fonctions et les objets adaptés
- ▶ Maîtriser le langage Python "de base" pour "faire la plomberie"

# Accès aux bibliothèques



[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. ... son sens de l'orientation dans l'*open source*
3. ... regarder régulièrement ce qui se fait.

Et surtout : lire la documentation et les exemples !

# Quelques bibliothèques

- ▶ Pandas, pour la manipulation de tableaux : ce sera la séance de la semaine prochaine
- ▶ Matplotlib, qui fixe une grammaire des visualisations 2D, la semaine d'après
- ▶ Numpy, caché dans presque toutes les bibliothèques, pour le calcul numérique



# Pandas



[About us](#) ▾ [Getting started](#) [Documentation](#) [Community](#) ▾ [Contribute](#)

## pandas

**pandas** is a fast, powerful, flexible and easy to use open source data analysis and manipulation tool,  
built on top of the Python programming language.

[Install pandas now!](#)

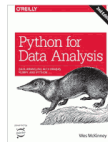
### Latest version: 1.4.2

- What's new in 1.4.2
- Release date:  
Apr 02, 2022
- Documentation (web)
- Documentation (pdf)
- Download source code

### Follow us

[Follow @pandas\\_dev](#)

### Get the book



### Getting started

- [Install pandas](#)
- [Getting started](#)

### Documentation

- [User guide](#)
- [API reference](#)
- [Contributing to pandas](#)
- [Release notes](#)

### Community

- [About pandas](#)
- [Ask a question](#)
- [Ecosystem](#)

With the support of:



### Previous versions

- 1.4.1 (Feb 12, 2022)

Scipy



Fundamental algorithms for scientific computing in Python

GET STARTED

## SciPy 1.7.2 released 2021-11-05

### FUNDAMENTAL ALGORITHMS

SciPy provides algorithms for optimization, integration, interpolation, eigenvalue problems, algebraic equations, differential equations, statistics and many other classes of problems.

### BROADLY APPLICABLE

The algorithms and data structures provided by SciPy are broadly applicable across domains.

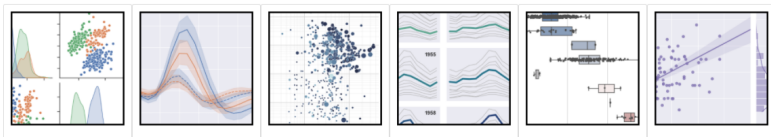
### FOUNDATIONAL

Extends NumPy providing additional tools for array computing and provides specialized data structures, such as sparse matrices and k-dimensional trees.

### PERFORMANT

SciPy wraps highly-optimized implementations written in low-level languages like Fortran, C, and C++. Enjoy the flexibility of Python with the speed of compiled code.

## seaborn: statistical data visualization



Seaborn is a Python data visualization library based on [matplotlib](#). It provides a high-level interface for drawing attractive and informative statistical graphics.

For a brief introduction to the ideas behind the library, you can read the [introductory notes](#) or the [paper](#). Visit the [installation page](#) to see how you can download the package and get started with it. You can browse the [example gallery](#) to see some of the things that you can do with seaborn, and then check out the [tutorial](#) or [API reference](#) to find out how.

To see the code or report a bug, please visit the [GitHub repository](#). General support questions are most at home on [stackoverflow](#) or [discourse](#), which have dedicated channels for seaborn.

### Contents

- [Introduction](#)
- [Release notes](#)
- [Installing](#)
- [Example gallery](#)
- [Tutorial](#)
- [API reference](#)

### Features

- Relational: [API](#) | [Tutorial](#)
- Distribution: [API](#) | [Tutorial](#)
- Categorical: [API](#) | [Tutorial](#)
- Regression: [API](#) | [Tutorial](#)
- Multiples: [API](#) | [Tutorial](#)
- Style: [API](#) | [Tutorial](#)
- Color: [API](#) | [Tutorial](#)

# Scikit-learn



Install User Guide API Examples More ▾

Go

# scikit-learn

*Machine Learning in Python*

Getting Started

Release Highlights for 1.0

GitHub

- Simple and efficient tools for predictive data analysis
- Accessible to everybody, and reusable in various contexts
- Built on NumPy, SciPy, and matplotlib
- Open source, commercially usable - BSD license

## Classification

Identifying which category an object belongs to.

**Applications:** Spam detection, image recognition.

**Algorithms:** SVM, nearest neighbors, random forest, and more...



## Regression

Predicting a continuous-valued attribute associated with an object.

**Applications:** Drug response, Stock prices.

**Algorithms:** SVR, [nearest neighbors](#), random forest, and more...



## Clustering

Automatic grouping of similar objects into sets.

**Applications:** Customer segmentation, Grouping experiment outcomes

**Algorithms:** k-Means, spectral clustering, mean-shift, and more...

Kmeans clustering on the digits dataset (PCA-reduced data). Centroids are marked with white cross



# Industrial-Strength Natural Language Processing

IN PYTHON

## Get things done

spaCy is designed to help you do real work — to build real products, or gather real insights. The library respects your time, and tries to avoid wasting it. It's easy to install, and its API is simple and productive.

GET STARTED

## Blazing fast

spaCy excels at large-scale information extraction tasks. It's written from the ground up in carefully memory-managed Cython. If your application needs to process entire web dumps, spaCy is the library you want to be using.

FACTS & FIGURES

## Awesome ecosystem

In the five years since its release, spaCy has become an industry standard with a huge ecosystem. Choose from a variety of plugins, integrate with your machine learning stack and build custom components and workflows.

