

---

# The Python Library Reference

*Release 3.7.2*

**Guido van Rossum  
and the Python development team**

**January 12, 2019**

**Python Software Foundation  
Email: [docs@python.org](mailto:docs@python.org)**



# CONTENTS

<b>1</b>	<b>Introduction</b>	<b>3</b>
1.1	Notes on availability . . . . .	3
<b>2</b>	<b>Built-in Functions</b>	<b>5</b>
<b>3</b>	<b>Built-in Constants</b>	<b>27</b>
3.1	Constants added by the <code>site</code> module . . . . .	28
<b>4</b>	<b>Built-in Types</b>	<b>29</b>
4.1	Truth Value Testing . . . . .	29
4.2	Boolean Operations — <code>and</code> , <code>or</code> , <code>not</code> . . . . .	29
4.3	Comparisons . . . . .	30
4.4	Numeric Types — <code>int</code> , <code>float</code> , <code>complex</code> . . . . .	30
4.5	Iterator Types . . . . .	36
4.6	Sequence Types — <code>list</code> , <code>tuple</code> , <code>range</code> . . . . .	37
4.7	Text Sequence Type — <code>str</code> . . . . .	43
4.8	Binary Sequence Types — <code>bytes</code> , <code>bytearray</code> , <code>memoryview</code> . . . . .	53
4.9	Set Types — <code>set</code> , <code>frozenset</code> . . . . .	74
4.10	Mapping Types — <code>dict</code> . . . . .	77
4.11	Context Manager Types . . . . .	81
4.12	Other Built-in Types . . . . .	82
4.13	Special Attributes . . . . .	84
<b>5</b>	<b>Built-in Exceptions</b>	<b>87</b>
5.1	Base classes . . . . .	87
5.2	Concrete exceptions . . . . .	88
5.3	Warnings . . . . .	94
5.4	Exception hierarchy . . . . .	94
<b>6</b>	<b>Text Processing Services</b>	<b>97</b>
6.1	<code>string</code> — Common string operations . . . . .	97
6.2	<code>re</code> — Regular expression operations . . . . .	108
6.3	<code>difflib</code> — Helpers for computing deltas . . . . .	128
6.4	<code>textwrap</code> — Text wrapping and filling . . . . .	139
6.5	<code>unicodedata</code> — Unicode Database . . . . .	142
6.6	<code>stringprep</code> — Internet String Preparation . . . . .	144
6.7	<code>readline</code> — GNU readline interface . . . . .	145
6.8	<code>rlcompleter</code> — Completion function for GNU readline . . . . .	150
<b>7</b>	<b>Binary Data Services</b>	<b>151</b>
7.1	<code>struct</code> — Interpret bytes as packed binary data . . . . .	151

7.2	<code>codecs</code> — Codec registry and base classes	156
<b>8</b>	<b>Data Types</b>	<b>175</b>
8.1	<code>datetime</code> — Basic date and time types	175
8.2	<code>calendar</code> — General calendar-related functions	206
8.3	<code>collections</code> — Container datatypes	210
8.4	<code>collections.abc</code> — Abstract Base Classes for Containers	227
8.5	<code>heapq</code> — Heap queue algorithm	231
8.6	<code>bisect</code> — Array bisection algorithm	235
8.7	<code>array</code> — Efficient arrays of numeric values	238
8.8	<code>weakref</code> — Weak references	240
8.9	<code>types</code> — Dynamic type creation and names for built-in types	248
8.10	<code>copy</code> — Shallow and deep copy operations	252
8.11	<code>pprint</code> — Data pretty printer	253
8.12	<code>reprlib</code> — Alternate <code>repr()</code> implementation	259
8.13	<code>enum</code> — Support for enumerations	261
<b>9</b>	<b>Numeric and Mathematical Modules</b>	<b>281</b>
9.1	<code>numbers</code> — Numeric abstract base classes	281
9.2	<code>math</code> — Mathematical functions	284
9.3	<code>cmath</code> — Mathematical functions for complex numbers	289
9.4	<code>decimal</code> — Decimal fixed point and floating point arithmetic	293
9.5	<code>fractions</code> — Rational numbers	321
9.6	<code>random</code> — Generate pseudo-random numbers	323
9.7	<code>statistics</code> — Mathematical statistics functions	329
<b>10</b>	<b>Functional Programming Modules</b>	<b>337</b>
10.1	<code>itertools</code> — Functions creating iterators for efficient looping	337
10.2	<code>functools</code> — Higher-order functions and operations on callable objects	352
10.3	<code>operator</code> — Standard operators as functions	359
<b>11</b>	<b>File and Directory Access</b>	<b>367</b>
11.1	<code>pathlib</code> — Object-oriented filesystem paths	367
11.2	<code>os.path</code> — Common pathname manipulations	384
11.3	<code>fileinput</code> — Iterate over lines from multiple input streams	389
11.4	<code>stat</code> — Interpreting <code>stat()</code> results	391
11.5	<code>filecmp</code> — File and Directory Comparisons	396
11.6	<code>tempfile</code> — Generate temporary files and directories	398
11.7	<code>glob</code> — Unix style pathname pattern expansion	403
11.8	<code>fnmatch</code> — Unix filename pattern matching	404
11.9	<code>linecache</code> — Random access to text lines	405
11.10	<code>shutil</code> — High-level file operations	406
11.11	<code>macpath</code> — Mac OS 9 path manipulation functions	414
<b>12</b>	<b>Data Persistence</b>	<b>415</b>
12.1	<code>pickle</code> — Python object serialization	415
12.2	<code>copyreg</code> — Register <code>pickle</code> support functions	428
12.3	<code>shelve</code> — Python object persistence	429
12.4	<code>marshal</code> — Internal Python object serialization	432
12.5	<code>dbm</code> — Interfaces to Unix “databases”	433
12.6	<code>sqlite3</code> — DB-API 2.0 interface for SQLite databases	437
<b>13</b>	<b>Data Compression and Archiving</b>	<b>459</b>
13.1	<code>zlib</code> — Compression compatible with <code>gzip</code>	459
13.2	<code>gzip</code> — Support for <code>gzip</code> files	463

