Appendix. New Features and Changes in Python 3.7 Through 3.11

The following tables enumerate language and standard library changes in Python versions 3.7 through 3.11 that are most likely to be found in Python code. Use these tables to plan your upgrade strategy, as constrained by your exposure to breaking changes in your codebase.

SETTINGS FOR ONLINE READING

To better view these tables you may need to reduce the font size in the settings of your device or content viewer.

The following types of changes are considered to be "breaking" and are marked with a **!** symbol in the last column:

- Introduces new keywords or built-ins (which may clash with names used in existing Python source code)
- Removes a method from a stdlib module or built-in type
- Changes a built-in or stdlib method signature in a way that is not backward-compatible (such as removing a parameter, or renaming a named parameter)

New warnings (including `DeprecatedWarning`) are also shown as "breaking," but marked with a * symbol in the last column.

Also see the table of proposed deprecations and removals from the standard library ("dead batteries") in PEP 594, which lists modules that are slated for deprecation or removal, the versions in which these changes are scheduled to be made (beginning with Python 3.12), and recommended replacements.

Python 3.7

The following table summarizes changes in Python version 3.7. For further details, see “What’s New in Python 3.7” in the **online docs**.

Python 3.7	**Added**	**Dep**
Functions accept > 255 arguments	+	
`argparse.ArgumentParser.parse_intermixed_args()`	+	
`ast.literal_eval()` no longer evaluates addition and subtraction		
`async` and **`await`** become reserved language keywords	+	
`asyncio.all_tasks()`, `asyncio.create_task()`, `asyncio.current_task()`, `asyncio.get_running_loop()`, `asyncio.Future.get_loop()`, `asyncio.Handle.cancelled()`, `asyncio.loop.sock_recv_into()`, `asyncio.loop.sock_sendfile()`, `asyncio.loop.start_tls()`, `asyncio.ReadTransport.is_reading()`, `asyncio.Server.is_serving()`, `asyncio.Server.get_loop()`, `asyncio.Task.get_loop()`, `asyncio.run()` (provisional)	+	
`asyncio.Server` is an async context manager	+	

Python 3.7	Added	Dep
`asyncio.loop.call_soon()`, `asyncio.loop.call_soon_threadsafe()`, `asyncio.loop.call_later()`, `asyncio.loop.call_at()`, and `asyncio.Future.add_done_callback()` all accept optional named `context` argument	+	
`asyncio.loop.create_server()`, `asyncio.loop.create_unix_server()`, `asyncio.Server.start_serving()`, and `asyncio.Server.serve_forever()` all accept optional named `start_serving` argument	+	
`asyncio.Task.current_task()` and `asyncio.Task.all_tasks()` are deprecated; use `asyncio.current_task()` and `asyncio.all_tasks()`		
`binascii.b2a_uu()` accepts named `backtick` argument	+	
`bool()` constructor no longer accepts a named argument (positional only)		
`breakpoint()` built-in function	+	
`bytearray.isascii()`	+	
`bytes.isascii()`	+	
`collections.namedtuple` supports default values	+	

Python 3.7	Added	Dep
`concurrent.Futures.ProcessPoolExecutor` and `concurrent.Futures.ThreadPoolExecutor` constructors accept optional `initializer` and `initargs` arguments	+	
`contextlib.AbstractAsyncContextManager`, `contextlib.asynccontextmanager()`, `contextlib.AsyncExitStack`, `contextlib.nullcontext()`	+	
`contextvars` module (similar to thread-local vars, with `asyncio` support)	+	
`dataclasses` module	+	
`datetime.datetime.fromisoformat()`	+	
`DeprecationWarning` shown by default in `__main__` module	+	
`dict` maintaining insertion order now guaranteed; `dict.popitem()` returns items in LIFO order	+	
`__dir__()` at module level	+	
`dis.dis()` method accepts named `depth` argument	+	
`float()` constructor no longer accepts a named argument (positional only)		
`fpectl` module removed		

