

vinta /
awesome-python

<> Code Pull requests 402 Actions Security Insights



master ▾

awesome-python / README.md



vinta update README

2 months ago



1198 lines (927 loc) · 76.3 KB

Preview

Code

Blame

Raw



Awesome Python

An opinionated list of awesome Python frameworks, libraries, software and resources.

Inspired by [awesome-php](#).

- [Awesome Python](#)
 - [Admin Panels](#)
 - [Algorithms and Design Patterns](#)
 - [ASGI Servers](#)
 - [Asynchronous Programming](#)
 - [Audio](#)
 - [Authentication](#)
 - [Build Tools](#)
 - [Built-in Classes Enhancement](#)
 - [Caching](#)
 - [ChatOps Tools](#)
 - [CMS](#)
 - [Code Analysis](#)
 - [Command-line Interface Development](#)
 - [Command-line Tools](#)
 - [Computer Vision](#)
 - [Configuration Files](#)
 - [Cryptography](#)
 - [Data Analysis](#)
 - [Data Validation](#)
 - [Data Visualization](#)

- [Database Drivers](#)
- [Database](#)
- [Date and Time](#)
- [Debugging Tools](#)
- [Deep Learning](#)
- [DevOps Tools](#)
- [Distributed Computing](#)
- [Distribution](#)
- [Documentation](#)
- [Downloader](#)
- [Editor Plugins and IDEs](#)
- [Email](#)
- [Environment Management](#)
- [File Manipulation](#)
- [Functional Programming](#)
- [Game Development](#)
- [Geolocation](#)
- [GUI Development](#)
- [Hardware](#)
- [HTML Manipulation](#)
- [HTTP Clients](#)
- [Image Processing](#)
- [Implementations](#)
- [Interactive Interpreter](#)
- [Internationalization](#)
- [Job Scheduler](#)
- [Logging](#)
- [Machine Learning](#)
- [Miscellaneous](#)
- [Natural Language Processing](#)
- [Network Virtualization](#)
- [News Feed](#)
- [ORM](#)
- [Package Management](#)
- [Package Repositories](#)
- [Penetration testing](#)
- [Permissions](#)
- [Processes](#)

- [Recommender Systems](#)
- [Refactoring](#)
- [RESTful API](#)
- [Robotics](#)
- [RPC Servers](#)
- [Science](#)
- [Search](#)
- [Serialization](#)
- [Serverless Frameworks](#)
- [Shell](#)
- [Specific Formats Processing](#)
- [Static Site Generator](#)
- [Tagging](#)
- [Task Queues](#)
- [Template Engine](#)
- [Testing](#)
- [Text Processing](#)
- [Third-party APIs](#)
- [URL Manipulation](#)
- [Video](#)
- [Web Asset Management](#)
- [Web Content Extracting](#)
- [Web Crawling](#)
- [Web Frameworks](#)
- [WebSocket](#)
- [WSGI Servers](#)
- [Resources](#)
 - [Newsletters](#)
 - [Podcasts](#)
- [Contributing](#)

Admin Panels

Libraries for administrative interfaces.

- [ajenti](#) - The admin panel your servers deserve.
- [django-grappelli](#) - A jazzy skin for the Django Admin-Interface.
- [flask-admin](#) - Simple and extensible administrative interface framework for Flask.

- [flower](#) - Real-time monitor and web admin for Celery.
- [jet-bridge](#) - Admin panel framework for any application with nice UI (ex Jet Django).
- [wooey](#) - A Django app which creates automatic web UIs for Python scripts.

Algorithms and Design Patterns

Python implementation of data structures, algorithms and design patterns. Also see [awesome-algorithms](#).

- Algorithms
 - [algorithms](#) - Minimal examples of data structures and algorithms.
 - [python-ds](#) - A collection of data structure and algorithms for coding interviews.
 - [sortedcontainers](#) - Fast and pure-Python implementation of sorted collections.
 - [thealgorithms](#) - All Algorithms implemented in Python.
- Design Patterns
 - [pypattnrn](#) - A simple yet effective library for implementing common design patterns.
 - [python-patterns](#) - A collection of design patterns in Python.
 - [transitions](#) - A lightweight, object-oriented finite state machine implementation.

ASGI Servers

[ASGI](#)-compatible web servers.

- [daphne](#) - A HTTP, HTTP2 and WebSocket protocol server for ASGI and ASGI-HTTP.
- [uvicorn](#) - A lightning-fast ASGI server implementation, using uvloop and httptools.
- [hypercorn](#) - An ASGI and WSGI Server based on Hyper libraries and inspired by Gunicorn.

Asynchronous Programming

Libraries for asynchronous, concurrent and parallel execution. Also see [awesome-asyncio](#).

- [asyncio](#) - (Python standard library) Asynchronous I/O, event loop, coroutines and tasks.
 - [awesome-asyncio](#)
- [concurrent.futures](#) - (Python standard library) A high-level interface for asynchronously executing callables.
- [multiprocessing](#) - (Python standard library) Process-based parallelism.
- [trio](#) - A friendly library for async concurrency and I/O.
- [twisted](#) - An event-driven networking engine.
- [uvloop](#) - Ultra fast asyncio event loop.
- [eventlet](#) - Asynchronous framework with WSGI support.
- [gevent](#) - A coroutine-based Python networking library that uses [greenlet](#).

Audio

Libraries for manipulating audio and its metadata.

- Audio
 - [audioread](#) - Cross-library (GStreamer + Core Audio + MAD + FFmpeg) audio decoding.
 - [audioFlux](#) - A library for audio and music analysis, feature extraction.
 - [dejavu](#) - Audio fingerprinting and recognition.
 - [kapre](#) - Keras Audio Preprocessors.
 - [librosa](#) - Python library for audio and music analysis.
 - [matchering](#) - A library for automated reference audio mastering.
 - [mingus](#) - An advanced music theory and notation package with MIDI file and playback support.
 - [pyaudioanalysis](#) - Audio feature extraction, classification, segmentation and applications.
 - [pydub](#) - Manipulate audio with a simple and easy high level interface.
 - [timeside](#) - Open web audio processing framework.
- Metadata
 - [beets](#) - A music library manager and [MusicBrainz](#) tagger.
 - [eyed3](#) - A tool for working with audio files, specifically MP3 files containing ID3 metadata.
 - [mutagen](#) - A Python module to handle audio metadata.
 - [tinytag](#) - A library for reading music meta data of MP3, OGG, FLAC and Wave files.

Authentication

Libraries for implementing authentications schemes.

