[CVE-2019-5010] Fix a NULL pointer deref in ssl module. The cert parser did not handle CRL distribution points with empty DP or URI correctly. A malicious or buggy certificate can result into segfault. Vulnerability (TALOS-2018-0758) reported by Colin Read and Nicolas Edet of Cisco.

The :option:`-I` command line option (run Python in isolated mode) is now also copied by the :mod:`multiprocessing` and :mod:`distutils` modules when spawning child processes. Previously, only :option:`-E` and :option:`-s` options (enabled by :option:`-I`) were copied.

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\(cpython-main) (Misc) (NEWS.d) 3.8.0a1.rst`, **line 19**); *backlink*
>
> Unknown interpreted text role "option".

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\(cpython-main) (Misc) (NEWS.d) 3.8.0a1.rst`, **line 19**); *backlink*
>
> Unknown interpreted text role "mod".

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\(cpython-main) (Misc) (NEWS.d) 3.8.0a1.rst`, **line 19**); *backlink*
>
> Unknown interpreted text role "mod".

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\(cpython-main) (Misc) (NEWS.d) 3.8.0a1.rst`, **line 19**); *backlink*
>
> Unknown interpreted text role "option".

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\(cpython-main) (Misc) (NEWS.d) 3.8.0a1.rst`, **line 19**); *backlink*
>
> Unknown interpreted text role "option".

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\(cpython-main) (Misc) (NEWS.d) 3.8.0a1.rst`, **line 19**); *backlink*
>
> Unknown interpreted text role "option".

The xml.sax and xml.dom.domreg no longer use environment variables to override parser implementations when sys.flags.ignore_environment is set by -E or -I arguments.

The xml.sax and xml.dom.minidom parsers no longer processes external entities by default. External DTD and ENTITY declarations no longer load files or create network connections.

CVE-2018-14647: The C accelerated _elementtree module now initializes hash randomization salt from _Py_HashSecret instead of libexpat's default CSPRNG.

Updated to OpenSSL 1.1.0i for Windows builds.

Fixed sending the part of the file in :func:`os.sendfile` on macOS. Using the *trailers* argument could cause sending more bytes from the input file than was specified.

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\(cpython-main) (Misc) (NEWS.d) 3.8.0a1.rst`, **line 72**); *backlink*
>
> Unknown interpreted text role "func".

Fixed thread-safety of error handling in _ssl.

Harden ssl module against LibreSSL CVE-2018-8970. X509_VERIFY_PARAM_set1_host() is called with an explicit namelen. A new test ensures that NULL bytes are not allowed.

Minimal fix to prevent buffer overrun in os.symlink on Windows

Regexes in difflib and poplib were vulnerable to catastrophic backtracking. These regexes formed potential DOS vectors (REDOS). They have been refactored. This resolves CVE-2018-1060 and CVE-2018-1061. Patch by Jamie Davis.

The ssl module now allows users to perform their own IDN en/decoding when using SNI.

Make parenthesis optional for named expressions in while statement. Patch by Karthikeyan Singaravelan.

Allow same right hand side expressions in annotated assignments as in normal ones. In particular, `x: Tuple[int, int] = 1, 2` (without parentheses on the right) is now allowed.

Add the option to parse PEP 484 type comments in the ast module. (Off by default.) This is merging the key functionality of the third party fork thereof, [typed_ast](https://github.com/python/typed_ast).

Reorganize Python initialization to get working exceptions and sys.stderr earlier.

Add end line and end column position information to the Python AST nodes. This is a C-level backwards incompatible change.

Fixed a minor memory leak in pymain_parse_cmdline_impl function in Modules/main.c

`func(**kwargs)` will now raise an error when `kwargs` is a mapping containing multiple entries with the same key. An error was already raised when other keyword arguments are passed before `**kwargs` since Python 3.6.

Fix a crash when sorting very long lists. Patch by Stephan Hohe.

clang Memory Sanitizer build instrumentation was added to work around false positives from posix, socket, time, test_io, and test_faulthandler.

Fix an assertion error in :func:`format` in debug build for floating point formatting with "n" format, zero padding and small width. Release build is not impacted. Patch by Karthikeyan Singaravelan.

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\(cpython-main)(Misc)(NEWS.d)3.8.0a1.rst`, **line 227**); *backlink*
>
> Unknown interpreted text role "func".

Format characters `%s` and `%V` in :c:func:`PyUnicode_FromFormat` and `%s` in :c:func:`PyBytes_FromFormat` no longer read memory past the limit if *precision* is specified.

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\(cpython-main)(Misc)(NEWS.d)3.8.0a1.rst`, **line 238**); *backlink*
>
> Unknown interpreted text role "c:func".

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\(cpython-main)(Misc)(NEWS.d)3.8.0a1.rst`, **line 238**); *backlink*
>
> Unknown interpreted text role "c:func".

Fix segfaults and :exc:`SystemError`s when deleting certain attributes. Patch by Zackery Spytz.

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\(cpython-main)(Misc)(NEWS.d)3.8.0a1.rst`, **line 249**); *backlink*
>
> Unknown interpreted text role "exc".

Fixed a SystemError when delete the characters_written attribute of an OSError.

Improved syntax error messages for unbalanced parentheses in f-string.

Fixed error handling in pickling methods when fail to look up builtin "getattr". Sped up pickling iterators.

Fix various issues with memory allocation error handling. Patch by Zackery Spytz.

Separate the signal handling trigger in the eval loop from the "pending calls" machinery. There is no semantic change and the difference in performance is insignificant.

Internal attributes' names of unittest.mock._Call and unittest.mock.MagicProxy (name, parent & from_kall) are now prefixed with _mock_ in order to prevent clashes with widely used object attributes. Fixed minor typo in test function name.

Fixed the code page decoder for input longer than 2 GiB containing undecodable bytes.

Fix PYTHONCOERCECLOCALE=1 environment variable: only coerce the C locale if the LC_CTYPE locale is "C".

The *lineno* and *col_offset* attributes of AST nodes for list comprehensions, generator expressions and tuples are now point to the opening parenthesis or square brace. For tuples without parenthesis they point to the position of the first item.

For :meth:`str.format`, :meth:`float.__format__` and :meth:`complex.__format__` methods for non-ASCII decimal point when using the "n" formatter.

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\(cpython-main)(Misc)(NEWS.d)3.8.0a1.rst`, **line 353**); *backlink*
>
> Unknown interpreted text role "meth".

Fix a possible segfault involving a newly-created coroutine. Patch by Zackery Spytz.

Implement PEP 572 (assignment expressions). Patch by Emily Morehouse.

Speed up :class:`namedtuple` attribute access by 1.6x using a C fast-path for the name descriptors. Patch by Pablo Galindo.

Fixed an out of bounds memory access when parsing a truncated unicode escape sequence at the end of a string such as `'\N'`. It would read one byte beyond the end of the memory allocation.

The interpreter and extension modules have had annotations added so that they work properly under clang's Memory Sanitizer. A new configure flag --with-memory-sanitizer has been added to make test builds of this nature easier to perform.

Fix an off by one error in the bytecode peephole optimizer where it could read bytes beyond the end of bounds of an array when removing unreachable code. This bug was present in every release of Python 3.6 and 3.7 until now.

Improved error messages for forbidden assignments.

Fix handling of hash-based bytecode files in :mod:`zipimport`. Patch by Elvis Pranskevichus.

Debug builds will no longer to attempt to import extension modules built for the ABI as they were never compatible to begin with. Patch by Stefano Rivera.

Clarify in the docstrings of :mod:`os` methods that path-like objects are also accepted as input parameters.

:mod:`socket`: Fix off-by-one bug in length check for `AF_ALG` name and type.

Raise :exc:`ValueError` instead of :exc:`OverflowError` in case of a negative `_length_` in a :class:`ctypes.Array` subclass. Also raise :exc:`TypeError` instead of :exc:`AttributeError` for non-integer `_length_`. Original patch by Oren Milman.

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\(cpython-main)(Misc)(NEWS.d)3.8.0a1.rst`, **line 477**); *backlink*
>
> Unknown interpreted text role "class".

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\(cpython-main)(Misc)(NEWS.d)3.8.0a1.rst`, **line 477**); *backlink*
>
> Unknown interpreted text role "exc".

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\(cpython-main)(Misc)(NEWS.d)3.8.0a1.rst`, **line 477**); *backlink*
>
> Unknown interpreted text role "exc".

Fix `lineno` and `col_offset` for multi-line string tokens.

:exc:`SyntaxWarning` raised as an exception at code generation time will be now replaced with a :exc:`SyntaxError` for better error reporting.

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\(cpython-main)(Misc)(NEWS.d)3.8.0a1.rst`, **line 498**); *backlink*
>
> Unknown interpreted text role "exc".

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\(cpython-main)(Misc)(NEWS.d)3.8.0a1.rst`, **line 498**); *backlink*
>
> Unknown interpreted text role "exc".

Expose :meth:`symtable.Symbol.is_nonlocal` in the symtable module. Patch by Pablo Galindo.

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\(cpython-main)(Misc)(NEWS.d)3.8.0a1.rst`, **line 508**); *backlink*
>
> Unknown interpreted text role "meth".

:class:`bytes` and :class:`bytearray` constructors no longer convert unexpected exceptions (e.g. :exc:`MemoryError` and :exc:`KeyboardInterrupt`) to :exc:`TypeError`.

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\(cpython-main)(Misc)(NEWS.d)3.8.0a1.rst`, **line 518**); *backlink*
>
> Unknown interpreted text role "class".

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\(cpython-main)(Misc)(NEWS.d)3.8.0a1.rst`, **line 518**); *backlink*
>
> Unknown interpreted text role "class".

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\(cpython-main)(Misc)(NEWS.d)3.8.0a1.rst`, **line 518**); *backlink*
>
> Unknown interpreted text role "exc".

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\(cpython-main)(Misc)(NEWS.d)3.8.0a1.rst`, **line 518**); *backlink*
>
> Unknown interpreted text role "exc".

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\(cpython-main)(Misc)(NEWS.d)3.8.0a1.rst`, **line 518**); *backlink*
>
> Unknown interpreted text role "exc".

Allow annotated names in module namespace that are declared global before the annotation happens. Patch by Pablo Galindo.

Fixed crash in :func:`bytes` when the :class:`list` argument is mutated while it is iterated.

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\(cpython-main)(Misc)(NEWS.d)3.8.0a1.rst, line 539`); *backlink*
>
> Unknown interpreted text role "func".

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\(cpython-main)(Misc)(NEWS.d)3.8.0a1.rst, line 539`); *backlink*
>
> Unknown interpreted text role "class".

The *lineno* and *col_offset* attributes of the AST for decorated function and class refer now to the position of the corresponding `def`, `async def` and `class` instead of the position of the first decorator. This leads to more correct line reporting in tracing. This is the only case when the position of child AST nodes can precede the position of the parent AST node.

Fix a possible null pointer dereference in bytesobject.c. Patch by Zackery Spytz.

Fix the implementation of PyStructSequence_NewType in order to create heap allocated StructSequences.

A :exc:`SyntaxWarning` is now emitted instead of a :exc:`DeprecationWarning` for invalid escape sequences in string and bytes literals.

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\(cpython-main)(Misc)(NEWS.d)3.8.0a1.rst, line 583`); *backlink*
>
> Unknown interpreted text role "exc".

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\(cpython-main)(Misc)(NEWS.d)3.8.0a1.rst, line 583`); *backlink*
>
> Unknown interpreted text role "exc".

Fixed a crash in compiling string annotations containing a lambda with a keyword-only argument that doesn't have a default value.

The compiler now produces a :exc:`SyntaxWarning` when identity checks (`is` and `is not`) are used with certain types of literals (e.g. strings, ints). These can often work by accident in CPython, but are not guaranteed by the language spec. The warning advises users to use equality tests (`==` and `!=`) instead.

