

# Python 资源大全中文版

[awesome-python](#) 是 vinta 发起维护的 Python 资源列表，内容包括：Web 框架、网络爬虫、网络内容提取、模板引擎、数据库、数据可视化、图片处理、文本处理、自然语言处理、机器学习、日志、代码分析等。

## 资源列表

### Python 资源大全中文版

#### 资源列表

- 环境管理
- 包管理
- 包仓库
- 分发
- 构建工具
- 交互式解析器
- 文件
- 日期和时间
- 文本处理
- 特殊文本格式处理
- 自然语言处理
- 文档
- 配置
- 命令行工具
- 下载器
- 图像处理
- OCR
- 音频
- Video
- 地理位置
- HTTP
- 数据库
- 数据库驱动
- ORM
- Web 框架
- WebSocket
- 权限
- CMS
- 电子商务
- RESTful API
- 验证
- 模板引擎
- 队列
- 搜索
- 动态消息
- Web 资源管理
- 缓存
- 电子邮件
- 国际化
- URL 处理
- HTML 处理

网页内容提取  
表单  
数据验证  
序列化  
反垃圾技术  
标记  
管理面板  
Serverless 框架  
静态站点生成器  
进程  
并发和并行  
网络  
WebSocket  
WSGI 服务器  
ASGI 服务器  
RPC 服务器  
密码学  
图形用户界面  
游戏开发  
日志  
测试  
渗透测试  
代码分析和 Lint 工具  
调试工具  
科学计算和数据分析  
数据可视化  
计算机视觉  
深度学习  
机器学习  
推荐系统  
分布式计算  
函数式编程  
第三方 API  
DevOps 工具  
任务调度  
外来函数接口  
重构  
高性能  
微软的 Windows 平台  
网络可视化和 SDN  
硬件  
兼容性  
算法和设计模式  
内置类的增强版实现  
机器人  
聊天工具  
编辑器插件和 IDE  
企业级应用集成  
GraphQL  
杂项

## 环境管理

管理 Python 版本和环境的工具

- [p](#): 非常简单的交互式 python 版本管理工具。

- [pyenv](#): 简单的 Python 版本管理工具。
- [Vex](#): 可以在虚拟环境中执行命令。
- [virtualenv](#): 创建独立 Python 环境的工具。
- [virtualenvwrapper](#): virtualenv 的一组扩展。
- [buildout](#): 在隔离环境初始化后使用声明性配置管理。

## 包管理

管理包和依赖的工具。

- [pip](#): Python 包和依赖关系管理工具。
- [pip-tools](#): 保证 Python 包依赖关系更新的一组工具。
- [PyPI](#): Python 正式的第三方包软件存储库。
- [pipenv](#): Python 官方推荐的新一代包管理工具。
- [poetry](#): 可完全取代 setup.py 的包管理工具。
- [conda](#): 跨平台的 Python 二进制包管理工具。
- [Curdling](#): 管理 Python 包的命令行工具。
- [wheel](#): Python 分发的新标准，意在取代 eggs。

## 包仓库

本地 PyPI 仓库服务和代理。

- [warehouse](#): 下一代 PyPI。
- [bandersnatch](#): PyPA 提供的 PyPI 镜像工具。
- [devpi](#): PyPI 服务和打包/测试/分发工具。
- [localshop](#): 本地 PyPI 服务（自定义包并且自动对 PyPI 镜像）。

## 分发

打包为可执行文件以便分发。

- [PyInstaller](#): 将 Python 程序转换成独立的执行文件（跨平台）。
- [cx\\_Freeze](#): 将python程序转换为带有一个动态链接库的可执行文件。
- [dh-virtualenv](#): 构建并将 virtualenv 虚拟环境作为一个 Debian 包来发布。
- [Nuitka](#): 将脚本、模块、包编译成可执行文件或扩展模块。
- [py2app](#): 将 Python 脚本变为独立软件包（Mac OS X）。
- [py2exe](#): 将 Python 脚本变为独立软件包（Windows）。
- [pynsist](#): 一个用来创建 Windows 安装程序的工具，可以在安装程序中打包 Python 本身。
- [pyarmor](#): 一个用于加密 python 脚本的工具，也可以将加密后的脚本绑定到固件上，或设置已加密脚本的有效期。
- [shiv](#): 一个命令行工具，可用于构建完全独立的 zip 应用（PEP 441 所描述的那种），同时包含了所有的依赖项。

## 构建工具

将源码编译成软件。

- [buildout](#): 一个构建系统，从多个组件来创建，组装和部署应用。
- [BitBake](#): 针对嵌入式 Linux 的类似 make 的构建工具。
- [fabricate](#): 对任何语言自动找到依赖关系的构建工具。
- [PlatformIO](#): 多平台命令行构建工具。
- [PyBuilder](#): 纯 Python 实现的持续化构建工具。
- [SCons](#): 软件构建工具。

## 交互式解析器

交互式 Python 解析器。

- [IPython](#): 功能丰富的工具, 非常有效的使用交互式 Python。
- [bpython](#): 界面丰富的 Python 解析器。
- [ptpython](#): 高级交互式 Python 解析器, 构建于 [python-prompt-toolkit](#) 之上。
- [Jupyter Notebook \(IPython\)](#): 一个能够让你最大限度地以交互式方式使用 Python 的丰富工具包。
  - [awesome-jupyter](#)

## 文件

文件管理和 MIME (多用途的网际邮件扩充协议) 类型检测。

- [aiofiles](#): 基于 [asyncio](#), 提供文件异步操作。
- [imgHDR](#): (Python 标准库) 检测图片类型。
- [mimetypes](#): (Python 标准库) 将文件名映射为 MIME 类型。
- [path.py](#): 对 `os.path` 进行封装的模块。
- [pathlib](#): (Python 3.4+ 标准库) 跨平台的、面向对象的路径操作库。
- [python-magic](#): 文件类型检测的第三方库 `libmagic` 的 Python 接口。
- [Unipath](#): 用面向对象的方式操作文件和目录。
- [watchdog](#): 管理文件系统事件的 API 和 shell 工具。
- [PyFilesystem2](#): Python 的文件系统抽象层。

## 日期和时间

操作日期和时间的类库。

- [arrow](#): 更好的 Python 日期时间操作类库。
- [Chronyk](#): Python 3 的类库, 用于解析手写格式的时间和日期。
- [dateutil](#): Python `datetime` 模块的扩展。
- [delorean](#): 解决 Python 中有关日期处理的棘手问题的库。
- [maya](#): 人性化的时间处理库。
- [moment](#): 一个用来处理时间和日期的 Python 库。灵感来自于 [Moment.js](#)。
- [pendulum](#): 一个比 [arrow](#) 更具有明确的, 可预测的行为的时间操作库。
- [PyTime](#): 一个简单易用的 Python 模块, 用于通过字符串来操作日期/时间。
- [pytz](#): 现代以及历史版本的世界时区定义。将时区数据库引入 Python。
- [when.py](#): 提供用户友好的函数来帮助用户进行常用的日期和时间操作。
- [dateutil](#): Python 标准包 `datetime` 的扩展。
- [moment](#): 一个处理日期/时间的库, 灵感来自 [Moment.js](#)。
- [pytz](#): 支持跨平台时区计算, 并将 [tz database](#) 引入 Python。

## 文本处理

用于解析和操作文本的库。

- 通用
  - [chardet](#): 字符编码检测器, 兼容 Python2 和 Python3。
  - [difflib](#): (Python 标准库) 帮助我们进行差异化比较。
  - [ftfy](#): 让 Unicode 文本更完整更连贯。
  - [fuzzywuzzy](#): 模糊字符串匹配。
  - [Levenshtein](#): 快速计算编辑距离以及字符串的相似度。

- [pangu.py](#): 在中日韩语字符和数字字母之间添加空格。
- [pypinyin](#): 汉字拼音转换工具 Python 版。
- [shortuuid](#): 一个生成器库, 用以生成简洁的, 明白的, URL 安全的 UUID。
- [simplejson](#): Python 的 JSON 编码、解码器。
- [unidecode](#): Unicode 文本的 ASCII 转换形式。
- [uniout](#): 打印可读的字符, 而不是转义的字符串。
- [xpinyin](#): 一个用于把汉字转换为拼音的库。
- [pyfiglet](#): figlet 的 Python 实现。
- [flashtext](#): 一个高效的文本查找替换库。
- [textdistance](#): 支持 30 多种算法来计算序列之间的距离。
- Slug 化
  - [awesome-slugify](#): 一个 Python slug 化库, 可以保持 Unicode。
  - [python-slugify](#): Python slug 化库, 可以把 unicode 转化为 ASCII。
  - [unicode-slugify](#): 一个 slug 工具, 可以生成 unicode slugs ,需要依赖 Django 。
- 解析器
  - [phonenumbers](#): 解析, 格式化, 储存, 验证电话号码。
  - [python-phonenumbers](#): 解析, 格式化, 存储, 校验国际电话号码。
  - [PLY](#): lex 和 yacc 解析工具的 Python 实现。
  - [Pygments](#): 通用语法高亮工具。
  - [pyparsing](#): 生成通用解析器的框架。
  - [python-nameparser](#): 把一个人名分解为几个独立的部分。
  - [python-user-agents](#): 浏览器 user agent 解析器。
  - [sqlparse](#): 一个无验证的 SQL 解析器。

