

`__future__` *Future statement definitions*  
`__main__`  
`_dummy_thread`  
`_thread`

*The environment where the top-level script is run.*  
*Drop-in replacement for the `_thread` module.*  
*Low-level threading API.*

## **a**

`abc` *Abstract base classes according to PEP 3119.*  
`aifc` *Read and write audio files in AIFF or AIFC format.*  
`argparse` *Command-line option and argument parsing library.*  
`array` *Space efficient arrays of uniformly typed numeric values.*  
`ast` *Abstract Syntax Tree classes and manipulation.*  
`asynchat` *Support for asynchronous command/response protocols.*  
`asyncio` *Asynchronous I/O.*  
`asyncore` *A base class for developing asynchronous socket handling services.*  
`atexit` *Register and execute cleanup functions.*  
`audioop` *Manipulate raw audio data.*

## **b**

`base64` *RFC 3548: Base16, Base32, Base64 Data Encodings; Base85 and Ascii85*  
`bdb` *Debugger framework.*  
`binascii` *Tools for converting between binary and various ASCII-encoded binary representations.*  
`binhex` *Encode and decode files in binhex4 format.*  
`bisect` *Array bisection algorithms for binary searching.*  
`builtins` *The module that provides the built-in namespace.*  
`bz2` *Interfaces for bzip2 compression and decompression.*

## **c**

`calendar` *Functions for working with calendars, including some emulation of the Unix `cal` program.*  
`cgi` *Helpers for running Python scripts via the Common Gateway Interface*  
`cgitb` *Configurable traceback handler for CGI scripts.*  
`chunk` *Module to read IFF chunks.*  
`cmath` *Mathematical functions for complex numbers.*  
`cmd` *Build line-oriented command interpreters.*

	<code>code</code>	<i>Facilities to implement read-eval-print loops.</i>
	<code>codecs</code>	<i>Encode and decode data and streams.</i>
	<code>codeop</code>	<i>Compile (possibly incomplete) Python code.</i>
+	<code>collections</code>	<i>Container datatypes</i>
	<code>colorsys</code>	<i>Conversion functions between RGB and other color systems.</i>
	<code>compileall</code>	<i>Tools for byte-compiling all Python source files in a directory tree.</i>
+	<code>concurrent</code>	
	<code>configparser</code>	<i>Configuration file parser.</i>
	<code>contextlib</code>	<i>Utilities for with-statement contexts.</i>
	<code>contextvars</code>	<i>Context Variables</i>
	<code>copy</code>	<i>Shallow and deep copy operations.</i>
	<code>copyreg</code>	<i>Register pickle support functions.</i>
	<code>cProfile</code>	
	<code>crypt (Unix)</code>	<i>The crypt() function used to check Unix passwords.</i>
	<code>csv</code>	<i>Write and read tabular data to and from delimited files.</i>
	<code>ctypes</code>	<i>A foreign function library for Python.</i>
+	<code>curses (Unix)</code>	<i>An interface to the curses library, providing portable terminal handling.</i>

## d

	<code>dataclasses</code>	<i>Generate special methods on user-defined classes.</i>
	<code>datetime</code>	<i>Basic date and time types.</i>
+	<code>dbm</code>	<i>Interfaces to various Unix "database" formats.</i>
	<code>decimal</code>	<i>Implementation of the General Decimal Arithmetic Specification.</i>
	<code>difflib</code>	<i>Helpers for computing differences between objects.</i>
	<code>dis</code>	<i>Disassembler for Python bytecode.</i>
+	<code>distutils</code>	<i>Support for building and installing Python modules into an existing Python installation.</i>
	<code>doctest</code>	<i>Test pieces of code within docstrings.</i>
	<code>dummy_threading</code>	<i>Drop-in replacement for the threading module.</i>

## e

+	<code>email</code>	<i>Package supporting the parsing, manipulating, and generating email messages.</i>
+	<code>encodings</code>	
	<code>ensurepip</code>	<i>Bootstrapping the "pip" installer into an existing Python installation or virtual environment.</i>
	<code>enum</code>	<i>Implementation of an enumeration class.</i>
	<code>errno</code>	<i>Standard errno system symbols.</i>

## f

<code>faulthandler</code>	<i>Dump the Python traceback.</i>
<code>fcntl (Unix)</code>	<i>The <code>fcntl()</code> and <code>ioctl()</code> system calls.</i>
<code>filecmp</code>	<i>Compare files efficiently.</i>
<code>fileinput</code>	<i>Loop over standard input or a list of files.</i>
<code>fnmatch</code>	<i>Unix shell style filename pattern matching.</i>
<code>formatter</code>	<b>Deprecated:</b> <i>Generic output formatter and device interface.</i>
<code>fractions</code>	<i>Rational numbers.</i>
<code>ftplib</code>	<i>FTP protocol client (requires sockets).</i>
<code>functools</code>	<i>Higher-order functions and operations on callable objects.</i>

## g

<code>gc</code>	<i>Interface to the cycle-detecting garbage collector.</i>
<code>getopt</code>	<i>Portable parser for command line options; support both short and long option names.</i>
<code>getpass</code>	<i>Portable reading of passwords and retrieval of the userid.</i>
<code>gettext</code>	<i>Multilingual internationalization services.</i>
<code>glob</code>	<i>Unix shell style pathname pattern expansion.</i>
<code>grp (Unix)</code>	<i>The group database (<code>getgrnam()</code> and friends).</i>
<code>gzip</code>	<i>Interfaces for gzip compression and decompression using file objects.</i>

## h

<code>hashlib</code>	<i>Secure hash and message digest algorithms.</i>
<code>heapq</code>	<i>Heap queue algorithm (a.k.a. priority queue).</i>
<code>hmac</code>	<i>Keyed-Hashing for Message Authentication (HMAC) implementation</i>
<code>html</code>	<i>Helpers for manipulating HTML.</i>
<code>http</code>	<i>HTTP status codes and messages</i>

## i

<code>imaplib</code>	<i>IMAP4 protocol client (requires sockets).</i>
<code>imghdr</code>	<i>Determine the type of image contained in a file or byte stream.</i>
<code>imp</code>	<b>Deprecated:</b> <i>Access the implementation of the <code>import</code> statement.</i>
<code>importlib</code>	<i>The implementation of the import machinery.</i>
<code>inspect</code>	<i>Extract information and source code from live objects.</i>
<code>io</code>	<i>Core tools for working with streams.</i>

`ipaddress`

*IPv4/IPv6 manipulation library.*

`itertools`

*Functions creating iterators for efficient looping.*

## **j**



`json`

*Encode and decode the JSON format.*

## **k**

`keyword`

*Test whether a string is a keyword in Python.*

## **l**

`lib2to3`

*the 2to3 library*

`linecache`

*This module provides random access to individual lines from text files.*

`locale`

*Internationalization services.*



`logging`

*Flexible event logging system for applications.*

`lzma`

*A Python wrapper for the liblzma compression library.*

## **m**

`macpath`

*Mac OS 9 path manipulation functions.*

`mailbox`

*Manipulate mailboxes in various formats*

`mailcap`

*Mailcap file handling.*

`marshal`

*Convert Python objects to streams of bytes and back (with different constraints).*

`math`

*Mathematical functions (sin() etc.).*

`mimetypes`

*Mapping of filename extensions to MIME types.*

`mmap`

*Interface to memory-mapped files for Unix and Windows.*

`modulefinder`

*Find modules used by a script.*

`msilib` (Windows)

*Creation of Microsoft Installer files, and CAB files.*

`msvcrt` (Windows)

*Miscellaneous useful routines from the MS VC++ runtime.*



`multiprocessing`

*Process-based parallelism.*

## **n**

`netrc`

*Loading of .netrc files.*

`nis` (Unix)

*Interface to Sun's NIS (Yellow Pages) library.*

`nntplib`

*NNTP protocol client (requires sockets).*

`numbers`

*Numeric abstract base classes (Complex, Real, Integral, etc.).*

## o

`operator`

*Functions corresponding to the standard operators.*

`optparse`

**Deprecated:** *Command-line option parsing library.*

`os`

*Miscellaneous operating system interfaces.*

`ossaudiodev` (*Linux, FreeBSD*)

*Access to OSS-compatible audio devices.*

## p

`parser`

*Access parse trees for Python source code.*

`pathlib`

*Object-oriented filesystem paths*

`pdb`

*The Python debugger for interactive interpreters.*

`pickle`

*Convert Python objects to streams of bytes and back.*

`pickletools`

*Contains extensive comments about the pickle protocols and pickle-machine opcodes, as well as some useful functions.*

`pipes` (*Unix*)

*A Python interface to Unix shell pipelines.*

`pkgutil`

*Utilities for the import system.*

`platform`

*Retrieves as much platform identifying data as possible.*

`plistlib`

*Generate and parse Mac OS X plist files.*

`poplib`

*POP3 protocol client (requires sockets).*

`posix` (*Unix*)

*The most common POSIX system calls (normally used via module `os`).*

`pprint`

*Data pretty printer.*

`profile`

*Python source profiler.*

`pstats`

*Statistics object for use with the profiler.*

`pty` (*Linux*)

*Pseudo-Terminal Handling for Linux.*

`pwd` (*Unix*)

*The password database (`getpwnam()` and friends).*

`py_compile`

*Generate byte-code files from Python source files.*

`pyclbr`

*Supports information extraction for a Python class browser.*

`pydoc`

*Documentation generator and online help system.*

## q

`queue`

*A synchronized queue class.*

`quopri`

*Encode and decode files using the MIME quoted-printable encoding.*

## r

`random`

*Generate pseudo-random numbers with various common distributions*

`re`



*Regular expression operations.*

<code>readline</code> (Unix)	GNU readline support for Python.
<code>reprlib</code>	Alternate <code>repr()</code> implementation with size limits.
<code>resource</code> (Unix)	An interface to provide resource usage information on the current process.
<code>rlcompleter</code>	Python identifier completion, suitable for the GNU readline library.
<code>runpy</code>	Locate and run Python modules without importing them first.



## **S**

<code>sched</code>	General purpose event scheduler.
<code>secrets</code>	Generate secure random numbers for managing secrets.
<code>select</code>	Wait for I/O completion on multiple streams.
<code>selectors</code>	High-level I/O multiplexing.
<code>shelve</code>	Python object persistence.
<code>shlex</code>	Simple lexical analysis for Unix shell-like languages.
<code>shutil</code>	High-level file operations, including copying.
<code>signal</code>	Set handlers for asynchronous events.
<code>site</code>	Module responsible for site-specific configuration.
<code>smtpd</code>	A SMTP server implementation in Python.
<code>smtplib</code>	SMTP protocol client (requires sockets).
<code>sndhdr</code>	Determine type of a sound file.
<code>socket</code>	Low-level networking interface.
<code>socketserver</code>	A framework for network servers.
<code>spwd</code> (Unix)	The shadow password database ( <code>getspnam()</code> and friends).
<code>sqlite3</code>	A DB-API 2.0 implementation using SQLite 3.x.
<code>ssl</code>	TLS/SSL wrapper for socket objects
<code>stat</code>	Utilities for interpreting the results of <code>os.stat()</code> , <code>os.lstat()</code> and <code>os.fstat()</code> .
<code>statistics</code>	mathematical statistics functions
<code>string</code>	Common string operations.
<code>stringprep</code>	String preparation, as per RFC 3453
<code>struct</code>	Interpret bytes as packed binary data.
<code>subprocess</code>	Subprocess management.
<code>sunau</code>	Provide an interface to the Sun AU sound format.
<code>symbol</code>	Constants representing internal nodes of the parse tree.
<code>symtable</code>	Interface to the compiler's internal symbol tables.
<code>sys</code>	Access system-specific parameters and functions.
<code>sysconfig</code>	Python's configuration information
<code>syslog</code> (Unix)	An interface to the Unix syslog library routines.

## **t**

<code>tabnanny</code>	<i>Tool for detecting white space related problems in Python source files in a directory tree.</i>
<code>tarfile</code>	<i>Read and write tar-format archive files.</i>
<code>telnetlib</code>	<i>Telnet client class.</i>
<code>tempfile</code>	<i>Generate temporary files and directories.</i>
<code>termios (Unix)</code>	<i>POSIX style tty control.</i>
 <code>test</code>	<i>Regression tests package containing the testing suite for Python.</i>
<code>textwrap</code>	<i>Text wrapping and filling</i>
<code>threading</code>	<i>Thread-based parallelism.</i>
<code>time</code>	<i>Time access and conversions.</i>
<code>timeit</code>	<i>Measure the execution time of small code snippets.</i>
 <code>tkinter</code>	<i>Interface to Tcl/Tk for graphical user interfaces</i>
<code>token</code>	<i>Constants representing terminal nodes of the parse tree.</i>
<code>tokenize</code>	<i>Lexical scanner for Python source code.</i>
<code>trace</code>	<i>Trace or track Python statement execution.</i>
<code>traceback</code>	<i>Print or retrieve a stack traceback.</i>
<code>tracemalloc</code>	<i>Trace memory allocations.</i>
<code>tty (Unix)</code>	<i>Utility functions that perform common terminal control operations.</i>
<code>turtle</code>	<i>An educational framework for simple graphics applications</i>
<code>turtledemo</code>	<i>A viewer for example turtle scripts</i>
<code>types</code>	<i>Names for built-in types.</i>
<code>typing</code>	<i>Support for type hints (see PEP 484).</i>

## **u**

<code>unicodedata</code>	<i>Access the Unicode Database.</i>
 <code>unittest</code>	<i>Unit testing framework for Python.</i>
 <code>urllib</code>	
<code>uu</code>	<i>Encode and decode files in uuencode format.</i>
<code>uuid</code>	<i>UUID objects (universally unique identifiers) according to RFC 4122</i>

## **v**

<code>venv</code>	<i>Creation of virtual environments.</i>
-------------------	--

## **w**

<code>warnings</code>	<i>Issue warning messages and control their disposition.</i>
-----------------------	--

	<code>wave</code>	<i>Provide an interface to the WAV sound format.</i>
	<code>weakref</code>	<i>Support for weak references and weak dictionaries.</i>
	<code>webbrowser</code>	<i>Easy-to-use controller for Web browsers.</i>
	<code>winreg</code> (Windows)	<i>Routines and objects for manipulating the Windows registry.</i>
	<code>winsound</code> (Windows)	<i>Access to the sound-playing machinery for Windows.</i>
+	<code>wsgiref</code>	<i>WSGI Utilities and Reference Implementation.</i>
<b>x</b>		
	<code>xdrlib</code>	<i>Encoders and decoders for the External Data Representation (XDR).</i>
+	<code>xml</code>	<i>Package containing XML processing modules</i>
+	<code>xmlrpc</code>	
<b>z</b>		
	<code>zipapp</code>	<i>Manage executable Python zip archives</i>
	<code>zipfile</code>	<i>Read and write ZIP-format archive files.</i>
	<code>zipimport</code>	<i>support for importing Python modules from ZIP archives.</i>
	<code>zlib</code>	<i>Low-level interface to compression and decompression routines compatible with gzip.</i>