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\(cpython-main)(Misc)(NEWS.d)3.8.0a1.rst, line 603`); *backlink*
>
> Unknown interpreted text role "exc".

Fix a possible null pointer dereference in Modules/_ssl.c. Patch by Zackery Spytz.

The C function `property_descr_get()` uses a "cached" tuple to optimize function calls. But this tuple can be discovered in debug mode with :func:`sys.getobjects()`. Remove the optimization, it's not really worth it and it causes 3 different crashes last years.

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\(cpython-main)(Misc)(NEWS.d)3.8.0a1.rst, line 626`); *backlink*
>
> Unknown interpreted text role "func".

Fix contextvars C API to use PyObject* pointer types.

The hash function for tuples is now based on xxHash which gives better collision results on (formerly) pathological cases. Additionally, on 64-bit systems it improves tuple hashes in general. Patch by Jeroen Demeyer with substantial contributions by Tim Peters.

Fix a memory leak in Modules/timemodule.c. Patch by Zackery Spytz.

Fixed a bug where some SyntaxError error pointed to locations that were off-by-one.

Only allow the main interpreter to fork. The avoids the possibility of affecting the main interpreter, which is critical to operation of the runtime.

Remove unused function PyParser_SimpleParseStringFilename.

Warn that line buffering is not supported if :func:`open` is called with binary mode and `buffering=1`.

Further restrict the syntax of the left-hand side of keyword arguments in function calls. In particular, `f((keyword)=arg)` is now disallowed.

Make the *start* argument to *sum()* visible as a keyword argument.

Do not assume signed integer overflow behavior (C undefined behavior) when performing set hash table resizing.

Fix an off-by-one in the recursive call pruning feature of traceback formatting.

On Windows, the LC_CTYPE is now set to the user preferred locale at startup. Previously, the LC_CTYPE locale was "C" at startup, but changed when calling setlocale(LC_CTYPE, "") or setlocale(LC_ALL, "").

Standard streams like sys.stdout now use the "surrogateescape" error handler, instead of "strict", on the POSIX locale (when the C locale is not coerced and the UTF-8 Mode is disabled).

Fix the error handler of standard streams like sys.stdout: PYTHONIOENCODING=":" is now ignored instead of setting the error handler to "strict".

Python now gets the locale encoding with C code to initialize the encoding of standard streams like sys.stdout. Moreover, the encoding is now initialized to the Python codec name to get a normalized encoding name and to ensure that the codec is loaded. The change avoids importing _bootlocale and _locale modules at startup by default.

On FreeBSD, Py_DecodeLocale() and Py_EncodeLocale() now also forces the ASCII encoding if the LC_CTYPE locale is "POSIX", not only if the LC_CTYPE locale is "C".

The UTF-8 Mode is now also enabled by the "POSIX" locale, not only by the "C" locale.

On HP-UX with C or POSIX locale, sys.getfilesystemencoding() now returns "ascii" instead of "roman8" (when the UTF-8 Mode is disabled and the C locale is not coerced).

The Python filesystem encoding is now read earlier during the Python initialization.

Tracebacks show now correct line number for subexpressions in multiline expressions. Tracebacks show now the line number of the first line for multiline expressions instead of the line number of the last subexpression.

Prevent a null pointer dereference and resource leakage in `PyInterpreterState_New()`.

Fix undefined behavior in parsetok.c. Patch by Zackery Spytz.

Added as_integer_ratio to ints to make them more interoperable with floats.

Update valgrind suppression list to use `_PyObject_Free`/`_PyObject_Realloc` instead of `PyObject_Free/PyObject_Realloc`.

Added the "socket" option in the *stat.filemode()* Python implementation to match the C implementation.

Fix `dict(od)` didn't copy iteration order of OrderedDict.

Fixed crash on debug builds when opcode stack was adjusted with negative numbers. Patch by Constantin Petrisor.

Compiler now merges constants in tuples and frozensets recursively. Code attributes like `co_names` are merged too.

Performance of list concatenation, repetition and slicing operations is slightly improved. Patch by Sergey Fedoseev.

-X dev: it is now possible to override the memory allocator using PYTHONMALLOC even if the developer mode is enabled.

Improved :exc:`AttributeError` message for partially initialized module.

Fix min and max functions to get default behavior when key is None.

Profiling of unbound built-in methods now works when `**kwargs` is given.

Optimized pickling atomic types (None, bool, int, float, bytes, str).

Fix crashes when profiling certain invalid calls of unbound methods. Patch by Jeroen Demeyer.

Fixed reading invalid memory when create the code object with too small varnames tuple or too large argument counts.

In :meth:`io.IOBase.close`, ensure that the :attr:`~io.IOBase.closed` attribute is not set with a live exception. Patch by Zackery Spytz and Serhiy Storchaka.

Unknown interpreted text role "meth".

Fix buffer overflow while converting unicode to numeric values.

Fixed a memory leak in the compiler when it raised some uncommon errors during tokenizing.

Disabled interruption by Ctrl-C between calling `open()` and entering a **with** block in `with open()`.

Fix dict.copy() to maintain correct total refcount (as reported by sys.gettotalrefcount()).

Fix potential memory leak in function object when it creates reference cycle.

Implement contextvars.ContextVar.name attribute.

Update vendored Expat library copy to version 2.2.5.

Decref the module object in :c:func:`PyRun_SimpleFileExFlags` before calling :c:func:`PyErr_Print()`. Patch by Zackery Spytz.

Close directly executed pyc files before calling `PyEval_EvalCode()`.

The hash of :class:`BuiltinMethodType` instances (methods of built-in classes) now depends on the hash of the identity of *__self__* instead of its value. The hash and equality of :class:`ModuleType` and :class:`MethodWrapperType` instances (methods of user-defined classes and some methods of built-in classes like `str.__add__`) now depend on the hash and equality of the identity of *__self__* instead of its value. :class:`MethodWrapperType` instances no longer support ordering.

Fix "LC_ALL=C python3.7 -V": reset properly the command line parser when the encoding changes after reading the Python configuration.

Fix a crash in hamt.c caused by enabling GC tracking for an object that hadn't all of its fields set to NULL.

Seven macro incompatibilities with the Limited API were fixed, and the macros :c:func:`PyIter_Check`, :c:func:`PyIndex_Check` and :c:func:`PyExceptionClass_Name` were added as functions. A script for automatic macro checks was added.

Unknown interpreted text role "c:func".

Unknown interpreted text role "c:func".

Unknown interpreted text role "c:func".

Fix asynchronous generators to handle GeneratorExit in athrow() correctly

`PyRun_SimpleFileExFlags` removes `__cached__` from module in addition to `__file__`.

Fix a crash in Python initialization when parsing the command line options. Thanks Christoph Gohlke for the bug report and the fix!

Reduce `PyGC_Head` size from 3 words to 2 words.

Fixed reset of the SIGINT handler to SIG_DFL on interpreter shutdown even when there was a custom handler set previously. Patch by Philipp Kerling.

Fixed a leak when the garbage collector fails to add an object with the `__del__` method or referenced by it into the :data:`gc.garbage` list. :c:func:`PyGC_Collect` can now be called when an exception is set and preserves it.

Unknown interpreted text role "data".

Unknown interpreted text role "c:func".

Make dict and dict views reversible. Patch by Rémi Lapeyre.

A :exc:`RuntimeError` is now raised when the custom metaclass doesn't provide the `__classcell__` entry in the namespace passed to `type.__new__`. A :exc:`DeprecationWarning` was emitted in Python 3.6--3.7.

Unknown interpreted text role "exc".

Unknown interpreted text role "exc".

Add :envvar:`PYTHONPYCACHEPREFIX` environment variable and :option:`-X` `pycache_prefix` command-line option to set an alternate root directory for writing module bytecode cache files.

Unknown interpreted text role "envvar".

Unknown interpreted text role "option".

The :mod:`zipimport` module has been rewritten in pure Python.

Unknown interpreted text role "mod".

Fix module_globals parameter of warnings.warn_explicit(): don't crash if module_globals is not a dict.

Fix signed/unsigned comparison warning in pyhash.c.

Fixed miscellaneous bugs in converting annotations to strings and optimized parentheses in the string representation.

Added support for the *setpgroup*, *resetids*, *setsigmask*, *setsigdef* and *scheduler* parameters of *posix_spawn*. Patch by Pablo Galindo.

Fix a leak in set_symmetric_difference().

Raise a SyntaxError for `async with` and `async for` statements outside of async functions.

Fix unaligned accesses in siphash24(). Patch by Rolf Eike Beer.

Fix a bug that causes PathFinder to appear twice on sys.meta_path. Patch by Pablo Galindo Salgado.

Modules imported last are now cleared first at interpreter shutdown.

Fixed clang ubsan (undefined behavior sanitizer) warnings in dictobject.c by adjusting how the internal struct _dictkeysobject shared keys structure is declared.

Improved syntax error messages for invalid numerical literals.

Improved syntax error messages for unbalanced parentheses.

The list constructor will pre-size and not over-allocate when the input length is known.

Intern the names for all anonymous code objects. Patch by Zackery Spytz.

The C and Python code and the documentation related to tokens are now generated from a single source file :file:`Grammar/Tokens`.

> **System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\(cpython-main)(Misc)(NEWS.d)3.8.0a1.rst`, line 1379);** *backlink*
>
> Unknown interpreted text role "file".

Add a `toreadonly()` method to memoryviews.

Fix potential memory leak in `normalizestring()`.

Change dict growth function from `round_up_to_power_2(used*2+hashtable_size/2)` to `round_up_to_power_2(used*3)`. Previously, dict is shrinked only when `used == 0`. Now dict has more chance to be shrinked.

Improved error messages in 'async with' when `__aenter__()` or `__aexit__()` return non-awaitable object.

Fix `ma_version_tag` in dict implementation is uninitialized when copying from key-sharing dict.

When using the -m switch, sys.path[0] is now explicitly expanded as the *starting* working directory, rather than being left as the empty path (which allows imports from the current working directory at the time of the import)

Changed standard error message for non-pickleable and non-copyable types. It now says "cannot pickle" instead of "can't pickle" or "cannot serialize".

Improve consistency of errors raised by `issubclass()` when called with a non-class and an abstract base class as the first and second arguments, respectively. Patch by Josh Bronson.

`math.factorial` no longer accepts arguments that are not int-like. Patch by Pablo Galindo.

Added new opcode :opcode:`END_ASYNC_FOR` and fixes the following issues:

> **System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\(cpython-main)(Misc)(NEWS.d)3.8.0a1.rst`, line 1482);** *backlink*
>
> Unknown interpreted text role "opcode".

- Setting global :exc:`StopAsyncIteration` no longer breaks `async for` loops.

  > **System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\(cpython-main)(Misc)(NEWS.d)3.8.0a1.rst`, line 1484);** *backlink*
  >
  > Unknown interpreted text role "exc".

- Jumping into an `async for` loop is now disabled.

- Jumping out of an `async for` loop no longer corrupts the stack.

Fix rare Python crash due to bad refcounting in `type_getattro()` if a descriptor deletes itself from the class. Patch by Jeroen Demeyer.

Fixed bytecode generation for "async for" with a complex target. A StopAsyncIteration raised on assigning or unpacking will be now propagated instead of stopping the iteration.

Fixed jumping out of "with" block by setting f_lineno.

Fix a crash on fork when using a custom memory allocator (ex: using PYTHONMALLOC env var). _PyGILState_Reinit() and _PyInterpreterState_Enable() now use the default RAW memory allocator to allocate a new interpreters mutex on fork.

Due to unexpected compatibility issues discovered during downstream beta testing, reverted :issue:\`29463\`. `docstring` field is removed from Module, ClassDef, FunctionDef, and AsyncFunctionDef ast nodes which was added in 3.7a1. Docstring expression is restored as a first statement in their body. Based on patch by Inada Naoki.