## 特殊文本格式处理

一些用来解析和操作特殊文本格式的库。

- 通用
  - [tablib](#): 一个用来处理中表格数据的模块。
- Office
  - [Marmir](#): 把输入的 Python 数据结构转换为电子表单。
  - [openpyxl](#): 一个用来读写 Excel 2010 xlsx/xlsm/xltx/xltm 文件的库。
  - [pyexcel](#): 一个提供统一 API, 用来读写, 操作 Excel 文件的库。
  - [python-docx](#): 读取, 查询以及修改 Microsoft Word 2007/2008 docx 文件。
  - [python-pptx](#): 可用于创建和修改 ppt 文件的 Python 库。
  - [relatorio](#): 模板化 OpenDocument 文件。
  - [unoconv](#): 在 LibreOffice/OpenOffice 支持的任意文件格式之间进行转换。
  - [XlsxWriter](#): 一个用于创建 Excel .xlsx 文件的 Python 模块。
  - [xlwings](#): 一个使得在 Excel 中方便调用 Python 的库 (反之亦然), 基于 BSD 协议。
  - [xlwt](#) / [xlrd](#): 读写 Excel 文件的数据和格式信息。
  - [docxtpl](#): 通过 jinja2 模版编辑 docx 文档。
- PDF
  - [PDFMiner](#): 一个用于从 PDF 文档中抽取信息的工具。
  - [PyPDF2](#): 一个可以分割, 合并和转换 PDF 页面的库。
  - [ReportLab](#): 快速创建富文本 PDF 文档。
- Markdown
  - [Mistune](#): 快速并且功能齐全的纯 Python 实现的 Markdown 解析器。
  - [Python-Markdown](#): John Gruber's Markdown 的 Python 版实现。
  - [Python-Markdown2](#): 纯 Python 实现的 Markdown 解析器, 比 Python-Markdown 更快, 更准确, 可扩展。

- [YAML](#)
  - [PyYAML](#): Python 版本的 YAML 解析器。
- [CSV](#)
  - [csvkit](#): 用于转换和操作 CSV 的工具。
- [Archive](#)
  - [unp](#): 一个用来方便解包归档文件的命令行工具。

## 自然语言处理

用来处理人类语言的库。

- [NLTK](#): 一个先进的平台，用以构建处理人类语言数据的 Python 程序。
- [gensim](#): 人性化的话题建模库。
- [jieba](#): 中文分词工具。
- [langid.py](#): 独立的语言识别系统。
- [Pattern](#): Python 网络信息挖掘模块。
- [SnowNLP](#): 一个用来处理中文文本的库。
- [TextBlob](#): 为进行普通自然语言处理任务提供一致的 API。
- [TextGrocery](#): 一简单高效的短文本分类工具，基于 LibLinear 和 Jieba。
- [thulac](#): 清华大学自然语言处理与社会人文计算实验室研制推出的一套中文词法分析工具包。
- [polyglot](#): 支持数百种语言的自然语言处理管道。
- [pytext](#): 基于 PyTouch 的自然语言模型框架。
- [PyTorch-NLP](#): 一个支持快速深度学习 NLP 原型研究的工具包。
- [spacy](#): Python 和 Cython 中用于工业级自然语言处理的库。
- [Stanza](#): 斯坦福 NLP 集团的官方 Python 库，支持60多种语言。
- [funNLP](#): 中文自然语言处理的工具和数据集。
- [pkuseg-python](#): 一个支持对不同领域进行中文分词的工具箱。

## 文档

用以生成项目文档的库。

- [Sphinx](#): Python 文档生成器。
  - [awesome-sphinxdoc](#)
- [MkDocs](#): 对 Markdown 友好的文档生成器。
- [pdoc](#): 一个可以替换 Epydoc 的库，可以自动生成 Python 库的 API 文档。
- [Pycco](#): 文学编程 (literate-programming) 风格的文档生成器。
- [readthedocs](#): 一个基于 Sphinx/MkDocs 的在线文档托管系统，对开源项目免费开放使用。

## 配置

用来保存和解析配置的库。

- [config](#): [logging](#) 模块作者写的分级配置模块。
- [ConfigObj](#): INI 文件解析器，带验证功能。
- [ConfigParser](#): (Python 标准库) INI 文件解析器。
- [profig](#): 通过多种格式进行配置，具有数值转换功能。
- [python-decouple](#): 将设置和代码完全隔离。
- [hydra](#): 一个优雅地配置复杂应用程序的框架。

## 命令行工具

用于创建命令行程序的库。

- 命令行程序开发
  - [cement](#): Python 的命令行程序框架。
  - [click](#): 一个通过组合的方式来创建精美命令行界面的包。
  - [cliff](#): 一个用于创建命令行程序的框架，可以创建具有多层命令的命令行程序。
  - [clint](#): Python 命令行程序工具。
  - [docopt](#): Python 风格的命令行参数解析器。
  - [Gooyey](#): 一条命令，将命令行程序变成一个 GUI 程序。
  - [python-prompt-toolkit](#): 一个用于构建强大的交互式命令行程序的库。
  - [python-fire](#): Google 出品的一个基于 Python 类的构建命令行界面的库。
  - [Pythonpy](#): 在命令行中直接执行任何 Python 指令。
- 终端呈现方式
  - [asciimatics](#): 跨平台，全屏终端包（即鼠标/键盘输入和彩色，定位文本输出），完整的复杂动画和特殊效果的高级 API。
  - [alive-progress](#): 一款新的进度条，具有实时吞吐量信息以及非常酷的动画。
  - [colorama](#): 跨平台彩色终端文本。
  - [bashplotlib](#): 在终端中进行基本绘图。
  - [rich](#): 一个在终端中支持富文本和格式美化的 Python 库，同时提供了 `RichHandler` 日志处理程序。
  - [tqdm](#): 一个可在循环和命令行中使用的快速、可扩展的进度条。
- 生产力工具
  - [aws-cli](#): Amazon Web Services 的通用命令行界面。
  - [caniusepython3](#): 判断是哪个项目妨碍你移植到 Python3。
  - [cookiecutter](#): 从 cookiecutters（项目模板）创建项目的一个命令行工具。
  - [doitlive](#): 一个用来在终端中进行现场演示的工具。
  - [pyftplib](#): 一个速度极快和可扩展的 Python FTP 服务库。
  - [howdoi](#): 通过命令行获取即时的编程问题解答。
  - [PathPicker](#): 从 bash 输出中选出文件。
  - [percol](#): 向 UNIX shell 传统管道概念中加入交互式选择功能。
  - [thefuck](#): 修正你之前的命令行指令。
  - [try](#): 一个极其简单的命令行工具，用来试用 python 库。
  - [copier](#): 用于呈现项目模板的库和命令行实用程序。
  - [Invoke](#): 用于管理面向 shell 的子进程，同时支持将可执行的 Python 代码组织成命令行可调用的状态。
  - [tmuxp](#): [tmux](#) 会话管理器。
- 高级 CLI
  - [httpie](#): 一个命令行 HTTP 客户端，cURL 的替代品，易用性更好。
  - [iredis](#): 支持自动补全和高亮显示的 redis 命令行工具。
  - [kube-shell](#): K8S 命令行集成的 shell 工具。
  - [litecli](#): 支持自动补全和语法高亮的 SQLite 命令行工具。
  - [mycli](#): 支持自动补全和语法高亮的 MySQL 命令行客户端
  - [pgcli](#): 支持自动补全和语法高亮的 Postgres 命令行工具。
  - [SAWS](#): 一个加强版的 AWS 命令行。
- Shell
  - [xonsh](#): 一种基于 python 的跨平台，面向 unix 的 shell 语言和命令提示符。

## 下载器

用来进行下载的库。



- [s3cmd](#): 一个用来管理 Amazon S3 和 CloudFront 的命令行工具。
- [s4cmd](#): 超级 S3 命令行工具, 性能更加强健。
- [you-get](#): 一个 YouTube/Youku/Niconico 视频下载器, 使用 Python3 编写。
- [youtube-dl](#): 一个小巧的命令行程序, 用来下载 YouTube 视频。
- [akshare](#): 为方便人使用而创建的金融数据接口库。

## 图像处理

用来操作图像的库。

- [pillow](#): Pillow 是一个更加易用版的 [PIL](#)。
- [hmap](#): 图像直方图映射。
- [imgSeek](#): 一个使用视觉相似性搜索一组图片集合的项目。
- [nude.py](#): 裸体检测。
- [python-barcode](#): 不借助其他库在 Python 程序中生成条形码。
- [pygram](#): 类似 Instagram 的图像滤镜。
- [python-qrcode](#): 一个纯 Python 实现的二维码生成器。
- [Quads](#): 基于四叉树的计算机艺术。
- [scikit-image](#): 一个用于 (科学) 图像处理的 Python 库。
- [thumbor](#): 一个小型图像服务, 具有剪裁, 尺寸重设和翻转功能。
- [wand](#): [MagickWand](#) 的 Python 绑定。MagickWand 是 ImageMagick 的 C API。
- [face\\_recognition](#): 简单易用的 python 人脸识别库。
- [pagan](#): 基于输入和哈希的复古风图标 (头像) 生成工具。
- [PyMatting](#): 支持 alpha matting 的库。
- [pywal](#): 由图像生成配色方案的工具。
- [pyvips](#): 低内存消耗且快速的图像处理库。

## OCR

光学字符识别库。

- [pyocr](#): Tesseract 和 Cuneiform 的一个封装。
- [pytesseract](#): [Google Tesseract OCR](#) 的一个封装。

## 音频

用来操作音频的库

- [audiolazy](#): Python 的数字信号处理包。
- [audioread](#): 交叉库 (GStreamer + Core Audio + MAD + FFmpeg) 音频解码。
- [beets](#): 一个音乐库管理工具及 [MusicBrainz](#) 标签添加工具。
- [dejavu](#): 音频指纹提取和识别。
- [django-elastic-transcoder](#): Django + [Amazon Elastic Transcoder](#)。
- [eyeD3](#): 一个用来操作音频文件的工具, 具体来讲就是包含 ID3 元信息的 MP3 文件。
- [id3reader](#): 一个用来读取 MP3 元数据的 Python 模块。
- [m3u8](#): 一个用来解析 m3u8 文件的模块。
- [mutagen](#): 一个用来处理音频元数据的 Python 模块。
- [pydub](#): 通过简单、简洁的高层接口来操作音频文件。
- [pyechonest](#): [Echo Nest](#) API 的 Python 客户端。
- [talkbox](#): 一个用来处理演讲/信号的 Python 库。
- [TimeSide](#): 开源 web 音频处理框架。
- [tinytag](#): 一个用来读取 MP3, OGG, FLAC 以及 Wave 文件音乐元数据的库。
- [mingus](#): 一个高级音乐理论和曲谱包, 支持 MIDI 文件和回放功能。
- [kapre](#): Keras 音频处理器。