13.3	<code>bz2</code> — Support for <code>bzip2</code> compression . . . . .	465
13.4	<code>lzma</code> — Compression using the LZMA algorithm . . . . .	468
13.5	<code>zipfile</code> — Work with ZIP archives . . . . .	474
13.6	<code>tarfile</code> — Read and write tar archive files . . . . .	481
<b>14</b>	<b>File Formats</b>	<b>493</b>
14.1	<code>csv</code> — CSV File Reading and Writing . . . . .	493
14.2	<code>configparser</code> — Configuration file parser . . . . .	499
14.3	<code>netrc</code> — netrc file processing . . . . .	518
14.4	<code>xdrlib</code> — Encode and decode XDR data . . . . .	519
14.5	<code>plistlib</code> — Generate and parse Mac OS X <code>.plist</code> files . . . . .	521
<b>15</b>	<b>Cryptographic Services</b>	<b>525</b>
15.1	<code>hashlib</code> — Secure hashes and message digests . . . . .	525
15.2	<code>hmac</code> — Keyed-Hashing for Message Authentication . . . . .	535
15.3	<code>secrets</code> — Generate secure random numbers for managing secrets . . . . .	537
<b>16</b>	<b>Generic Operating System Services</b>	<b>541</b>
16.1	<code>os</code> — Miscellaneous operating system interfaces . . . . .	541
16.2	<code>io</code> — Core tools for working with streams . . . . .	588
16.3	<code>time</code> — Time access and conversions . . . . .	600
16.4	<code>argparse</code> — Parser for command-line options, arguments and sub-commands . . . . .	610
16.5	<code>getopt</code> — C-style parser for command line options . . . . .	643
16.6	<code>logging</code> — Logging facility for Python . . . . .	645
16.7	<code>logging.config</code> — Logging configuration . . . . .	661
16.8	<code>logging.handlers</code> — Logging handlers . . . . .	672
16.9	<code>getpass</code> — Portable password input . . . . .	684
16.10	<code>curses</code> — Terminal handling for character-cell displays . . . . .	685
16.11	<code>curses.textpad</code> — Text input widget for curses programs . . . . .	703
16.12	<code>curses.ascii</code> — Utilities for ASCII characters . . . . .	704
16.13	<code>curses.panel</code> — A panel stack extension for curses . . . . .	707
16.14	<code>platform</code> — Access to underlying platform’s identifying data . . . . .	708
16.15	<code>errno</code> — Standard errno system symbols . . . . .	711
16.16	<code>ctypes</code> — A foreign function library for Python . . . . .	717
<b>17</b>	<b>Concurrent Execution</b>	<b>751</b>
17.1	<code>threading</code> — Thread-based parallelism . . . . .	751
17.2	<code>multiprocessing</code> — Process-based parallelism . . . . .	763
17.3	The concurrent package . . . . .	807
17.4	<code>concurrent.futures</code> — Launching parallel tasks . . . . .	807
17.5	<code>subprocess</code> — Subprocess management . . . . .	813
17.6	<code>sched</code> — Event scheduler . . . . .	831
17.7	<code>queue</code> — A synchronized queue class . . . . .	832
17.8	<code>_thread</code> — Low-level threading API . . . . .	836
17.9	<code>_dummy_thread</code> — Drop-in replacement for the <code>_thread</code> module . . . . .	838
17.10	<code>dummy_threading</code> — Drop-in replacement for the <code>threading</code> module . . . . .	838
<b>18</b>	<b><code>contextvars</code> — Context Variables</b>	<b>839</b>
18.1	Context Variables . . . . .	839
18.2	Manual Context Management . . . . .	840
18.3	asyncio support . . . . .	842
<b>19</b>	<b>Networking and Interprocess Communication</b>	<b>843</b>
19.1	<code>asyncio</code> — Asynchronous I/O . . . . .	843
19.2	<code>socket</code> — Low-level networking interface . . . . .	927