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\(cpython-main)(Misc)(NEWS.d)3.8.0a1.rst`, **line 1538);** *backlink*
>
> Unknown interpreted text role "issue".

Prevent jumps from 'return' and 'exception' trace events.

Importing names from already imported module with "from ... import ..." is now 30% faster if the module is not a package.

Make error message more revealing when there are non-str objects in `__all__`.

Optimized iterating and containing test for literal lists consisting of non-constants: `x in [a, b]` and `for x in [a, b]`. The case of all constant elements already was optimized.

Update Valgrind suppression list to account for the rename of `Py_ADDRESS_IN_RANG` to `address_in_range`.

Don't use temporary variables in cases of list/dict/set comprehensions

Remove the new API added in bpo-31356 (gc.ensure_disabled() context manager).

For namespace packages, ensure that both `__file__` and `__spec__`.origin are set to None.

Make sure `__spec__`.loader matches `__loader__` for namespace packages.

Fix the warning messages for Python/ast_unparse.c. Patch by Stéphane Wirtel

Fix possible crashing in builtin Unicode decoders caused by write out-of-bound errors when using customized decode error handlers.

A :keyword:\`continue\` statement is now allowed in the :keyword:\`finally\` clause.

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\(cpython-main)(Misc)(NEWS.d)3.8.0a1.rst`, **line 1658);** *backlink*
>
> Unknown interpreted text role "keyword".

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\(cpython-main)(Misc)(NEWS.d)3.8.0a1.rst`, **line 1658);** *backlink*
>
> Unknown interpreted text role "keyword".

Simplified the interpreter loop by moving the logic of unrolling the stack of blocks into the compiler. The compiler emits now explicit instructions for adjusting the stack of values and calling the cleaning up code for :keyword:\`break\`, :keyword:\`continue\` and :keyword:\`return\`.

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\(cpython-main)(Misc)(NEWS.d)3.8.0a1.rst`, **line 1668);** *backlink*
>
> Unknown interpreted text role "keyword".

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\(cpython-main)(Misc)(NEWS.d)3.8.0a1.rst`, **line 1668);** *backlink*
>
> Unknown interpreted text role "keyword".

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\(cpython-main)(Misc)(NEWS.d)3.8.0a1.rst`, **line 1668);** *backlink*
>
> Unknown interpreted text role "keyword".

Removed opcodes :opcode:`BREAK_LOOP`, :opcode:`CONTINUE_LOOP`, :opcode:`SETUP_LOOP` and :opcode:`SETUP_EXCEPT`. Added new opcodes :opcode:`ROT_FOUR`, :opcode:`BEGIN_FINALLY` and :opcode:`CALL_FINALLY` and :opcode:`POP_FINALLY`. Changed the behavior of :opcode:`END_FINALLY` and :opcode:`WITH_CLEANUP_START`.

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\(cpython-main)(Misc)(NEWS.d)3.8.0a1.rst`, **line 1673**); *backlink*

Unknown interpreted text role "opcode".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\(cpython-main)(Misc)(NEWS.d)3.8.0a1.rst`, **line 1673**); *backlink*

Unknown interpreted text role "opcode".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\(cpython-main)(Misc)(NEWS.d)3.8.0a1.rst`, **line 1673**); *backlink*

Unknown interpreted text role "opcode".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\(cpython-main)(Misc)(NEWS.d)3.8.0a1.rst`, **line 1673**); *backlink*

Unknown interpreted text role "opcode".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\(cpython-main)(Misc)(NEWS.d)3.8.0a1.rst`, **line 1673**); *backlink*

Unknown interpreted text role "opcode".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\(cpython-main)(Misc)(NEWS.d)3.8.0a1.rst`, **line 1673**); *backlink*

Unknown interpreted text role "opcode".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\(cpython-main)(Misc)(NEWS.d)3.8.0a1.rst`, **line 1673**); *backlink*

Unknown interpreted text role "opcode".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\(cpython-main)(Misc)(NEWS.d)3.8.0a1.rst`, **line 1673**); *backlink*

Unknown interpreted text role "opcode".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\(cpython-main)(Misc)(NEWS.d)3.8.0a1.rst`, **line 1673**); *backlink*

Unknown interpreted text role "opcode".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\(cpython-main)(Misc)(NEWS.d)3.8.0a1.rst`, **line 1673**); *backlink*

Unknown interpreted text role "opcode".

New function unicodedata.is_normalized, which can check whether a string is in a specific normal form.

Yield expressions are now disallowed in comprehensions and generator expressions except the expression for the outermost iterable.

Iterable unpacking is now allowed without parentheses in yield and return statements, e.g. `yield 1, 2, 3, *rest`. Thanks to David Cuthbert for the change and Jordan Chapman for added tests.

Fix the `col_offset` attribute for ast nodes `ast.AsyncFor`, `ast.AsyncFunctionDef`, and `ast.AsyncWith`. Previously, `col_offset` pointed to the keyword after `async`.

Fix assertion failures in the `tell()` method of `io.TextIOWrapper`. Patch by Zackery Spytz.

Fix a crash in *ctypes.cast()* in case the type argument is a ctypes structured data type. Patch by Eryk Sun and Oren Milman.

Fix a crash in *os.utime()* in case of a bad ns argument. Patch by Oren Milman.

Remove references to 'getsockaddrarg' from various socket error messages. Patch by Oren Milman.

Add 'order' parameter to memoryview.tobytes().

The _asdict() method for collections.namedtuple now returns a regular dict instead of an OrderedDict.

An ExitStack is now used internally within subprocess.Popen to clean up pipe file handles. No behavior change in normal operation. But if closing one handle were ever to cause an exception, the others will now be closed instead of leaked. (patch by Giampaolo Rodola)

RISC-V needed the CTYPES_PASS_BY_REF_HACK. Fixes ctypes Structure test_pass_by_value.

Shared memory submodule added to multiprocessing to avoid need for serialization between processes

Fix lru_cache() errors arising in recursive, reentrant, or multi-threaded code. These errors could result in orphan links and in the cache being trapped in a state with fewer than the specified maximum number of links. Fix handling of negative maxsize which should have been treated as zero. Fix errors in toggling the "full" status flag. Fix misordering of links when errors are encountered. Sync-up the C code and pure Python code for the space saving path in functions with a single positional argument. In this common case, the space overhead of an lru cache entry is reduced by almost half. Fix counting of cache misses. In error cases, the miss count was out of sync with the actual number of times the underlying user function was called.

:func:`os.posix_spawn` and :func:`os.posix_spawnp` now have a *setsid* parameter.

> **System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\(cpython-main) (Misc) (NEWS.d) 3.8.0a1.rst`, line 1838); *backlink***
>
> Unknown interpreted text role "func".

> **System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\(cpython-main) (Misc) (NEWS.d) 3.8.0a1.rst`, line 1838); *backlink***
>
> Unknown interpreted text role "func".

:class:`asyncio.ProactorEventLoop` now catches and logs send errors when the self-pipe is full.

> **System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\(cpython-main) (Misc) (NEWS.d) 3.8.0a1.rst`, line 1848); *backlink***
>
> Unknown interpreted text role "class".

:mod:`asyncio`: Enhance `IocpProactor.close()` log: wait 1 second before the first log, then log every second. Log also the number of seconds since `close()` was called.

> **System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\(cpython-main) (Misc) (NEWS.d) 3.8.0a1.rst`, line 1858); *backlink***
>
> Unknown interpreted text role "mod".

Add a new :func:`os.posix_spawnp` function. Patch by Joannah Nanjekye.

> **System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\(cpython-main) (Misc) (NEWS.d) 3.8.0a1.rst`, line 1869); *backlink***
>
> Unknown interpreted text role "func".

`ast.Constant(boolean)` no longer an instance of :class:`ast.Num`. Patch by Anthony Sottile.

> **System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\(cpython-main) (Misc) (NEWS.d) 3.8.0a1.rst`, line 1878); *backlink***
>
> Unknown interpreted text role "class".

QueueHandler.prepare() now makes a copy of the record before modifying and enqueueing it, to avoid affecting other handlers in the chain.

Sped up multi-argument :mod:`math` functions atan2(), copysign(), remainder() and hypot() by 1.3--2.5 times.

Fix KeyError exception raised when using enums and compile. Patch contributed by RÃ©mi Lapeyre.

Fixed detection of Visual Studio Build Tools 2017 in distutils

Fix memory leaks in asyncio ProactorEventLoop on overlapped operation failure.

The :data:`time.CLOCK_UPTIME_RAW` constant is now available for macOS 10.12.

Fix a memory leak in asyncio in the ProactorEventLoop when `ReadFile()` or `WSASend()` overlapped operation fail immediately: release the internal buffer.

Fix `asyncio.ProactorEventLoop.sendfile()`: don't attempt to set the result of an internal future if it's already done.

Add a deprecated warning for the :meth:`threading.Thread.isAlive` method. Patch by Dong-hee Na.

Improve operator.itemgetter() performance by 33% with optimized argument handling and with adding a fast path for the common case of a single non-negative integer index into a tuple (which is the typical use case in the standard library).

Fixed a SyntaxWarning: invalid escape sequence in Modules/_sha3/cleanup.py. Patch by MickaÃ«l Schoentgen.

Improved support of custom data descriptors in :func:`help` and :mod:`pydoc`.

The *crypt* module now internally uses the *crypt_r()* library function instead of *crypt()* when available.

Fixed help() on metaclasses. Patch by Sanyam Khurana.

Expose `raise(signum)` as *raise_signal*

The floor division and modulo operations and the :func:`divmod` function on :class:`fractions.Fraction` types are 2--4x faster. Patch by Stefan Behnel.

Speed-up building enums by value, e.g. http.HTTPStatus(200).

random.gammavariate(1.0, beta) now computes the same result as random.expovariate(1.0 / beta). This synchronizes the two algorithms and eliminates some idiosyncrasies in the old implementation. It does however produce a difference stream of random variables than it used to.

The :mod:`subprocess` module can now use the :func:`os.posix_spawn` function in some cases for better performance.

> **System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\(cpython-main) (Misc) (NEWS.d) 3.8.0a1.rst`, line 2068);** *backlink*
>
> Unknown interpreted text role "mod".

> **System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\(cpython-main) (Misc) (NEWS.d) 3.8.0a1.rst`, line 2068);** *backlink*
>
> Unknown interpreted text role "func".

Delaying the 'joke' of barry_as_FLUFL.mandatory to Python version 4.0

Remove :mod:`ctypes` callback workaround: no longer create a callback at startup. Avoid SELinux alert on `import ctypes` and `import uuid`.

> **System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\(cpython-main) (Misc) (NEWS.d) 3.8.0a1.rst`, line 2087);** *backlink*
>
> Unknown interpreted text role "mod".

:func:`uuid.uuid1` now calls :func:`time.time_ns` rather than `int(time.time() * 1e9)`.

> **System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\(cpython-main) (Misc) (NEWS.d) 3.8.0a1.rst`, line 2097);** *backlink*
>
> Unknown interpreted text role "func".

> **System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\(cpython-main) (Misc) (NEWS.d) 3.8.0a1.rst`, line 2097);** *backlink*
>
> Unknown interpreted text role "func".

:class:`~unittest.runner.TextTestRunner` of :mod:`unittest.runner` now uses :func:`time.perf_counter` rather than :func:`time.time` to measure the execution time of a test: :func:`time.time` can go backwards, whereas :func:`time.perf_counter` is monotonic.

> **System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\(cpython-main) (Misc) (NEWS.d) 3.8.0a1.rst`, line 2107);** *backlink*
>
> Unknown interpreted text role "class".