Python 3.7	Added	Dep
from __future__ **import** annotations enables referencing as-yet-undefined types in type annotations without enclosing in quotes	+	
gc.freeze()	+	
__getattr__() at module level	+	
hmac.digest()	+	
http.client.HTTPConnection and http.client.HTTPSConnection constructors accept optional blocksize argument	+	
http.server.ThreadingHTTPServer	+	
importlib.abc.ResourceReader, importlib.resources module, importlib.source_hash()	+	
int() constructor no longer accepts a named *x* argument (positional only; named base argument is still supported)		
io.TextIOWrapper.reconfigure()	+	
ipaddress.IPv*Network.subnet_of(), ipaddress.IPv*Network.supernet_of()	+	
list() constructor no longer accepts a named argument (positional only)		

Python 3.7	**Added**	**Dep**
`logging.StreamHandler.setStream()`	+	
`math.remainder()`	+	
`multiprocessing.Process.close()`, `multiprocessing.Process.kill()`	+	
`ntpath.splitunc()` removed; use `ntpath.splitdrive()`		
`os.preadv()`, `os.pwritev()`, `os.register_at_fork()`	+	
`os.stat_float_times()` removed (compatibility function with Python 2; all timestamps in `stat` result are `float`s in Python 3)		
`pathlib.Path.is_mount()`	+	
`pdb.set_trace()` accepts named `header` argument	+	
`plist.Dict`, `plist.Plist`, and `plist._InternalDict` removed		
`queue.SimpleQueue`	+	
`re` compiled expressions and match objects can be copied with `copy.copy` and `copy.deepcopy`	+	
`re.sub()` no longer supports unknown escapes of \ and an ASCII letter		

Python 3.7	Added	Dep
socket.close(), socket.getblocking(), socket.TCP_CONGESTION, socket.TCP_USER_TIMEOUT, socket.TCP_NOTSENT_LOWAT (Linux platforms only)	+	
sqlite3.Connection.backup()	+	
StopIteration handling in generators	+	
str.isascii()	+	
subprocess.run() named argument capture_output=**True** for simplified stdin/stdout capture	+	
subprocess.run() and subprocess.Popen() named argument text, alias for universal_newlines	+	
subprocess.run(), subprocess.call(), and subprocess.Popen() improved KeyboardInterrupt handling	+	
sys.breakpointhook(), sys.getandroidapilevel(), sys.get_coroutine_origin_tracking_depth(), sys.set_coroutine_origin_tracking_depth()	+	
time.clock_gettime_ns(), time.clock_settime_ns(), time.monotonic_ns(), time.perf_counter_ns(), time.process_time_ns(), time.time_ns(),	+	

Python 3.7	Added	Dep
`time.CLOCK_BOOTTIME`, `time.CLOCK_PROF`, `time.CLOCK_UPTIME`		
`time.thread_time()` and `time.thread_time_ns()` for per-thread CPU timing	+	
`tkinter.ttk.Spinbox`	+	
`tuple()` constructor no longer accepts a named argument (positional only)		
`types.ClassMethodDescriptorType`, `types.MethodDescriptorType`, `types.MethodWrapperType`, `types.WrapperDescriptorType`	+	
`types.resolve_bases()`	+	
`uuid.UUID.is_safe`	+	
`yield` and **`yield from`** in comprehensions or generator expressions are deprecated		
`zipfile.ZipFile` constructor accepts named `compresslevel` argument	+	

Python 3.8

The following table summarizes changes in Python version 3.8. For further details, see “What’s New in Python 3.8” in the **online docs**.