19.3	<code>ssl</code> — TLS/SSL wrapper for socket objects . . . . .	950
19.4	<code>select</code> — Waiting for I/O completion . . . . .	985
19.5	<code>selectors</code> — High-level I/O multiplexing . . . . .	992
19.6	<code>asyncore</code> — Asynchronous socket handler . . . . .	995
19.7	<code>asynchat</code> — Asynchronous socket command/response handler . . . . .	1000
19.8	<code>signal</code> — Set handlers for asynchronous events . . . . .	1002
19.9	<code>mmap</code> — Memory-mapped file support . . . . .	1009
<b>20</b>	<b>Internet Data Handling</b>	<b>1013</b>
20.1	<code>email</code> — An email and MIME handling package . . . . .	1013
20.2	<code>json</code> — JSON encoder and decoder . . . . .	1072
20.3	<code>mailcap</code> — Mailcap file handling . . . . .	1082
20.4	<code>mailbox</code> — Manipulate mailboxes in various formats . . . . .	1083
20.5	<code>mimetypes</code> — Map filenames to MIME types . . . . .	1100
20.6	<code>base64</code> — Base16, Base32, Base64, Base85 Data Encodings . . . . .	1103
20.7	<code>binhex</code> — Encode and decode binhex4 files . . . . .	1106
20.8	<code>binascii</code> — Convert between binary and ASCII . . . . .	1107
20.9	<code>quopri</code> — Encode and decode MIME quoted-printable data . . . . .	1109
20.10	<code>uu</code> — Encode and decode uuencode files . . . . .	1110
<b>21</b>	<b>Structured Markup Processing Tools</b>	<b>1111</b>
21.1	<code>html</code> — HyperText Markup Language support . . . . .	1111
21.2	<code>html.parser</code> — Simple HTML and XHTML parser . . . . .	1111
21.3	<code>html.entities</code> — Definitions of HTML general entities . . . . .	1116
21.4	XML Processing Modules . . . . .	1117
21.5	<code>xml.etree.ElementTree</code> — The ElementTree XML API . . . . .	1118
21.6	<code>xml.dom</code> — The Document Object Model API . . . . .	1134
21.7	<code>xml.dom.minidom</code> — Minimal DOM implementation . . . . .	1144
21.8	<code>xml.dom.pulldom</code> — Support for building partial DOM trees . . . . .	1149
21.9	<code>xml.sax</code> — Support for SAX2 parsers . . . . .	1151
21.10	<code>xml.sax.handler</code> — Base classes for SAX handlers . . . . .	1152
21.11	<code>xml.sax.saxutils</code> — SAX Utilities . . . . .	1157
21.12	<code>xml.sax.xmlreader</code> — Interface for XML parsers . . . . .	1158
21.13	<code>xml.parsers.expat</code> — Fast XML parsing using Expat . . . . .	1162
<b>22</b>	<b>Internet Protocols and Support</b>	<b>1173</b>
22.1	<code>webbrowser</code> — Convenient Web-browser controller . . . . .	1173
22.2	<code>cgi</code> — Common Gateway Interface support . . . . .	1175
22.3	<code>cgitb</code> — Traceback manager for CGI scripts . . . . .	1182
22.4	<code>wsgiref</code> — WSGI Utilities and Reference Implementation . . . . .	1183
22.5	<code>urllib</code> — URL handling modules . . . . .	1192
22.6	<code>urllib.request</code> — Extensible library for opening URLs . . . . .	1193
22.7	<code>urllib.response</code> — Response classes used by urllib . . . . .	1210
22.8	<code>urllib.parse</code> — Parse URLs into components . . . . .	1211
22.9	<code>urllib.error</code> — Exception classes raised by urllib.request . . . . .	1218
22.10	<code>urllib.robotparser</code> — Parser for robots.txt . . . . .	1218
22.11	<code>http</code> — HTTP modules . . . . .	1220
22.12	<code>http.client</code> — HTTP protocol client . . . . .	1222
22.13	<code>ftplib</code> — FTP protocol client . . . . .	1228
22.14	<code>poplib</code> — POP3 protocol client . . . . .	1234
22.15	<code>imaplib</code> — IMAP4 protocol client . . . . .	1236
22.16	<code>nntplib</code> — NNTP protocol client . . . . .	1243
22.17	<code>smtplib</code> — SMTP protocol client . . . . .	1250
22.18	<code>smtpd</code> — SMTP Server . . . . .	1256