> **System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\(cpython-main) (Misc) (NEWS.d) 3.8.0a1.rst`, line 2107);** *backlink*
>
> Unknown interpreted text role "mod".

> **System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\(cpython-main) (Misc) (NEWS.d) 3.8.0a1.rst`, line 2107);** *backlink*
>
> Unknown interpreted text role "func".

> **System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\(cpython-main) (Misc) (NEWS.d) 3.8.0a1.rst`, line 2107);** *backlink*
>
> Unknown interpreted text role "func".

> **System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\(cpython-main) (Misc) (NEWS.d) 3.8.0a1.rst`, line 2107);** *backlink*
>
> Unknown interpreted text role "func".

> **System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\(cpython-main) (Misc) (NEWS.d) 3.8.0a1.rst`, line 2107);** *backlink*
>
> Unknown interpreted text role "func".

Fixed reference leaks in :class:`xml.etree.ElementTree.TreeBuilder` in case of unfinished building of the tree (in particular when an error was raised during parsing XML).

> **System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\(cpython-main)(Misc)(NEWS.d)3.8.0a1.rst`, line 2119);** *backlink*
>
> Unknown interpreted text role "class".

Make :func:`platform.architecture` parsing of `file` command output more reliable: add the `-b` option to the `file` command to omit the filename, force the usage of the C locale, and search also the "shared object" pattern.

> **System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\(cpython-main)(Misc)(NEWS.d)3.8.0a1.rst`, line 2130);** *backlink*
>
> Unknown interpreted text role "func".

:mod:`multiprocessing`: Add `Pool.__repr__()` and enhance `BaseProcess.__repr__()` (add pid and parent pid) to ease debugging. Pool state constant values are now strings instead of integers, for example `RUN` value becomes `'RUN'` instead of `0`.

> **System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\(cpython-main)(Misc)(NEWS.d)3.8.0a1.rst`, line 2142);** *backlink*
>
> Unknown interpreted text role "mod".

:meth:`multiprocessing.Pool.__enter__` now fails if the pool is not running: `with pool:` fails if used more than once.

> **System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\(cpython-main)(Misc)(NEWS.d)3.8.0a1.rst`, line 2154);** *backlink*
>
> Unknown interpreted text role "meth".

Copy command line that was passed to CreateProcessW since this function can change the content of the input buffer.

Python 2.4 dropped MacOS 9 support. The macpath module was deprecated in Python 3.7. The module is now removed.

Unblock Proactor event loop when keyboard interrupt is received on Windows

Fix xml.dom.minidom cloneNode() on a document with an entity: pass the correct arguments to the user data handler of an entity.

Allow repeated assignment deletion of :class:`unittest.mock.Mock` attributes. Patch by Pablo Galindo.

> **System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\(cpython-main)(Misc)(NEWS.d)3.8.0a1.rst`, line 2203);** *backlink*
>
> Unknown interpreted text role "class".

Set `__signature__` on mock for :mod:`inspect` to get signature. Patch by Karthikeyan Singaravelan.

> **System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\(cpython-main)(Misc)(NEWS.d)3.8.0a1.rst`, line 2213);** *backlink*
>
> Unknown interpreted text role "mod".

Memory errors during creating posix.environ no longer ignored.

Validate fileno= argument to socket.socket().

:class:`multiprocessing.Pool` destructor now emits :exc:`ResourceWarning` if the pool is still running.

> **System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\(cpython-main)(Misc)(NEWS.d)3.8.0a1.rst`, line 2241);** *backlink*
>
> Unknown interpreted text role "class".

> **System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\(cpython-main)(Misc)(NEWS.d)3.8.0a1.rst`, line 2241);** *backlink*
>
> Unknown interpreted text role "exc".

When a :class:`Mock` instance was used to wrap an object, if *side_effect* is used in one of the mocks of it methods, don't call the original implementation and return the result of using the side effect the same way that it is done with return_value.

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\(cpython-main) (Misc) (NEWS.d)3.8.0a1.rst`, **line 2251**); *backlink*
>
> Unknown interpreted text role "class".

Drop Mac OS 9 and Rhapsody support from the :mod:`platform` module. Rhapsody last release was in 2000. Mac OS 9 last release was in 2001.

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\(cpython-main) (Misc) (NEWS.d)3.8.0a1.rst`, **line 2263**); *backlink*
>
> Unknown interpreted text role "mod".

:func:`~distutils.utils.check_environ` of :mod:`distutils.utils` now catches :exc:`KeyError` on calling :func:`pwd.getpwuid`: don't create the HOME environment variable in this case.

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\(cpython-main) (Misc) (NEWS.d)3.8.0a1.rst`, **line 2273**); *backlink*
>
> Unknown interpreted text role "func".

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\(cpython-main) (Misc) (NEWS.d)3.8.0a1.rst`, **line 2273**); *backlink*
>
> Unknown interpreted text role "mod".

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\(cpython-main) (Misc) (NEWS.d)3.8.0a1.rst`, **line 2273**); *backlink*
>
> Unknown interpreted text role "exc".

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\(cpython-main) (Misc) (NEWS.d)3.8.0a1.rst`, **line 2273**); *backlink*
>
> Unknown interpreted text role "func".

:func:`posixpath.expanduser` now returns the input *path* unchanged if the HOME environment variable is not set and the current user has no home directory (if the current user identifier doesn't exist in the password database). This change fix the :mod:`site` module if the current user doesn't exist in the password database (if the user has no home directory).

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\(cpython-main) (Misc) (NEWS.d)3.8.0a1.rst`, **line 2284**); *backlink*
>
> Unknown interpreted text role "func".

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\(cpython-main) (Misc) (NEWS.d)3.8.0a1.rst`, **line 2284**); *backlink*
>
> Unknown interpreted text role "mod".

:func:`platform.libc_ver` now uses os.confstr('CS_GNU_LIBC_VERSION') if available and the *executable* parameter is not set.

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\(cpython-main) (Misc) (NEWS.d)3.8.0a1.rst`, **line 2297**); *backlink*
>
> Unknown interpreted text role "func".

Add empty slots to asyncio abstract protocols.

Fix a bug in :func:`select.select` where, in some cases, the file descriptor sequences were returned unmodified after a signal interruption, even though the file descriptors might not be ready yet. :func:`select.select` will now always return empty lists if a timeout has occurred. Patch by Oran Avraham.

Enable TCP_NODELAY on Windows for proactor asyncio event loop.

Add generic version of `collections.OrderedDict` to the `typing` module. Patch by Ismo Toijala.

Fixed possible crash in `os.utime()` on Windows when pass incorrect arguments.

:func:`platform.uname` now redirects `stderr` to :data:`os.devnull` when running external programs like `cmd /c ver`.

Previously, calling the strftime() method on a datetime object with a trailing '%' in the format string would result in an exception. However, this only occurred when the datetime C module was being used; the python implementation did not match this behavior. Datetime is now PEP-399 compliant, and will not throw an exception on a trailing '%'.

The function *platform.popen* has been removed, it was deprecated since Python 3.3: use :func:`os.popen` instead.

On macOS, :func:`platform.platform` now uses :func:`platform.mac_ver`, if it returns a non-empty release string, to get the macOS version rather than the darwin version.

Make `lib2to3.pgen2.parse.ParseError` round-trip pickle-able. Patch by Anthony Sottile.

Fix regression in `webbrowser` where default browsers may be preferred over browsers in the `BROWSER` environment variable.

Avoid stripping trailing whitespace in doctest fancy diff. Original patch by R. David Murray & Jairo Trad. Enhanced by Sanyam Khurana.

:func:`locale.localeconv` now sets temporarily the `LC_CTYPE` locale to the `LC_MONETARY` locale if the two locales are different and monetary strings are non-ASCII. This temporary change affects other threads.

Update ensurepip to install pip 18.1 and setuptools 40.6.2.

Adds IPv6 support when invoking http.server directly.

Recursively check arguments when testing for equality of :class:`unittest.mock.call` objects and add note that tracking of parameters used to create ancestors of mocks in mock_calls is not possible.

> **System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\(cpython-main) (Misc) (NEWS.d) 3.8.0a1.rst, line 2460);** *backlink*
>
> Unknown interpreted text role "class".

The warnings module now suggests to enable tracemalloc if the source is specified, the tracemalloc module is available, but tracemalloc is not tracing memory allocations.

Modify the following fnctl function to retry if interrupted by a signal (EINTR): flock, lockf, fnctl

Use add_done_callback() in sock_* asyncio API to unsubscribe reader/writer early on calcellation.

Removed the "built with" comment added when setup.py upload is used with either bdist_rpm or bdist_dumb.

Allow sending more than 2 GB at once on a multiprocessing connection on non-Windows systems.

Fix incorrect parsing of :class:`_io.IncrementalNewlineDecoder`'s *translate* argument.

> **System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\(cpython-main) (Misc) (NEWS.d) 3.8.0a1.rst, line 2522);** *backlink*
>
> Unknown interpreted text role "class".

Remove *StreamReaderProtocol._untrack_reader*. The call to *_untrack_reader* is currently performed too soon, causing the protocol to forget about the reader before *connection_lost* can run and feed the EOF to the reader.

ElementTree and minidom now preserve the attribute order specified by the user.

Improve difflib.SequenceManager.get_matching_blocks doc by adding 'non-overlapping' and changing '!=' to '<'.

Deprecated l*gettext() functions and methods in the :mod:`gettext` module. They return encoded bytes instead of Unicode strings and are artifacts from Python 2 times. Also deprecated functions and methods related to setting the charset for l*gettext() functions and methods.

> **System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\(cpython-main) (Misc) (NEWS.d) 3.8.0a1.rst, line 2563);** *backlink*
>
> Unknown interpreted text role "mod".

:meth:`socketserver.BaseServer.serve_forever` now exits immediately if it's :meth:`~socketserver.BaseServer.shutdown` method is called while it is polling for new events.

> **System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\(cpython-main) (Misc) (NEWS.d) 3.8.0a1.rst, line 2575);** *backlink*
>
> Unknown interpreted text role "meth".

> **System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\(cpython-main) (Misc) (NEWS.d) 3.8.0a1.rst, line 2575);** *backlink*
>
> Unknown interpreted text role "meth".

importlib no longer logs *wrote <bytecode path>* redundantly after *(created|could not create) <bytecode path>* is already logged. Patch by Quentin Agren.

unittest.mock now includes mock calls in exception messages if assert_not_called, assert_called_once, or assert_called_once_with fails. Patch by Petter Strandmark.

Fix ntpath.abspath regression where it didn't remove a trailing separator on Windows. Patch by Tim Graham.

tracemalloc now tries to update the traceback when an object is reused from a "free list" (optimization for faster object creation, used by the builtin list type for example).

Add the --json-lines option to json.tool. Patch by hongweipeng.

Fixed a leak in Tkinter when pass the Python wrapper around Tcl_Obj back to Tcl/Tk.

Enum: fix grandchildren subclassing when parent mixed with concrete data types.

:class:`unittest.mock.MagicMock` now supports the __fspath__ method (from :class:`os.PathLike`).

Fixed references leaks when call the `__setstate__()` method of :class:`xml.etree.ElementTree.Element` in the C implementation for already initialized element.

Verify the value for the parameter '-s' of the cProfile CLI. Patch by Robert Kuska

dataclasses now handle recursive reprs without raising RecursionError.

Make :func:`inspect.iscoroutinefunction`, :func:`inspect.isgeneratorfunction` and :func:`inspect.isasyncgenfunction` work with :func:`functools.partial`. Patch by Pablo Galindo.

Use :func:`socket.CMSG_SPACE` to calculate ancillary data size instead of :func:`socket.CMSG_LEN` in :func:`multiprocessing.reduction.recvfds` as RFC 3542 requires the use of the former for portable applications.

The *mailbox.mbox.get_string* function *from_* parameter can now successfully be set to a non-default value.

