Outside of the Standard Library

Although the Python standard library is extensive, there is also a robust ecosystem of modules provided by third-party developers and available from the Python Package Index. This appendix describes some of these modules, and the situations when you might want to use them to supplement or even replace the standard library.

Text

The `string` module includes a very basic template tool. Many web frameworks include more powerful template tools, but Jinja2 and Mako are popular standalone alternatives. Both support looping and conditional control structures as well as other features for combining data with a template to produce text output.

The `re` module includes functions for searching and parsing text using formally described patterns called regular expressions. It is not the only way to parse text, though.

The PLY package supports building parsers in the style of the GNU tools lexx and yacc, commonly used for building language compilers. By providing inputs describing the valid tokens, a grammar, and actions to take when each are encountered, it is possible to build fully functional compilers and interpreters, as well as more straightforward data parsers.

PyParsing is a another tool for building parsers. The inputs are instances of classes that can be chained together using operators and method calls to build up a grammar.

Finally, NLTK is a package for processing natural language text – human languages instead of computer languages. It supports parsing sentences into parts of speech, finding the root form of words, and basic semantic processing.

Algorithms

The `functools` module includes some tools for creating decorators, functions that wrap other functions to change how they behave. The wrapt package goes further than `functools.wrap()` to ensure that a decorator is constructed properly and works for all edge-cases.

Dates and Times

The `time` and `datetime` modules provide functions and classes for manipulating time and date values. Both include functions for parsing strings to turn them into internal representations. The dateutil package includes a more flexible parser that makes it easier to build robust applications that are more forgiving of different input formats.

The `datetime` module includes a timezone-aware class for representing a specific time on a specific day. It does not, however, include a full timezone database. The pytz package does provide such a database. It is distributed separately from the standard library because it is maintained by other authors and it is updated frequently when timezone and daylight savings time values are changed by the political institutions that control them.

Mathematics

The `math` module contains fast implementations of advanced mathematical functions. NumPy expands the set of functions supported to include linear algebra and Fourier transform functions. It also includes a fast multi-dimensional array implementation, improving on the version in `array`.

Data Persistence and Exchange

The examples in the `sqlite3` section run SQL statements directly and work with low-level data structures. For large applications, it is often desirable to map classes to tables in the database using an *object relational mapper* or ORM. The sqlalchemy ORM library provides APIs for associating classes with tables, building queries, and connecting to different types of production-grade relational databases.

The lxml package wraps the libxml2 and libxslt libraries to create an alternative to the XML parser in `xml.etree.ElementTree`. Developers familiar with using those libraries from other languages may find lxml easier to adopt in Python.

The defusedxml package contains fixes for “Billion Laughs” and other entity expansion denial of service vulnerabilities in Python’s XML libraries and makes working with untrusted XML safer than using the standard library alone.

Cryptography

The team building the cryptography package says “Our goal is for it to be your ‘cryptographic standard library’.” The cryptography package exposes high-level APIs to make it easy to add cryptographic features to applications and the package

cryptography package exposes high-level APIs to make it easy to add cryptographic features to applications and the package is actively maintained with frequent releases to address vulnerabilities in the underlying libraries such as OpenSSL.

Concurrency with Processes, Threads, and Coroutines

The event loop built into `asyncio` is a reference implementation based on the abstract API defined by the module. It is possible to replace the event loop with a library such as uvloop, which gives better performance in exchange for adding extra application dependencies.

The curio package is another concurrency package similar to asyncio but with a smaller API that treats everything as a coroutine and does not support callbacks in the way asyncio does.

The Twisted library provides an extensible framework for Python programming, with special focus on event-based network programming and multiprotocol integration. It is mature, robust, and well-documented.

The Internet

The requests package is a very popular replacement for `urllib.request`. It provides a consistent API for working with remote resources addressable via HTTP, includes robust SSL support, and can use connection pooling for better performance in multi-threaded applications. It also provides features that make it well suited for accessing REST APIs, such as built-in JSON parsing.