Python 3.8	**Added**	**Deprecated**
Assignment expressions (`:=` “walrus” operator)	+	
Positional-only and named-only parameters (/ and * arg separators)	+	
F-string trailing = for debugging	+	
`is` and **`is not`** tests against `str` and `int` literals emit `SyntaxWarning`		
`ast` AST nodes `end_lineno` and `end_col_offset` attributes	+	
`ast.get_source_segment()`	+	
`ast.parse()` accepts named arguments `type_comments`, `mode`, and `feature_version`	+	
`async` REPL can be run using **`python -m asyncio`**	+	
`asyncio` tasks can be named	+	
`asyncio.coroutine` decorator deprecated		—
`asyncio.run()` to execute a coroutine directly	+	
`asyncio.Task.get_coro()`	+	

Python 3.8	**Added**	**Deprecated**
`bool.as_integer_ratio()`	+	
`collections.namedtuple._asdict()` returns `dict` instead of `OrderedDict`	+	
`continue` permitted in **`finally`** block	+	

`cgi.parse_qs`, `cgi.parse_qsl`, and `cgi.escape` removed;
import from `urllib.parse` and `html` modules		

`csv.DictReader` returns `dicts` instead of `OrderedDicts`	+	
`datetime.date.fromisocalendar()`, `datetime.datetime.fromisocalendar()`	+	
`dict` comprehensions compute key first, value second		
`dict` and `dictviews` returned from `dict.keys()`, `dict.values()` and `dict.items()` now iterable with `reversed()`	+	
`fractions.Fraction.as_integer_ratio()`	+	
`functools.cached_property()` decorator (see cautionary notes **here** and **here**)	+	

Python 3.8	**Added**	**Deprecated**
`functools.lru_cache` can be used as a decorator without `()`	+	
`functools.singledispatchmethod` decorator	+	
`gettext.pgettext()`	+	
`importlib.metadata` module	+	
`int.as_integer_ratio()`	+	
`itertools.accumulate()` accepts named `initial` argument	+	
`macpath` module removed		
`math.comb()`, `math.dist()`, `math.isqrt()`, `math.perm()`, `math.prod()`	+	
`math.hypot()` added support for > 2 dimensions	+	
`multiprocessing.shared_memory` module	+	
`namedtuple._asdict()` returns `dict` instead of `OrderedDict`	+	
`os.add_dll_directory()` on Windows	+	
`os.memfd_create()`	+	

Python 3.8	**Added**	**Deprecated**
`pathlib.Path.link_to()`	+	
`platform.popen()` removed; use `os.popen()`		
`pprint.pp()`	+	
pyvenv script removed; use **`python -m venv`**		
`re` regular expression patterns support `\N{`*`name`*`}` escapes	+	
`shlex.join()` (inverse of `shlex.split()`)	+	
`shutil.copytree()` accepts named `dirs_exist_ok` argument	+	
`__slots__` accepts a `dict` of `{`*`name`*`:` *`docstring`*`}`	+	
`socket.create_server()`, `socket.has_dualstack_ipv6()`	+	
`socket.if_nameindex()`, `socket.if_nametoindex()`, and `socket.if_indextoname()` are all supported on Windows	+	
`sqlite3` `Cache` and `Statement` objects no longer user-visible		

Python 3.8	Added	Deprecated
`ssl.post_handshake_auth()`, `ssl.verify_client_post_handshake()`	+	
`statistics.fmean()`, `statistics.geometric_mean()`, `statistics.multimode()`, `statistics.NormalDist`, `statistics.quantiles()`	+	
`sys.get_coroutine_wrapper()` and `sys.set_coroutine_wrapper()` removed		
`sys.unraisablehook()`	+	
`tarfile.filemode()` removed		
`threading.excepthook()`, `threading.get_native_id()`, `threading.Thread.native_id`	+	
`time.clock()` removed; use `time.perf_counter()`		
`tkinter.Canvas.moveto()`, `tkinter.PhotoImage.transparency_get()`, `tkinter.PhotoImage.transparency_set()`, `tkinter.Spinbox.selection_from()`, `tkinter.Spinbox.selection_present()`, `tkinter.Spinbox.selection_range()`, `tkinter.Spinbox.selection_to()`	+	