Protect tasks weak set manipulation in `asyncio.all_tasks()`

gzip: Add --fast, --best on the gzip CLI, these parameters will be used for the fast compression method (quick) or the best method compress (slower, but smaller file). Also, change the default compression level to 6 (tradeoff).

The :term:`2to3` :2to3fixer:`execfile` fixer now opens the file with mode `'rb'`. Patch by Zackery Spytz.

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\(cpython-main)(Misc)(NEWS.d)3.8.0a1.rst`, **line 2750**); *backlink*
>
> Unknown interpreted text role "term".

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\(cpython-main)(Misc)(NEWS.d)3.8.0a1.rst`, **line 2750**); *backlink*
>
> Unknown interpreted text role "2to3fixer".

:mod:`pydoc` now supports aliases not only to methods defined in the end class, but also to inherited methods. The docstring is not duplicated for aliases.

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\(cpython-main)(Misc)(NEWS.d)3.8.0a1.rst`, **line 2760**); *backlink*
>
> Unknown interpreted text role "mod".

:meth:`mimetypes.MimeTypes.guess_type` now accepts :term:`path-like object` in addition to url strings. Patch by Mayank Asthana.

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\(cpython-main)(Misc)(NEWS.d)3.8.0a1.rst`, **line 2771**); *backlink*
>
> Unknown interpreted text role "meth".

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\(cpython-main)(Misc)(NEWS.d)3.8.0a1.rst`, **line 2771**); *backlink*
>
> Unknown interpreted text role "term".

Add `moveto()` method to the `tkinter.Canvas` widget. Patch by Juliette Monsel.

Methods `find()`, `findtext()` and `findall()` of the `Element` class in the :mod:`xml.etree.ElementTree` module are now able to find children which are instances of `Element` subclasses.

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\(cpython-main)(Misc)(NEWS.d)3.8.0a1.rst`, **line 2791**); *backlink*
>
> Unknown interpreted text role "mod".

:class:`smtplib.SMTP` objects now always have a *sock* attribute present

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\(cpython-main)(Misc)(NEWS.d)3.8.0a1.rst`, **line 2802**); *backlink*
>
> Unknown interpreted text role "class".

Fix for async generators not finalizing when event loop is in debug mode and garbage collector runs in another thread.

Fix `TclError` in `tkinter.Spinbox.selection_element()`. Patch by Juliette Monsel.

Add methods `selection_from`, `selection_range`, `selection_present` and `selection_to` to the `tkinter.Spinbox` for consistency with the `tkinter.Entry` widget. Patch by Juliette Monsel.

Added *secure_protocols* argument to *http.cookiejar.DefaultCookiePolicy* to allow for tweaking of protocols and also to add support by default for *wss*, the secure websocket protocol.

Fixed integer overflow in the :meth:`~hashlib.shake.digest()` and :meth:`~hashlib.shake.hexdigest()` methods for the SHAKE algorithm in the :mod:`hashlib` module.

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\(cpython-main)(Misc)(NEWS.d)3.8.0a1.rst`, **line 2853**); *backlink*
>
> Unknown interpreted text role "meth".

25% speedup in argument parsing for the functions in the bisect module.

Fixed :meth:`unittest.TestCase.debug` when used to call test methods with subtests. Patch by Bruno Oliveira.

logging.Formatter enhancement - Ensure styles and fmt matches in logging.Formatter - Added validate method in each format style class: StrFormatStyle, PercentStyle, StringTemplateStyle. - This method is called in the constructor of logging.Formatter class - Also re-raise the KeyError in the format method of each style class, so it would a bit clear that it's an error with the invalid format fields.

Adjust test.support.missing_compiler_executable check so that a nominal command name of "" is ignored. Patch by Michael Felt.

Fix inspect module polluted `sys.modules` when parsing `__text_signature__` of callable.

Add *mtime* argument to *gzip.compress* for reproducible output. Patch by Guo Ci Teo.

On Cygwin and MinGW, ensure that `sys.executable` always includes the full filename in the path, including the `.exe` suffix (unless it is a symbolic link).

Adding `max_num_fields` to `cgi.FieldStorage` to make DOS attacks harder by limiting the number of `MiniFieldStorage` objects created by `FieldStorage`.

http.server ensures it reports HTTPStatus.NOT_FOUND when the local path ends with "/" and is not a directory, even if the underlying OS (e.g. AIX) accepts such paths as a valid file reference. Patch by Michael Felt.

Fix self-cancellation in C implementation of asyncio.Task

Don't log waiting for `selector.select` in asyncio loop iteration. The waiting is pretty normal for any asyncio program, logging its time just adds a noise to logs without any useful information provided.

The :envvar:`SOURCE_DATE_EPOCH` environment variable no longer overrides the value of the *invalidation_mode* argument to :func:`py_compile.compile`, and determines its default value instead.

Use a monotonic clock to compute timeouts in :meth:`Executor.map` and :func:`as_completed`, in order to prevent timeouts from deviating when the system clock is adjusted.

Add .wasm -> application/wasm to list of recognized file types and content type headers

:func:`xml.sax.make_parser` now accepts any iterable as its *parser_list* argument. Patch by André Delfino.

In :class:`QueueHandler`, clear *exc_text* from :class:`LogRecord` to prevent traceback from being written twice.

On Windows, asyncio now uses ProactorEventLoop, instead of SelectorEventLoop, by default.

Support reading zip files with archive comments in :mod:`zipimport`.

The parser now represents all constants as :class:`ast.Constant` instead of using specific constant AST types (`Num`, `Str`, `Bytes`, `NameConstant` and `Ellipsis`). These classes are considered deprecated and will be removed in future Python versions.

Add deprecation warning when *loop* is used in methods: *asyncio.sleep*, *asyncio.wait* and *asyncio.wait_for*.

ZIP files created by :mod:`distutils` will now include entries for directories.

Add an optional *initial* argument to itertools.accumulate().

Support multiple mixin classes when creating Enums.

Add SSLContext.post_handshake_auth and SSLSocket.verify_client_post_handshake for TLS 1.3's post handshake authentication feature.

The Activate.ps1 script from venv works with PowerShell Core 6.1 and is now available under all operating systems.

Fix bug that prevented using :meth:`reset_mock <unittest.mock.Mock.reset_mock>` on mock instances with deleted attributes

Add a workaround, so the `'z'` :func:`time.strftime` specifier on the musl C library can work in some cases.

Implement `asyncio.StreamWriter.awrite` and `asyncio.StreamWriter.aclose()` coroutines. Methods are needed for

providing a consistent stream API with control flow switched on by default.

Acquire the logging module's commonly used internal locks while fork()ing to avoid deadlocks in the child process.

Fix a rare interpreter unhandled exception state SystemError only seen when using subprocess with a preexec_fn while an after_parent handler has been registered with os.register_at_fork and the fork system call fails.

Ensure :func:`os.lchmod` is never defined on Linux.

Store a weak reference to stream reader to break strong references loop between reader and protocol. It allows to detect and close the socket if the stream is deleted (garbage collected) without `close()` call.

*Enum._missing_*: raise *ValueError* if None returned and *TypeError* if non-member is returned.

Speed up re scanning of many non-matching characters for s w and d within bytes objects. (microoptimization)

Add :func:`~unittest.addModuleCleanup()` and :meth:`~unittest.TestCase.addClassCleanup()` to unittest to support cleanups for :func:`~unittest.setUpModule()` and :meth:`~unittest.TestCase.setUpClass()`. Patch by Lisa Roach.

Don't log SSL certificate errors in asyncio code (connection error logging is skipped already).

Prevent filename duplication in :mod:`subprocess` exception messages. Patch by Zackery Spytz.

dataclasses.asdict() and .astuple() now handle namedtuples correctly.

Update vendorized expat library version to 2.2.6.

The subprocess module no longer mistakenly closes redirected fds even when they were in pass_fds when outside of the default {0, 1, 2} set.

Create a dedicated `asyncio.CancelledError`, `asyncio.InvalidStateError` and `asyncio.TimeoutError` exception classes. Inherit them from corresponding exceptions from `concurrent.futures` package. Extract `asyncio` exceptions into a separate file.

Fixed iterator of :class:`multiprocessing.managers.DictProxy`.

Fix distutils logging for non-ASCII strings. This caused installation issues on Windows.

Fix possible mojibake in the error message of *pwd.getpwnam* and *grp.getgrnam* using string representation because of invisible characters or trailing whitespaces. Patch by William Grzybowski.

Make uuid.UUID use `__slots__` to reduce its memory footprint. Based on original patch by Wouter Bolsterlee.

OrderedDict iterators are not exhausted during pickling anymore. Patch by Sergey Fedoseev.

Refactored :mod:`subprocess` to check for Windows-specific modules rather than `sys.platform == 'win32'`.

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\(cpython-main)(Misc)(NEWS.d)3.8.0a1.rst`, **line 3336**); *backlink*
>
> Unknown interpreted text role "mod".

`distutils.spawn.find_executable()` now falls back on :data:`os.defpath` if the `PATH` environment variable is not set.

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\(cpython-main)(Misc)(NEWS.d)3.8.0a1.rst`, **line 3346**); *backlink*
>
> Unknown interpreted text role "data".

On Windows, fix multiprocessing.Connection for very large read: fix _winapi.PeekNamedPipe() and _winapi.ReadFile() for read larger than INT_MAX (usually `2**31-1`).

Correct typo in Lib/ctypes/_aix.py

Move `Enum._convert` to `EnumMeta._convert_` and fix enum members getting shadowed by parent attributes.

When the queue is closed, :exc:`ValueError` is now raised by :meth:`multiprocessing.Queue.put` and :meth:`multiprocessing.Queue.get` instead of :exc:`AssertionError` and :exc:`OSError`, respectively. Patch by Zackery Spytz.

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\(cpython-main)(Misc)(NEWS.d)3.8.0a1.rst`, **line 3386**); *backlink*
>
> Unknown interpreted text role "exc".

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\(cpython-main)(Misc)(NEWS.d)3.8.0a1.rst`, **line 3386**); *backlink*
>
> Unknown interpreted text role "meth".

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\(cpython-main)(Misc)(NEWS.d)3.8.0a1.rst`, **line 3386**); *backlink*
>
> Unknown interpreted text role "meth".

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\(cpython-main)(Misc)(NEWS.d)3.8.0a1.rst`, **line 3386**); *backlink*
>
> Unknown interpreted text role "exc".

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\(cpython-main)(Misc)(NEWS.d)3.8.0a1.rst`, **line 3386**); *backlink*
>
> Unknown interpreted text role "exc".

Fix parsing non-ASCII identifiers in :mod:`lib2to3.pgen2.tokenize` (PEP 3131).

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\(cpython-main)(Misc)(NEWS.d)3.8.0a1.rst`, **line 3398**); *backlink*
>
> Unknown interpreted text role "mod".

Avoids a possible integer underflow (undefined behavior) in the time module's year handling code when passed a very low negative year value.

Improved compatibility for streamed files in :mod:`zipfile`. Previously an optional signature was not being written and certain ZIP applications were not supported. Patch by Silas Sewell.

Fix the .fromisoformat() methods of datetime types crashing when given unicode with non-UTF-8-encodable code points. Specifically, datetime.fromisoformat() now accepts surrogate unicode code points used as the separator. Report and tests by Alexey Izbyshev, patch by Paul Ganssle.

Fix inspect.getsourcelines for module level frames/tracebacks. Patch by Vladimir Matveev.

Running the :mod:`trace` module no longer creates the `trace.cover` file.

Fix crash when an `ABC`-derived class with invalid `__subclasses__` is passed as the second argument to :func:`issubclass()`. Patch by Alexey Izbyshev.

Fix infinite loop in `a.extend(a)` for `MutableSequence` subclasses.