Python's `html` module includes a basic parser for well-formed HTML data. However, real world data is rarely well structured, making parsing it problematic. The BeautifulSoup and PyQuery libraries are alternatives to `html` that are more robust in the face of messy data. Both define APIs for parsing, modifying, and constructing HTML.

The built-in `http.server` package includes base classes for creating simple HTTP servers from scratch. It does not offer much support beyond that for building web-based applications, though. The Django and Pyramid packages are two popular web application frameworks that provide more support for advanced features like request parsing, URL routing, and cookie handling.

Many existing libraries do not work with `asyncio` because they do not integrate with the event loop. A new set of libraries such as aiohttp is being created to fill this gap as part of the aio-libs project.

Email

The API for `imaplib` is relatively low-level, requiring the caller to understand the IMAP protocol to build queries and parse results. The imapclient package provides a higher-level API that is easier to work with for building applications that need to manipulate IMAP mailboxes.

Application Building Blocks

The two standard library modules for building command line interfaces, `argparse` and `getopt`, both separate the definition of command line arguments from their parsing and value processing. Alternatively, click (the "Command Line Interface Construction Kit"), works by defining command processing functions and then associating option and prompt definitions with those commands using decorators.

cliff ("Command Line Interface Formulation Framework") provides a set of base classes for defining commands and a plugin system for extending applications with multiple sub-commands that can be distributed in separate packages. It uses `argparse` to build the help text and argument parser, so the command line processing is familiar.

The docopt package reverses the typical flow by asking the developer to write the help text for a program, which it then parses to understand the sub-commands, valid combinations of options, and sub-commands.

For interactive terminal-based programs, prompt_toolkit includes advanced features like color support, syntax highlighting, input editing, mouse support, and searchable history. It can be used to build command-oriented programs with a prompt loop like the `cmd` module, or full-screen applications like text editors.

While INI files such as used by `configparser` continue to be popular for application configuration, the YAML file format is also very popular. YAML provides many of the data structure features of JSON in a format that is easier for people to read. The PyYAML library provides access to a YAML parser and serializer.

Developer Tools

The standard library module venv is new in Python 3. For similar application isolation under both Python 2 and 3, use virtualenv.

The fixtures package provides several test resource management classes tailor made to work with the `addCleanup()` method of test cases from the `unittest` module. The provided fixture classes can manage loggers, environment variables, temporary files, and more in a consistent and safe way that ensures each test case is completely isolated from others in the suite.

The `distutils` module in the standard library for packaging Python modules for distribution and reuse is deprecated. The

The distutils module in the standard library for packaging Python modules for distribution and reuse is deprecated. The replacement, setuptools, is packaged separately from the standard library to make it easier to deliver new versions frequently. The API for setuptools includes tools for building the list of files to include in a package. There are extensions to automatically build the list from the set of files managed by a version control system. For example, using setuptools-git with source in a git repository causes all of the tracked files to be included in the package by default. After a package is built, the twine application will upload it to the package index to be shared with other developers.

Tools like tabnanny are good at finding common formatting mistakes in Python code. The Python Code Quality Authority maintains an extensive range of more advanced *static analysis tools*, including tools that enforce style guidelines, find common programming errors, and even help avoid excessive complexity.

See also

- Python Package Index or PyPI – The site for finding and downloading extension modules distributed separately from the Python runtime.

Porting Notes

About Python Module of the Week

Quick Links

Text
Algorithms
Dates and Times
Mathematics
Data Persistence and Exchange
Cryptography
Concurrency with Processes, Threads, and Coroutines
The Internet
Email
Application Building Blocks
Developer Tools

This page was last updated 2017-01-29.

Navigation

Porting Notes
About Python Module of the Week

The output from all the example programs from PyMOTW-3 has been generated with Python 3.7.1, unless otherwise noted. Some of the features described here may not be available in earlier versions of Python.

Looking for examples for Python 2?

This Site

Module Index
Index

Other Writing

- Blog
- The Python Standard Library By Example