Python 3.8	**Added**	**Deprecated**
`typing.Final`, `typing.get_args()`, `typing.get_origin()`, `typing.Literal`, `typing.Protocol`, `typing.SupportsIndex`, `typing.TypedDict`	+	
`typing.NamedTuple._field_types` deprecated		—
`unicodedata.is_normalized()`	+	
`unittest` supports coroutines as test cases	+	
`unittest.addClassCleanup()`, `unittest.addModuleCleanup()`, `unittest.AsyncMock`	+	
`xml.etree.Element.getchildren()`, `xml.etree.Element.getiterator()`, `xml.etree.ElementTree.getchildren()`, and `xml.etree.ElementTree.getiterator()` deprecated		—
`XMLParser.doctype()` removed		
`xmlrpc.client.ServerProxy` accepts named `headers` argument	+	
`yield` and **`return`** unpacking no longer requires enclosing parentheses	+	
`yield` and **`yield from`** no longer allowed in comprehensions or generator expressions		

Python 3.9

The following table summarizes changes in Python version 3.9. For further details, see “What’s New in Python 3.9” in the online docs.

Python 3.9	Added	Deprecated
Type annotations can now use built-in types in generics (e.g., `list[int]` instead of `List[int]`)	+	
`array.array.tostring()` and `array.array.fromstring()` removed; use `tobytes()` and `frombytes()`		
`ast.unparse()`	+	
`asyncio.loop.create_datagram_endpoint()` argument `reuse_address` disabled		
`asyncio.PidfdChild Watcher`, `asyncio.shutdown_default_executor()`, `asyncio.to_thread()`	+	
`asyncio.Task.all_asks` removed; use `asyncio.all_tasks()`		
`asyncio.Task.current_task` removed; use `asyncio.current_task()`		
`base64.encodestring()` and `base64.decodestring()` removed; use `base64.encodebytes()` and `base64.decodebytes()`		

Python 3.9	Added	Deprecated
`concurrent.futures.Executor.shutdown()` accepts named `cancel_futures` argument	+	
`curses.get_escdelay()`, `curses.get_tabsize()`, `curses.set_escdelay()`, `curses.set_tabsize()`	+	
`dict` supports union operators | and `|=`	+	
`fcntl.F_OFD_GETLK`, `fcntl.F_OFD_SETLK`, `fcntl.F_OFD_SETKLW`	+	
`fractions.gcd()` removed; use `math.gcd()`		
`functools.cache()` (lightweight/faster version of `lru_cache`)	+	
`gc.is_finalized()`	+	
`graphlib` module with `TopologicalSorter` class	+	
`html.parser.HTMLParser.unescape()` removed		
`imaplib.IMAP4.unselect()`	+	
`importlib.resources.files()`	+	

Python 3.9	**Added**	**Deprecated**
`inspect.BoundArguments.arguments` returns `dict` instead of `OrderedDict`	+	
`ipaddress` module does not accept leading zeros in IPv4 address strings		
`logging.getLogger('root')` returns the root logger	+	
`math.gcd()` accepts multiple arguments	+	
`math.lcm()`, `math.nextafter()`, `math.ulp()`	+	
`multiprocessing.SimpleQueue.close()`	+	
`nntplib.NNTP.xpath()` and `nntplib.xgtitle()` removed		
`os.pidfd_open()`	+	
`os.unsetenv()` available on Windows	+	
`os.waitstatus_to_exitcode()`	+	
`parser` module deprecated		—
`pathlib.Path.readlink()`	+	
`plistlib` API removed		
`pprint` supports `types.SimpleNamespace`	+	