Make :func:`signal.strsignal` work on HP-UX. Patch by Michael Osipov.

shutil.copytree now accepts a new `dirs_exist_ok` keyword argument. Patch by Josh Bronson.

Associate `.mjs` file extension with `application/javascript` MIME Type.

:func:`os.readlink` now accepts :term:`path-like <path-like object>` and :class:`bytes` objects on Windows.

The UTF-7 decoder now raises :exc:`UnicodeDecodeError` for ill-formed sequences starting with "+" (as specified in RFC 2152). Patch by Zackery Spytz.

The :meth:`mmap.flush() <mmap.mmap.flush>` method now returns `None` on success, raises an exception on error under all platforms.

Appending to the ZIP archive with the ZIP64 extension no longer grows the size of extra fields of existing entries.

Fix %-formatting in :meth:`pathlib.PurePath.with_suffix` when formatting an error message.

The :class:`imaplib.IMAP4` and :class:`imaplib.IMAP4_SSL` classes now resolve to the local host IP correctly when the default value of *host* parameter (`''`) is used.

Implement `traceback.FrameSummary.__len__()` method to preserve compatibility with the old tuple API.

:func:`~unittest.TestCase.assertRaises`, :func:`~unittest.TestCase.assertRaisesRegex`, :func:`~unittest.TestCase.assertWarns` and :func:`~unittest.TestCase.assertWarnsRegex` no longer success if the passed callable is None. They no longer ignore unknown keyword arguments in the context manager mode. A DeprecationWarning was raised in these cases since Python 3.5.

Deprecate :meth:`__getitem__` methods of :class:`xml.dom.pulldom.DOMEventStream`, :class:`wsgiref.util.FileWrapper` and :class:`fileinput.FileInput`.

Fix a race condition in `multiprocessing.semaphore_tracker` when the tracker receives SIGINT before it can register signal handlers for ignoring it.

Report filename in the exception raised when the database file cannot be opened by :func:`dbm.gnu.open` and :func:`dbm.ndbm.open` due to OS-related error. Patch by Zsolt Cserna.

Add math.dist() to compute the Euclidean distance between two points.

:meth:`smtplib.SMTP.send_message` no longer modifies the content of the *mail_options* argument. Patch by Pablo S. Blum de Aguiar.

Fix `ntpath.abspath` for invalid paths on windows. Patch by Franz Woellert.

Add pure Python fallback for functools.reduce. Patch by Robert Wright.

The default asyncio task class now always has a name which can be get or set using two new methods (:meth:`~asyncio.Task.get_name()` and :meth:`~asyncio.Task.set_name`) and is visible in the :func:`repr` output. An initial name can also be set using the new `name` keyword argument to :func:`asyncio.create_task` or the :meth:`~asyncio.AbstractEventLoop.create_task` method of the event loop. If no initial name is set, the default Task implementation generates a name like `Task-1` using a monotonic counter.

Unknown interpreted text role "meth".

asyncio's event loop will not pass timeouts longer than one day to epoll/select etc.

Fix several AttributeError in zipfile seek() methods. Patch by Mickaël Schoentgen.

Fix performance regression in :mod:`sqlite3` when a DML statement appeared in a different line than the rest of the SQL query.

Deprecate passing non-ThreadPoolExecutor instances to :meth:`AbstractEventLoop.set_default_executor`.

Restore `msilib.Win64` to preserve backwards compatibility since it's already used by :mod:`distutils`` bdist_msi` command.

Ignore errors caused by missing / non-writable homedir while writing history during exit of an interactive session. Patch by Anthony Sottile.

Enhanced math.hypot() to support more than two dimensions.

tracemalloc: PYTHONTRACEMALLOC=0 environment variable and -X tracemalloc=0 command line option are now allowed to disable explicitly tracemalloc at startup.

Use :func:`shutil.get_terminal_size` to calculate the terminal width correctly in the `argparse.HelpFormatter` class. Initial patch by Zbyszek Jędrzejewski-Szmek.

Allow frozen dataclasses to have a field named "object". Previously this conflicted with an internal use of "object".

:meth:`sqlite3.Connection.create_aggregate`, :meth:`sqlite3.Connection.create_function`, :meth:`sqlite3.Connection.set_authorizer`, :meth:`sqlite3.Connection.set_progress_handler` methods raises TypeError when unhashable objects are passed as callable. These methods now don't pass such objects to SQLite API. Previous behavior could lead to segfaults. Patch by Sergey Fedoseev.

Attributes *skipinitialspace*, *doublequote* and *strict* of the *dialect* attribute of the :mod:`csv` reader are now :class:`bool` instances instead of integers 0 or 1.

> **System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\(cpython-main) (Misc) (NEWS.d) 3.8.0a1.rst`, line 3798);** *backlink*
>
> Unknown interpreted text role "mod".

> **System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\(cpython-main) (Misc) (NEWS.d) 3.8.0a1.rst`, line 3798);** *backlink*
>
> Unknown interpreted text role "class".

Errors other than :exc:`TypeError` raised in methods `__adapt__()` and `__conform__()` in the :mod:`sqlite3` module are now propagated to the user.

> **System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\(cpython-main) (Misc) (NEWS.d) 3.8.0a1.rst`, line 3809);** *backlink*
>
> Unknown interpreted text role "exc".

> **System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\(cpython-main) (Misc) (NEWS.d) 3.8.0a1.rst`, line 3809);** *backlink*
>
> Unknown interpreted text role "mod".

The :2to3fixer:`reload` fixer now uses :func:`importlib.reload` instead of deprecated :func:`imp.reload`.

> **System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\(cpython-main) (Misc) (NEWS.d) 3.8.0a1.rst`, line 3820);** *backlink*
>
> Unknown interpreted text role "2to3fixer".

> **System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\(cpython-main) (Misc) (NEWS.d) 3.8.0a1.rst`, line 3820);** *backlink*
>
> Unknown interpreted text role "func".

> **System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\(cpython-main) (Misc) (NEWS.d) 3.8.0a1.rst`, line 3820);** *backlink*
>
> Unknown interpreted text role "func".

pydoc's `Helper.showtopic()` method now prints the cross references of a topic correctly.

:func:`base64.b32decode` could raise UnboundLocalError or OverflowError for incorrect padding. Now it always raises :exc:`base64.Error` in these cases.

> **System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\(cpython-main) (Misc) (NEWS.d) 3.8.0a1.rst`, line 3840);** *backlink*
>
> Unknown interpreted text role "func".

> **System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\(cpython-main) (Misc) (NEWS.d) 3.8.0a1.rst`, line 3840);** *backlink*
>
> Unknown interpreted text role "exc".

Fixed issues with arguments parsing in :mod:`hashlib`.

> **System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\(cpython-main) (Misc) (NEWS.d) 3.8.0a1.rst`, line 3850);** *backlink*
>
> Unknown interpreted text role "mod".

ZipFile can zip files older than 1980-01-01 and newer than 2107-12-31 using a new `strict_timestamps` parameter at the cost of

setting the timestamp to the limit.

Remove extraneous CR in 2to3 refactor.

Make sure to only check if the handle is a tty, when opening a file with `buffering=-1`.

Reverted :issue:`27494`. 2to3 rejects now a trailing comma in generator expressions.

functools.singledispatch now raises TypeError instead of IndexError when no positional arguments are passed.

Add the parameter *deterministic* to the :meth:`sqlite3.Connection.create_function` method. Patch by Sergey Fedoseev.

Ensure the loader shim created by `imp.load_module` always returns bytes from its `get_data()` function. This fixes using `imp.load_module` with PEP 552 hash-based pycs.

The multiprocessing module now uses the monotonic clock :func:`time.monotonic` instead of the system clock :func:`time.time` to implement timeout.

Optimize tarfile uncompress performance about 15% when gzip is used.

`subprocess.Popen` now copies the *startupinfo* argument to leave it unchanged: it will modify the copy, so that the same `STARTUPINFO` object can be used multiple times.

Fixed a performance regression for reading streams with tarfile. The buffered read should use a list, instead of appending to a bytes object.

webbrowser: Correct the arguments passed to Opera Browser when opening a new URL using the `webbrowser` module. Patch by Bumsik Kim.

csv.DictReader now creates dicts instead of OrderedDicts. Patch by Michael Selik.

Closed existing logging handlers before reconfiguration via fileConfig and dictConfig. Patch by Karthikeyan Singaravelan.

Make minor tweaks to turtledemo. The 'wikipedia' example is now 'rosette', describing what it draws. The 'penrose' print output is reduced. The '1024' output of 'tree' is eliminated.

Fixed passing lists and tuples of strings containing special characters `"`, `\`, `{`, `}` and `\n` as options to :mod:`~tkinter.ttk` widgets.

Fix getaddrinfo to resolve IPv6 addresses correctly.

Improve random.choices() to handle subnormal input weights that could occasionally trigger an IndexError.

Fixed integer overflow in :func:`os.readv`, :func:`os.writev`, :func:`os.preadv` and :func:`os.pwritev` and in :func:`os.sendfile` with *headers* or *trailers* arguments (on BSD-based OSes and macOS).

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\(cpython-main)(Misc)(NEWS.d)3.8.0a1.rst`, line 4042); *backlink*

Unknown interpreted text role "func".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\(cpython-main)(Misc)(NEWS.d)3.8.0a1.rst`, line 4042); *backlink*

Unknown interpreted text role "func".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\(cpython-main)(Misc)(NEWS.d)3.8.0a1.rst`, line 4042); *backlink*

Unknown interpreted text role "func".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\(cpython-main)(Misc)(NEWS.d)3.8.0a1.rst`, line 4042); *backlink*

Unknown interpreted text role "func".

Add :func:`copy.copy` and :func:`copy.deepcopy` support to zlib compressors and decompressors. Patch by Zackery Spytz.

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\(cpython-main)(Misc)(NEWS.d)3.8.0a1.rst`, line 4053); *backlink*

Unknown interpreted text role "func".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\(cpython-main)(Misc)(NEWS.d)3.8.0a1.rst`, line 4053); *backlink*

Unknown interpreted text role "func".

multiprocessing: Fix a race condition in Popen of multiprocessing.popen_spawn_win32. The child process now duplicates the read end of pipe instead of "stealing" it. Previously, the read end of pipe was "stolen" by the child process, but it leaked a handle if the child process had been terminated before it could steal the handle from the parent process.

Tokenize module now implicitly emits a NEWLINE when provided with input that does not have a trailing new line. This behavior now matches what the C tokenizer does internally. Contributed by Ammar Askar.

Added a 'force' keyword argument to logging.basicConfig().

:func:`shutil.copytree` uses :func:`os.scandir` function and all copy functions depending from it use cached :func:`os.stat` values. The speedup for copying a directory with 8000 files is around +9% on Linux, +20% on Windows and + 30% on a Windows SMB share. Also the number of :func:`os.stat` syscalls is reduced by 38% making :func:`shutil.copytree` especially faster on network filesystems. (Contributed by Giampaolo Rodola' in :issue:`33695`.)

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\(cpython-main)(Misc)(NEWS.d)3.8.0a1.rst`, line 4097); *backlink*

Unknown interpreted text role "func".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\(cpython-main)(Misc)(NEWS.d)3.8.0a1.rst`, line 4097); *backlink*

Unknown interpreted text role "func".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\(cpython-main)(Misc)(NEWS.d)3.8.0a1.rst`, line 4097); *backlink*

Unknown interpreted text role "func".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\(cpython-main)(Misc)(NEWS.d)3.8.0a1.rst`, line 4097); *backlink*

Unknown interpreted text role "func".

bz2 and lzma: When Decompressor.__init__() is called twice, free the old lock to not leak memory.

Make select.epoll() and its documentation consistent regarding *sizehint* and *flags*.

Fixed bug in asyncio where ProactorSocketTransport logs AssertionError if force closed during write.