READ MORE

## NLTK

## Search

NLTK Documentation

**API Reference**

**Example Usage**

**Module Index**

**Wiki**

**FAQ**

Installation

**Installing NLTK**

**Installing NLTK Data**

More

**Release Notes**

**Contributing to NLTK**

**NLTK Team**

## Documentation

## Natural Language Toolkit


NLTK is a leading platform for building Python programs to work with human language data. It provides easy-to-use interfaces to **over 50 corpora and lexical resources** such as WordNet, along with a suite of text processing libraries for classification, tokenization, stemming, tagging, parsing, and semantic reasoning, wrappers for industrial-strength NLP libraries, and an active **discussion forum**.


Thanks to a hands-on guide introducing programming fundamentals alongside topics in computational linguistics, plus comprehensive API documentation, NLTK is suitable for linguists, engineers, students, educators, researchers, and industry users alike. NLTK is available for Windows, Mac OS X, and Linux. Best of all, NLTK is a free, open source, community-driven project.


NLTK has been called “a wonderful tool for teaching, and working in, computational linguistics using Python,” and “an amazing library to play with natural language.”

**Natural Language Processing with Python** provides a practical introduction to programming for language processing. Written by the creators of NLTK, it guides the reader through the fundamentals of writing Python programs, working with corpora, categorizing text, analyzing linguistic structure, and more. The online version of the book has been updated for Python 3 and NLTK 3. (The original Python 2 version is still available at [https://www.nltk.org/book\\_1ed.](https://www.nltk.org/book_1ed.))

# Statsmodel

 statsmodels v0.13.1

 Search

 statsmodels  
6.8k Stars · 2.4k Forks

Versions

statistical models, hypothesis tests, and data exploration




[statsmodels](#) is a Python module that provides classes and functions for the estimation of many different statistical models, as well as for conducting statistical tests, and statistical data exploration. An extensive list of result statistics are available for each estimator. The results are tested against existing statistical packages to ensure that they are correct. The package is released under the open source Modified BSD (3-clause) license. The online documentation is hosted at [statsmodels.org](https://statsmodels.org).

## Contents

- [Introduction](#)
- [Citation](#)
- [Index](#)
- [Show Source](#)

# BeautifulSoup

 Beautiful Soup  
latest

Search docs

[Beautiful Soup Documentation](#)  
[Quick Start](#)  
[Installing Beautiful Soup](#)  
[Making the soup](#)  
[Kinds of objects](#)  
[Navigating the tree](#)  
[Searching the tree](#)  
[Modifying the tree](#)  
[Output](#)  
[Specifying the parser to use](#)  
[Encodings](#)  
[Line numbers](#)  
[Comparing objects for equality](#)  
[Copying BeautifulSoup objects](#)  
[Parsing only part of a document](#)  
[Troubleshooting](#)

[Docs](#) » [Beautiful Soup Documentation](#)

[View page source](#)

## Beautiful Soup Documentation

[Beautiful Soup](#) is a Python library for pulling data out of HTML and XML files. It works with your favorite parser to provide idiomatic ways of navigating, searching, and modifying the parse tree. It commonly saves programmers hours or days of work.

These instructions illustrate all major features of BeautifulSoup 4, with examples. I show you what the library is good for, how it works, how to use it, how to make it do what you want, and what to do when it violates your expectations.

This document covers BeautifulSoup version 4.8.1. The examples in this documentation should work the same way in Python 2.7 and Python 3.2.

You might be looking for the documentation for [Beautiful Soup 3](#). If so, you should know that BeautifulSoup 3 is no longer being developed and that support for it will be dropped on or after December 31, 2020. If you want to learn about the differences between BeautifulSoup 3 and BeautifulSoup 4, see [Porting code to BS4](#).





## OpenCV-Python Tutorials

- [Introduction to OpenCV](#)

Learn how to setup OpenCV-Python on your computer!

- [Gui Features in OpenCV](#)

Here you will learn how to display and save images and videos, control mouse events and create trackbar.

- [Core Operations](#)

In this section you will learn basic operations on image like pixel editing, geometric transformations, code optimization, some mathematical tools etc.

- [Image Processing in OpenCV](#)

In this section you will learn different image processing functions inside OpenCV.

- [Feature Detection and Description](#)

In this section you will learn about feature detectors and descriptors

- [Video analysis \(video module\)](#)

In this section you will learn different techniques to work with videos like object tracking etc.



# A faster way to build and share data apps

Streamlit turns data scripts into shareable web apps in minutes.

All in pure Python. No front-end experience required.

[Try Streamlit now](#)[Sign up for \*\*Streamlit Cloud\*\*](#)

## À chaque fois :

- ▶ Identifier la bibliothèque pertinente (savoir antérieur ou recherche sur internet - attention, les choses peuvent changer)
- ▶ Lire la documentation / installer
- ▶ Tester sur un petit exemple
- ▶ Intégrer les fonctions dont on a besoin
- ▶ Éventuellement : développer de la virtuosité

# Application

Quelle place a la programmation / l'informatique dans différentes disciplines ?

Prenons le cas de la présentation publique des disciplines sur Wikipédia.

# Programmer c'est décomposer

Pour essayer de répondre à cette question, il faut

- ▶ Définir ce qu'on appelle "place de l'informatique" : des termes dans des textes
- ▶ Récupérer les pages à étudier
- ▶ Tester la présence de ces termes

Comment récupérer une page ? A la main ? Sûrement qu'il y a des outils pour ça.

# Différents niveaux d'abstraction

- ▶ Tout refaire à la main : non !
- ▶ Utiliser des modules standards bas-niveau
- ▶ Utiliser des bibliothèques de haut niveau