Python 3.9	Added	Deprecated
`random.choices()` with `weights` argument raises `ValueError` if `weights` are all `0`		
`random.Random.randbytes()`	+	
`socket.CAN_RAW_JOIN_FILTERS`, `socket.send_fds()`, `socket.recv_fds()`	+	
`str.removeprefix()`, `str.removesuffix()`	+	
`symbol` module deprecated		—
`sys.callstats()`, `sys.getcheckinterval()`, `sys.getcounts()`, and `sys.setcheckinterval()` removed		
`sys.getcheckinterval()` and `sys.setcheckinterval()` removed; use `sys.getswitchinterval()` and `sys.setswitchinterval()`		
`sys.platlibdir` attribute	+	
`threading.Thread.isAlive()` removed; use `threading.Thread.is_alive()`		
`tracemalloc.reset_peak()`	+	
`typing.Annotated` type	+	
`typing.Literal` deduplicates values; equality matching is order independent		

Python 3.9	Added	Deprecated
(3.9.1)		
`typing.NamedTuple._field_types` removed; use `__annotations__`		
`urllib.parse.parse_qs()` and `urllib.parse.parse_qsl()` accept `;` or `&` query parameter separator, but not both (3.9.2)		
`urllib.parse.urlparse()` changed handling of numeric paths; a string like `'path:80'` is no longer parsed as a path but as a scheme (`'path'`) and a path (`'80'`)		
`with` `(`**`await`** `asyncio.Condition)` and **`with`** `(`**`yield from`** `asyncio.Condition)` removed; use **`async with`** `condition`		
`with` `(`**`await`** `asyncio.lock)` and **`with`** `(`**`yield from`** `asyncio.lock)` removed; use **`async with`** `lock`		
`with` `(`**`await`** `asyncio.Semaphore)` and **`with`** `(`**`yield from`** `asyncio.Semaphore)` removed; use **`async with`** `semaphore`		
`xml.etree.Element.getchildren()`, `xml.etree.Element.getiterator()`, `xml.etree.ElementTree.getchildren()`, and		

Python 3.9	**Added**	**Deprecated**
`xml.etree.ElementTree.getiterator()` removed		
`zoneinfo` module for IANA time zone support	+	

Python 3.10

The following table summarizes changes in Python version 3.10. For further details, see “What’s New in Python 3.10” in the **online docs**.

Python 3.10	**Added**	**Deprecat**
Building requires OpenSSL 1.1.1 or newer	+	
Debugging improved with precise line numbers	+	
Structural pattern matching using **`match`**, **`case`**, and **`_`** soft keywords[a]	+	
`aiter()` and `anext()` built-ins	+	
`array.array.index()` accepts optional arguments `start` and `stop`	+	
`ast.literal_eval(`*`s`*`)` strips leading spaces and tabs from input string *s*	+	
`asynchat` module deprecated		—

Python 3.10	Added	Deprecat
asyncio functions remove loop parameter		
asyncio.connect_accepted_socket()	+	
asyncore module deprecated		—
base64.b32hexdecode, base64.b32hexencode	+	
bdb.clearBreakpoints()	+	
bisect.bisect, bisect.bisect_left, bisect.bisect_right, bisect.insort, bisect.insort_left, and bisect.insert_right all accept optional key argument	+	
cgi.log deprecated		—
codecs.unregister()	+	
collections module compatibility definitions of ABCs removed; use collections.abc		
collections.Counter.total()	+	
contextlib.aclosing() decorator, contextlib.AsyncContextDecorator	+	
curses.has_extended_color_support()	+	
dataclasses.dataclass() decorator accepts optional slots argument	+	

Python 3.10	**Added**	**Deprecat**
`dataclasses.KW_ONLY`	+	
`distutils` deprecated, to be removed in Python 3.12		—
`enum.StrEnum`	+	
`fileinput.input()` and `fileinput.FileInput` accept optional `encoding` and `errors` arguments	+	
`formatter` module removed		
`glob.glob()` and `glob.iglob()` accept optional `root_dir` and `dir_fd` arguments to specify root search directory	+	
`importlib.metadata.package_distributions()`	+	
`inspect.get_annotations()`	+	
`int.bit_count()`	+	
`isinstance(obj, (atype, btype))` can be written `isinstance(obj, atype|btype)`	+	
`issubclass(cls, (atype, btype))` can be written `issubclass(cls, atype|btype)`	+	
`itertools.pairwise()`	+	
`os.eventfd()`, `os.splice()`	+	

