

The `__class__` cell used by zero-argument `super()` is now initialized from type, `__new__` rather than `__build_class__`, so class methods relying on that will now work correctly when called from metaclass methods during class creation. Patch by Martin Teichmann.

Fix corrupted result from `PyLong_FromLong(0)` when Python is compiled with `NSMALLPOSINTS = 0`.

Implement formatting support for [PEP 515](#). Initial patch by Chris Angelico.

In `tarfile`, expose `copyfileobj` bufsize to improve throughput. Patch by Jason Fried.

In f-strings, only allow backslashes inside the braces (where the expressions are). This is a breaking change from the 3.6 alpha releases, where backslashes are allowed anywhere in an f-string. Also, require that expressions inside f-strings be enclosed within literal braces, and not escapes like `f'\x7b"hi"\x7d'`.

Remove platform-specific directories from `sys.path`.

Add early-out for differencing from an empty set.

Prevents `zipimport` from unnecessarily encoding a filename (patch by Eryk Sun)

The `__module__` attribute of extension classes and functions now is interned. This leads to more compact pickle data with protocol 4.

Rework `CALL_FUNCTION*` opcodes to produce shorter and more efficient bytecode. Patch by Demur Rumed, design by Serhiy Storchaka, reviewed by Serhiy Storchaka and Victor Stinner.

Implement tokenizing support for [PEP 515](#). Patch by Georg Brandl.

Make "global after use" a `SyntaxError`, and ditto for `nonlocal`. Patch by Ivan Levkivskiy.

Implement [PEP 525](#) -- Asynchronous Generators.

Implement [PEP 526](#) -- Syntax for Variable Annotations. Patch by Ivan Levkivskiy.

Add a new private version to the builtin dict type, incremented at each dictionary creation and at each dictionary change.

Implementation of the [PEP 509](#).

A backslash-character pair that is not a valid escape sequence now generates a `DeprecationWarning`. Patch by Emanuel Barry.

`dict` implementation is changed like PyPy. It is more compact and preserves insertion order. (Concept developed by Raymond Hettinger and patch by Inada Naoki.)

Remove unnecessary error checks in `exec_builtin_or_dynamic()`.

Added `BUILD_STRING` opcode. Optimized f-strings evaluation.

Python now requires systems with `inttypes.h` and `stdint.h`

Require platforms to support `long long`. Python hasn't compiled without `long long` for years, so this is basically a formality.

Removed support for Windows CE. It was never finished, and Windows CE is no longer a relevant platform for Python.

Implement [PEP 523](#).

A left shift of zero by a large integer no longer attempts to allocate large amounts of memory.

In int-to-decimal-string conversion, improve the estimate of the intermediate memory required, and remove an unnecessarily strict overflow check. Patch by Serhiy Storchaka.

In `long_invert`, be more careful about modifying object returned by `long_add`, and remove an unnecessary check for small longs. Thanks Oren Milman for analysis and patch.

Support passing the `bytes/bytearray.translate()` "delete" argument by keyword.

Properly clear out a generator's frame's backreference to the generator to prevent crashes in `frame.clear()`.

Fix a crash when a coroutine that has not been awaited is finalized with `warnings-as-errors` enabled.

Fix another issue found by PVS-Studio: Null pointer check after use of `'def' in _PyState_AddModule()`. Initial patch by Christian Heimes.

The modulo operation applied to `bool` and other `int` subclasses now always returns an `int`. Previously the return type depended on the input values. Patch by Xiang Zhang.

`int()` now always returns an instance of exact `int`.

Fix a minor bug in integer true division; this bug could potentially have caused off-by-one-ulp results on platforms with unreliable `ldexp` implementations.

Make class definition namespace ordered by default.

Fix an overflow check in `List_New`: the original code was checking against `Py_SIZE_MAX` instead of the correct upper bound of `Py_SSIZE_T_MAX`. Patch by Xiang Zhang.

Multi-phase extension module `import` now correctly allows the `m_methods` field to be used to add module level functions to instances of non-module types returned from `Py_create_mod`. Patch by Xiang Zhang.

The `round()` function accepted a second `None` argument for some types but not for others. Fixed the inconsistency by accepting `None` for all numeric types.