22.19	<code>telnetlib</code> — Telnet client . . . . .	1260
22.20	<code>uuid</code> — UUID objects according to <b>RFC 4122</b> . . . . .	1262
22.21	<code>socketserver</code> — A framework for network servers . . . . .	1266
22.22	<code>http.server</code> — HTTP servers . . . . .	1274
22.23	<code>http.cookies</code> — HTTP state management . . . . .	1280
22.24	<code>http.cookiejar</code> — Cookie handling for HTTP clients . . . . .	1284
22.25	<code>xmlrpc</code> — XMLRPC server and client modules . . . . .	1292
22.26	<code>xmlrpc.client</code> — XML-RPC client access . . . . .	1292
22.27	<code>xmlrpc.server</code> — Basic XML-RPC servers . . . . .	1300
22.28	<code>ipaddress</code> — IPv4/IPv6 manipulation library . . . . .	1306
<b>23</b>	<b>Multimedia Services</b> . . . . .	<b>1321</b>
23.1	<code>audioop</code> — Manipulate raw audio data . . . . .	1321
23.2	<code>aifc</code> — Read and write AIFF and AIFC files . . . . .	1324
23.3	<code>sunau</code> — Read and write Sun AU files . . . . .	1326
23.4	<code>wave</code> — Read and write WAV files . . . . .	1329
23.5	<code>chunk</code> — Read IFF chunked data . . . . .	1332
23.6	<code>colorsys</code> — Conversions between color systems . . . . .	1333
23.7	<code>imghdr</code> — Determine the type of an image . . . . .	1334
23.8	<code>sndhdr</code> — Determine type of sound file . . . . .	1334
23.9	<code>ossaudiodev</code> — Access to OSS-compatible audio devices . . . . .	1335
<b>24</b>	<b>Internationalization</b> . . . . .	<b>1341</b>
24.1	<code>gettext</code> — Multilingual internationalization services . . . . .	1341
24.2	<code>locale</code> — Internationalization services . . . . .	1349
<b>25</b>	<b>Program Frameworks</b> . . . . .	<b>1357</b>
25.1	<code>turtle</code> — Turtle graphics . . . . .	1357
25.2	<code>cmd</code> — Support for line-oriented command interpreters . . . . .	1392
25.3	<code>shlex</code> — Simple lexical analysis . . . . .	1397
<b>26</b>	<b>Graphical User Interfaces with Tk</b> . . . . .	<b>1403</b>
26.1	<code>tkinter</code> — Python interface to Tcl/Tk . . . . .	1403
26.2	<code>tkinter.ttk</code> — Tk themed widgets . . . . .	1414
26.3	<code>tkinter.tix</code> — Extension widgets for Tk . . . . .	1433
26.4	<code>tkinter.scrolledtext</code> — Scrolled Text Widget . . . . .	1438
26.5	<code>IDLE</code> . . . . .	1438
26.6	Other Graphical User Interface Packages . . . . .	1449
<b>27</b>	<b>Development Tools</b> . . . . .	<b>1451</b>
27.1	<code>typing</code> — Support for type hints . . . . .	1451
27.2	<code>pydoc</code> — Documentation generator and online help system . . . . .	1467
27.3	<code>doctest</code> — Test interactive Python examples . . . . .	1468
27.4	<code>unittest</code> — Unit testing framework . . . . .	1493
27.5	<code>unittest.mock</code> — mock object library . . . . .	1521
27.6	<code>unittest.mock</code> — getting started . . . . .	1558
27.7	<code>2to3</code> - Automated Python 2 to 3 code translation . . . . .	1579
27.8	<code>test</code> — Regression tests package for Python . . . . .	1584
27.9	<code>test.support</code> — Utilities for the Python test suite . . . . .	1587
27.10	<code>test.support.script_helper</code> — Utilities for the Python execution tests . . . . .	1598
<b>28</b>	<b>Debugging and Profiling</b> . . . . .	<b>1601</b>
28.1	<code>bdb</code> — Debugger framework . . . . .	1601
28.2	<code>faulthandler</code> — Dump the Python traceback . . . . .	1605
28.3	<code>pdb</code> — The Python Debugger . . . . .	1607

28.4	The Python Profilers	1614
28.5	<code>timeit</code> — Measure execution time of small code snippets	1622
28.6	<code>trace</code> — Trace or track Python statement execution	1627
28.7	<code>tracemalloc</code> — Trace memory allocations	1630
<b>29</b>	<b>Software Packaging and Distribution</b>	<b>1641</b>
29.1	<code>distutils</code> — Building and installing Python modules	1641
29.2	<code>ensurepip</code> — Bootstrapping the <code>pip</code> installer	1641
29.3	<code>venv</code> — Creation of virtual environments	1643
29.4	<code>zipapp</code> — Manage executable Python zip archives	1652
<b>30</b>	<b>Python Runtime Services</b>	<b>1659</b>
30.1	<code>sys</code> — System-specific parameters and functions	1659
30.2	<code>sysconfig</code> — Provide access to Python’s configuration information	1676
30.3	<code>builtins</code> — Built-in objects	1680
30.4	<code>__main__</code> — Top-level script environment	1681
30.5	<code>warnings</code> — Warning control	1681
30.6	<code>dataclasses</code> — Data Classes	1687
30.7	<code>contextlib</code> — Utilities for <code>with</code> -statement contexts	1696
30.8	<code>abc</code> — Abstract Base Classes	1709
30.9	<code>atexit</code> — Exit handlers	1714
30.10	<code>traceback</code> — Print or retrieve a stack traceback	1715
30.11	<code>__future__</code> — Future statement definitions	1722
30.12	<code>gc</code> — Garbage Collector interface	1723
30.13	<code>inspect</code> — Inspect live objects	1726
30.14	<code>site</code> — Site-specific configuration hook	1742
<b>31</b>	<b>Custom Python Interpreters</b>	<b>1747</b>
31.1	<code>code</code> — Interpreter base classes	1747
31.2	<code>codeop</code> — Compile Python code	1749
<b>32</b>	<b>Importing Modules</b>	<b>1751</b>
32.1	<code>zipimport</code> — Import modules from Zip archives	1751
32.2	<code>pkgutil</code> — Package extension utility	1753
32.3	<code>modulefinder</code> — Find modules used by a script	1755
32.4	<code>runpy</code> — Locating and executing Python modules	1757
32.5	<code>importlib</code> — The implementation of <code>import</code>	1759
<b>33</b>	<b>Python Language Services</b>	<b>1781</b>
33.1	<code>parser</code> — Access Python parse trees	1781
33.2	<code>ast</code> — Abstract Syntax Trees	1785
33.3	<code>symtable</code> — Access to the compiler’s symbol tables	1791
33.4	<code>symbol</code> — Constants used with Python parse trees	1793
33.5	<code>token</code> — Constants used with Python parse trees	1793
33.6	<code>keyword</code> — Testing for Python keywords	1795
33.7	<code>tokenize</code> — Tokenizer for Python source	1795
33.8	<code>tabnanny</code> — Detection of ambiguous indentation	1799
33.9	<code>pyclbr</code> — Python class browser support	1800
33.10	<code>py_compile</code> — Compile Python source files	1801
33.11	<code>compileall</code> — Byte-compile Python libraries	1803
33.12	<code>dis</code> — Disassembler for Python bytecode	1806
33.13	<code>pickletools</code> — Tools for pickle developers	1819
<b>34</b>	<b>Miscellaneous Services</b>	<b>1821</b>
34.1	<code>formatter</code> — Generic output formatting	1821