Convert content length to string before putting to header.

:mod:`os.path` functions that return a boolean result like :func:`~os.path.exists`, :func:`~os.path.lexists`, :func:`~os.path.isdir`, :func:`~os.path.isfile`, :func:`~os.path.islink`, and :func:`~os.path.ismount`, and :mod:`pathlib.Path` methods that return a boolean result like :meth:`~pathlib.Path.exists()`, :meth:`~pathlib.Path.is_dir()`, :meth:`~pathlib.Path.is_file()`, :meth:`~pathlib.Path.is_mount()`, :meth:`~pathlib.Path.is_symlink()`, :meth:`~pathlib.Path.is_block_device()`, :meth:`~pathlib.Path.is_char_device()`, :meth:`~pathlib.Path.is_fifo()`, :meth:`~pathlib.Path.is_socket()` now return `False` instead of raising :exc:`ValueError` or its subclasses :exc:`UnicodeEncodeError` and :exc:`UnicodeDecodeError` for paths that contain characters or bytes unrepresentable at the OS level.

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\(cpython-main) (Misc) (NEWS.d)3.8.0a1.rst`, line 4151); *backlink*

Unknown interpreted text role "meth".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\(cpython-main) (Misc) (NEWS.d)3.8.0a1.rst`, line 4151); *backlink*

Unknown interpreted text role "meth".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\(cpython-main) (Misc) (NEWS.d)3.8.0a1.rst`, line 4151); *backlink*

Unknown interpreted text role "meth".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\(cpython-main) (Misc) (NEWS.d)3.8.0a1.rst`, line 4151); *backlink*

Unknown interpreted text role "meth".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\(cpython-main) (Misc) (NEWS.d)3.8.0a1.rst`, line 4151); *backlink*

Unknown interpreted text role "meth".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\(cpython-main) (Misc) (NEWS.d)3.8.0a1.rst`, line 4151); *backlink*

Unknown interpreted text role "meth".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\(cpython-main) (Misc) (NEWS.d)3.8.0a1.rst`, line 4151); *backlink*

Unknown interpreted text role "meth".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\(cpython-main) (Misc) (NEWS.d)3.8.0a1.rst`, line 4151); *backlink*

Unknown interpreted text role "meth".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\(cpython-main) (Misc) (NEWS.d)3.8.0a1.rst`, line 4151); *backlink*

Unknown interpreted text role "meth".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\(cpython-main) (Misc) (NEWS.d)3.8.0a1.rst`, line 4151); *backlink*

Unknown interpreted text role "exc".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\(cpython-main) (Misc) (NEWS.d)3.8.0a1.rst`, line 4151); *backlink*

Unknown interpreted text role "exc".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\(cpython-main) (Misc) (NEWS.d)3.8.0a1.rst`, line 4151); *backlink*

Unknown interpreted text role "exc".

Fixed implementation of :func:`platform.libc_ver`. It almost always returned version '2.9' for glibc.

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\(cpython-main) (Misc) (NEWS.d)3.8.0a1.rst`, line 4172); *backlink*

Unknown interpreted text role "func".

Remove deprecated `cgi.escape`, `cgi.parse_qs` and `cgi.parse_qsl`.

Remove `tarfile.filemode` which is deprecated since Python 3.3.

Prevent site.main() exception if PYTHONSTARTUP is set. Patch by Steve Weber.

Improve error message of dataclasses.replace() when an InitVar is not specified

Fix the call to `os.chmod()` for `uu.decode()` if a mode is given or decoded. Patch by Timo Furrer.

Datetime instance d with non-None tzinfo, but with d.tzinfo.utcoffset(d) returning None is now treated as naive by the astimezone() method.

In configparser, don't clear section when it is assigned to itself.

Make email module properly handle invalid-length base64 strings.

Implement multibyte encoder/decoder state methods

Avoid race condition with debug logging

Fix _header_value_parser.py when address group is missing final ';'. Contributed by Enrique Perez-Terron

asyncio: Fix a race condition causing data loss on pause_reading()/resume_reading() when using the ProactorEventLoop.

Correct test for `uuid_enc_be` availability in `configure.ac`. Patch by Michael Felt.

Add asyncio.WindowsSelectorEventLoopPolicy and asyncio.WindowsProactorEventLoopPolicy.

W3C DOM Level 1 specifies return value of Element.removeAttributeNode() as "The Attr node that was removed." xml.dom.minidom now complies with this requirement.

Update `unicodedata`'s database to Unicode version 11.0.0.

Added a stacklevel parameter to logging calls to allow use of wrapper/helper functions for logging APIs.

improve base64 exception message for encoded inputs of invalid length

asyncio/start_tls: Fix error message; cancel callbacks in case of an unhandled error; mark SSLTransport as closed if it is aborted.

The concatenation (+) and repetition (*) sequence operations now raise :exc:`TypeError` instead of :exc:`SystemError` when performed on :class:`mmap.mmap` objects. Patch by Zackery Spytz.

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\(cpython-main)(Misc)(NEWS.d)3.8.0a1.rst`, line 4364); *backlink*
>
> Unknown interpreted text role "exc".

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\(cpython-main)(Misc)(NEWS.d)3.8.0a1.rst`, line 4364); *backlink*
>
> Unknown interpreted text role "exc".

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\(cpython-main)(Misc)(NEWS.d)3.8.0a1.rst`, line 4364); *backlink*
>
> Unknown interpreted text role "class".

asyncio/ssl: Fix AttributeError, increase default handshake timeout

Fixed creating a controller for :mod:`webbrowser` when a user specifies a path to an entry in the BROWSER environment variable. Based on patch by John Still.

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\(cpython-main)(Misc)(NEWS.d)3.8.0a1.rst`, line 4384); *backlink*
>
> Unknown interpreted text role "mod".

Add gettext.pgettext() and variants.

Add description property for _ParameterKind

When cancelling the task due to a timeout, :meth:`asyncio.wait_for` will now wait until the cancellation is complete.

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-`

main\Misc\NEWS.d\(cpython-main)(Misc)(NEWS.d)3.8.0a1.rst, line 4413); *backlink*

Unknown interpreted text role "meth".

Fix gather to propagate cancellation of itself even with return_exceptions.

Support protocol type switching in SSLTransport.set_protocol().

Pause the transport as early as possible to further reduce the risk of data_received() being called before connection_made().

:func:`shutil.copyfile`, :func:`shutil.copy`, :func:`shutil.copy2`, :func:`shutil.copytree` and :func:`shutil.move` use platform-specific fast-copy syscalls on Linux and macOS in order to copy the file more efficiently. On Windows :func:`shutil.copyfile` uses a bigger default buffer size (1 MiB instead of 16 KiB) and a :func:`memoryview`-based variant of :func:`shutil.copyfileobj` is used. The speedup for copying a 512MiB file is about +26% on Linux, +50% on macOS and +40% on Windows. Also, much less CPU cycles are consumed. (Contributed by Giampaolo Rodola' in :issue:`25427`.)

> **System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\(cpython-main)(Misc)(NEWS.d)3.8.0a1.rst**, line 4451); *backlink*
>
> Unknown interpreted text role "func".

> **System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\(cpython-main)(Misc)(NEWS.d)3.8.0a1.rst**, line 4451); *backlink*
>
> Unknown interpreted text role "func".

> **System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\(cpython-main)(Misc)(NEWS.d)3.8.0a1.rst**, line 4451); *backlink*
>
> Unknown interpreted text role "func".

> **System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\(cpython-main)(Misc)(NEWS.d)3.8.0a1.rst**, line 4451); *backlink*
>
> Unknown interpreted text role "func".

> **System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\(cpython-main)(Misc)(NEWS.d)3.8.0a1.rst**, line 4451); *backlink*
>
> Unknown interpreted text role "func".

> **System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\(cpython-main)(Misc)(NEWS.d)3.8.0a1.rst**, line 4451); *backlink*
>
> Unknown interpreted text role "func".

> **System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\(cpython-main)(Misc)(NEWS.d)3.8.0a1.rst**, line 4451); *backlink*
>
> Unknown interpreted text role "func".

> **System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\(cpython-main)(Misc)(NEWS.d)3.8.0a1.rst**, line 4451); *backlink*
>
> Unknown interpreted text role "func".

> **System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\(cpython-main)(Misc)(NEWS.d)3.8.0a1.rst**, line 4451); *backlink*
>
> Unknown interpreted text role "issue".

Fix a race condition in SSLProtocol.connection_made() of asyncio.sslproto: start immediately the handshake instead of using call_soon(). Previously, data_received() could be called before the handshake started, causing the handshake to hang or fail.

Fixed bug where calling write_eof() on a _SelectorSocketTransport after it's already closed raises AttributeError.

Make asyncio.all_tasks() return only pending tasks.

Avoid blocking on file IO in sendfile fallback code

Fix RuntimeError after closing loop that used run_in_executor

Fix Task.__repr__ crash with Cython's bogus coroutines

Fix transport.set_protocol() to support switching between asyncio.Protocol and asyncio.BufferedProtocol. Fix loop.start_tls() to work with asyncio.BufferedProtocols.

Pickles of type variables and subscripted generics are now future-proof and compatible with older Python versions.

Fixed :func:`uuid.uuid1` on FreeBSD.

> **System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\(cpython-main) (Misc) (NEWS.d) 3.8.0a1.rst, line 4547);** *backlink*
>
> Unknown interpreted text role "func".

Add `InvalidStateError` to :mod:`concurrent.futures`. `Future.set_result` and `Future.set_exception` now raise `InvalidStateError` if the futures are not pending or running. Patch by Jason Haydaman.

> **System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\(cpython-main) (Misc) (NEWS.d) 3.8.0a1.rst, line 4556);** *backlink*
>
> Unknown interpreted text role "mod".

Finalize and document preliminary and experimental TLS 1.3 support with OpenSSL 1.1.1

Release GIL on *grp.getgrnam*, *grp.getgrgid*, *pwd.getpwnam* and *pwd.getpwuid* if reentrant variants of these functions are available. Patch by William Grzybowski.

Fix possible SIGSGV when asyncio.Future is created in __del__

Use a better regex when breaking usage into wrappable parts. Avoids bogus assertion errors from custom metavar strings.

Fixed a bug in the Python implementation of the JSON decoder that prevented the cache of parsed strings from clearing after finishing the decoding. Based on patch by c-fos.

Remove HMAC default to md5 marked for removal in 3.8 (removal originally planned in 3.6, bump to 3.8 in gh-7062).

Emit a deprecation warning for inspect.formatargspec

Add `functools.cached_property` decorator, for computed properties cached for the life of the instance.

Change TLS 1.3 cipher suite settings for compatibility with OpenSSL 1.1.1-pre6 and newer. OpenSSL 1.1.1 will have TLS 1.3 ciphers enabled by default.

Do not simplify arguments to *typing.Union*. Now *Union[Manager, Employee]* is not simplified to *Employee* at runtime. Such simplification previously caused several bugs and limited possibilities for introspection.

:func:`tokenize.generate_tokens` is now documented as a public API to tokenize unicode strings. It was previously present but undocumented.

> **System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\(cpython-main) (Misc) (NEWS.d) 3.8.0a1.rst, line 4670);** *backlink*
>
> Unknown interpreted text role "func".

Add a new `block_on_close` class attribute to `ForkingMixIn` and `ThreadingMixIn` classes of :mod:`socketserver`.

> **System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\(cpython-main) (Misc) (NEWS.d) 3.8.0a1.rst, line 4680);** *backlink*
>
> Unknown interpreted text role "mod".

tempfile._candidate_tempdir_list should consider common TEMP locations

argparse subparsers are once again not required by default, reverting the change in behavior introduced by bpo-26510 in 3.7.0a2.

