

The Python Standard Library

While [The Python Language Reference](#) describes the exact syntax and semantics of the Python language, this library reference manual describes the standard library that is distributed with Python. It also describes some of the optional components that are commonly included in Python distributions.

Python's standard library is very extensive, offering a wide range of facilities as indicated by the long table of contents listed below. The library contains built-in modules (written in C) that provide access to system functionality such as file I/O that would otherwise be inaccessible to Python programmers, as well as modules written in Python that provide standardized solutions for many problems that occur in everyday programming. Some of these modules are explicitly designed to encourage and enhance the portability of Python programs by abstracting away platform-specifics into platform-neutral APIs.

The Python installers for the Windows platform usually include the entire standard library and often also include many additional components. For Unix-like operating systems Python is normally provided as a collection of packages, so it may be necessary to use the packaging tools provided with the operating system to obtain some or all of the optional components.

In addition to the standard library, there is a growing collection of several thousand components (from individual programs and modules to packages and entire application development frameworks), available from the [Python Package Index](#).

- [1. Introduction](#)
- [2. Built-in Functions](#)
- [3. Non-essential Built-in Functions](#)
- [4. Built-in Constants](#)
 - [4.1. Constants added by the `site` module](#)
- [5. Built-in Types](#)
 - [5.1. Truth Value Testing](#)
 - [5.2. Boolean Operations — `and`, `or`, `not`](#)
 - [5.3. Comparisons](#)
 - [5.4. Numeric Types — `int`, `float`, `long`, `complex`](#)
 - [5.5. Iterator Types](#)
 - [5.6. Sequence Types — `str`, `unicode`, `list`, `tuple`, `bytearray`, `buffer`, `xrange`](#)
 - [5.7. Set Types — `set`, `frozenset`](#)
 - [5.8. Mapping Types — `dict`](#)
 - [5.9. File Objects](#)
 - [5.10. `memoryview` type](#)
 - [5.11. Context Manager Types](#)
 - [5.12. Other Built-in Types](#)
 - [5.13. Special Attributes](#)
- [6. Built-in Exceptions](#)
 - [6.1. Exception hierarchy](#)
- [7. String Services](#)
 - [7.1. `string` — Common string operations](#)
 - [7.2. `re` — Regular expression operations](#)
 - [7.3. `struct` — Interpret strings as packed binary data](#)
 - [7.4. `difflib` — Helpers for computing deltas](#)
 - [7.5. `StringIO` — Read and write strings as files](#)

- 7.6. **cStringIO** — Faster version of **StringIO**
- 7.7. **textwrap** — Text wrapping and filling
- 7.8. **codecs** — Codec registry and base classes
- 7.9. **unicodedata** — Unicode Database
- 7.10. **stringprep** — Internet String Preparation
- 7.11. **fpformat** — Floating point conversions
- 8. Data Types
 - 8.1. **datetime** — Basic date and time types
 - 8.2. **calendar** — General calendar-related functions
 - 8.3. **collections** — High-performance container datatypes
 - 8.4. **heapq** — Heap queue algorithm
 - 8.5. **bisect** — Array bisection algorithm
 - 8.6. **array** — Efficient arrays of numeric values
 - 8.7. **sets** — Unordered collections of unique elements
 - 8.8. **sched** — Event scheduler
 - 8.9. **mutex** — Mutual exclusion support
 - 8.10. **Queue** — A synchronized queue class
 - 8.11. **weakref** — Weak references
 - 8.12. **UserDict** — Class wrapper for dictionary objects
 - 8.13. **UserList** — Class wrapper for list objects
 - 8.14. **UserString** — Class wrapper for string objects
 - 8.15. **types** — Names for built-in types
 - 8.16. **new** — Creation of runtime internal objects
 - 8.17. **copy** — Shallow and deep copy operations
 - 8.18. **pprint** — Data pretty printer
 - 8.19. **repr** — Alternate **repr()** implementation
- 9. Numeric and Mathematical Modules
 - 9.1. **numbers** — Numeric abstract base classes
 - 9.2. **math** — Mathematical functions
 - 9.3. **cmath** — Mathematical functions for complex numbers
 - 9.4. **decimal** — Decimal fixed point and floating point arithmetic
 - 9.5. **fractions** — Rational numbers
 - 9.6. **random** — Generate pseudo-random numbers
 - 9.7. **itertools** — Functions creating iterators for efficient looping
 - 9.8. **functools** — Higher-order functions and operations on callable objects
 - 9.9. **operator** — Standard operators as functions
- 10. File and Directory Access
 - 10.1. **os.path** — Common pathname manipulations
 - 10.2. **fileinput** — Iterate over lines from multiple input streams
 - 10.3. **stat** — Interpreting **stat()** results
 - 10.4. **statvfs** — Constants used with **os.statvfs()**
 - 10.5. **filecmp** — File and Directory Comparisons
 - 10.6. **tempfile** — Generate temporary files and directories
 - 10.7. **glob** — Unix style pathname pattern expansion
 - 10.8. **fnmatch** — Unix filename pattern matching
 - 10.9. **linecache** — Random access to text lines
 - 10.10. **shutil** — High-level file operations
 - 10.11. **dircache** — Cached directory listings
 - 10.12. **macpath** — Mac OS 9 path manipulation functions

- 11. Data Persistence
 - 11.1. `pickle` — Python object serialization
 - 11.2. `cPickle` — A faster `pickle`
 - 11.3. `copy_reg` — Register `pickle` support functions
 - 11.4. `shelve` — Python object persistence
 - 11.5. `marshal` — Internal Python object serialization
 - 11.6. `anydbm` — Generic access to DBM-style databases
 - 11.7. `whichdb` — Guess which DBM module created a database
 - 11.8. `dbm` — Simple “database” interface
 - 11.9. `gdbm` — GNU’s reinterpretation of `dbm`
 - 11.10. `dbhash` — DBM-style interface to the BSD database library
 - 11.11. `bsddb` — Interface to Berkeley DB library
 - 11.12. `dumbdbm` — Portable DBM implementation
 - 11.13. `sqlite3` — DB-API 2.0 interface for SQLite databases
- 12. Data Compression and Archiving
 - 12.1. `zlib` — Compression compatible with `gzip`
 - 12.2. `gzip` — Support for `gzip` files
 - 12.3. `bz2` — Compression compatible with `bzip2`
 - 12.4. `zipfile` — Work with ZIP archives
 - 12.5. `tarfile` — Read and write tar archive files
- 13. File Formats
 - 13.1. `csv` — CSV File Reading and Writing
 - 13.2. `ConfigParser` — Configuration file parser
 - 13.3. `robotparser` — Parser for `robots.txt`
 - 13.4. `netrc` — `netrc` file processing
 - 13.5. `xdrlib` — Encode and decode XDR data
 - 13.6. `plistlib` — Generate and parse Mac OS X `.plist` files
- 14. Cryptographic Services
 - 14.1. `hashlib` — Secure hashes and message digests
 - 14.2. `hmac` — Keyed-Hashing for Message Authentication
 - 14.3. `md5` — MD5 message digest algorithm
 - 14.4. `sha` — SHA-1 message digest algorithm
- 15. Generic Operating System Services
 - 15.1. `os` — Miscellaneous operating system interfaces
 - 15.2. `io` — Core tools for working with streams
 - 15.3. `time` — Time access and conversions
 - 15.4. `argparse` — Parser for command-line options, arguments and sub-commands
 - 15.5. `optparse` — Parser for command line options
 - 15.6. `getopt` — C-style parser for command line options
 - 15.7. `logging` — Logging facility for Python
 - 15.8. `logging.config` — Logging configuration
 - 15.9. `logging.handlers` — Logging handlers
 - 15.10. `getpass` — Portable password input
 - 15.11. `curses` — Terminal handling for character-cell displays
 - 15.12. `curses.textpad` — Text input widget for `curses` programs
 - 15.13. `curses.ascii` — Utilities for ASCII characters
 - 15.14. `curses.panel` — A panel stack extension for `curses`
 - 15.15. `platform` — Access to underlying platform’s identifying data
 - 15.16. `errno` — Standard `errno` system symbols
 - 15.17. `ctypes` — A foreign function library for Python

- 16. Optional Operating System Services
 - 16.1. `select` — Waiting for I/O completion
 - 16.2. `threading` — Higher-level threading interface
 - 16.3. `thread` — Multiple threads of control
 - 16.4. `dummy_threading` — Drop-in replacement for the `threading` module
 - 16.5. `dummy_thread` — Drop-in replacement for the `thread` module
 - 16.6. `multiprocessing` — Process-based “threading” interface
 - 16.7. `mmap` — Memory-mapped file support
 - 16.8. `readline` — GNU readline interface
 - 16.9. `rlcompleter` — Completion function for GNU readline
- 17. Interprocess Communication and Networking
 - 17.1. `subprocess` — Subprocess management
 - 17.2. `socket` — Low-level networking interface
 - 17.3. `ssl` — TLS/SSL wrapper for socket objects
 - 17.4. `signal` — Set handlers for asynchronous events
 - 17.5. `popen2` — Subprocesses with accessible I/O streams
 - 17.6. `asyncore` — Asynchronous socket handler
 - 17.7. `asynchat` — Asynchronous socket command/response handler
- 18. Internet Data Handling
 - 18.1. `email` — An email and MIME handling package
 - 18.2. `json` — JSON encoder and decoder
 - 18.3. `mailcap` — Mailcap file handling
 - 18.4. `mailbox` — Manipulate mailboxes in various formats
 - 18.5. `mhlib` — Access to MH mailboxes
 - 18.6. `mimetools` — Tools for parsing MIME messages
 - 18.7. `mimetypes` — Map filenames to MIME types
 - 18.8. `MimeWriter` — Generic MIME file writer
 - 18.9. `mimify` — MIME processing of mail messages
 - 18.10. `multifile` — Support for files containing distinct parts
 - 18.11. `rfc822` — Parse RFC 2822 mail headers
 - 18.12. `base64` — RFC 3548: Base16, Base32, Base64 Data Encodings
 - 18.13. `binhex` — Encode and decode binhex4 files
 - 18.14. `binascii` — Convert between binary and ASCII
 - 18.15. `quopri` — Encode and decode MIME quoted-printable data
 - 18.16. `uu` — Encode and decode uuencode files
- 19. Structured Markup Processing Tools
 - 19.1. `HTMLParser` — Simple HTML and XHTML parser
 - 19.2. `sgmlib` — Simple SGML parser
 - 19.3. `htmlib` — A parser for HTML documents
 - 19.4. `htmlentitydefs` — Definitions of HTML general entities
 - 19.5. XML Processing Modules
 - 19.6. XML vulnerabilities
 - 19.7. `xml.etree.ElementTree` — The ElementTree XML API
 - 19.8. `xml.dom` — The Document Object Model API
 - 19.9. `xml.dom.minidom` — Minimal DOM implementation
 - 19.10. `xml.dom.pulldom` — Support for building partial DOM trees
 - 19.11. `xml.sax` — Support for SAX2 parsers
 - 19.12. `xml.sax.handler` — Base classes for SAX handlers
 - 19.13. `xml.sax.saxutils` — SAX Utilities
 - 19.14. `xml.sax.xmlreader` — Interface for XML parsers

- 19.15. `xml.parsers.expat` — Fast XML parsing using Expat
- 20. Internet Protocols and Support
 - 20.1. `webbrowser` — Convenient Web-browser controller
 - 20.2. `cgi` — Common Gateway Interface support
 - 20.3. `cgitb` — Traceback manager for CGI scripts
 - 20.4. `wsgiref` — WSGI Utilities and Reference Implementation
 - 20.5. `urllib` — Open arbitrary resources by URL
 - 20.6. `urllib2` — extensible library for opening URLs
 - 20.7. `httplib` — HTTP protocol client
 - 20.8. `ftplib` — FTP protocol client
 - 20.9. `poplib` — POP3 protocol client
 - 20.10. `imaplib` — IMAP4 protocol client
 - 20.11. `nnplib` — NNTP protocol client
 - 20.12. `smtplib` — SMTP protocol client
 - 20.13. `smtpd` — SMTP Server
 - 20.14. `telnetlib` — Telnet client
 - 20.15. `uuid` — UUID objects according to RFC 4122
 - 20.16. `urlparse` — Parse URLs into components
 - 20.17. `SocketServer` — A framework for network servers
 - 20.18. `BaseHTTPServer` — Basic HTTP server
 - 20.19. `SimpleHTTPServer` — Simple HTTP request handler
 - 20.20. `CGIHTTPServer` — CGI-capable HTTP request handler
 - 20.21. `cookielib` — Cookie handling for HTTP clients
 - 20.22. `Cookie` — HTTP state management
 - 20.23. `xmlrpclib` — XML-RPC client access
 - 20.24. `SimpleXMLRPCServer` — Basic XML-RPC server
 - 20.25. `DocXMLRPCServer` — Self-documenting XML-RPC server
- 21. Multimedia Services
 - 21.1. `audioop` — Manipulate raw audio data
 - 21.2. `imageop` — Manipulate raw image data
 - 21.3. `aifc` — Read and write AIFF and AIFC files
 - 21.4. `sunau` — Read and write Sun AU files
 - 21.5. `wave` — Read and write WAV files
 - 21.6. `chunk` — Read IFF chunked data
 - 21.7. `coloursys` — Conversions between color systems
 - 21.8. `imghdr` — Determine the type of an image
 - 21.9. `sndhdr` — Determine type of sound file
 - 21.10. `ossaudiodev` — Access to OSS-compatible audio devices
- 22. Internationalization
 - 22.1. `gettext` — Multilingual internationalization services
 - 22.2. `locale` — Internationalization services
- 23. Program Frameworks
 - 23.1. `cmd` — Support for line-oriented command interpreters
 - 23.2. `shlex` — Simple lexical analysis
- 24. Graphical User Interfaces with Tk
 - 24.1. `Tkinter` — Python interface to Tcl/Tk
 - 24.2. `ttk` — Tk themed widgets
 - 24.3. `Tix` — Extension widgets for Tk
 - 24.4. `ScrolledText` — Scrolled Text Widget
 - 24.5. `turtle` — Turtle graphics for Tk

- 24.6. IDLE
- 24.7. Other Graphical User Interface Packages
- 25. Development Tools
 - 25.1. **pydoc** — Documentation generator and online help system
 - 25.2. **doctest** — Test interactive Python examples
 - 25.3. **unittest** — Unit testing framework
 - 25.4. 2to3 - Automated Python 2 to 3 code translation
 - 25.5. **test** — Regression tests package for Python
 - 25.6. **test.test_support** — Utility functions for tests
- 26. Debugging and Profiling
 - 26.1. **bdb** — Debugger framework
 - 26.2. **pdb** — The Python Debugger
 - 26.3. Debugger Commands
 - 26.4. The Python Profilers
 - 26.5. **hotshot** — High performance logging profiler
 - 26.6. **timeit** — Measure execution time of small code snippets
 - 26.7. **trace** — Trace or track Python statement execution
- 27. Software Packaging and Distribution
 - 27.1. **distutils** — Building and installing Python modules
 - 27.2. **ensurepip** — Bootstrapping the **pip** installer
- 28. Python Runtime Services
 - 28.1. **sys** — System-specific parameters and functions
 - 28.2. **sysconfig** — Provide access to Python's configuration information
 - 28.3. **__builtin__** — Built-in objects
 - 28.4. **future_builtins** — Python 3 builtins
 - 28.5. **__main__** — Top-level script environment
 - 28.6. **warnings** — Warning control
 - 28.7. **contextlib** — Utilities for **with**-statement contexts
 - 28.8. **abc** — Abstract Base Classes
 - 28.9. **atexit** — Exit handlers
 - 28.10. **traceback** — Print or retrieve a stack traceback
 - 28.11. **__future__** — Future statement definitions
 - 28.12. **gc** — Garbage Collector interface
 - 28.13. **inspect** — Inspect live objects
 - 28.14. **site** — Site-specific configuration hook
 - 28.15. **user** — User-specific configuration hook
 - 28.16. **fpectl** — Floating point exception control
- 29. Custom Python Interpreters
 - 29.1. **code** — Interpreter base classes
 - 29.2. **codeop** — Compile Python code
- 30. Restricted Execution
 - 30.1. **rexec** — Restricted execution framework
 - 30.2. **Bastion** — Restricting access to objects
- 31. Importing Modules
 - 31.1. **imp** — Access the **import** internals
 - 31.2. **importlib** — Convenience wrappers for **__import__()**
 - 31.3. **importlib** — Import utilities
 - 31.4. **zipimport** — Import modules from Zip archives
 - 31.5. **pkgutil** — Package extension utility
 - 31.6. **modulefinder** — Find modules used by a script

- 31.7. **runpy** — Locating and executing Python modules
- 32. Python Language Services
 - 32.1. **parser** — Access Python parse trees
 - 32.2. **ast** — Abstract Syntax Trees
 - 32.3. **symtable** — Access to the compiler's symbol tables
 - 32.4. **symbol** — Constants used with Python parse trees
 - 32.5. **token** — Constants used with Python parse trees
 - 32.6. **keyword** — Testing for Python keywords
 - 32.7. **tokenize** — Tokenizer for Python source
 - 32.8. **tabnanny** — Detection of ambiguous indentation
 - 32.9. **pyclbr** — Python class browser support
 - 32.10. **py_compile** — Compile Python source files
 - 32.11. **compileall** — Byte-compile Python libraries
 - 32.12. **dis** — Disassembler for Python bytecode
 - 32.13. **pickletools** — Tools for pickle developers
- 33. Python compiler package
 - 33.1. The basic interface
 - 33.2. Limitations
 - 33.3. Python Abstract Syntax
 - 33.4. Using Visitors to Walk ASTs
 - 33.5. Bytecode Generation
- 34. Miscellaneous Services
 - 34.1. **formatter** — Generic output formatting
- 35. MS Windows Specific Services
 - 35.1. **msilib** — Read and write Microsoft Installer files
 - 35.2. **msvcrt** — Useful routines from the MS VC++ runtime
 - 35.3. **_winreg** — Windows registry access
 - 35.4. **winsound** — Sound-playing interface for Windows
- 36. Unix Specific Services
 - 36.1. **posix** — The most common POSIX system calls
 - 36.2. **pwd** — The password database
 - 36.3. **spwd** — The shadow password database
 - 36.4. **grp** — The group database
 - 36.5. **crypt** — Function to check Unix passwords
 - 36.6. **dl** — Call C functions in shared objects
 - 36.7. **termios** — POSIX style tty control
 - 36.8. **tty** — Terminal control functions
 - 36.9. **pty** — Pseudo-terminal utilities
 - 36.10. **fcntl** — The **fcntl** and **ioctl** system calls
 - 36.11. **pipes** — Interface to shell pipelines
 - 36.12. **posixfile** — File-like objects with locking support
 - 36.13. **resource** — Resource usage information
 - 36.14. **nis** — Interface to Sun's NIS (Yellow Pages)
 - 36.15. **syslog** — Unix syslog library routines
 - 36.16. **commands** — Utilities for running commands
- 37. Mac OS X specific services
 - 37.1. **ic** — Access to the Mac OS X Internet Config
 - 37.2. **MacOS** — Access to Mac OS interpreter features
 - 37.3. **macostools** — Convenience routines for file manipulation
 - 37.4. **findertools** — The **finder**'s Apple Events interface

- 37.5. **EasyDialogs** — Basic Macintosh dialogs
- 37.6. **FrameWork** — Interactive application framework
- 37.7. **autoGIL** — Global Interpreter Lock handling in event loops
- 37.8. Mac OS Toolbox Modules
- 37.9. **ColorPicker** — Color selection dialog
- 38. MacPython OSA Modules
 - 38.1. **gensuitemodule** — Generate OSA stub packages
 - 38.2. **aetools** — OSA client support
 - 38.3. **aepack** — Conversion between Python variables and AppleEvent data containers
 - 38.4. **aetypes** — AppleEvent objects
 - 38.5. **MiniAETFrame** — Open Scripting Architecture server support
- 39. SGI IRIX Specific Services
 - 39.1. **al** — Audio functions on the SGI
 - 39.2. **AL** — Constants used with the **al** module
 - 39.3. **cd** — CD-ROM access on SGI systems
 - 39.4. **fl** — FORMS library for graphical user interfaces
 - 39.5. **FL** — Constants used with the **fl** module
 - 39.6. **flp** — Functions for loading stored FORMS designs
 - 39.7. **fm** — *Font Manager* interface
 - 39.8. **gl** — *Graphics Library* interface
 - 39.9. **DEVICE** — Constants used with the **gl** module
 - 39.10. **GL** — Constants used with the **gl** module
 - 39.11. **imgfile** — Support for SGI imglib files
 - 39.12. **jpeg** — Read and write JPEG files
- 40. SunOS Specific Services
 - 40.1. **sunaudiodev** — Access to Sun audio hardware
 - 40.2. **SUNAUDIODEV** — Constants used with **sunaudiodev**
- 41. Undocumented Modules
 - 41.1. Miscellaneous useful utilities
 - 41.2. Platform specific modules
 - 41.3. Multimedia
 - 41.4. Undocumented Mac OS modules
 - 41.5. Obsolete
 - 41.6. SGI-specific Extension modules