- OAuth
 - [authlib](#) - JavaScript Object Signing and Encryption draft implementation.
 - [django-allauth](#) - Authentication app for Django that "just works."
 - [django-oauth-toolkit](#) - OAuth 2 goodies for Django.
 - [oauthlib](#) - A generic and thorough implementation of the OAuth request-signing logic.
- JWT
 - [pyjwt](#) - JSON Web Token implementation in Python.
 - [python-jose](#) - A JOSE implementation in Python.

Build Tools

Compile software from source code.

- [bitbake](#) - A make-like build tool for embedded Linux.
- [buildout](#) - A build system for creating, assembling and deploying applications from multiple parts.
- [platformio](#) - A console tool to build code with different development platforms.
- [pybuilder](#) - A continuous build tool written in pure Python.
- [scons](#) - A software construction tool.

Built-in Classes Enhancement

Libraries for enhancing Python built-in classes.

- [attrs](#) - Replacement for `__init__`, `__eq__`, `__repr__`, etc. boilerplate in class definitions.
- [bidict](#) - Efficient, Pythonic bidirectional map data structures and related functionality..
- [box](#) - Python dictionaries with advanced dot notation access.
- [dataclasses](#) - (Python standard library) Data classes.
- [dotteddict](#) - A library that provides a method of accessing lists and dicts with a dotted path notation.

CMS

Content Management Systems.

- [feincms](#) - One of the most advanced Content Management Systems built on Django.
- [indico](#) - A feature-rich event management system, made @ [CERN](#).
- [wagtail](#) - A Django content management system.

Caching

Libraries for caching data.

- [beaker](#) - A WSGI middleware for sessions and caching.
- [django-cache-machine](#) - Automatic caching and invalidation for Django models.
- [django-cacheops](#) - A slick ORM cache with automatic granular event-driven invalidation.
- [dogpile.cache](#) - dogpile.cache is a next generation replacement for Beaker made by the same authors.
- [hermescache](#) - Python caching library with tag-based invalidation and dogpile effect prevention.
- [pylibmc](#) - A Python wrapper around the [libmemcached](#) interface.
- [python-diskcache](#) - SQLite and file backed cache backend with faster lookups than memcached and redis.

ChatOps Tools

Libraries for chatbot development.

- [errbot](#) - The easiest and most popular chatbot to implement ChatOps.

Code Analysis

Tools of static analysis, linters and code quality checkers. Also see [awesome-static-analysis](#).

- Code Analysis
 - [code2flow](#) - Turn your Python and JavaScript code into DOT flowcharts.
 - [prospector](#) - A tool to analyse Python code.
 - [vulture](#) - A tool for finding and analysing dead Python code.
- Code Linters
 - [flake8](#) - A wrapper around `pycodestyle`, `pyflakes` and McCabe.
 - [awesome-flake8-extensions](#)
 - [pylint](#) - A fully customizable source code analyzer.
- Code Formatters
 - [black](#) - The uncompromising Python code formatter.
 - [isort](#) - A Python utility / library to sort imports.
 - [yapf](#) - Yet another Python code formatter from Google.
- Static Type Checkers, also see [awesome-python-typing](#)
 - [mypy](#) - Check variable types during compile time.
 - [pyre-check](#) - Performant type checking.
 - [typeshed](#) - Collection of library stubs for Python, with static types.
- Static Type Annotations Generators
 - [monkeytype](#) - A system for Python that generates static type annotations by collecting runtime types.
 - [pytype](#) - Pytype checks and infers types for Python code - without requiring type annotations.

Command-line Interface Development

Libraries for building command-line applications.

- Command-line Application Development
 - [cement](#) - CLI Application Framework for Python.
 - [click](#) - A package for creating beautiful command line interfaces in a composable way.
 - [cliff](#) - A framework for creating command-line programs with multi-level commands.
 - [python-fire](#) - A library for creating command line interfaces from absolutely any Python object.
 - [python-prompt-toolkit](#) - A library for building powerful interactive command lines.
- Terminal Rendering
 - [alive-progress](#) - A new kind of Progress Bar, with real-time throughput, eta and very cool animations.
 - [asciimatics](#) - A package to create full-screen text UIs (from interactive forms to ASCII animations).
 - [bashplotlib](#) - Making basic plots in the terminal.
 - [colorama](#) - Cross-platform colored terminal text.
 - [rich](#) - Python library for rich text and beautiful formatting in the terminal. Also provides a great `RichHandler` log handler.
 - [tqdm](#) - Fast, extensible progress bar for loops and CLI.

Command-line Tools

Useful CLI-based tools for productivity.

- Productivity Tools
 - [copier](#) - A library and command-line utility for rendering projects templates.
 - [cookiecutter](#) - A command-line utility that creates projects from cookiecutters (project templates).
 - [doitlive](#) - A tool for live presentations in the terminal.
 - [howdoi](#) - Instant coding answers via the command line.
 - [invoke](#) - A tool for managing shell-oriented subprocesses and organizing executable Python code into CLI-invokable tasks.
 - [pathpicker](#) - Select files out of bash output.
 - [thefuck](#) - Correcting your previous console command.
 - [tmuxp](#) - A [tmux](#) session manager.
 - [try](#) - A dead simple CLI to try out python packages - it's never been easier.
- CLI Enhancements
 - [httpie](#) - A command line HTTP client, a user-friendly cURL replacement.
 - [iredis](#) - Redis CLI with autocompletion and syntax highlighting.
 - [litecli](#) - SQLite CLI with autocompletion and syntax highlighting.
 - [mycli](#) - MySQL CLI with autocompletion and syntax highlighting.
 - [pgcli](#) - PostgreSQL CLI with autocompletion and syntax highlighting.

Computer Vision

Libraries for Computer Vision.

- [easyocr](#) - Ready-to-use OCR with 40+ languages supported.
- [kornia](#) - Open Source Differentiable Computer Vision Library for PyTorch.
- [opencv](#) - Open Source Computer Vision Library.
- [pytesseract](#) - A wrapper for [Google Tesseract OCR](#).
- [tesseractocr](#) - Another simple, Pillow-friendly, wrapper around the `tesseract-ocr` API for OCR.

Configuration Files

Libraries for storing and parsing configuration options.

- [configparser](#) - (Python standard library) INI file parser.
- [configobj](#) - INI file parser with validation.
- [hydra](#) - Hydra is a framework for elegantly configuring complex applications.
- [python-decouple](#) - Strict separation of settings from code.

Cryptography

- [cryptography](#) - A package designed to expose cryptographic primitives and recipes to Python developers.
- [paramiko](#) - The leading native Python SSHv2 protocol library.
- [pynacl](#) - Python binding to the Networking and Cryptography (NaCl) library.

Data Analysis

Libraries for data analyzing.