Warn if a submodule argument to `"python -m"` or `runpy.run_module()` is found in `sys.modules` after parent packages are imported, but before the submodule is executed.

Make only `type()` itself accept the one-argument form. Patch by Eryk Sun and Emanuel Barry.

Fix a `SystemError` in the implementation of `"raise"` statement. In a brand new thread, raise a `RuntimeError` since there is no active exception to reraise. Patch written by Xiang Zhang.

Implement [PEP 530](#) -- asynchronous comprehensions.

Fix memory leak in `codeobject.c`

Fix crash in `os.spawnv()` with no elements in `args`

Always raise `ValueError` for negative `compileall.compile_dir(workers=...)` parameter, even when multithreading is unavailable.

Use `sqlite3._get_autocommit()` instead of setting `Connection->inTransaction` manually.

Attributes `tm_gmtoff` and `tm_zone` are now available on all platforms in the return values of `time.localtime()` and `time.gmtime()`.

Regular expression match object groups are now accessible using `__getitem__`. `"mo[x]"` is equivalent to `"mo.group(x)"`.

`sqlite3` no longer implicitly commit an open transaction before DDL statements.

Add a `module` parameter to `collections.namedtuple()`.

Inline flags now should be used only at the start of the regular expression. Deprecation warning is emitted if uses them in the middle of the regular expression.

`xmlrpc` now supports unmarshalling additional data types used by Apache XML-RPC implementation for numerics and `None`.

Fixed parsing inline verbose flag in regular expressions.

Add client-side SSL session resumption to the `ssl` module.

Deprecate `ssl`-related arguments in favor of `SSLContext`. The deprecation include manual creation of `SSLSocket` and `certfile/keyfile` (or similar) in `ftplib`, `httplib`, `imaplib`, `smtplib`, `poplib` and `urllib`.

`SSLContext` has improved default settings: `OP_NO_SSLv2`, `OP_NO_SSLv3`, `OP_NO_COMPRESSION`, `OP_CIPHER_SERVER_PREFERENCE`, `OP_SINGLE_DH_USE`, `OP_SINGLE_ECDH_USE` and HIGH ciphers without MD5.

Changed some `RuntimeError`'s in the `zipfile` module to more appropriate types. Improved some error messages and debugging output.

`json.load` and `json.loads` now support binary input encoded as UTF-8, UTF-16 or UTF-32. Patch by Serhiy Storchaka.

the pure Python fallback implementation of `functools.partial` now matches the behaviour of its accelerated C counterpart for subclassing, pickling and text representation purposes. Patch by Emanuel Barry and Serhiy Storchaka.

Fix possible integer overflows and crashes in the `mmap` module with unusual usage patterns.

Fix the ability to pass the `--link-objects` option to the `distutils build_ext` command.

`itertools.count()` no longer rounds non-integer step in range between 1.0 and 2.0 to 1.

`Pdb` now supports the `'readrc'` keyword argument to control whether `.pdbrc` files should be read. Patch by Martin Matusiak and Sam Kimbrel.

Update the `lib2to3` grammar to handle the unpacking generalizations added in 3.5.

`mailcap` now respects the order of the lines in the `mailcap` files ("first match"), as required by RFC 1542. Patch by Michael Lazar.

Convert `re` flag constants to `IntFlag`.

Convert all `ssl` module constants to `IntEnum` and `IntFlags`. `SSLContext` properties now return flags and enums.

Add `Flag`, `IntFlag`, and `auto()` to `enum` module.

Added support of modifier spans in regular expressions.

Validates `persist` parameter when opening MSI database

`xml.etree.ElementTree` now preserves whitespaces in attributes (Patch by Duane Griffin. Reviewed and approved by Stefan Behnel.)

Fixed calculation of line length used for the base64 CTE in the new email policies.

Fix call order in `OrderedDict.__init__()`.

`email.generator.DecodedGenerator` now supports the `policy` keyword.

Remove undocumented modules from `Lib/plat-*`: `IN`, `CDROM`, `DLFCN`, `TYPES`, `CDIO`, and `STROPTS`.

Don't pass `str(_charset)` to `MIMEText.set_payload()`. Patch by Claude Paroz.

The new email API is no longer provisional, and the docs have been reorganized and rewritten to emphasize the new API.

urllib now includes an `Accept: */*` header among the default headers. This makes the results of REST API requests more consistent and predictable especially when proxy servers are involved.