Python 3.10	Added	Deprecat
`os.path.realpath()` accepts optional `strict` argument	+	
`os.EVTONLY`, `os.O_FSYNC`, `os.O_SYMLINK`, and `os.O_NOFOLLOW_ANY` all added on macOS	+	
`parser` module removed		
`pathlib.Path.chmod()` and `pathlib.Path.stat()` accept optional `follow_symlinks` keyword argument	+	
`pathlib.Path.hardlink_to()`	+	
`pathlib.Path.link_to()` deprecated; use `hardlink_to()`		—
`platform.freedesktop_os_release()`	+	
`pprint.pprint()` accepts optional `underscore_numbers` keyword argument	+	
`smtpd` module deprecated		—
`ssl.get_server_certificate` accepts optional `timeout` argument	+	
`statistics.correlation()`, `statistics.covariance()`, `statistics.linear_regression()`	+	

Python 3.10	Added	Deprecat
SyntaxError.end_line_no and SyntaxError.end_offset attributes	+	
sys.flags.warn_default_encoding to emit EncodingWarning	+	
sys.orig_argv and sys.stdlib_module_names attributes	+	
threading.__excepthook__	+	
threading.getprofile(), threading.gettrace()	+	
threading.Thread appends '(<target.__name__>)' to generated thread names	+	
traceback.format_exception(), traceback.format_exception_only(), and traceback.print_exception() signature changes		
types.EllipsisType, types.NoneType, types.NotImplementedType	+	
typing module includes parameter specification variables for specifying Callable types	+	
typing.io module deprecated; use typing		—
typing.is_typeddict()	+	

Python 3.10	Added	Deprecat
`typing.Literal` deduplicates values; equality matching is order-independent		
`typing.Optional[X]` can be written as `X | None`	+	
`typing.re` module deprecated; use `typing`		—
`typing.TypeAlias` for defining explicit type aliases	+	
`typing.TypeGuard`	+	
`typing.Union[X, Y]` can use | operator as `X | Y`	+	
`unittest.assertNoLogs()`	+	
`urllib.parse.parse_qs()` and `urllib.parse.parse_qsl()` accept `;` or `&` query parameter separator, but not both		
`with` statement accepts parenthesized context managers: **`with`**`(ctxmgr, ctxmgr, ...)`	+	
`xml.sax.handler.LexicalHandler`	+	
`zip` built-in accepts optional `strict` named argument for length-checking	+	

Python 3.10	Added	Deprecat
`zipimport.find_spec()`, `zipimport.zipimporter.create_module()`, `zipimport.zipimporter.exec_module()`, `zipimport.zipimporter.invalidate_caches()`	+	

[a] Since these are defined as *soft* keywords, they do not break existing code using

Python 3.11

The following table summarizes changes in Python version 3.11. For further details, see "What's New in Python 3.11" in the online docs.

Python 3.11	Added
Security patch released in Python 3.11.0 and backported to versions 3.7–3.10: `int` conversion to `str` and `str` conversion to `int` in bases other than 2, 4, 8, 16, or 32 raises `ValueError` if the resulting string > 4,300 digits (addresses CVE-2020-10735)	
General performance improvements	+
Improved error messages	+
New syntax: **`for`** *`x`* **`in`** `*`*`values`*	+
`aifc` module deprecated	

Python 3.11	Added
`asynchat` and `asyncore` modules deprecated	
`asyncio.Barrier`, `asyncio.start_tls()`, `asyncio.TaskGroup`	+
`asyncio.coroutine` decorator removed	
`asyncio.loop.create_datagram_endpoint()` argument `reuse_address` removed	
`asyncio.TimeoutError` deprecated; use `TimeoutError`	
`audioop` module deprecated	
`BaseException.add_note()`, `BaseException.__notes__` attribute	+
`binascii.a2b_hqx()`, `binascii.b2a_hqx()`, `binascii.rlecode_hqx()`, and `binascii.rledecode_hqx()` removed	
`binhex` module removed	
`cgi` and `cgitb` modules deprecated	
`chunk` module deprecated	
`concurrent.futures.ProcessPoolExecutor()` `max_tasks_per_child` argument	+