- [pandas](#) - A library providing high-performance, easy-to-use data structures and data analysis tools.
- [aws-sdk-pandas](#) - Pandas on AWS.
- [datasette](#) - An open source multi-tool for exploring and publishing data.
- [optimus](#) - Agile Data Science Workflows made easy with PySpark.

Data Validation

Libraries for validating data. Used for forms in many cases.

- [cerberus](#) - A lightweight and extensible data validation library.
- [colander](#) - Validating and deserializing data obtained via XML, JSON, an HTML form post.
- [jsonschema](#) - An implementation of [JSON Schema](#) for Python.
- [schema](#) - A library for validating Python data structures.
- [schematics](#) - Data Structure Validation.
- [voluptuous](#) - A Python data validation library.
- [pydantic](#) - Data validation using Python type hints.

Data Visualization

Libraries for visualizing data. Also see [awesome-javascript](#).

- [altair](#) - Declarative statistical visualization library for Python.
- [bokeh](#) - Interactive Web Plotting for Python.
- [bqplot](#) - Interactive Plotting Library for the Jupyter Notebook.
- [cartopy](#) - A cartographic python library with matplotlib support.
- [diagrams](#) - Diagram as Code.
- [matplotlib](#) - A Python 2D plotting library.
- [plotnine](#) - A grammar of graphics for Python based on ggplot2.
- [pygal](#) - A Python SVG Charts Creator.
- [pygraphviz](#) - Python interface to [Graphviz](#).

- [pyqtgraph](#) - Interactive and realtime 2D/3D/Image plotting and science/engineering widgets.
- [seaborn](#) - Statistical data visualization using Matplotlib.
- [vispy](#) - High-performance scientific visualization based on OpenGL.

Database

Databases implemented in Python.

- [pickleDB](#) - A simple and lightweight key-value store for Python.
- [tinydb](#) - A tiny, document-oriented database.
- [zodb](#) - A native object database for Python. A key-value and object graph database.

Database Drivers

Libraries for connecting and operating databases.

- MySQL - [awesome-mysql](#)
 - [mysqlclient](#) - MySQL connector with Python 3 support ([mysql-python](#) fork).
 - [pymysql](#) - A pure Python MySQL driver compatible to mysql-python.
- PostgreSQL - [awesome-postgres](#)
 - [psycopg](#) - The most popular PostgreSQL adapter for Python.
- SQLite - [awesome-sqlite](#)
 - [sqlite3](#) - (Python standard library) SQLite interface compliant with DB-API 2.0.
 - [sqlite-utils](#) - Python CLI utility and library for manipulating SQLite databases.
- Other Relational Databases
 - [pymssql](#) - A simple database interface to Microsoft SQL Server.
 - [clickhouse-driver](#) - Python driver with native interface for ClickHouse.
- NoSQL Databases
 - [cassandra-driver](#) - The Python Driver for Apache Cassandra.
 - [happybase](#) - A developer-friendly library for Apache HBase.
 - [kafka-python](#) - The Python client for Apache Kafka.
 - [pymongo](#) - The official Python client for MongoDB.
 - [motor](#) - The async Python driver for MongoDB.
 - [redis-py](#) - The Python client for Redis.

Date and Time

Libraries for working with dates and times.

- [arrow](#) - A Python library that offers a sensible and human-friendly approach to creating, manipulating, formatting and converting dates, times and timestamps.
- [dateutil](#) - Extensions to the standard Python [datetime](#) module.

- [pendulum](#) - Python datetimes made easy.
- [pytz](#) - World timezone definitions, modern and historical. Brings the [tz database](#) into Python.

Debugging Tools

Libraries for debugging code.

- pdb-like Debugger
 - [ipdb](#) - IPython-enabled [pdb](#).
 - [pudb](#) - A full-screen, console-based Python debugger.
- Tracing
 - [manhole](#) - Debugging UNIX socket connections and present the stacktraces for all threads and an interactive prompt.
 - [python-hunter](#) - A flexible code tracing toolkit.
- Profiler
 - [py-spy](#) - A sampling profiler for Python programs. Written in Rust.
 - [vprof](#) - Visual Python profiler.
- Others
 - [django-debug-toolbar](#) - Display various debug information for Django.
 - [flask-debugtoolbar](#) - A port of the django-debug-toolbar to flask.
 - [icecream](#) - Inspect variables, expressions, and program execution with a single, simple function call.
 - [pyelftools](#) - Parsing and analyzing ELF files and DWARF debugging information.

Deep Learning

Frameworks for Neural Networks and Deep Learning. Also see [awesome-deep-learning](#).

- [keras](#) - A high-level neural networks library and capable of running on top of either TensorFlow or Theano.
- [pytorch](#) - Tensors and Dynamic neural networks in Python with strong GPU acceleration.
- [pytorch-lightning](#) - Deep learning framework to train, deploy, and ship AI products Lightning fast.
- [tensorflow](#) - The most popular Deep Learning framework created by Google.
- [theano](#) - A library for fast numerical computation.

DevOps Tools

Software and libraries for DevOps.

- Configuration Management
 - [ansible](#) - A radically simple IT automation platform.
 - [cloudinit](#) - A multi-distribution package that handles early initialization of a cloud instance.

- [openstack](#) - Open source software for building private and public clouds.
- [pyinfra](#) - A versatile CLI tools and python libraries to automate infrastructure.
- [saltstack](#) - Infrastructure automation and management system.
- SSH-style Deployment
 - [cuisine](#) - Chef-like functionality for Fabric.
 - [fabric](#) - A simple, Pythonic tool for remote execution and deployment.
- Process Management
 - [supervisor](#) - Supervisor process control system for UNIX.
- Monitoring
 - [psutil](#) - A cross-platform process and system utilities module.
- Backup
 - [borg](#) - A deduplicating archiver with compression and encryption.

Distributed Computing

Frameworks and libraries for Distributed Computing.

- Batch Processing
 - [dask](#) - A flexible parallel computing library for analytic computing.
 - [luigi](#) - A module that helps you build complex pipelines of batch jobs.
 - [PySpark](#) - [Apache Spark](#) Python API.
 - [Ray](#) - A system for parallel and distributed Python that unifies the machine learning ecosystem.
- Stream Processing
 - [faust](#) - A stream processing library, porting the ideas from [Kafka Streams](#) to Python.
 - [streamparse](#) - Run Python code against real-time streams of data via [Apache Storm](#).

Distribution

Libraries to create packaged executables for release distribution.

- [py2app](#) - Freezes Python scripts (Mac OS X).
- [py2exe](#) - Freezes Python scripts (Windows).
- [pyarmor](#) - A tool used to obfuscate python scripts, bind obfuscated scripts to fixed machine or expire obfuscated scripts.
- [pyinstaller](#) - Converts Python programs into stand-alone executables (cross-platform).
- [shiv](#) - A command line utility for building fully self-contained zipapps (PEP 441), but with all their dependencies included.

Documentation

Libraries for generating project documentation.

- [sphinx](#) - Python Documentation generator.
 - [awesome-sphinxdoc](#)
- [pdoc](#) - Epydoc replacement to auto generate API documentation for Python libraries.

Downloader

Libraries for downloading.

- [akshare](#) - A financial data interface library, built for human beings!
- [s3cmd](#) - A command line tool for managing Amazon S3 and CloudFront.
- [youtube-dl](#) - A command-line program to download videos from YouTube and other video sites.

Editor Plugins and IDEs

- Emacs
 - [elpy](#) - Emacs Python Development Environment.
- Vim
 - [jedi-vim](#) - Vim bindings for the Jedi auto-completion library for Python.
 - [python-mode](#) - An all in one plugin for turning Vim into a Python IDE.
 - [YouCompleteMe](#) - Includes [Jedi](#)-based completion engine for Python.
- Visual Studio
 - [PTVS](#) - Python Tools for Visual Studio.
- Visual Studio Code
 - [Python](#) - The official VSCode extension with rich support for Python.
- IDE
 - [PyCharm](#) - Commercial Python IDE by JetBrains. Has free community edition available.
 - [spyder](#) - Open Source Python IDE.

Email

Libraries for sending and parsing email.

- Mail Servers
 - [modoboa](#) - A mail hosting and management platform including a modern Web UI.
 - [salmon](#) - A Python Mail Server.
- Clients
 - [imbox](#) - Python IMAP for Humans.
 - [yagmail](#) - Yet another Gmail/SMTP client.
- Others
 - [flanker](#) - An email address and Mime parsing library.
 - [mailer](#) - High-performance extensible mail delivery framework.

Environment Management

Libraries for Python version and virtual environment management.

- [pyenv](#) - Simple Python version management.
- [virtualenv](#) - A tool to create isolated Python environments.

File Manipulation

Libraries for file manipulation.

- [mimetypes](#) - (Python standard library) Map filenames to MIME types.
- [pathlib](#) - (Python standard library) An cross-platform, object-oriented path library.
- [path.py](#) - A module wrapper for [os.path](#).
- [python-magic](#) - A Python interface to the libmagic file type identification library.
- [watchdog](#) - API and shell utilities to monitor file system events.

Functional Programming

Functional Programming with Python.

- [coconut](#) - A variant of Python built for simple, elegant, Pythonic functional programming.
- [fancy](#) - A fancy and practical functional tools.
- [more-itertools](#) - More routines for operating on iterables, beyond `itertools`.
- [returns](#) - A set of type-safe monads, transformers, and composition utilities.
- [cytoolz](#) - Cython implementation of `Toolz`: High performance functional utilities.
- [toolz](#) - A collection of functional utilities for iterators, functions, and dictionaries.

GUI Development

Libraries for working with graphical user interface applications.

- [curses](#) - Built-in wrapper for [ncurses](#) used to create terminal GUI applications.
- [Eel](#) - A library for making simple Electron-like offline HTML/JS GUI apps.
- [enaml](#) - Creating beautiful user-interfaces with Declarative Syntax like QML.
- [Flexx](#) - Flexx is a pure Python toolkit for creating GUI's, that uses web technology for its rendering.
- [Gooeey](#) - Turn command line programs into a full GUI application with one line.
- [kivy](#) - A library for creating NUI applications, running on Windows, Linux, Mac OS X, Android and iOS.
- [pyglet](#) - A cross-platform windowing and multimedia library for Python.
- [PyGObject](#) - Python Bindings for GLib/GObject/GIO/GTK+ (GTK+3).
- [PyQt](#) - Python bindings for the [Qt](#) cross-platform application and UI framework.

- [PySimpleGUI](#) - Wrapper for tkinter, Qt, WxPython and Remi.
- [pywebview](#) - A lightweight cross-platform native wrapper around a webview component.
- [Tkinter](#) - Tkinter is Python's de-facto standard GUI package.
- [Toga](#) - A Python native, OS native GUI toolkit.
- [urwid](#) - A library for creating terminal GUI applications with strong support for widgets, events, rich colors, etc.
- [wxPython](#) - A blending of the wxWidgets C++ class library with the Python.
- [DearPyGui](#) - A Simple GPU accelerated Python GUI framework

GraphQL

Libraries for working with GraphQL.

- [graphene](#) - GraphQL framework for Python.

Game Development

Awesome game development libraries.

- [Arcade](#) - Arcade is a modern Python framework for crafting games with compelling graphics and sound.
- [Cocos2d](#) - cocos2d is a framework for building 2D games, demos, and other graphical/interactive applications.
- [Harfang3D](#) - Python framework for 3D, VR and game development.
- [Panda3D](#) - 3D game engine developed by Disney.
- [Pygame](#) - Pygame is a set of Python modules designed for writing games.
- [PyOgre](#) - Python bindings for the Ogre 3D render engine, can be used for games, simulations, anything 3D.
- [PyOpenGL](#) - Python ctypes bindings for OpenGL and it's related APIs.
- [PySDL2](#) - A ctypes based wrapper for the SDL2 library.
- [RenPy](#) - A Visual Novel engine.

Geolocation

Libraries for geocoding addresses and working with latitudes and longitudes.

- [django-countries](#) - A Django app that provides a country field for models and forms.
- [geodjango](#) - A world-class geographic web framework.
- [geojson](#) - Python bindings and utilities for GeoJSON.
- [geopy](#) - Python Geocoding Toolbox.

HTML Manipulation

Libraries for working with HTML and XML.

- [beautifulsoup](#) - Providing Pythonic idioms for iterating, searching, and modifying HTML or XML.
- [bleach](#) - A whitelist-based HTML sanitization and text linkification library.
- [cssutils](#) - A CSS library for Python.
- [html5lib](#) - A standards-compliant library for parsing and serializing HTML documents and fragments.
- [lxml](#) - A very fast, easy-to-use and versatile library for handling HTML and XML.
- [markupsafe](#) - Implements a XML/HTML/XHTML Markup safe string for Python.
- [pyquery](#) - A jQuery-like library for parsing HTML.
- [untangle](#) - Converts XML documents to Python objects for easy access.
- [WeasyPrint](#) - A visual rendering engine for HTML and CSS that can export to PDF.
- [xmldataset](#) - Simple XML Parsing.
- [xmldict](#) - Working with XML feel like you are working with JSON.

HTTP Clients

Libraries for working with HTTP.

- [httpx](#) - A next generation HTTP client for Python.
- [requests](#) - HTTP Requests for Humans.
- [treq](#) - Python requests like API built on top of Twisted's HTTP client.
- [urllib3](#) - A HTTP library with thread-safe connection pooling, file post support, sanity friendly.

Hardware

Libraries for programming with hardware.

- [keyboard](#) - Hook and simulate global keyboard events on Windows and Linux.
- [mouse](#) - Hook and simulate global mouse events on Windows and Linux.
- [pynput](#) - A library to control and monitor input devices.
- [scapy](#) - A brilliant packet manipulation library.

Image Processing

Libraries for manipulating images.

- [pillow](#) - Pillow is the friendly [PIL](#) fork.
- [python-barcode](#) - Create barcodes in Python with no extra dependencies.
- [pymatting](#) - A library for alpha matting.
- [python-qrcode](#) - A pure Python QR Code generator.
- [pywal](#) - A tool that generates color schemes from images.

- [pyvips](#) - A fast image processing library with low memory needs.
- [quads](#) - Computer art based on quadtrees.
- [scikit-image](#) - A Python library for (scientific) image processing.
- [thumbor](#) - A smart imaging service. It enables on-demand crop, re-sizing and flipping of images.
- [wand](#) - Python bindings for [MagickWand](#), C API for ImageMagick.

Implementations

Implementations of Python.

- [cpython](#) - Default, most widely used implementation of the Python programming language written in C.
- [cython](#) - Optimizing Static Compiler for Python.
- [clpython](#) - Implementation of the Python programming language written in Common Lisp.
- [ironpython](#) - Implementation of the Python programming language written in C#.
- [micropython](#) - A lean and efficient Python programming language implementation.
- [numba](#) - Python JIT compiler to LLVM aimed at scientific Python.
- [peachpy](#) - x86-64 assembler embedded in Python.
- [pypy](#) - A very fast and compliant implementation of the Python language.
- [pyston](#) - A Python implementation using JIT techniques.

Interactive Interpreter

Interactive Python interpreters (REPL).

- [bpython](#) - A fancy interface to the Python interpreter.
- [Jupyter Notebook \(IPython\)](#) - A rich toolkit to help you make the most out of using Python interactively.
 - [awesome-jupyter](#)
- [ptpython](#) - Advanced Python REPL built on top of the [python-prompt-toolkit](#).

Internationalization

Libraries for working with i18n.

- [Babel](#) - An internationalization library for Python.
- [PyICU](#) - A wrapper of International Components for Unicode C++ library ([ICU](#)).

Job Scheduler

Libraries for scheduling jobs.

- [Airflow](#) - Airflow is a platform to programmatically author, schedule and monitor workflows.

- [APScheduler](#) - A light but powerful in-process task scheduler that lets you schedule functions.
- [django-schedule](#) - A calendaring app for Django.
- [doit](#) - A task runner and build tool.
- [gunnery](#) - Multipurpose task execution tool for distributed systems with web-based interface.
- [Joblib](#) - A set of tools to provide lightweight pipelining in Python.
- [Plan](#) - Writing crontab file in Python like a charm.
- [Prefect](#) - A modern workflow orchestration framework that makes it easy to build, schedule and monitor robust data pipelines.
- [schedule](#) - Python job scheduling for humans.
- [Spiff](#) - A powerful workflow engine implemented in pure Python.
- [TaskFlow](#) - A Python library that helps to make task execution easy, consistent and reliable.

Logging

Libraries for generating and working with logs.

- [logbook](#) - Logging replacement for Python.
- [logging](#) - (Python standard library) Logging facility for Python.
- [loguru](#) - Library which aims to bring enjoyable logging in Python.
- [sentry-python](#) - Sentry SDK for Python.
- [structlog](#) - Structured logging made easy.

Machine Learning

Libraries for Machine Learning. Also see [awesome-machine-learning](#).

- [gym](#) - A toolkit for developing and comparing reinforcement learning algorithms.
- [H2O](#) - Open Source Fast Scalable Machine Learning Platform.
- [Metrics](#) - Machine learning evaluation metrics.
- [NuPIC](#) - Numenta Platform for Intelligent Computing.
- [scikit-learn](#) - The most popular Python library for Machine Learning.
- [Spark ML](#) - [Apache Spark](#)'s scalable Machine Learning library.
- [vowpal_porpoise](#) - A lightweight Python wrapper for [Vowpal Wabbit](#).
- [xgboost](#) - A scalable, portable, and distributed gradient boosting library.
- [MindsDB](#) - MindsDB is an open source AI layer for existing databases that allows you to effortlessly develop, train and deploy state-of-the-art machine learning models using standard queries.

Microsoft Windows

Python programming on Microsoft Windows.

- [Python\(x,y\)](#) - Scientific-applications-oriented Python Distribution based on Qt and Spyder.

- [pythonlibs](#) - Unofficial Windows binaries for Python extension packages.
- [PythonNet](#) - Python Integration with the .NET Common Language Runtime (CLR).
- [PyWin32](#) - Python Extensions for Windows.
- [WinPython](#) - Portable development environment for Windows 7/8.

Miscellaneous

Useful libraries or tools that don't fit in the categories above.

- [blinker](#) - A fast Python in-process signal/event dispatching system.
- [boltons](#) - A set of pure-Python utilities.
- [itsdangerous](#) - Various helpers to pass trusted data to untrusted environments.
- [magenta](#) - A tool to generate music and art using artificial intelligence.
- [pluginbase](#) - A simple but flexible plugin system for Python.
- [tryton](#) - A general purpose business framework.

Natural Language Processing

Libraries for working with human languages.

- General
 - [gensim](#) - Topic Modeling for Humans.
 - [langid.py](#) - Stand-alone language identification system.
 - [nltk](#) - A leading platform for building Python programs to work with human language data.
 - [pattern](#) - A web mining module.
 - [polyglot](#) - Natural language pipeline supporting hundreds of languages.
 - [pytext](#) - A natural language modeling framework based on PyTorch.
 - [PyTorch-NLP](#) - A toolkit enabling rapid deep learning NLP prototyping for research.
 - [spacy](#) - A library for industrial-strength natural language processing in Python and Cython.
 - [Stanza](#) - The Stanford NLP Group's official Python library, supporting 60+ languages.
- Chinese
 - [funNLP](#) - A collection of tools and datasets for Chinese NLP.
 - [jieba](#) - The most popular Chinese text segmentation library.
 - [pkuseg-python](#) - A toolkit for Chinese word segmentation in various domains.
 - [snownlp](#) - A library for processing Chinese text.