- [librosa](#): 音频音乐分析 Python 库。
- [matchering](#): 用于音频母带制作的库。
- [pyAudioAnalysis](#): 音频特征提取, 分类, 分段和应用。
- [beets](#): 一个音乐库管理器和 [MusicBrainz](#) 标记器。

## Video

用来操作视频和 GIF 的库。

- [moviepy](#): 一个用来进行基于脚本的视频编辑模块, 适用于多种格式, 包括动图 GIFs。
- [scikit-video](#): SciPy 视频处理常用程序。
- [vidgear](#): 强大的多线程视频处理框架。

## 地理位置

地理编码地址以及用来处理经纬度的库。

- [GeoDjango](#): 世界级地理图形 web 框架。
- [GeoIP](#): MaxMind GeoIP Legacy 数据库的 Python API。
- [geojson](#): GeoJSON 的 Python 绑定及工具。
- [geopy](#): Python 地址编码工具箱。
- [GeoIP2](#): GeoIP2 Webservice 客户端与数据库 Python API。
- [django-countries](#): 一个 Django 应用程序, 提供用于表格的国家选择功能, 国旗图标静态文件以及模型中的国家字段。
- [pygeoip](#): Python GeoIP 接口。

## HTTP

使用 HTTP 的库。

- [aiohttp](#): 基于 asyncio 的异步 HTTP 网络库。
- [requests](#): 人性化的 HTTP 请求库。
- [grequests](#): requests 库 + gevent, 用于异步 HTTP 请求。
- [httplib2](#): 全面的 HTTP 客户端库。
- [treq](#): 类似 requests 的 Python API 构建于 Twisted HTTP 客户端之上。
- [urllib3](#): 一个具有线程安全连接池, 支持文件 post, 清晰友好的 HTTP 库。
- [httpx](#): 下一代 Python HTTP 客户端。

## 数据库

Python 实现的数据库。

- [pickleDB](#): 一个简单, 轻量级键值储存数据库。
- [PipelineDB](#): 流式 SQL 数据库。
- [TinyDB](#): 一个微型的, 面向文档型数据库。
- [ZODB](#): 一个 Python 原生对象数据库。一个键值和对象图数据库。

## 数据库驱动

用来连接和操作数据库的库。

- MySQL: [awesome-mysql](#) 系列
  - [aiomysql](#): 基于 asyncio 的异步 MySQL 数据库操作库。
  - [mysql-python](#): Python 的 MySQL 数据库连接器。
  - sqlalchemy: [mysql-python](#) 分支, 支持 Python 3。
  - [oursql](#): 一个更好的 MySQL 连接器, 支持原生预编译指令和 BLOBs。

- [PyMySQL](#): 纯 Python MySQL 驱动, 兼容 mysql-python。
- PostgreSQL
  - [psycopg2](#): Python 中最流行的 PostgreSQL 适配器。
  - [queries](#): psycopg2 库的封装, 用来和 PostgreSQL 进行交互。
  - [txpostgres](#): 基于 Twisted 的异步 PostgreSQL 驱动。
- 其他关系型数据库
  - [apsw](#): 另一个 Python SQLite 封装。
  - dataset: 在数据库中存储 Python 字典
  - [pymssql](#): 一个简单的 Microsoft SQL Server 数据库接口。
- NoSQL 数据库
  - [asyncio-redis](#): 基于 asyncio 的 redis 客户端 (PEP 3156)。
  - [cassandra-python-driver](#): Cassandra 的 Python 驱动。
  - [HappyBase](#): 一个为 Apache HBase 设计的, 对开发者友好的库。
  - [Plyvel](#): 一个快速且功能丰富的 LevelDB 的 Python 接口。
  - [py2neo](#): Neo4j restful 接口的 Python 封装客户端。
  - [pycassa](#): Cassandra 的 Python Thrift 驱动。
  - [PyMongo](#): MongoDB 的官方 Python 客户端。
  - [redis-py](#): Redis 的 Python 客户端。
  - [telephus](#): 基于 Twisted 的 Cassandra 客户端。
  - [txRedis](#): 基于 Twisted 的 Redis 客户端。
  - [kafka-python](#): Apache Kafka Python 客户端。
- 异步客户端
  - [motor](#): 支持 MongoDB 的异步 Python 驱动程序。

## ORM

实现对象关系映射或数据映射技术的库。

- 关系型数据库
  - [Django Models](#): Django 的一部分。
  - [SQLAlchemy](#): Python SQL 工具以及对象关系映射工具。
    - [awesome-sqlalchemy](#)
  - [awesome-sqlalchemy](#) 系列
  - [Peewee](#): 一个小巧, 富有表达力的 ORM。
  - [PonyORM](#): 提供面向生成器的 SQL 接口的 ORM。
  - [python-sql](#): 编写 Python 风格的 SQL 查询。
  - [dataset](#): 在数据库中存储字典, 支持 SQLite, MySQL 和 PostgreSQL。
  - [orator](#): Orator ORM, 提供了一个简单而美观的 ActiveRecord 实现。
  - [orm](#): 一个异步的 ORM。
  - [peewee](#): 一个小但是很有表现力的 ORM。
  - [pony](#): 提供面向生成器的SQL接口的ORM。
  - [pydal](#): 纯 Python 数据库抽象接口层。
- NoSQL 数据库
  - [django-mongodb-engine](#): Django MongoDB 后端。
  - [PynamoDB](#): [Amazon DynamoDB](#) 的一个 Python 风格接口。
  - [flywheel](#): Amazon DynamoDB 的对象映射工具。
  - [MongoEngine](#): 一个 Python 对象文档映射工具, 用于 MongoDB。

- [hot-redis](#): 为 Redis 提供 Python 丰富的数据类型。
- [redisco](#): 一个 Python 库, 提供可以持续存在在 Redis 中的简单模型和容器。
- 其他
  - [butterdb](#): Google Drive 电子表格的 Python ORM。

## Web 框架

全栈 Web 框架。

- [Django](#): Python 界最流行的 web 框架。
  - [awesome-django\(by\\_shahraizali\)](#) 系列
  - [awesome-django\(by\\_wsvincent\)](#) 系列
- [Flask](#): 一个 Python 微型框架。
  - [awesome-flask](#) 系列
- [Pyramid](#): 一个小巧, 快速, 接地气的开源 Python web 框架。
  - [awesome-pyramid](#) 系列
- [Bottle](#): 一个快速小巧, 轻量级的 WSGI 微型 web 框架。
- [CherryPy](#): 一个极简的 Python web 框架, 服从 HTTP/1.1 协议且具有 WSGI 线程池。
- [TurboGears](#): 一个可以扩展为全栈解决方案的微型框架。
- [web.py](#): 一个 Python 的 web 框架, 既简单, 又强大。
- [web2py](#): 一个全栈 web 框架和平台, 专注于简单易用。
- [Tornado](#): 一个 web 框架和异步网络库。
- [sanic](#): 基于 Python3.5+ 的异步网络框架。
- [starlette](#): 一款轻量级, 高性能的 ASGI 框架。
- [Masonite](#): 以开发者为中心的现代 Python Web 框架。

## WebSocket

Web socket 相关库。

- [autobahn-python](#): 适用于 Twisted 和 asyncio 的 Python WebSocket 和 WAMP。
- [channels](#): 开发者友好的 Django 异步工具。
- [websockets](#): 一个用于构建 WebSocket 服务器和客户端的库, 着重于正确性和简单性。

## 权限

允许或拒绝用户访问数据或功能的库。

- [Carteblanche](#): 站在用户和设计者角度开发的一个代码对齐模块, 很好地处理了代码导航及权限。
- [django-guardian](#): Django 1.2+, 实现了单个对象权限。
- [django-rules](#): 一个小巧但是强大的应用, 提供对象级别的权限管理, 且不需要使用数据库。

## CMS

内容管理系统。

- [odoo-cms](#): 一个开源的, 企业级 CMS, 基于 odoo。
- [django-cms](#): 一个开源的, 企业级 CMS, 基于 Django。
- [djed CMS](#): 一个轻量级但却非常强大的 Django CMS, 考虑到了插件, 内联编辑以及性能。
- [FeinCMS](#): 基于 Django 构建的最先进的内容管理系统之一。

- [Kotti](#): 一个高级的, Python 范的 web 应用框架, 基于 Pyramid 构建。
- [Mezzanine](#): 一个强大的, 持续的, 灵活的内容管理平台。
- [Opps](#): 一个为杂志, 报纸网站以及大流量门户网站设计的 CMS 平台, 基于 Django。
- [Plone](#): 一个构建于开源应用服务器 Zope 之上的 CMS。
- [Quokka](#): 灵活, 可扩展的小型 CMS, 基于 Flask 和 MongoDB。
- [Wagtail](#): 一个 Django 内容管理系统。
- [Widgy](#): 最新的 CMS 框架, 基于 Django。
- [indico](#): 一个功能丰富的事件管理系统, 由 @CERN 开发。

## 电子商务

用于电子商务以及支付的框架和库。

- [django-oscar](#): 一个用于 Django 的开源的电子商务框架。
- [django-shop](#): 一个基于 Django 的店铺系统。
- [Cartridge](#): 一个基于 Mezzanine 构建的购物车应用。
- [shoop](#): 一个基于 Django 的开源电子商务平台。
- [alipay](#): 非官方的 Python 支付宝 API。
- [merchant](#): 一个可以接收来自多种支付平台支付的 Django 应用。
- [money](#): 一个货币类库。带有可选的 CLDR 后端本地化格式, 提供可扩展的货币兑换解决方案。
- [python-currencies](#): 显示货币格式以及它的数值。
- [forex-python](#): 外汇汇率, 比特币价格指数和货币换算。
- [saleor](#): 一款兼容 Django 的电子商务平台。