`lib2to3.pgen3.driver.load_grammar()` now creates a stable cache file between runs given the same `Grammar.txt` input regardless of the hash randomization setting.

Allow `ImportErrors` in encoding implementation to propagate.

Support path-like objects in `importlib.util`.

Avoid zero-length `memcpy()` etc calls with null source pointers in the `"ctypes"` and `"array"` modules.

Break email header lines *only* on the RFC specified CR and LF characters, not on arbitrary unicode line breaks. This also fixes a bug in HTTP header parsing.

The `email.mime` classes now all accept an optional policy keyword.

Fix email `iter_attachments` incorrect mutation of payload list.

Add SHA-3 and SHAKE support to `hashlib` module.

Eliminate a tautological-pointer-compare warning in `_scproxy.c`.

The `:func:'os.urandom'` function does now block on Linux 3.17 and newer until the system `urandom` entropy pool is initialized to increase the security. This change is part of the [PEP 524](#).

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\ [cpython-main] [Misc] [NEWS.d] 3.6.0b1.rst, line 861); [backlink](#)**

Unknown interpreted text role "func".

Expose the Linux `getrandom()` syscall as a new `:func:'os.getrandom'` function. This change is part of the [PEP 524](#).

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\ [cpython-main] [Misc] [NEWS.d] 3.6.0b1.rst, line 872); [backlink](#)**

Unknown interpreted text role "func".

Fix `ssl` module's parsing of `GEN_RID` subject alternative name fields in X.509 certs.

Add `random.choices()`.

Improved error reporting about truncated pickle data in C implementation of unpickler. `UnpicklingError` is now raised instead of `AttributeError` and `ValueError` in some cases.

Add BLAKE2 (blake2b and blake2s) to `hashlib`.

Optimized globbing in `pathlib` by using `os.scandir()`; it is now about 1.5--4 times faster.

Optimized `glob()` and `iglob()` functions in the `glob` module; they are now about 3--6 times faster.

Add `sCrypt` (password-based key derivation function) to `hashlib` module (requires OpenSSL 1.1.0).

Remove 3DES from `ssl` module's default cipher list to counter measure sweet32 attack (CVE-2016-2183).

Add ChaCha20 Poly1305 to `ssl` module's default cipher list. (Required OpenSSL 1.1.0 or LibreSSL).

Check return value of `winsound.MessageBeep`.

Add `SSLContext.get_ciphers()` method to get a list of all enabled ciphers.

Add `AF_ALG` (Linux Kernel crypto) to `socket` module.

Port `ssl` and `hashlib` module to OpenSSL 1.1.0.

Fix support for `SND_MEMORY` in `winsound.PlaySound`. Based on a patch by Tim Leshner.

Add support for IEEE 754 half-precision floats to the `struct` module. Based on a patch by Eli Stevens.

Deprecated `extra_path` distribution option in `distutils` packaging.

Add new `cmath` constants: `cmath.inf` and `cmath.nan` to match `math.inf` and `math.nan`, and also `cmath.infj` and `cmath.nanj` to match the format used by complex repr.

The `csv.DictReader` now returns rows of type `OrderedDict`. (Contributed by Steve Holden.)

Remove support for passing a file descriptor to `os.access`. It never worked but previously didn't raise.

Fix error when `distutils` encounters symlink.

Fixed possible bugs when setting `sqlite3.Connection.isolation_level`. Based on patch by Xiang Zhang.

Fixed a crash in `sqlite3.Connection.cursor()` when a factory creates not a cursor. Patch by Xiang Zhang.

Avoid spurious output on OS X with Gnu Readline.

Restore deterministic behavior of `random.Random().seed()` for string seeds using seeding version 1. Allows sequences of calls to `random()` to exactly match those obtained in Python 2. Patch by Nofar Schnider.

Fix a regression in `Connection.commit()`. Statements should not be reset after a commit.

Chunked transfer encoding support added to `http.client.HTTPConnection` requests. The `urllib.request.AbstractHTTPHandler` class does not enforce a Content-Length header any more. If a HTTP request has a file or iterable body, but no Content-Length header, the library now falls back to use chunked transfer-encoding.

A new version of `typing.py` from <https://github.com/python/typing>: - Collection (only for 3.6) (Issue #27598) - Add `Frozenset` to `__all__` (upstream #261) - fix crash in `__get_type_vars()` (upstream #259) - Remove the dict constraint in `ForwardRef._eval_type` (upstream #252)

Make `_normalize` parameter to `Fraction` constructor keyword-only, so that `Fraction(2, 3, 4)` now raises `TypeError`.

Fix unnormalised `Fraction.__pow__` result in the case of negative exponent and negative base.

`cursor.description` is now available for queries using CTEs.

In `distutils.sdist`s, simply produce the "gztar" (gzipped tar format) distributions on all platforms unless "formats" is supplied.

`posixpath.ismount` now correctly recognizes mount points which the user does not have permission to access.

On Linux, `ctypes.util.find_library` now looks in `LD_LIBRARY_PATH` for shared libraries.

`exit` message for `code.interact` is now configurable.

Improved behaviour of `logging.handlers.QueueListener`. Thanks to Paulo Andrade and Petr Viktorin for the analysis and patch.

Distributed reference counting added to multiprocessing to support nesting of shared values / proxy objects.

Improves readability of multiprocessing error message. Thanks to Wojciech Walczak for patch.

asyncio: Add `set_protocol / get_protocol` to `Transports`.

asyncio: Set `TCP_NODELAY` by default.

Add 'interrupt execution' (^C) to Shell menu. Patch by Roger Serwy, updated by Bayard Randel.

Stop IDLE tests from 'flashing' gui widgets on the screen.

Consistently group and sort imports within `idlelib` modules.

add larger font sizes for classroom projection.

Add version to title of IDLE help window.

In section on IDLE -- console differences, mention that using `exec` means that `__builtins__` is defined for each statement.

Fix 3.6.0a3 regression that prevented custom key sets from being selected when no custom theme was defined.

Excluded underscored names and other private API from limited API.

Add support for path-like objects in `PyUnicode_FSConverter()` & `PyUnicode_FSDecoder()`.

Additional tests for the `math` module. Patch by Francisco Couzo.

Skip `math` and `cmath` tests that fail on OS X 10.4 due to a poor `libm` implementation of `tan`.

Improve `test_math` and `test_cmath` coverage and rigour. Patch by Jeff Allen.

Call `gc.collect()` before checking each test for "dangling threads", since the dangling threads are weak references.

Fix clean target in freeze makefile (patch by Lisa Roach)

Update message in `validate_ucrtbase.py`

Deprecate building `_ctypes` with the bundled copy of `libffi` on non-OSX UNIX platforms.

Cause lack of `llvm-profdata` tool when using `clang` as required for PGO linking to be a configure time error rather than make time when `--with-optimizations` is enabled. Also improve our ability to find the `llvm-profdata` tool on MacOS and some Linuxes.

Support for DTrace and SystemTap probes.

The profile-opt build now applies PGO to the built-in modules.

Add the `--with-optimizations` flag to turn on LTO and PGO build support when available.

Set platform triplets for Android builds.

Update references to the `$(LIBPL)` installation path on AIX. This path was changed in 3.2a4.

Update OS X installer to use SQLite 3.14.1 and XZ 5.2.2.

Fix LTO builds on OS X.

Build OS X installer with a private copy of OpenSSL. Also provide a sample Install Certificates command script to install a set of root certificates from the third-party `certifi` module.

Get Tools/scripts/fixcid.py working with Python 3 and the current "re" module, avoid invalid Python backslash escapes, and fix a bug parsing escaped C quote signs.

Update xz dependency to 5.2.2 and build it from source.

Ensures TargetDir is set before continuing with custom install.

Windows console doesn't input or print Unicode (PEP 528)

Change file system encoding on Windows to UTF-8 (PEP 529)

Opt-out of MAX\_PATH on Windows 10

Adds encoding and errors parameters to subprocess.

Adds oem encoding, alias ansi to mbc, move aliasmbcs to codec lookup.

The functions of the winsound module now accept keyword arguments.

Build full text search support into SQLite on Windows.

Adds new icons for Python files and processes on Windows. Designs by Cherry Wang.

Update sqlite to 3.14.1.0 on Windows.