Network Virtualization

Tools and libraries for Virtual Networking and SDN (Software Defined Networking).

- [mininet](#) - A popular network emulator and API written in Python.

- [napalm](#) - Cross-vendor API to manipulate network devices.
- [pox](#) - A Python-based SDN control applications, such as OpenFlow SDN controllers.

News Feed

Libraries for building user's activities.

- [django-activity-stream](#) - Generating generic activity streams from the actions on your site.
- [Stream Framework](#) - Building news feed and notification systems using Cassandra and Redis.

ORM

Libraries that implement Object-Relational Mapping or data mapping techniques.

- Relational Databases
 - [Django Models](#) - The Django ORM.
 - [SQLAlchemy](#) - The Python SQL Toolkit and Object Relational Mapper.
 - [awesome-sqlalchemy](#)
 - [dataset](#) - Store Python dicts in a database - works with SQLite, MySQL, and PostgreSQL.
 - [orator](#) - The Orator ORM provides a simple yet beautiful ActiveRecord implementation.
 - [orm](#) - An async ORM.
 - [peewee](#) - A small, expressive ORM.
 - [pony](#) - ORM that provides a generator-oriented interface to SQL.
 - [pydal](#) - A pure Python Database Abstraction Layer.
- NoSQL Databases
 - [hot-redis](#) - Rich Python data types for Redis.
 - [mongoengine](#) - A Python Object-Document-Mapper for working with MongoDB.
 - [PynamoDB](#) - A Pythonic interface for [Amazon DynamoDB](#).
 - [redisco](#) - A Python Library for Simple Models and Containers Persisted in Redis.

Package Management

Libraries for package and dependency management.

- [pip](#) - The package installer for Python.
 - [pip-tools](#) - A set of tools to keep your pinned Python dependencies fresh.
 - [PyPI](#)
- [conda](#) - Cross-platform, Python-agnostic binary package manager.
- [poetry](#) - Python dependency management and packaging made easy.

Package Repositories

Local PyPI repository server and proxies.

- [bandersnatch](#) - PyPI mirroring tool provided by Python Packaging Authority (PyPA).
- [devpi](#) - PyPI server and packaging/testing/release tool.
- [localshop](#) - Local PyPI server (custom packages and auto-mirroring of pypi).
- [warehouse](#) - Next generation Python Package Repository (PyPI).

Penetration Testing

Frameworks and tools for penetration testing.

- [fsociety](#) - A Penetration testing framework.
- [setoolkit](#) - A toolkit for social engineering.
- [sqlmap](#) - Automatic SQL injection and database takeover tool.

Permissions

Libraries that allow or deny users access to data or functionality.

- [django-guardian](#) - Implementation of per object permissions for Django 1.2+
- [django-rules](#) - A tiny but powerful app providing object-level permissions to Django, without requiring a database.

Processes

Libraries for starting and communicating with OS processes.

- [delegator.py](#) - [Subprocesses](#) for Humans 2.0.
- [sarge](#) - Yet another wrapper for subprocess.
- [sh](#) - A full-fledged subprocess replacement for Python.

Recommender Systems

Libraries for building recommender systems.

- [annoy](#) - Approximate Nearest Neighbors in C++/Python optimized for memory usage.
- [fastFM](#) - A library for Factorization Machines.
- [implicit](#) - A fast Python implementation of collaborative filtering for implicit datasets.
- [libffm](#) - A library for Field-aware Factorization Machine (FFM).
- [lightfm](#) - A Python implementation of a number of popular recommendation algorithms.
- [spotlight](#) - Deep recommender models using PyTorch.
- [Surprise](#) - A scikit for building and analyzing recommender systems.
- [tensorrec](#) - A Recommendation Engine Framework in TensorFlow.

Refactoring

Refactoring tools and libraries for Python

- [Bicycle Repair Man](#) - Bicycle Repair Man, a refactoring tool for Python.
- [Bowler](#) - Safe code refactoring for modern Python.
- [Rope](#) - Rope is a python refactoring library.

RESTful API

Libraries for building RESTful APIs.

- Django
 - [django-rest-framework](#) - A powerful and flexible toolkit to build web APIs.
 - [django-tastypie](#) - Creating delicious APIs for Django apps.
- Flask
 - [eve](#) - REST API framework powered by Flask, MongoDB and good intentions.
 - [flask-api](#) - Browsable Web APIs for Flask.
 - [flask-restful](#) - Quickly building REST APIs for Flask.
- Pyramid
 - [cornice](#) - A RESTful framework for Pyramid.
- Framework agnostic
 - [falcon](#) - A high-performance framework for building cloud APIs and web app backends.
 - [fastapi](#) - A modern, fast, web framework for building APIs with Python 3.6+ based on standard Python type hints.
 - [hug](#) - A Python 3 framework for cleanly exposing APIs.
 - [sandman2](#) - Automated REST APIs for existing database-driven systems.
 - [sanic](#) - A Python 3.6+ web server and web framework that's written to go fast.

Robotics

Libraries for robotics.

- [PythonRobotics](#) - This is a compilation of various robotics algorithms with visualizations.
- [rospy](#) - This is a library for ROS (Robot Operating System).

RPC Servers

RPC-compatible servers.

- [RPyC](#) (Remote Python Call) - A transparent and symmetric RPC library for Python
- [zeroRPC](#) - zerorpc is a flexible RPC implementation based on [ZeroMQ](#) and [MessagePack](#).

Science

Libraries for scientific computing. Also see [Python-for-Scientists](#).

- [astropy](#) - A community Python library for Astronomy.
- [bcbio-nextgen](#) - Providing best-practice pipelines for fully automated high throughput sequencing analysis.
- [bccb](#) - Collection of useful code related to biological analysis.
- [Biopython](#) - Biopython is a set of freely available tools for biological computation.
- [cclib](#) - A library for parsing and interpreting the results of computational chemistry packages.
- [Colour](#) - Implementing a comprehensive number of colour theory transformations and algorithms.
- [Karate Club](#) - Unsupervised machine learning toolbox for graph structured data.
- [NetworkX](#) - A high-productivity software for complex networks.
- [NIPY](#) - A collection of neuroimaging toolkits.
- [NumPy](#) - A fundamental package for scientific computing with Python.
- [ObsPy](#) - A Python toolbox for seismology.
- [Open Babel](#) - A chemical toolbox designed to speak the many languages of chemical data.
- [PyDy](#) - Short for Python Dynamics, used to assist with workflow in the modeling of dynamic motion.
- [PyMC](#) - Markov Chain Monte Carlo sampling toolkit.
- [QuTiP](#) - Quantum Toolbox in Python.
- [RDKit](#) - Cheminformatics and Machine Learning Software.
- [SciPy](#) - A Python-based ecosystem of open-source software for mathematics, science, and engineering.
- [SimPy](#) - A process-based discrete-event simulation framework.
- [statsmodels](#) - Statistical modeling and econometrics in Python.
- [SymPy](#) - A Python library for symbolic mathematics.
- [Zipline](#) - A Pythonic algorithmic trading library.

Search

Libraries and software for indexing and performing search queries on data.

- [django-haystack](#) - Modular search for Django.
- [elasticsearch-dsl-py](#) - The official high-level Python client for Elasticsearch.
- [elasticsearch-py](#) - The official low-level Python client for [Elasticsearch](#).
- [pysolr](#) - A lightweight Python wrapper for [Apache Solr](#).
- [whoosh](#) - A fast, pure Python search engine library.

Serialization

Libraries for serializing complex data types

- [marshmallow](#) - A lightweight library for converting complex objects to and from simple Python datatypes.
- [pysimdjson](#) - A Python bindings for [simdjson](#).
- [python-rapidjson](#) - A Python wrapper around [RapidJSON](#).
- [ultrajson](#) - A fast JSON decoder and encoder written in C with Python bindings.

Serverless Frameworks

Frameworks for developing serverless Python code.

- [python-lambda](#) - A toolkit for developing and deploying Python code in AWS Lambda.
- [Zappa](#) - A tool for deploying WSGI applications on AWS Lambda and API Gateway.

Shell

Shells based on Python.

- [xonsh](#) - A Python-powered, cross-platform, Unix-gazing shell language and command prompt.

Specific Formats Processing

Libraries for parsing and manipulating specific text formats.

- General
 - [tablib](#) - A module for Tabular Datasets in XLS, CSV, JSON, YAML.
- Office
 - [docxtpl](#) - Editing a docx document by jinja2 template
 - [openpyxl](#) - A library for reading and writing Excel 2010 xlsx/xlsm/xltx/xltn files.
 - [pyexcel](#) - Providing one API for reading, manipulating and writing csv, ods, xls, xlsx and xlsm files.
 - [python-docx](#) - Reads, queries and modifies Microsoft Word 2007/2008 docx files.
 - [python-pptx](#) - Python library for creating and updating PowerPoint (.pptx) files.
 - [unoconv](#) - Convert between any document format supported by LibreOffice/OpenOffice.
 - [XlsxWriter](#) - A Python module for creating Excel .xlsx files.
 - [xlwings](#) - A BSD-licensed library that makes it easy to call Python from Excel and vice versa.
 - [xlwt](#) / [xlrd](#) - Writing and reading data and formatting information from Excel files.
- PDF
 - [PDFMiner](#) - A tool for extracting information from PDF documents.
 - [PyPDF2](#) - A library capable of splitting, merging and transforming PDF pages.
 - [ReportLab](#) - Allowing Rapid creation of rich PDF documents.
- Markdown

- [Mistune](#) - Fastest and full featured pure Python parsers of Markdown.
- [Python-Markdown](#) - A Python implementation of John Gruber's Markdown.
- YAML
 - [PyYAML](#) - YAML implementations for Python.
- CSV
 - [csvkit](#) - Utilities for converting to and working with CSV.
- Archive
 - [unp](#) - A command line tool that can unpack archives easily.

Static Site Generator

Static site generator is a software that takes some text + templates as input and produces HTML files on the output.

- [lektor](#) - An easy to use static CMS and blog engine.
- [mkdocs](#) - Markdown friendly documentation generator.
- [makesite](#) - Simple, lightweight, and magic-free static site/blog generator (< 130 lines).
- [nikola](#) - A static website and blog generator.
- [pelican](#) - Static site generator that supports Markdown and reST syntax.

Tagging

Libraries for tagging items.

- [django-taggit](#) - Simple tagging for Django.

Task Queues

Libraries for working with task queues.

- [celery](#) - An asynchronous task queue/job queue based on distributed message passing.
- [dramatiq](#) - A fast and reliable background task processing library for Python 3.
- [huey](#) - Little multi-threaded task queue.
- [mrq](#) - A distributed worker task queue in Python using Redis & gevent.
- [rq](#) - Simple job queues for Python.

Template Engine

Libraries and tools for templating and lexing.

- [Genshi](#) - Python templating toolkit for generation of web-aware output.
- [Jinja2](#) - A modern and designer friendly templating language.
- [Mako](#) - Hyperfast and lightweight templating for the Python platform.

Testing

Libraries for testing codebases and generating test data.

- Testing Frameworks
 - [hypothesis](#) - Hypothesis is an advanced Quickcheck style property based testing library.
 - [nose2](#) - The successor to `nose`, based on `unittest2`.
 - [pytest](#) - A mature full-featured Python testing tool.
 - [Robot Framework](#) - A generic test automation framework.
 - [unittest](#) - (Python standard library) Unit testing framework.
- Test Runners
 - [green](#) - A clean, colorful test runner.
 - [mamba](#) - The definitive testing tool for Python. Born under the banner of BDD.
 - [tox](#) - Auto builds and tests distributions in multiple Python versions
- GUI / Web Testing
 - [locust](#) - Scalable user load testing tool written in Python.
 - [PyAutoGUI](#) - PyAutoGUI is a cross-platform GUI automation Python module for human beings.
 - [Schemathesis](#) - A tool for automatic property-based testing of web applications built with Open API / Swagger specifications.
 - [Selenium](#) - Python bindings for [Selenium](#) WebDriver.
 - [sixpack](#) - A language-agnostic A/B Testing framework.
 - [splinter](#) - Open source tool for testing web applications.
- Mock
 - [doublex](#) - Powerful test doubles framework for Python.
 - [freezegun](#) - Travel through time by mocking the datetime module.
 - [httmock](#) - A mocking library for requests for Python 2.6+ and 3.2+.
 - [httpretty](#) - HTTP request mock tool for Python.
 - [mock](#) - (Python standard library) A mocking and patching library.
 - [mocket](#) - A socket mock framework with gevent/asyncio/SSL support.
 - [responses](#) - A utility library for mocking out the requests Python library.
 - [VCR.py](#) - Record and replay HTTP interactions on your tests.
- Object Factories
 - [factory_boy](#) - A test fixtures replacement for Python.
 - [mixer](#) - Another fixtures replacement. Supports Django, Flask, SQLAlchemy, Peewee and etc.
 - [model_mommy](#) - Creating random fixtures for testing in Django.
- Code Coverage
 - [coverage](#) - Code coverage measurement.
- Fake Data
 - [fake2db](#) - Fake database generator.
 - [faker](#) - A Python package that generates fake data.