## RESTful API

用来开发 RESTful APIs 的库

- Django
  - [django-rest-framework](#): 一个强大灵活的工具, 用来构建 web API。
  - [django-tastypie](#): 为 Django 应用开发 API。
  - [django-formapi](#): 为 Django 的表单验证, 创建 JSON APIs。
- Flask
  - [flask-api](#): 为 flask 开发的, 可浏览 Web APIs。
  - [flask-restful](#): 为 flask 快速创建 REST APIs。
  - [flask-restless](#): 为 SQLAlchemy 定义的数据库模型创建 RESTful APIs。
  - [flask-api-utils](#): 为 Flask 处理 API 表示和验证。
  - [eve](#): REST API 框架, 由 Flask, MongoDB 等驱动。
- Pyramid
  - [cornice](#): 一个 Pyramid 的 REST 框架。
- 与框架无关
  - [falcon](#): 一个用来建立云 API 和 web app 后端的高性能框架。
  - [sandman](#): 为现存的数据库驱动系统自动创建 REST APIs。
  - [restless](#): 框架无关的 REST 框架, 基于从 Tastypie 学到的知识。
  - [ripozo](#): 快速创建 REST/HATEOAS/Hypermedia APIs。
  - [apistar](#): 专为 Python 3 设计的智能 Web API 框架。
  - [fastapi](#): 一个现代, 快速, 基于标准 Python 类型注解的 web 框架, 可使用 Python 3.6+ 版本构建 API。
  - [hug](#): 一个为纯净公开的 API 打造的 Python 3 框架。
  - [sandman2](#): 为数据库驱动的系统自动生成 REST API。
  - [vibora](#): 快速高效且支持异步的 Web 框架, 灵感来源于 Flask。

## 验证

实现验证方案的库。

- OAuth
  - [authlib](#): 一个强大的Python库, 用来构建 OAuth 和 OpenID 服务端。包括: JWS, JWK, JWA, JWT。
  - [Authomatic](#): 简单但是强大的框架, 身份验证/授权客户端。
  - [django-allauth](#): Django 的验证应用。
  - [django-oauth-toolkit](#): 为 Django 用户准备的 OAuth2。
  - [django-oauth2-provider](#): 为 Django 应用提供 OAuth2 接入。
  - [Flask-OAuthlib](#): OAuth 1.0/a, 2.0 客户端实现, 供 Flask 使用。
  - [OAuthLib](#): 一个 OAuth 请求-签名逻辑通用、完整的实现。
  - [python-oauth2](#): 一个完全测试的抽象接口。用来创建 OAuth 客户端和服务端。
  - [python-social-auth](#): 一个设置简单的社会化验证方式。
  - [rauth](#): OAuth 1.0/a, 2.0, 和 Ofly 的 Python 库。
  - [sanction](#): 一个超级简单的 OAuth2 客户端实现。
- 其他
  - [PyJWT](#): JSON Web 令牌草案 01。
  - [python-jws](#): JSON Web 签名草案 02 的实现。
  - [python-jwt](#): 一个用来生成和验证 JSON Web 令牌的模块。
  - [python-jose](#): python 版 JOSE 实现。

## 模板引擎

模板生成和词法解析的库和工具。

- [Jinja2](#): 一个现代的, 对设计师友好的模板引擎。
- [Chameleon](#): 一个 HTML/XML 模板引擎。模仿了 ZPT (Zope Page Templates), 进行了速度上的优化。
- [Genshi](#): Python 模板工具, 用以生成 web 感知的结果。
- [Mako](#): Python 平台的超高速轻量级模板。

## 队列

处理事件以及任务队列的库。

- [celery](#): 一个异步任务队列/作业队列, 基于分布式消息传递
- [dramatiq](#): 适用于Python 3的快速可靠的后台任务处理库。
- [huey](#): 小型多线程任务队列。
- [mrq](#): 一个 Python 的分布式 worker 任务队列, 使用 Redis 和 gevent。
- [rq](#): 简单的 Python 作业队列。
- [simpleq](#): 一个简单的, 可无限扩张的, 基于亚马逊 SQS 的队列。

## 搜索

对数据进行索引和执行搜索查询的库和软件。

- [django-haystack](#): Django 模块化搜索。
- [elasticsearch-py](#): Elasticsearch 的官方底层 Python 客户端。
- [elasticsearch-dsl-py](#): Elasticsearch 的官方高级 Python 客户端。
- [solrpy](#): [solr](#) 的 Python 客户端。
- [pysolr](#): 支持 [Apache Solr](#) 的轻量级 Python 装饰器。
- [Whoosh](#): 一个快速的纯 Python 搜索引擎库。

## 动态消息

用来创建用户活动的库。

- [django-activity-stream](#): 从你的站点行为中生成通用活动信息流。
- [Stream-Framework](#): 使用 Cassandra 和 Redis 创建动态消息和通知系统。

## Web 资源管理

管理、压缩、缩小网站资源的工具。

- [django-compressor](#): 将链接和内联的 JavaScript 或 CSS 压缩到一个单独的缓存文件中。
- [django-pipeline](#): Django 的资源包装库。
- [django-storages](#): 一个针对 Django 的自定义存储后端的工具集合。
- [fanstatic](#): 打包、优化, 并且把静态文件依赖作为 Python 的包来提供。
- [File Conveyor](#): 一个后台驻留的程序, 用来发现和同步文件到 CDNs, S3 和 FTP。
- [Flask-Assets](#): 帮你将 web 资源整合到你的 Flask app 中。
- [jinja-assets-compressor](#): 一个 Jinja 扩展, 用来编译和压缩你的资源。
- [webassets](#): 为你的静态资源打包、优化和管理生成独一无二的缓存 URL。

## 缓存

缓存数据的库。

- [Beaker](#): 一个缓存和会话库, 可以用在 web 应用和独立 Python 脚本和应用上。
- [django-cache-machine](#): Django 模型的自动缓存和失效。
- [django-cacheops](#): 具有自动颗粒化事件驱动失效功能的 ORM。
- [django-viewlet](#): 渲染模板, 同时具有额外的缓存控制功能。
- [dogpile.cache](#): dogpile.cache 是 Beaker 的下一代替品, 由同一作者开发。
- [HermesCache](#): Python 缓存库, 具有基于标签的失效和 dogpile effect 保护功能。
- [johnny-cache](#): django 应用缓存框架。
- [pylibmc](#): [libmemcached](#) 接口的 Python 封装。
- [python-diskcache](#): SQLite 和文件支持的缓存后端, 具有比 memcached 和 redis 更快的查找速度。

## 电子邮件

用来发送和解析电子邮件的库。

- [django-celery-ses](#): 带有 AWS SES 和 Celery 的 Django email 后端。
- [envelopes](#): 供人类使用的电子邮件库。
- [flanker](#): 一个 email 地址和 Mime 解析库。
- [inbox](#): Python IMAP 库。
- [inbox.py](#): Python SMTP 服务器。
- [inbox](#): 一个开源电子邮件工具箱。
- [lamson](#): Python 风格的 SMTP 应用服务器。
- [mailjet](#): Mailjet API 实现, 用来提供批量发送邮件, 统计等功能。
- [marrow.mailer](#): 高性能可扩展邮件分发框架。
- [modoboa](#): 一个邮件托管和管理平台, 具有现代的、简约的 Web UI。
- [pyzmail](#): 创建, 发送和解析电子邮件。
- [Talon](#): Mailgun 库, 用来抽取信息和签名。
- [yagmail](#): yagmail是一个GMAIL / SMTP客户端, 旨在使其尽可能简单地发送电子邮件。
- [salmon](#): 一个 Python 邮件服务器。
- [mailer](#): 一款高性能可扩展的邮件投递框架。

## 国际化

用来进行国际化的库。

- [Babel](#): 一个 Python 的国际化库。
- [Korean](#): 一个韩语词态库。
- [PyICU](#): 一个封装了 [ICU](#) C++ 库的 Python 扩展。

## URL 处理

解析 URLs 的库

- [furl](#): 一个让处理 URL 更简单小型 Python 库。
- [purl](#): 一个简单的, 不可变的 URL 类, 具有简洁的 API 来进行询问和处理。
- [pyshorteners](#): 一个纯 Python URL 缩短库。
- [shorturl](#): 生成短小 URL 和类似 bit.ly 短链的 Python 实现。
- [webargs](#): 一个解析 HTTP 请求参数的库, 内置对流行 web 框架的支持, 包括 Flask, Django, Bottle, Tornado 和 Pyramid。

## HTML 处理

处理 HTML 和 XML 的库。

- [BeautifulSoup](#): 以 Python 风格的方式来对 HTML 或 XML 进行迭代, 搜索和修改。
- [bleach](#): 一个基于白名单的 HTML 清理和文本链接库。
- [cssutils](#): 一个 Python 的 CSS 库。
- [html5lib](#): 一个兼容标准的 HTML 文档和片段解析及序列化库。
- [lxml](#): 一个非常快速, 简单易用, 功能齐全的库, 用来处理 HTML 和 XML。
- [MarkupSafe](#): 为 Python 实现 XML/HTML/XHTML 标记安全字符串。
- [pyquery](#): 一个解析 HTML 的库, 类似 jQuery。
- [requests-html](#): 人性化的, Pythonic 的 HTML 解析库。
- [untangle](#): 将 XML 文档转换为 Python 对象, 使其可以方便的访问。
- [xhtml2pdf](#): HTML/CSS 转 PDF 工具。
- [xmldict](#): 像处理 JSON 一样处理 XML。
- [WeasyPrint](#): 用于HTML和CSS的可视化呈现引擎, 并可以导出为PDF。
- [xmldataset](#): 简单 XML 解析。

爬取网络站点的库

- [Scrapy](#): 一个快速高级的屏幕爬取及网页采集框架。
- [ScrapydWeb](#): 一个用于 Scrapyd 集群管理的全功能 web UI, 支持 Scrapy 日志分析和可视化, 自动打包, 定时器任务和邮件通知等特色功能。
- [cola](#): 一个分布式爬虫框架。
- [Demiurge](#): 基于 PyQuery 的爬虫微型框架。
- [feedparser](#): 通用 feed 解析器。
- [Grab](#): 站点爬取框架。
- [MechanicalSoup](#): 用于自动和网络站点交互的 Python 库。
- [portia](#): Scrapy 可视化爬取。
- [pyspider](#): 一个强大的爬虫系统。
- [RoboBrowser](#): 一个简单的, Python 风格的库, 用来浏览网站, 而不需要一个独立安装的浏览器。

## 网页内容提取

用于进行网页内容提取的库。



- [Haul](#): 一个可以扩展的图像爬取工具。
- [html2text](#): 将 HTML 转换为 Markdown 格式文本。
- [lassie](#): 人性化的网页内容检索库。
- [micawber](#): 一个小型网页内容提取库, 用来从 URLs 提取富内容。
- [newspaper](#): 使用 Python 进行新闻提取, 文章提取以及内容策展。
- [opengraph](#): 一个用来解析开放内容协议(Open Graph Protocol)的 Python 模块。
- [python-goose](#): HTML 内容/文章提取器(python2)。
- [goose3](#): HTML 内容/文章提取器(python3)。
- [python-readability](#): arc90 公司 readability 工具的 Python 高速端口。
- [sanitize](#): 为杂乱的数据世界带来调理性。
- [sumy](#): 一个为文本文件和 HTML 页面进行自动摘要的模块。
- [textextract](#): 从任何格式的文档中提取文本, Word, PowerPoint, PDFs 等等。

## 表单

进行表单操作的库。

- [Deform](#): Python HTML 表单生成库, 受到了 formish 表单生成库的启发。
- [django-bootstrap3](#): 集成了 Bootstrap 3 的 Django。
- [django-bootstrap4](#): 集成了 Bootstrap 4 的 Django。
- [django-crispy-forms](#): 一个 Django 应用, 他可以让你以一种非常优雅且 DRY (Don't repeat yourself) 的方式来创建美观的表单。
- [django-remote-forms](#): 一个平台独立的 Django 表单序列化工具。
- [WTForms](#): 一个灵活的表单验证和呈现库。
- [WTForms-JSON](#): 一个 WTForms 扩展, 用来处理 JSON 数据。

## 数据验证

数据验证库。多用于表单验证。

- [Cerberus](#): 一个映射验证器 (mappings-validator) 。支持多种规则, 提供归一化功能, 可以方便地定制为 Python 风格的 schema 定义。
- [colander](#): 一个用于对从 XML, JSON, HTML 表单获取的数据或其他同样简单的序列化数据进行验证和反序列化的系统。
- [kmatch](#): 一种用于匹配/验证/筛选 Python 字典的语言。
- [schema](#): 一个用于对 Python 数据结构进行验证的库。
- [Schematics](#): 数据结构验证。
- [valideer](#): 轻量级可扩展的数据验证和适配库。
- [voluptuous](#): 一个 Python 数据验证库。主要是为了验证传入 Python 的 JSON, YAML 等数据。
- [jsonschema](#): [JSON Schema](#) 的 python 实现, 用于 JSON 数据的验证。

## 序列化

复杂数据类型序列化相关库。

- [marshmallow](#): 一个轻量级的库, 用于将复杂对象与简单 Python 数据类型相互转换。
- [pysimdjson](#): 与 Python 绑定的 [simdjson](#) 。
- [python-rapidjson](#): [RapidJSON](#) 的 Python 封装。
- [ultrajson](#): 使用 Python 绑定的, 用 C 编写的快速 JSON 解码器和编码器。

## 反垃圾技术

帮助你和电子垃圾进行战斗的库。

- [django-simple-captcha](#): 一个简单、高度可定制的 Django 应用，可以为任何 Django 表单添加验证码。
- [django-simple-spam-blocker](#): 一个用于 Django 的简单的电子垃圾屏蔽工具。

## 标记

用来进行标记的库。

- [django-taggit](#): 简单的 Django 标记工具。

## 管理面板

管理界面库。

- [Ajenti](#): 一个你的服务器值得拥有的管理面板。
- [django-suit](#): Django 管理界面的一个替代品 (仅对于非商业用途是免费的)。
- [django-xadmin](#): Django admin 的一个替代品，具有很多不错的功能。
- [flask-admin](#): 一个用于 Flask 的简单可扩展的管理界面框架。
- [flower](#): 一个对 Celery 集群进行实时监控和提供 web 管理界面的工具。
- [Grappelli](#): Django 管理界面的一个漂亮的皮肤。
- [Wooye](#): 一个 Django 应用，可以为 Python 脚本创建 web 用户界面。
- [django-grappelli](#): 拥有绚丽外观的 Django Admin 界面。
- [django-jet](#): 具有改进功能的现代响应式 Django 管理界面模板。
- [jet-bridge](#): 管理面板框架，适用于任何具有良好 UI 的应用 (例如 Django) 。

## Serverless 框架

使用 Python 开发 Serverless 模型相关的库。

- [python-lambda](#): 在 AWS Lambda 开发和部署 Python 代码的工具包。
- [Zappa](#): 在 AWS Lambda 和 API Gateway 部署 WSGI 应用的工具。

## 静态站点生成器

静态站点生成器是一个软件，它把文本和模板作为输入，然后输出 HTML 文件。

- [Pelican](#): 使用 Markdown 或 ReST 来处理内容，Jinja 2 来制作主题。支持 DVCS, Disqus., AGPL 许可。
- [Cactus](#): 为设计师设计的静态站点生成器。
- [Hyde](#): 基于 Jinja2 的静态站点生成器。
- [Nikola](#): 一个静态网站和博客生成器。
- [Tinkerer](#): Tinkerer 是一个博客引擎/静态站点生成器，由 Sphinx 驱动。
- [Lektor](#): 一个简单易用的静态 CMS 和博客引擎。
- [makesite](#): 简单轻量的站点/博客生成器 (小于 130 行代码)。

## 进程

操作系统进程启动及通信库。

- [envoy](#): 比 Python [subprocess](#) 模块更人性化。
- [sarge](#): 另一种 subprocess 模块的封装。
- [sh](#): 一个完备的 subprocess 替代库。
- [delegator.py](#): 人性化的 [Subprocesses](#) 2.0 版本。

## 并发和并行

用以进行并发和并行操作的库。

- [multiprocessing](#): (Python 标准库) 基于进程的“线程”接口。
- [threading](#): (Python 标准库)更高层的线程接口。
- [eventlet](#): 支持 WSGI 的异步框架。
- [gevent](#): 一个基于协程的 Python 网络库, 使用 [greenlet](#)。
- [Tomorrow](#): 用于产生异步代码的神奇的装饰器语法实现。
- [uvloop](#): 在 libuv 之上超快速实现 asyncio 事件循环。
- [concurrent.futures](#): (Python 标准库) 异步执行可调用对象的高级接口。
- [gevent](#): 使用 [greenlet](#) 且基于协程的 Python 网络库。
- [scoop](#): 支持在 Python 中进行可伸缩并行操作。

## 网络

用于网络编程的库。

- [asyncio](#): (Python 标准库) 异步 I/O, 事件循环, 协程以及任务。
- [trio](#): 异步并发和 I/O 友好的库。
- [Twisted](#): 一个事件驱动的网络引擎。
- [pulsar](#): 事件驱动的并发框架。
- [diesel](#): 基于 Greenlet 的事件 I/O 框架。
- [pyzmq](#): 一个 ZeroMQ 消息库的 Python 封装。
- [Toapi](#): 一个轻巧, 简单, 快速的 Flask 库, 致力于为所有网站提供 API 服务。
- [txZMQ](#): 基于 Twisted 的 ZeroMQ 消息库的 Python 封装。

## WebSocket

帮助使用 WebSocket 的库。

- [AutobahnPython](#): 给 Python、使用的 WebSocket & WAMP 基于 Twisted 和 [asyncio](#)。
- [Crossbar](#): 开源统一应用路由(WebSocket & WAMP for Python on Autobahn)。
- [django-socketio](#): 给 Django 用的 WebSockets。
- [WebSocket-for-Python](#): 为 Python2/3 以及 PyPy 编写的 WebSocket 客户端和服务端库。

## WSGI 服务器

兼容 WSGI 的 web 服务器

- [gunicorn](#): Pre-forked, 部分是由 C 语言编写的。
- [uwsgi](#): uwsgi 项目的目的是开发一组全栈工具, 用来建立托管服务, 由 C 语言编写。
- [bjoern](#): 异步, 非常快速, 由 C 语言编写。
- [fapws3](#): 异步 (仅对于网络端), 由 C 语言编写。
- [meinheld](#): 异步, 部分是由 C 语言编写的。
- [netius](#): 异步, 非常快速。
- [paste](#): 多线程, 稳定, 久经考验。
- [rocket](#): 多线程。
- [waitress](#): 多线程, 是它驱动着 Pyramid 框架。
- [Werkzeug](#): 一个 WSGI 工具库, 驱动着 Flask, 而且可以很方便地嵌入到你的项目中去。

## ASGI 服务器

兼容 ASGI 的 web 服务器。

- [daphne](#): 用于 ASGI 和 ASGI-HTTP 的, 支持 HTTP, HTTP2 和 WebSocket 协议的服务器。
- [uvicorn](#): 使用 uvloop 和 httptools 实现的闪电般快速的 ASGI 服务器。

## RPC 服务器

兼容 RPC 的服务器。

- [SimpleJSONRPCServer](#): 这个库是 JSON-RPC 规范的一个实现。
- [SimpleXMLRPCServer](#): (Python 标准库) 简单的 XML-RPC 服务器实现, 单线程。
- [zeroRPC](#): zerorpc 是一个灵活的 RPC 实现, 基于 [ZeroMQ](#) 和 [MessagePack](#)。
- [RPyC](#) (Remote Python Call): 适用于 Python 的透明且对称的 RPC 库。

## 密码学

- [cryptography](#): 这个软件包意在提供密码学基本内容和方法提供给 Python 开发者。
- [hashids](#): 在 Python 中实现 [hashids](#)。
- [Paramiko](#): SSHv2 协议的 Python (2.6+, 3.3+) , 提供客户端和服务端的功能。
- [Passlib](#): 安全密码存储 / 哈希库。
- [PyCrypto](#): Python 密码学工具箱。
- [PyNaCl](#): 网络和密码学(NaCl) 库的 Python 绑定。

## 图形用户界面

用来创建图形用户界面程序的库。

- [curses](#): 内建的 [ncurses](#) 封装, 用来创建终端图形用户界面。
- [enaml](#): 使用类似 QML 的 Declarative 语法来创建美观的用户界面。
- [kivy](#): 一个用来创建自然用户交互 (NUI) 应用程序的库, 可以运行在 Windows, Linux, Mac OS X, Android 以及 iOS 平台上。
- [pyglet](#): 一个 Python 的跨平台窗口及多媒体库。
- [PyQt](#): 跨平台用户界面框架 [Qt](#) 的 Python 绑定, 支持 Qt v4 和 Qt v5。
- [PySide](#): 跨平台用户界面框架 [Qt](#) 的 Python 绑定, 支持 Qt v4。
- [Tkinter](#): Tkinter 是 Python GUI 的一个事实标准库。
- [Toga](#): 一个 Python 原生的, 操作系统原生的 GUI 工具包。
- [urwid](#): 一个用来创建终端 GUI 应用的库, 支持组件, 事件和丰富的色彩等。
- [wxPython](#): wxPython 是 wxWidgets C++ 类库和 Python 语言混合的产物。
- [PyGObject](#): GLib/GObject/GIO/GTK+ (GTK+3) 的 Python 绑定。
- [Flexx](#): Flexx 是一个纯 Python 语言编写的用来创建 GUI 程序的工具集, 它使用 web 技术进行界面的展示。
- [Eel](#): 用于制作简单离线 HTML/JS GUI 应用的库。
- [PySimpleGUI](#): tkinter, Qt, WxPython 和 Remi 的封装。
- [pywebview](#): 围绕网页视图组件的轻量级跨平台的原生包装。
- [DearPyGui](#): 一个简单的可使用 GPU 加速的 Python GUI 框架。

## 游戏开发

超赞的游戏开发库。

- [Cocos2d](#): cocos2d 是一个用来开发 2D 游戏, 示例和其他图形/交互应用的框架。基于 pyglet。
- [Panda3D](#): 由迪士尼开发的 3D 游戏引擎, 并由卡内基梅隆娱乐技术中心负责维护。使用 C++ 编写, 针对 Python 进行了完全的封装。
- [Pygame](#): Pygame 是一组 Python 模块, 用来编写游戏。
- [PyOgre](#): Ogre 3D 渲染引擎的 Python 绑定, 可以用来开发游戏和仿真程序等任何 3D 应用。
- [PyOpenGL](#): OpenGL 的 Python 绑定及其相关 APIs。
- [PySDL2](#): SDL2 库的封装, 基于 ctypes。
- [RenPy](#): 一个视觉小说 (visual novel) 引擎。
- [Arcade](#): 一个现代 Python 框架, 用于制作具有引人入胜的图形与声音的游戏。

- [Harfang3D](#): 支持3D, VR 与游戏开发的 Python 框架。

## 日志

用来生成和操作日志的库。

- [logging](#): (Python 标准库) 为 Python 提供日志功能。
- [logbook](#): Logging 库的替代品。
- [Eliot](#): 为复杂的和分布式系统创建日志。
- [Raven](#): Sentry 的 Python 客户端。
- [Sentry](#): 实时记录和收集日志的服务器。
- [sentry-python](#): Python 版 Sentry SDK。
- [loguru](#): 旨在带来愉悦体验的 Python 日志库。
- [structlog](#): 结构化日志, 让日志变得简单。

## 测试

进行代码库测试和生成测试数据的库。

- 测试框架
  - [unittest](#): (Python 标准库) 单元测试框架。
  - [nose](#): nose 扩展了 unittest 的功能。
  - [nose2](#) nose 的继任者, 基于 [unittest2](#)。
  - [contexts](#): 一个 Python 3.3+ 的 BDD 框架。受到 C# Machine.Specifications 的启发。
  - [hypothesis](#): Hypothesis 是一个基于先进的 Quickcheck 风格特性的测试库。
  - [mamba](#): Python 的终极测试工具, 拥护 BDD。
  - [pyshould](#): Should 风格的断言, 基于 [PyHamcrest](#)。
  - [pytest](#): 一个成熟的全功能 Python 测试工具。
  - [green](#): 干净, 多彩的测试工具。
  - [pyvows](#): BDD 风格的测试工具, 受 Vows.js 的启发。
  - [Robot Framework](#): 一个通用的自动化测试框架。
  - [tox](#): 自动化测试与发布的工具, 支持多个 Python 版本。
- GUI / Web 测试
  - [Selenium](#): [Selenium](#) WebDriver 的 Python 绑定。
  - [PyAutoGUI](#): PyAutoGUI 是一个人性化的跨平台 GUI 自动测试模块。
  - [locust](#): 使用 Python 编写的, 可扩展的用户加载测试工具。
  - [sixpack](#): 一个和语言无关的 A/B 测试框架。
  - [splinter](#): 开源的 web 应用测试工具。
  - [Schemathesis](#): 基于属性的自动测试工具, 用于测试使用 Open API / Swagger 规范构建的 Web 应用程序。
- Mock 测试
  - [mock](#): (Python 标准库) 一个用于伪造测试的库。
  - [doublex](#): Python 的一个功能强大的 doubles 测试框架。
  - [freezeGUN](#): 通过伪造日期模块来生成不同的时间。
  - [httmock](#): 针对 Python 2.6+ 和 3.2+ 生成 伪造请求的库。
  - [httpretty](#): Python 的 HTTP 请求 mock 工具。
  - [responses](#): 伪造 Python 中的 requests 库的一个通用库。
  - [VCR.py](#): 在你的测试中记录和重放 HTTP 交互。
  - [mocket](#): gevent/asyncio/SSL 支持的 socket mock 框架。
- 对象工厂
  - [factoryboy](#): 一个 Python 用的测试固件 (test fixtures) 替代库。

- [mixer](#): 另外一个测试固件 (test fixtures) 替代库, 支持 Django, Flask, SQLAlchemy, Peewee 等。
- [modelmommy](#): 为 Django 测试创建随机固件。
- 代码覆盖率
  - [coverage](#): 代码覆盖率测量。
  - [Codecov](#): 一个代码覆盖率测试工具, 为开源项目提供免费代码覆盖率测试服务。
- 伪数据
  - [faker](#): 一个 Python 库, 用来生成伪数据。
  - [fake2db](#): 伪数据库生成器。
  - [mimesis](#): 一个帮助你生成伪数据的 Python 库。
  - [radar](#): 生成随机的日期/时间。
- 错误处理
  - [FuckIt.py](#): 使用最先进的技术来保证你的 Python 代码无论对错都能继续运行。

## 渗透测试

渗透测试相关框架和工具。

- [fsociety](#): 一款渗透测试框架。
- [setoolkit](#): 社会工程工具包。
- [sqlmap](#): 自动 SQL 注入和数据库接管工具。

## 代码分析和 Lint 工具

进行代码分析、解析和操作代码库的库和工具。

- 代码分析
  - [coala](#): 语言独立和易于扩展的代码分析应用程序。
  - [code2flow](#): 把你的 Python 和 JavaScript 代码转换为流程图。
  - [pycallgraph](#): 这个库可以把你的 Python 应用的流程(调用图)进行可视化。
  - [pysonar2](#): Python 类型推断和检索工具。
  - [prospector](#): 分析 Python 代码的工具。
  - [vulture](#): 用于发现和分析无效 Python 代码的工具。
- Lint 工具
  - [Flake8](#): 模块化源码检查工具, 提供与 `pycodestyle`、`pyflakes`、McCabe 相关的装饰器。
    - [awesome-flake8-extensions](#)
  - [Pylint](#): 一个完全可定制的源码分析器。
  - [YAPF](#): Google 的 Python 代码格式化工具。
  - [pylama](#): Python 和 JavaScript 的代码审查工具。
  - [wemake-python-styleguide](#): 有史以来最严格的 Python 代码审查工具。
- 代码格式化
  - [autopep8](#): 自动格式化 Python 代码, 以使其符合 PEP8 规范。
  - [black](#): 一个坚定的 Python 代码格式化工具。
  - [isort](#): 用于纠正包导入顺序的 Python 库。
- 静态类型检查, 也可以参考 [awesome-python-typing](#)
  - [mypy](#): 在编译期间检查变量类型。
  - [pyre-check](#): 性能类型检查。
  - [typeshed](#): 带有静态类型的 Python 库存根的集合。
- 静态类型注释生成器



- [MonkeyType](#): 通过收集运行时的类型来为 Python 生成静态类型注释的系统。
- [pyannotate](#): 自动生成符合 PEP-484 的注解。
- [pytype](#): 检查和推断 Python 代码中的类型, 无需添加注解。

## 调试工具

用来进行代码调试的库。

- 调试器
  - [ipdb](#): IPython 启用的 [pdb](#)。
  - [pudb](#): 全屏, 基于控制台的 Python 调试器。
  - [pdb++](#): 另一种 pdb 的替代。
  - [pyringe](#): 可以在 Python 进程中附加和注入代码的调试器。
  - [wdb](#): 一个奇异的 web 调试器, 通过 WebSockets 工作。
  - [winpdb](#): 一个具有图形用户界面的 Python 调试器, 可以进行远程调试, 基于 rpdb2。
- 追踪器
  - [lptrace](#): 为 Python 程序打造的 [strace](#)。
  - [manhole](#): 调试UNIX套接字连接, 并显示所有线程的堆栈跟踪和交互式提示。
  - [pyringe](#): 能够附加到 Python 进程并将代码注入Python进程的调试器。
  - [python-hunter](#): 一个灵活的代码追踪工具包。
- 性能分析器
  - [lineprofiler](#): 逐行性能分析。
  - [Memory Profiler](#)、[内存](#): 监控 Python 代码的内存使用。
  - [profiling](#): 一个交互式 Python 性能分析工具。
  - [py-spy](#): Python 程序采样分析器, 使用 Rust 实现。
  - [pyflame](#): 用于 Python 的跟踪分析器。
  - [vprof](#): 视觉 Python 分析器。
- 其他
  - [pyelftools](#): 解析和分析 ELF 文件以及 DWARF 调试信息。
  - [python-statsd](#): [statsd](#) 服务器的 Python 客户端。
  - [django-debug-toolbar](#): 为 Django 显示各种调试信息。
  - [django-devserver](#): 一个 Django 运行服务器的替代品。
  - [flask-debugtoolbar](#): django-debug-toolbar 的 flask 版。
  - [icecream](#): 通过一个简单的函数调用检查变量、表达式和程序执行情况。

## 科学计算和数据分析

用来进行科学计算和数据分析的库。

- [astropy](#): 一个天文学 Python 库。
- [bcbio-nextgen](#): 这个工具箱为全自动高通量测序分析提供符合最佳实践的处理流程。
- [bccb](#): 生物分析相关代码集合。
- [Biopython](#): Biopython 是一组可以免费使用的用来进行生物计算的工具。
- [blaze](#): NumPy 和 Pandas 的大数据接口。
- [cclib](#): 一个用来解析和解释计算化学软件包输出结果的库。
- [NetworkX](#): 一个为复杂网络设计的高性能软件。
- [Neupy](#): 执行和测试各种不同的人工神经网络算法。
- [NumPy](#): 使用 Python 进行科学计算的基础包。
- [Open Babel](#): 一个化学工具箱, 用来描述多种化学数据。
- [Open Mining](#): 使用 Python 挖掘商业情报 (BI) (Pandas web 接口)。
- [orange](#): 通过可视化编程或 Python 脚本进行数据挖掘, 数据可视化, 分析和机器学习。
- [Pandas](#): 提供高性能, 易用的数据结构和数据分析工具。



- [PyDy](#): PyDy 是 Python Dynamics 的缩写，用来为动力学运动建模工作流程提供帮助，基于 NumPy, SciPy, IPython 和 matplotlib。
- [PyMC](#): 马尔科夫链蒙特卡洛采样工具。
- [RDKit](#): 化学信息学和机器学习软件。
- [SciPy](#): 由一些基于 Python，用于数学，科学和工程的开源软件构成的生态系统。
- [statsmodels](#): 统计建模和计量经济学。
- [SymPy](#): 一个用于符号数学的 Python 库。
- [zipline](#): 一个 Python 算法交易库。
- [Bayesian-belief-networks](#): 优雅的贝叶斯理念网络框架。
- [AWS Data Wrangler](#): AWS 平台上使用的 Pandas。
- [Optimus](#): 在使用 PySpark 时，让敏捷数据科学工作流程变得简单。
- [Colour](#): 大量色彩理论转换和算法的实现。
- [Karate Club](#): 用于图形结构化数据的无监督机器学习工具箱。
- [NIPY](#): 神经影像学工具箱集合。
- [ObsPy](#): 地震学 Python 工具箱。
- [QuTiP](#): Python 版 Quantum 工具箱。
- [SimPy](#): 一个基于过程的离散事件模拟框架。

## 数据可视化

进行数据可视化的库。参见: [awesome-javascript](#)。

- [matplotlib](#): 一个 Python 2D 绘图库。
- [bokeh](#): 用 Python 进行交互式 web 绘图。
- [ggplot](#): ggplot2 给 R 提供的 API 的 Python 版本。
- [plotly](#): 协同 Python 和 matplotlib 工作的 web 绘图库。
- [pyecharts](#): 基于百度 Echarts 的数据可视化库。
- [pygal](#): 一个 Python SVG 图表创建工具。
- [pygraphviz](#): Graphviz 的 Python 接口。
- [PyQtGraph](#): 交互式实时 2D/3D/ 图像绘制及科学/工程学组件。
- [SnakeViz](#): 一个基于浏览器的 Python's cProfile 模块输出结果查看工具。
- [vincent](#): 把 Python 转换为 Vega 语法的转换工具。
- [VisPy](#): 基于 OpenGL 的高性能科学可视化工具。
- [Altair](#): 用于 Python 的声明式统计可视化库。
- [bqplot](#): Jupyter Notebook 的交互式绘图库。
- [Cartopy](#): 具有 matplotlib 支持的 python 制图库。
- [Dash](#): 构建在 Flask、React 和 Plotly 之上，旨在用于分析 Web 应用程序。
  - [awesome-dash](#)
- [diagrams](#): 用图表作为代码。
- [plotnine](#): 基于 ggplot2 的 Python 图形语法。
- [PyGraphviz](#): [Graphviz](#) 的 Python 接口。
- [Seaborn](#): 使用 Matplotlib 进行统计数据可视化。

## 计算机视觉

计算机视觉相关库。

- [OpenCV](#): 开源计算机视觉库。

- [pyocr](#): Tesseract 和 Cuneiform 的包装库。
- [pytesseract](#): [Google Tesseract OCR](#) 的另一包装库。
- [SimpleCV](#): 一个用来创建计算机视觉应用的开源框架。
- [EasyOCR](#): 支持40多种语言的即用型 OCR。
- [Face Recognition](#): 简单的面部识别库。
- [Kornia](#): PyTorch 的开源差异化计算机视觉库。
- [tesseractocr](#): 另一个简单的, 兼容 Pillow 的 `tesseract-ocr` API 装饰器, 可用于 OCR。

## 深度学习

神经网络和深度学习相关框架。也可以参考 [awesome-deep-learning](#)。

- [Caffe](#): 一个 [Caffe](#) 的 python 接口。
- [Caffe2](#): 一个轻量级的, 模块化的, 可扩展的深度学习框架。
- [keras](#): 以 tensorflow/theano/CNTK 为后端的深度学习封装库, 快速上手神经网络。
- [MXNet](#): 一个高效和灵活的深度学习框架。
- [Pytorch](#): 一个具有张量和动态神经网络, 并有强大 GPU 加速能力的深度学习框架。
- [SerpentAI](#): 游戏代理框架, 可使用任意视频游戏作为深度学习沙箱。
- [Theano](#): 一个快速数值计算库。
- [TensorFlow](#): 谷歌开源的最受欢迎的深度学习框架。
- [skflow](#): 一个 [TensorFlow](#) 的简化接口(模仿 scikit-learn)。
- [hebel](#): GPU 加速的深度学习库。
- [pydeep](#): Python 深度学习库。

## 机器学习

机器学习相关库, 也可以参考 [awesome-machine-learning](#)。

- [Crab](#): 灵活、快速的推荐引擎。
- [NuPIC](#): 智能计算 Numenta 平台。
- [pattern](#): Python 网络挖掘模块。
- [PyBrain](#): 另一个 Python 机器学习库。
- [Pylearn2](#): 一个基于 [Theano](#) 的机器学习库。
- [python-recsys](#): 一个用来实现推荐系统的 Python 库。
- [scikit-learn](#): 基于 SciPy 构建的机器学习 Python 模块。
- [vowpalporpoise](#): 轻量级 [Vowpal Wabbit](#) 的 Python 封装。
- [gym](#): 开发和比较强化学习算法的工具包。
- [H2O](#): 开源快速可扩展的机器学习平台。
- [Metrics](#): 机器学习的评估指标。
- [vowpal\\_porpoise](#): Python 版 [Vowpal Wabbit](#) 装饰器。
- [xgboost](#): 可扩展, 便携式和分布式梯度提升库。
- [MindsDB](#): MindsDB是现有数据库的开源AI层, 可让使用标准查询轻松地进行开发, 训练和部署最新的机器学习模型。

## 推荐系统

用于构建推荐系统的相关库。

- [annoy](#): 对 C++/Python 实现的近似近邻算法进行了内存优化。
- [fastFM](#): Factorization Machine 相关库。
- [implicit](#): 对隐式数据集进行协作过滤的快速 Python 实现。
- [libffm](#): Field-aware Factorization Machine (FFM) 相关库。
- [lightfm](#): 很多流行的推荐算法的 Python 实现。
- [spotlight](#): 使用 PyTorch 实现的深度推荐模型。

- [Surprise](#): 用于构建和分析推荐系统的科学工具。
- [tensorrec](#): TensorFlow 的推荐引擎框架。

## 分布式计算

分布式计算相关的框架和库。

- [dpark](#): Spark 的 Python 克隆版, 一个类似 MapReduce 的框架。
- [dumbo](#): 这个 Python 模块可以让人轻松的编写和运行 Hadoop 程序。
- [luigi](#): 这个模块帮你构建批处理作业的复杂流水线。
- [mrjob](#): 在 Hadoop 或 Amazon Web Services 上运行 MapReduce 任务。
- [dask](#): 用于分析计算的灵活的并行计算库。
- [PySpark](#): Spark 的 Python API。
- [Ray](#): 一个用于并行和分布式 Python 的系统, 它统一了机器学习生态系统。
- [faust](#): 一个 Python 流处理库, 核心思想来源 [Kafka Streams](#)。
- [streamparse](#): 运行针对事实数据流的 Python 代码。集成了 [Apache Storm](#)。

## 函数式编程

使用 Python 进行函数式编程。

- [CyToolz](#): Toolz 的 Cython 实现: 高性能函数式工具。
- [fn.py](#): 在 Python 中进行函数式编程: 实现了一些享受函数式编程缺失的功能。
- [fancy](#): 炫酷又实用的函数式工具。
- [Toolz](#): 一组用于迭代器, 函数和字典的函数式编程工具。
- [Coconut](#): 为了简单、优雅、更 Pythonic 的函数式编程而构建的 Python 变体。
- [more-itertools](#): 比 `itertools` 拥有更多的可迭代对象的操作方式。
- [returns](#): 一个类型安全的单元、转换器与合成工具集合。

## 第三方 API

用来访问第三方 API 的库。参见: [List of Python API Wrappers and Libraries](#)。

- [apache-libcloud](#): 一个为各种云设计的 Python 库。
- [boto3](#): Amazon Web Services 的 Python 接口。
- [django-wordpress](#): Django 的 WordPress 模型与视图。
- [facebook-sdk](#): Facebook 平台的 Python SDK。
- [facepy](#): Facepy 让和 Facebook's Graph API 的交互变得更容易。
- [gmail](#): Gmail 的 Python 接口。
- [google-api-python-client](#): Python 用的 Google APIs 客户端库。
- [gspreed](#): Google 电子表格的 Python API。
- [twython](#): Twitter API 的封装。