Python 3.11	**Added**
`concurrent.futures.TimeoutError` deprecated; use built-in `TimeoutError`	
`contextlib.chdir` context manager (change current working dir and then restore it)	+
`crypt` module deprecated	
`dataclasses` check for mutable defaults disallows any value that is not hashable (formerly allowed any value that was not a `dict`, `list`, or `set`)	
`datetime.UTC` as a convenience alias for `datetime.timezone.utc`	+
`enum.Enum` `str()` output just gives name	+
`enum.EnumCheck`, `enum.FlagBoundary`, `enum.global_enum()` decorator, `enum.member()` decorator, `enum.nonmember()` decorator, `enum.property`, `enum.ReprEnum`, `enum.StrEnum`, and `enum.verify()`	+
`ExceptionGroups` and `except*`	+
`fractions.Fraction` initialization from string	+
`gettext.l*gettext()` methods removed	
`glob.glob()` and `glob.iglob()` accept optional `include_hidden` argument	+

Python 3.11	Added
`hashlib.file_digest()`	+
`imghdr` module deprecated	
`inspect.formatargspec()` and `inspect.getargspec()` removed; use `inspect.signature()`	
`inspect.getmembers_static()`, `inspect.ismethodwrapper()`	+
`locale.getdefaultlocale()` and `locale.resetlocale()` deprecated	
`locale.getencoding()`	+
`logging.getLevelNamesMapping()`	+
`mailcap` module deprecated	
`math.cbrt()` (cube root), `math.exp2()` (computes 2^n)	+
`msilib` module deprecated	
`nis` module deprecated	
`nntplib` module deprecated	
`operator.call`	+
`ossaudiodev` module deprecated	

Python 3.11	**Added**
pipes module deprecated	
re pattern syntax supports *+, ++, ?+, and {m,n}+ possessive quantifiers, and (?>...) atomic grouping	+
re.template() deprecated	
smtpd module deprecated	
sndhdr module deprecated	
spwd module deprecated	
sqlite3.Connection.blobopen(), sqlite3.Connection.create_window_function(), sqlite3.Connection.deserialize(), sqlite3.Connection.getlimit(), sqlite3.Connection.serialize(), sqlite3.Connection.setlimit()	+
sre_compile, sre_constants, and sre_parse deprecated	
statistics.fmean() optional weights argument	+
sunau module deprecated	
sys.exception() (equivalent to sys.exc_info()[1])	+
telnetlib module deprecated	
time.nanosleep() (Unix-like systems only)	+

Python 3.11	**Added**
`tomllib` TOML parser module	+
`typing.assert_never()`, `typing.assert_type()`, `typing.LiteralString`, `typing.Never`, `typing.reveal_type()`, `typing.Self`	+
`typing.Text` deprecated; use `str`	
`typing.TypedDict` items can be marked as `Required` or `NotRequired`	+
`typing.TypedDict(a=int, b=str)` form deprecated	
`unicodedata` updated to Unicode 14.0.0	+
`unittest.enterModuleContext()`, `unittest.IsolatedAsyncioTestCase.enterAsyncContext()`, `unittest.TestCase.enterClassContext()`, `unittest.TestCase.enterContext()`	+
`unittest.findTestCases()`, `unittest.getTestCaseName()`, and `unittest.makeSuite()` deprecated; use methods of `unittest.TestLoader`	
`uu` module deprecated	
`with` statement now raises `TypeError` instead of `AttributeError` for objects that do not support the context manager protocol	

Python 3.11	Added
`xdrlib` module deprecated	
`z` string format specifier added, for negative sign of values close to zero	+
`zipfile.ZipFile.mkdir()` added	+

Add your own notes here: