

Learner's Library – Abhilash Shukla Blog

PROGRAMMING

50 PYTHON LIBRARIES AND ADD-ONS PROGRAMMERS CANNOT LIVE WITHOUT

OCTOBER 8, 2013 | ABHILASH SHUKLA

There are plenty of Python libraries who really counts in projects and dramatically help-out programmers to reach a standard maturity level.

The list below is with my over all experience with python. I fond these 50 libraries to be basic for a programmer to use and build efficient applications. Many of these are now industry standards, and others are some nice geek stuffs. Just explore all of these and I am sure if you have these on your tips than you are among the best developers out there. And yeah you are fabulously costly priced 😊

NOTE: *My intention with this blog is to give you an idea on libraries that I personally use or have used in one or many projects, and found them to be awesome and rapidly easy to build and deploy applications. You may can use alternatives as your preference and there are many great too. So just consider this list to be a general composition of a python developer. I have used them for years now.*

1. **Requests** – Requests allow you to send HTTP/1.1 requests. You can add headers, form data, multipart files, and parameters with simple Python dictionaries, and access the response data in the same way. It's powered by httplib and urllib3, but it does all the hard work and crazy hacks for you.
2. **Scrapy** – Scrapy is a fast high-level screen scraping and web crawling framework, used to crawl websites and extract structured data from their pages. It can be used for a wide range of purposes, from data mining to monitoring and automated testing.
3. **Pillow** – Pillow is the 'friendly' fork of Python Imaging Library (PIL), which adds image processing capabilities to your Python interpreter. This library supports many file formats, and provides powerful image processing and graphics capabilities.
4. **Twisted** – Twisted implements a variety of networking and communication protocols and exposes them all as method-calls on your Python objects.
5. **Numpy** – NumPy is a general-purpose array-processing package designed to efficiently manipulate large multi-dimensional arrays of arbitrary records without sacrificing too much speed for small multi-dimensional arrays.
6. **SciPy** – SciPy is an open-source software for mathematics, science, and engineering. The SciPy library depends on NumPy, which provides convenient and fast N-dimensional array

manipulation.

7. **Matplotlib** – matplotlib strives to produce publication quality 2D graphics for interactive graphing, scientific publishing, user interface development and web application servers targeting multiple user interfaces and hardcopy output formats. There is a 'pylab' mode which emulates matlab graphics.
8. **Nose** – Nose is basically an extension to unittest in-built library of Python to make testing more easier.
9. **SymPy** – SymPy is a Python library for symbolic mathematics. It aims to become a full-featured computer algebra system (CAS) while keeping the code as simple as possible in order to be comprehensible and easily extensible. SymPy is written entirely in Python and does not require any external libraries.
10. **IPython** – IPython provides a rich toolkit to help you make the most out of using Python interactively, like providing Powerful interactive Python shells, high-performance library and many more.
11. **pyquery** – pyquery allows you to make jquery queries on xml documents. The API is as much as possible the similar to jquery. pyquery uses lxml for fast xml and html manipulation.
12. **dateutil** – The dateutil module provides powerful extensions to the datetime module available in the Python standard library.
13. **py-amqp** – Python client for the Advanced Message Queuing Protocol (AMQP), using for RabbitMQ.
14. **path.py** – It implements a path objects as first-class entities, allowing common operations on files to be invoked on those path objects directly.
15. **libgmail** – The 'libgmail' project is a pure Python binding to provide access to Google's Gmail web-mail service.
16. **SetupTools** – Setuptools is a fully-featured, actively-maintained, and stable library designed to facilitate packaging Python projects and helps you to easily download, build, install, upgrade, and uninstall Python packages
17. **VirtualEnv** – It is a tool to create isolated Python environments.
18. **pip** – A tool for installing and managing Python packages.
19. **boto** – Boto is a Python package that provides interfaces to Amazon Web Services, like EC2, Dynamo and all.
20. **lxml** – The lxml XML toolkit is a Pythonic binding for the C libraries libxml2 and libxslt. It is unique in that it combines the speed and XML feature completeness of these libraries with the simplicity of a native Python API, mostly compatible but superior to the well-known ElementTree API.
21. **Coverage** – Coverage.py measures code coverage, typically during test execution. It uses the code analysis tools and tracing hooks provided in the Python standard library to determine which lines are executable, and which have been executed.
22. **pycrypto** – This is a collection of both secure hash functions (such as SHA256 and RIPEMD160), and various encryption algorithms (AES, DES, RSA, ElGamal, etc.). The package is structured to make adding new modules easy.
23. **Celery** – Celery is a Distributed Task Queue system. Celery communicates via messages, usually using a broker to mediate between clients and workers. To initiate a task a client puts a message on the queue, the broker then delivers the message to a worker.