<b>35 MS Windows Specific Services</b>	<b>1827</b>
35.1 <code>msilib</code> — Read and write Microsoft Installer files . . . . .	1827
35.2 <code>msvcrt</code> — Useful routines from the MS VC++ runtime . . . . .	1832
35.3 <code>winreg</code> — Windows registry access . . . . .	1834
35.4 <code>winsound</code> — Sound-playing interface for Windows . . . . .	1842
<b>36 Unix Specific Services</b>	<b>1845</b>
36.1 <code>posix</code> — The most common POSIX system calls . . . . .	1845
36.2 <code>pwd</code> — The password database . . . . .	1846
36.3 <code>spwd</code> — The shadow password database . . . . .	1847
36.4 <code>grp</code> — The group database . . . . .	1848
36.5 <code>crypt</code> — Function to check Unix passwords . . . . .	1848
36.6 <code>termios</code> — POSIX style tty control . . . . .	1850
36.7 <code>tty</code> — Terminal control functions . . . . .	1852
36.8 <code>pty</code> — Pseudo-terminal utilities . . . . .	1852
36.9 <code>fcntl</code> — The <code>fcntl</code> and <code>ioctl</code> system calls . . . . .	1853
36.10 <code>pipes</code> — Interface to shell pipelines . . . . .	1855
36.11 <code>resource</code> — Resource usage information . . . . .	1856
36.12 <code>nis</code> — Interface to Sun's NIS (Yellow Pages) . . . . .	1860
36.13 <code>syslog</code> — Unix syslog library routines . . . . .	1861
<b>37 Superseded Modules</b>	<b>1863</b>
37.1 <code>optparse</code> — Parser for command line options . . . . .	1863
37.2 <code>imp</code> — Access the import internals . . . . .	1890
<b>38 Undocumented Modules</b>	<b>1897</b>
38.1 Platform specific modules . . . . .	1897
<b>A Glossary</b>	<b>1899</b>
<b>B About these documents</b>	<b>1913</b>
B.1 Contributors to the Python Documentation . . . . .	1913
<b>C History and License</b>	<b>1915</b>
C.1 History of the software . . . . .	1915
C.2 Terms and conditions for accessing or otherwise using Python . . . . .	1916
C.3 Licenses and Acknowledgements for Incorporated Software . . . . .	1919
<b>D Copyright</b>	<b>1933</b>
<b>Bibliography</b>	<b>1935</b>
<b>Python Module Index</b>	<b>1937</b>
<b>Index</b>	<b>1941</b>



While `reference-index` 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](#).



## INTRODUCTION

The “Python library” contains several different kinds of components.

It contains data types that would normally be considered part of the “core” of a language, such as numbers and lists. For these types, the Python language core defines the form of literals and places some constraints on their semantics, but does not fully define the semantics. (On the other hand, the language core does define syntactic properties like the spelling and priorities of operators.)

The library also contains built-in functions and exceptions — objects that can be used by all Python code without the need of an `import` statement. Some of these are defined by the core language, but many are not essential for the core semantics and are only described here.

The bulk of the library, however, consists of a collection of modules. There are many ways to dissect this collection. Some modules are written in C and built in to the Python interpreter; others are written in Python and imported in source form. Some modules provide interfaces that are highly specific to Python, like printing a stack trace; some provide interfaces that are specific to particular operating systems, such as access to specific hardware; others provide interfaces that are specific to a particular application domain, like the World Wide Web. Some modules are available in all versions and ports of Python; others are only available when the underlying system supports or requires them; yet others are available only when a particular configuration option was chosen at the time when Python was compiled and installed.

This manual is organized “from the inside out:” it first describes the built-in functions, data types and exceptions, and finally the modules, grouped in chapters of related modules.

This means that if you start reading this manual from the start, and skip to the next chapter when you get bored, you will get a reasonable overview of the available modules and application areas that are supported by the Python library. Of course, you don’t *have* to read it like a novel — you can also browse the table of contents (in front of the manual), or look for a specific function, module or term in the index (in the back). And finally, if you enjoy learning about random subjects, you choose a random page number (see module *random*) and read a section or two. Regardless of the order in which you read the sections of this manual, it helps to start with chapter *Built-in Functions*, as the remainder of the manual assumes familiarity with this material.

Let the show begin!

### 1.1 Notes on availability

- An “Availability: Unix” note means that this function is commonly found on Unix systems. It does not make any claims about its existence on a specific operating system.
- If not separately noted, all functions that claim “Availability: Unix” are supported on Mac OS X, which builds on a Unix core.