- [mimesis](#) - is a Python library that help you generate fake data.
- [radar](#) - Generate random datetime / time.

Text Processing

Libraries for parsing and manipulating plain texts.

- General
 - [chardet](#) - Python 2/3 compatible character encoding detector.
 - [difflib](#) - (Python standard library) Helpers for computing deltas.
 - [ftfy](#) - Makes Unicode text less broken and more consistent automatically.
 - [fuzzywuzzy](#) - Fuzzy String Matching.
 - [Levenshtein](#) - Fast computation of Levenshtein distance and string similarity.
 - [pangu.py](#) - Paranoid text spacing.
 - [pyfiglet](#) - An implementation of figlet written in Python.
 - [pypinyin](#) - Convert Chinese hanzi (漢字) to pinyin (拼音).
 - [textdistance](#) - Compute distance between sequences with 30+ algorithms.
 - [unidecode](#) - ASCII transliterations of Unicode text.
- Slugify
 - [awesome-slugify](#) - A Python slugify library that can preserve unicode.
 - [python-slugify](#) - A Python slugify library that translates unicode to ASCII.
 - [unicode-slugify](#) - A slugifier that generates unicode slugs with Django as a dependency.
- Unique identifiers
 - [hashids](#) - Implementation of [hashids](#) in Python.
 - [shortuuid](#) - A generator library for concise, unambiguous and URL-safe UUIDs.
- Parser
 - [ply](#) - Implementation of lex and yacc parsing tools for Python.
 - [pygments](#) - A generic syntax highlighter.
 - [pyparsing](#) - A general purpose framework for generating parsers.
 - [python-nameparser](#) - Parsing human names into their individual components.
 - [python-phonenumbers](#) - Parsing, formatting, storing and validating international phone numbers.
 - [python-user-agents](#) - Browser user agent parser.
 - [sqlparse](#) - A non-validating SQL parser.

Third-party APIs

Libraries for accessing third party services APIs. Also see [List of Python API Wrappers and Libraries](#).

- [apache-libcloud](#) - One Python library for all clouds.
- [boto3](#) - Python interface to Amazon Web Services.

- [django-wordpress](#) - WordPress models and views for Django.
- [facebook-sdk](#) - Facebook Platform Python SDK.
- [google-api-python-client](#) - Google APIs Client Library for Python.
- [gspread](#) - Google Spreadsheets Python API.
- [twython](#) - A Python wrapper for the Twitter API.

URL Manipulation

Libraries for parsing URLs.

- [furl](#) - A small Python library that makes parsing and manipulating URLs easy.
- [purl](#) - A simple, immutable URL class with a clean API for interrogation and manipulation.
- [pyshorteners](#) - A pure Python URL shortening lib.
- [webargs](#) - A friendly library for parsing HTTP request arguments with built-in support for popular web frameworks.

Video

Libraries for manipulating video and GIFs.

- [moviepy](#) - A module for script-based movie editing with many formats, including animated GIFs.
- [scikit-video](#) - Video processing routines for SciPy.
- [vidgear](#) - Most Powerful multi-threaded Video Processing framework.

Web Asset Management

Tools for managing, compressing and minifying website assets.

- [django-compressor](#) - Compresses linked and inline JavaScript or CSS into a single cached file.
- [django-pipeline](#) - An asset packaging library for Django.
- [django-storages](#) - A collection of custom storage back ends for Django.
- [fanstatic](#) - Packages, optimizes, and serves static file dependencies as Python packages.
- [fileconveyor](#) - A daemon to detect and sync files to CDNs, S3 and FTP.
- [flask-assets](#) - Helps you integrate webassets into your Flask app.
- [webassets](#) - Bundles, optimizes, and manages unique cache-busting URLs for static resources.

Web Content Extracting

Libraries for extracting web contents.

- [html2text](#) - Convert HTML to Markdown-formatted text.
- [lassie](#) - Web Content Retrieval for Humans.

- [micawber](#) - A small library for extracting rich content from URLs.
- [newspaper](#) - News extraction, article extraction and content curation in Python.
- [python-readability](#) - Fast Python port of arc90's readability tool.
- [requests-html](#) - Pythonic HTML Parsing for Humans.
- [sumy](#) - A module for automatic summarization of text documents and HTML pages.
- [texttract](#) - Extract text from any document, Word, PowerPoint, PDFs, etc.
- [toapi](#) - Every web site provides APIs.

Web Crawling

Libraries to automate web scraping.

- [feedparser](#) - Universal feed parser.
- [grab](#) - Site scraping framework.
- [mechanicalsoup](#) - A Python library for automating interaction with websites.
- [scrapy](#) - A fast high-level screen scraping and web crawling framework.

Web Frameworks

Traditional full stack web frameworks. Also see [RESTful API](#).

- Synchronous
 - [django](#) - The most popular web framework in Python.
 - [awesome-django](#)
 - [awesome-django](#)
 - [flask](#) - A microframework for Python.
 - [awesome-flask](#)
 - [pyramid](#) - A small, fast, down-to-earth, open source Python web framework.
 - [awesome-pyramid](#)
 - [masonite](#) - The modern and developer centric Python web framework.
- Asynchronous
 - [tornado](#) - A web framework and asynchronous networking library.

WebSocket

Libraries for working with WebSocket.

- [autobahn-python](#) - WebSocket & WAMP for Python on Twisted and [asyncio](#).
- [channels](#) - Developer-friendly asynchrony for Django.
- [websockets](#) - A library for building WebSocket servers and clients with a focus on correctness and simplicity.

WSGI Servers

WSGI-compatible web servers.

- [gunicorn](#) - Pre-forked, ported from Ruby's Unicorn project.
- [uwsgi](#) - A project aims at developing a full stack for building hosting services, written in C.
- [waitress](#) - Multi-threaded, powers Pyramid.
- [werkzeug](#) - A WSGI utility library for Python that powers Flask and can easily be embedded into your own projects.

Resources

Where to discover learning resources or new Python libraries.

Newsletters

- [Awesome Python Newsletter](#)
- [Pycoder's Weekly](#)
- [Python Tricks](#)
- [Python Weekly](#)

Podcasts

- [Django Chat](#)
- [Python Bytes](#)
- [Talk Python To Me](#)
- [Python Test](#)
- [The Real Python Podcast](#)

Contributing

Your contributions are always welcome! Please take a look at the [contribution guidelines](#) first.

If you have any question about this opinionated list, do not hesitate to contact me [@VintaChen](#) on Twitter or open an issue on GitHub.