## DevOps 工具

用于 DevOps 的软件和库。

- [Ansible](#): 一个非常简单的 IT 自动化平台。
- [SaltStack](#): 基础设施自动化和管理系统。
- [OpenStack](#): 用于构建私有和公有云的开源软件。
- [Docker Compose](#): 快速, 分离的开发环境, 使用 Docker。
- [Fabric](#): 一个简单的, Python 风格的工具, 用来进行远程执行和部署。
- [cuisine](#): 为 Fabric 提供一系列高级函数。
- [Fabtools](#): 一个用来编写超赞的 Fabric 文件的工具。
- [gitapi](#): Git 的纯 Python API。

- [hgapi](#): Mercurial 的纯 Python API。
- [honcho](#): [Foreman](#) 的 Python 克隆版, 用来管理基于 [Procfile](#) 的应用。
- [pexpect](#): 在一个伪终端中控制交互程序, 就像 GNU expect 一样。
- [psutil](#): 一个跨平台进程和系统工具模块。
- [supervisor](#): UNIX 的进程控制系统。
- [cloudinit](#): 一个多分发包, 用于处理云实例的早期初始化。
- [pyinfra](#): 一个通用的 CLI 工具包和 python 库, 用于自动化的基础设施。
- [honcho](#): [Foreman](#) 的 Python 克隆版, 用于管理基于 Procfile 的应用。
- [BorgBackup](#): 具有压缩和加密功能的重复数据删除存档器。
- [docker-compose](#): 使用 [Docker](#) 的快速独立的开发环境。

## 任务调度

任务调度库。

- [APScheduler](#): 轻巧但强大的进程内任务调度, 使你可以调度函数。
- [django-schedule](#): 一个 Django 排程应用。
- [doit](#): 一个任务执行和构建工具。
- [gunnery](#): 分布式系统使用的多用途任务执行工具, 具有 web 交互界面。
- [Joblib](#): 一组为 Python 提供轻量级作业流水线的工具。
- [Plan](#): 如有神助地编写 crontab 文件。
- [schedule](#): 人性化的 Python 任务调度库。
- [Spiff](#): 使用纯 Python 实现的强大的工作流引擎。
- [TaskFlow](#): 一个可以让你方便执行任务的 Python 库, 一致并且可靠。
- [Airflow](#): 是一个工作流分配管理系统, 通过有向非循环图的方式管理任务流程, 设置任务依赖关系和时间调度。
- [Prefect](#): 一个现代的工作流程编排框架, 使构建、计划和监视健壮的数据管道变得容易。

## 外来函数接口

使用外来函数接口的库。

- [cffi](#): 用来调用 C 代码的外来函数接口。
- [ctypes](#): (Python 标准库) 用来调用 C 代码的外来函数接口。
- [PyCUDA](#): Nvidia CUDA API 的封装。
- [SWIG](#): 简化的封装和接口生成器。

## 重构

Python 重构相关库和工具。

- [Bicycle Repair Man](#): Python 的重构工具。
- [Bowler](#): 适用于现代Python的安全代码重构。
- [Rope](#): 一个 Python 的重构库。

## 高性能

让 Python 更快的库。

- [Cython](#): 优化的 Python 静态编译器。使用类型混合使 Python 编译成 C 或 C++ 模块来获得性能的极大提升。
- [CLPython](#): 用 Common Lisp 编写的 Python 编程语言的实现。
- [Grumpy](#): 编译器比解释器更强大的 cpython2.7 替代品 (alpha) 。
- [IronPython](#): 用 C# 编写的 Python 编程语言的实现。
- [Jython](#): 为 JVM 用 Java 编写的 Python 编程语言的实现。

- [MicroPython](#): 精简高效的 Python 编程语言实现。
- [Pyjion](#): 基于 CoreCLR 的 Python JIT。
- [Numba](#): Python JIT (just in time) 编译器, 针对科学用的 Python, 由 Cython 和 NumPy 的开发者开发。
- [PeachPy](#): 嵌入 Python 的 x86-64 汇编器。可以被用作 Python 内联的汇编器或者是独立的汇编器, 用于 Windows, Linux, OS X, Native Client 或者 Go。
- [PyPy](#): 使用 Python 实现的 Python。解释器使用黑魔法加快 Python 运行速度且不需要加入额外的类型信息。
- [Pyston](#): 使用 LLVM 和现代 JIT 技术构建的 Python 实现, 目标是为了获得很好的性能。
- [Stackless Python](#): 一个强化版的 Python。

## 微软的 Windows 平台

在 Windows 平台上进行 Python 编程。

- [Python\(x,y\)](#): 面向科学应用的 Python 发行版, 基于 Qt 和 Spyder。
- [pythonlibs](#): 非官方的 Windows 平台 Python 扩展二进制包。
- [PythonNet](#): Python 与 .NET 公共语言运行库 (CLR) 的集成。
- [PyWin32](#): 针对 Windows 的 Python 扩展。
- [WinPython](#): Windows 7/8 系统下便携式开发环境。

## 网络可视化和 SDN

用来进行网络可视化和 SDN(软件定义网络)的工具和库。

- [Mininet](#): 一款流行的网络模拟器以及用 Python 编写的 API。
- [POX](#): 一个针对基于 Python 的软件定义网络应用 (例如 OpenFlow SDN 控制器) 的开源开发平台。
- [Pyretic](#): 火热的 SDN 编程语言中的一员, 为网络交换机和模拟器提供强大的抽象能力。
- [SDX Platform](#): 基于 SDN 的 IXP 实现, 影响了 Mininet, POX 和 Pyretic。
- [NRU](#): 一个基于组件的软件定义网络框架。
- [napalm](#): 可跨供应商 API 来操纵网络设备。

## 硬件

用来对硬件进行编程的库。

- [ino](#): 操作 [Arduino](#) 的命令行工具。
- [Pyro](#): Python 机器人编程库。
- [PyUserInput](#): 跨平台的, 控制鼠标和键盘的模块。
- [scapy](#): 一个非常棒的操作数据包的库。
- [wifi](#): 一个 Python 库和命令行工具用来在 Linux 平台上操作 WiFi。
- [Pingo](#): Pingo 为类似 Raspberry Pi, pcDuino, Intel Galileo 等设备提供统一的 API 用以编程。
- [keyboard](#): 在 Windows 和 Linux 上挂钩并模拟全局键盘事件。
- [mouse](#): 在 Windows 和 Linux 上挂钩并模拟全局鼠标事件。

## 兼容性

帮助从 Python 2 向 Python 3 迁移的库。

- [Python-Future](#): 这就是 Python 2 和 Python 3 之间丢失的那个兼容性层。
- [Six](#): Python 2 和 3 的兼容性工具。
- [modernize](#): 使 Python 代码更加现代化以便最终迁移到 Python 3。

## 算法和设计模式

数据结构、算法和设计模式的 Python 实现。也可以参考 [awesome-algorithms](#)。

- 算法
  - [algorithms](#): 数据结构和算法的简单示例。
  - [python-ds](#): 用于面试的数据结构和算法的集合。
  - [sortedcontainers](#): 排序集合的快速的纯 Python 实现。
  - [TheAlgorithms](#): 所有算法的 Python 实现。
- 设计模式
  - [PyPattyrn](#): 一个简单而有效的库，用于实现常见的设计模式。
  - [python-patterns](#): 一个 Python 设计模式集合。
  - [transitions](#): 轻量级的，面向对象的有限状态机实现。

## 内置类的增强版实现

一些 Python 内置类的增强版实现库。

- [attrs](#): 一个在类定义时可替换 `__init__`, `__eq__`, `__repr__` 等方法的样板。
- [bidict](#): 高效的 Pythonic 的双向映射数据结构和相关功能。
- [Box](#): 具有高级点符号访问权限的 Python 字典。
- [dataclasses](#): (Python 标准库) 数据类。
- [DottedDict](#): 提供一种使用点路径符号访问列表和字典的方法的库。

## 机器人

机器人相关库。

- [PythonRobotics](#): 各种具有可视化效果的机器人算法的汇总。
- [rospy](#): ROS (Robot Operating System) 库。

## 聊天工具

聊天机器人开发相关的库。

- [errbot](#): 实现 ChatOps 的最简单最受欢迎的聊天机器人。

## 编辑器插件和 IDE

- Emacs
  - [elpy](#): Emacs Python 开发环境。
- Sublime Text
  - [anaconda](#): Anaconda 可将功能齐全的 Python 开发 IDE 转换为 Sublime Text 3。
  - [SublimeJEDI](#): 一个很棒的自动补全库 Jedi 的 Sublime Text 插件。
- Vim
  - [jedi-vim](#): 用于 Python 的 Jedi 自动补全库的 Vim 绑定。
  - [python-mode](#): 一个将 Vim 转换为 Python IDE 的多合一插件。
  - [YouCompleteMe](#): 包含 Jedi 补全的 Python 引擎。
- Visual Studio
  - [PTVS](#): Visual Studio Python 工具。
- Visual Studio Code
  - [Python](#): 对 Python 有丰富支持的官方 VSCode 扩展。
- IDE

- [PyCharm](#): JetBrains 提供的商业 Python IDE, 也有免费的社区版。
- [spyder](#): 开源 Python IDE。

## 企业级应用集成

企业级环境中用于集成的平台与工具。

- [Zato](#): ESB, SOA, REST, APIs 以及云的 Python 整合。

## GraphQL

GraphQL 相关库。

- [graphql](#): Python GraphQL 框架。
- [tartiflette-aiohttp](#): Tartiflette 的基于 aiohttp 的装饰器, 用于通过 HTTP 公开 GraphQL API。
- [tartiflette-asgi](#): Tartiflette GraphQL 引擎的 ASGI 支持。
- [tartiflette](#): 支持 Python 3.6+ 和 asyncio 的 SDL 优先的 GraphQL 引擎实现。

## 杂项

不属于上面任何一个类别, 但是非常有用的库。

- [blinker](#): 快速的 Python 运行时信号/事件分配系统。
- [boltons](#): 一组纯 Python 实用工具。
- [itsdangerous](#): 将受信任的数据传递到不受信任的环境的帮助工具。
- [magenta](#): 使用人工智能生成音乐与艺术的工具。
- [pluginbase](#): 一个简单但灵活的 Python 插件系统。
- [tryton](#): 一个通用业务框架。