24. **Pygments** – Pygments is a syntax highlighting package written in Python. It is a generic syntax highlighter for general use in all kinds of software such as forum systems, wikis or other applications that need to prettify source code.
25. **Selenium** – The selenium package is used automate web browser interaction from Python. I am sharing the link for Python language bindings for Selenium WebDriver.
26. **Cython** – The Cython language makes writing C extensions for the Python language as easy as Python itself. Cython is a source code translator based on the well-known Pyrex, but supports more cutting edge functionality and optimizations.
27. **OpenCV** – OpenCV was designed for computational efficiency and with a strong focus on real-time applications. Written in optimized C/C++, the library can take advantage of multi-core processing. Enabled with OpenCL, it can take advantage of the hardware acceleration of the underlying heterogeneous compute platform.
28. **Sphinx** – Sphinx is a tool that makes it easy to create intelligent and beautiful documentation for Python projects (or other documents consisting of multiple reStructuredText sources)
29. **Markdown** – This is a Python implementation of John Gruber's Markdown. Markdown is a plain text formatting syntax designed so that it optionally can be converted to HTML using a tool by the same name. Markdown is popularly used to format readme files, for writing messages in online discussion forums or in text editors for the quick creation of rich text documents.
30. **Memcached** – This software is a 100% Python interface to the memcached memory cache daemon. Free & open source, high-performance, distributed memory object caching system, generic in nature, but intended for use in speeding up dynamic web applications by alleviating database load.
31. **Plone** – Usually people know WordPress, Plone is same a free and open source content management system built on top of the Zope application server. In principle, Plone can be used for any kind of website, including blogs, internet sites, webshops and internal websites.
32. **Feedparser** – Parse Atom and RSS feeds in Python. Universal feed parser, handles RSS 0.9x, RSS 1.0, RSS 2.0, CDF, Atom 0.3, and Atom 1.0 feeds.
33. **oAuthLib** – OAuthLib is a generic utility which implements the logic of OAuth without assuming a specific HTTP request object or web framework. Use it to graft OAuth client support onto your favorite HTTP library, or provider support onto your favourite web framework. If you're a maintainer of such a library, write a thin veneer on top of OAuthLib and get OAuth support for very little effort.
34. **bson** – BSON, short for Binary JSON, is a binary-encoded serialization of JSON-like documents. Like JSON, BSON supports the embedding of documents and arrays within other documents and arrays. BSON also contains extensions that allow representation of data types that are not part of the JSON spec.
35. **sh** – A a full-fledged subprocess interface for Python that allows you to call any program as if it were a function.
36. **Feedly** – Feedly allows you to build complex feed and caching structures using Redis.
37. **PDFMiner** – PDFMiner is a tool for extracting information from PDF documents. Unlike other PDF-related tools, it focuses entirely on getting and analyzing text data. PDFMiner allows to obtain the exact location of texts in a page, as well as other information such as fonts or lines. It includes a PDF converter that can transform PDF files into other text formats (such as HTML).

38. **Tweepy** – An easy-to-use Python library for accessing the Twitter API.
39. **Read the Docs** – Life-line of python packages documentation. ReadTheDocs.org documentation for the open source community. It supports Sphinx docs written with reStructuredText, and can pull from your Subversion, Bazaar, Git, and Mercurial repositories.
40. **pyWin32** – It provides various extensions for interacting with windows and to perform different operations.
41. **Sentry** – Sentry is a realtime event logging and aggregation platform. It specializes in monitoring errors and extracting all the information needed to do a proper post-mortem without any of the hassle of the standard user feedback loop.
42. **Raven** – Raven is a standalone (and the official) Python client for Sentry.
43. **Nltk** – NLTK is a leading platform for building Python programs to work with human language data. It provides easy-to-use interfaces to over 50 corpora and lexical resources such as WordNet, along with a suite of text processing libraries for classification, tokenization, stemming, tagging, parsing, and semantic reasoning, and an active discussion forum.
44. **Elastic Search** – Official low-level client for Elasticsearch. Its goal is to provide common ground for all Elasticsearch-related code in Python; because of this it tries to be opinion-free and very extendable.
45. **Redis** – Redis is an open source, BSD licensed, advanced key-value cache and store. It is often referred to as a data structure server since keys can contain strings, hashes, lists, sets, sorted sets, bitmaps and hyperloglogs.
46. **Pymongo** – PyMongo is a Python distribution containing tools for working with MongoDB, and is the recommended way to work with MongoDB from Python. This documentation attempts to explain everything you need to know to use PyMongo.
47. **pymysqld** – This package contains a pure-Python MySQL client library. The goal of PyMySQL is to be a drop-in replacement for MySQLdb and work on CPython, PyPy, IronPython and Jython.
48. **Cassandra-python-driver** – A Python client driver for Apache Cassandra. This driver works exclusively with the Cassandra Query Language v3 (CQL3) and Cassandra's native protocol.
49. **GeoIP** – MaxMind GeoIP Legacy Database – Python API
50. **Django** – Django is a high-level Python Web framework that encourages rapid development and clean, pragmatic design. Django should have a separate blog for its libraries, but lets just add some to make this post utilize django more sensibly 😊
 1. **TastyPie** – Tastypie is a web service API framework for Django. It provides a convenient, yet powerful and highly customizable abstraction for creating REST-style interfaces.
 2. **Django Rest Framework** – Django REST framework is a powerful and flexible toolkit that makes it easy to build Web APIs.
 3. **Django-haystack** – Search doesn't have to be hard. Haystack lets you write your search code once and choose the search engine you want it to run on. With a familiar API that should make any Django developer feel right at home and an architecture that allows you to swap things in and out as you need to, it's how search ought to be.
 4. **Django-debug-toolbar** – The Django Debug Toolbar is a configurable set of panels that display various debug information about the current request/response and when clicked, display more details about the panel's content.

5. **Piston** – Piston is a relatively small Django application that lets you create application programming interfaces (API) for your sites.

In final note, my intention was to list down the power of python, kindly explore each on google and read their respective documents for more information.

Connect with me via twitter or contact me from [here](#). Cheers!

SHARE THIS:

[Tweet](#)[Share 0](#)[G+](#)[Share](#)[2](#)[Save](#)[submit](#)[Email](#)[Print](#)

Related:

Setting up Python framework
Django and no-sql database
Cassandra for web development -
Step by Step Tutorial
December 12, 2013
In "Database"

Guidelines for young python
developers - PART 1 | Things you
should know being a programmer
April 20, 2013
In "Programming"

Guidelines for young python
developers - PART 2 | Fun with
happy coding
May 10, 2013
In "Programming"