Remove unused private method `_strptime.LocaleTime.__pad` (a.k.a. `_LocaleTime__pad`).

dataclasses.make_dataclass now checks for invalid field names and duplicate fields. Also, added a check for invalid field specifications.

Prevent `uuid.get_node` from using a DUID instead of a MAC on Windows. Patch by Zvi Effron

Fix race condition with *ReadTransport.resume_reading* in Windows proactor event loop.

Fix failure in *typing.get_type_hints()* when ClassVar was provided as a string forward reference.

:class:`unittest.mock.MagicMock` now supports the `__round__` magic method.

Added support for Site Maps to urllib's `RobotFileParser` as :meth:`RobotFileParser.site_maps() <urllib.robotparser.RobotFileParser.site_maps>`. Patch by Lady Red, based on patch by Peter Wirtz.

Remove platform.linux_distribution, which was deprecated since 3.5.

Switch the default dictionary implementation for :mod:`configparser` from :class:`collections.OrderedDict` to the standard :class:`dict` type.

Optimize asyncio.ensure_future() by reordering if checks: 1.17x faster.

Add errors param to cgi.parse_multipart and make an encoding in FieldStorage use the given errors (needed for Twisted). Patch by Amber Brown.

The :class:`cProfile.Profile` class can now be used as a context manager. Patch by Scott Sanderson.

Change dataclasses.Fields repr to use the repr of each of its members, instead of str. This makes it more clear what each field actually represents. This is especially true for the 'type' member.

Correct `inspect.isdatadescriptor` to look for `__set__` or `__delete__`. Patch by Aaron Hall.

Removed the `doctype()` method and the *html* parameter of the constructor of :class:`~xml.etree.ElementTree.XMLParser`. The `doctype()` method defined in a subclass will no longer be called. Deprecated methods `getchildren()` and `getiterator()` in the :mod:`~xml.etree.ElementTree` module emit now a :exc:`DeprecationWarning` instead of :exc:`PendingDeprecationWarning`.

Fix dataclasses to work if using literal string type annotations or if using PEP 563 "Postponed Evaluation of Annotations". Only specific string prefixes are detected for both ClassVar ("ClassVar" and "typing.ClassVar") and InitVar ("InitVar" and "dataclasses.InitVar").

Minor fixes in typing module: add annotations to `NamedTuple.__new__`, pass `*args` and `**kwds` in `Generic.__new__`. Original PRs by Paulius Å arka and Chad Dombrova.

Print the header values besides the header keys instead just the header keys if *debuglevel* is set to >0 in :mod:`http.client`. Patch by Marco Strigl.

Updated alias mapping with glibc 2.27 supported locales.

Fix trailing quotation marks getting deleted when looking up byte/string literals on pydoc. Patch by André Delfino.

The function `platform.linux_distribution` and `platform.dist` now trigger a `DeprecationWarning` and have been marked for removal in Python 3.8

Fix ctypes.util.find_library regression on macOS.

Text and html output generated by cgitb does not display parentheses if the current call is done directly in the module. Patch by Stéphane Blondon.

The file classes in *tempfile* now accept an *errors* parameter that complements the already existing *encoding*. Patch by Stephan Hohe.

:func:`unittest.mock.mock_open` now supports iteration over the file contents. Patch by Tony Flury.

Raise :exc:`TypeError` when looking up non-Enum objects in Enum classes and Enum members.

Update error message when constructing invalid inspect.Parameters Patch by Dong-hee Na.

Fixed crash in the get() method of the :mod:`dbm.ndbm` database object when it is called with a single argument.

The warnings module now finds the Python file associated with a warning from the code object, rather than the frame's global namespace. This is consistent with how tracebacks and pdb find filenames, and should work better for dynamically executed code.

`imaplib` now allows `MOVE` command in `IMAP4.uid()` (RFC 6851: IMAP MOVE Extension) and potentially as a name of supported method of `IMAP4` object.

Added *jump* parameter to :func:`dis.stack_effect`.

Rename and deprecate undocumented functions in :func:`urllib.parse`.

Add signal.valid_signals() to expose the POSIX sigfillset() functionality.

ConfigParser.items() was fixed so that key-value pairs passed in via *vars* are not included in the resulting output.

Fix multiprocessing regression on newer glibcs

:func:`dis.stack_effect` now supports all defined opcodes including NOP and EXTENDED_ARG.

Fix quoting of the Comment attribute of :class:`http.cookies.SimpleCookie`.

Upgrade bundled version of pip to 10.0.1.

Fixed a crash in the :mod:`parser` module when converting an ST object to a tree of tuples or lists with line_info=False and col_info=True.

lib2to3 now uses pickle protocol 4 for pre-computed grammars.

lib2to3 now recognizes rf'...' strings.

Ensure line-endings are respected when using lib2to3.

Have :func:`importlib.resources.contents` and :meth:`importlib.abc.ResourceReader.contents` return an :term:`iterable` instead of an :term:`iterator`.

`contextlib.ExitStack` and `contextlib.AsyncExitStack` now use a method instead of a wrapper function for exit callbacks.

Fix FD leak in *_SelectorSocketTransport* Patch by Vlad Starostin.

Fix display of `<module>` call in the html produced by `cgitb.html()`. Patch by StÃ©phane Blondon.

`random.Random()` and its subclassing mechanism got optimized to check only once at class/subclass instantiation time whether its `getrandbits()` method can be relied on by other methods, including `randrange()`, for the generation of arbitrarily large random integers. Patch by Wolfgang Maier.

Fixed regression when running pydoc with the :option:`-m` switch. (The regression was introduced in 3.7.0b3 by the resolution of :issue:`33053`)

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\(cpython-main)(Misc)(NEWS.d)3.8.0a1.rst`, line 5184); *backlink*
>
> Unknown interpreted text role "option".

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\(cpython-main)(Misc)(NEWS.d)3.8.0a1.rst`, line 5184); *backlink*
>
> Unknown interpreted text role "issue".

This fix also changed pydoc to add `os.getcwd()` to :data:`sys.path` when necessary, rather than adding `"."`.

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\(cpython-main)(Misc)(NEWS.d)3.8.0a1.rst`, line 5187); *backlink*
>
> Unknown interpreted text role "data".

Added support for the `SameSite` cookie flag to the `http.cookies` module.

Delete entries of `None` in :data:`sys.path_importer_cache` when :meth:`importlib.machinery.invalidate_caches` is called.

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\(cpython-main)(Misc)(NEWS.d)3.8.0a1.rst`, line 5207); *backlink*
>
> Unknown interpreted text role "data".

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\(cpython-main)(Misc)(NEWS.d)3.8.0a1.rst`, line 5207); *backlink*
>
> Unknown interpreted text role "meth".

`random.Random.choice()` now raises `IndexError` for empty sequences consistently even when called from subclasses without a `getrandbits()` implementation.

Update difflib.mdiff() for PEP 479. Convert an uncaught StopIteration in a generator into a return-statement.

End framing at the end of C implementation of :func:`pickle.Pickler.dump`.

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\(cpython-main)(Misc)(NEWS.d)3.8.0a1.rst`, line 5238); *backlink*
>
> Unknown interpreted text role "func".

The urllib.robotparser's `__str__` representation now includes wildcard entries and the "Crawl-delay" and "Request-rate" fields. Also removes extra newlines that were being appended to the end of the string. Patch by Michael Lazar.

`DEFAULT_PROTOCOL` in :mod:`pickle` was bumped to 4. Protocol 4 is described in PEP 3154 and available since Python 3.4. It offers better performance and smaller size compared to protocol 3 introduced in Python 3.0.

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\(cpython-main)(Misc)(NEWS.d)3.8.0a1.rst`, line 5259); *backlink*
>
> Unknown interpreted text role "mod".

Improved error handling and fixed a reference leak in :func:`os.posix_spawn()`.

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-

Deleting a key from a read-only dbm database raises module specific error instead of KeyError.

In dataclasses, Field.__set_name__ now looks up the __set_name__ special method on the class, not the instance, of the default value.

Create functools.singledispatchmethod to support generic single dispatch on descriptors and methods.

Have Field objects pass through __set_name__ to their default values, if they have their own __set_name__.

Allow ttk.Treeview.insert to insert iid that has a false boolean value. Note iid=0 and iid=False would be same. Patch by Garvit Khatri.

Treat type variables and special typing forms as immutable by copy and pickle. This fixes several minor issues and inconsistencies, and improves backwards compatibility with Python 3.6.

When computing dataclass's __hash__, use the lookup table to contain the function which returns the __hash__ value. This is an improvement over looking up a string, and then testing that string to see what to do.

The ssl module now compiles with LibreSSL 2.7.1.

Raise TypeError if a member variable of a dataclass is of type Field, but doesn't have a type annotation.

Fix the failure on OSX caused by the tests relying on sem_getvalue

Add 'Field' to dataclasses.__all__.

Fix an error where subclassing a dataclass with a field that uses a default_factory would generate an incorrect class.

Dataclasses: If a field has a default value that's a MemberDescriptorType, then it's from that field being in __slots__, not an actual default value.

If a non-dataclass inherits from a frozen dataclass, allow attributes to be added to the derived class. Only attributes from the frozen dataclass cannot be assigned to. Require all dataclasses in a hierarchy to be either all frozen or all non-frozen.

Raise RuntimeError when executor.submit is called during interpreter shutdown.

Modulo and floor division involving Fraction and float should return float.

Add missing NoReturn to __all__ in typing.py

Fix the size handling in multiprocessing.Queue when a pickling error occurs.

lib2to3 now properly supports trailing commas after *args and **kwargs in function signatures.

FIX properly close leaking fds in concurrent.futures.ProcessPoolExecutor.

Release the GIL during fstat() calls, avoiding hang of all threads when calling mmap.mmap(), os.urandom(), and random.seed(). Patch by Nir Soffer.

Avoid failing in multiprocessing.Process if the standard streams are closed or None at exit.

Providing an explicit error message when casting the port property to anything that is not an integer value using urlparse() and urlsplit(). Patch by Matt Eaton.

Improve struct.unpack_from() exception messages for problems with the buffer size and offset.

Skip sending/receiving data after SSL transport closing.

Fix a regression in :mod:`ipaddress` that result of :meth:`hosts` is empty when the network is constructed by a tuple containing an integer mask and only 1 bit left for addresses.

Add the strsignal() function in the signal module that returns the system description of the given signal, as returned by strsignal(3).

Fix C implementation of ABC.__subclasscheck__(cls, subclass) crashed when subclass is not a type object.

Fix inspect.signature() for single-parameter partialmethods.

Expose several missing constants in zlib and fix corresponding documentation.

Improved exceptions raised for invalid number of channels and sample width when read an audio file in modules :mod:`aifc`, :mod:`wave` and :mod:`sunau`.

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\(cpython-main) (Misc) (NEWS.d)3.8.0a1.rst`, line 5578); *backlink*
>
> Unknown interpreted text role "mod".

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\(cpython-main) (Misc) (NEWS.d)3.8.0a1.rst`, line 5578); *backlink*
>
> Unknown interpreted text role "mod".

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\(cpython-main) (Misc) (NEWS.d)3.8.0a1.rst`, line 5578); *backlink*
>
> Unknown interpreted text role "mod".

Improved disassembly of the MAKE_FUNCTION instruction.

Fix wrong redirection of a low descriptor (0 or 1) to stderr in subprocess if another low descriptor is closed.

For dataclasses, disallow inheriting frozen from non-frozen classes, and also disallow inheriting non-frozen from frozen classes. This restriction will be relaxed at a future date.

Fixed tarfile.itn handling of out-of-bounds float values. Patch by Joffrey Fuhrer.

The ssl module now contains OP_NO_RENEGOTIATION constant, available with OpenSSL 1.1.0h or 1.1.1.

Direct instantiation of SSLSocket and SSLObject objects is now prohibited. The constructors were never documented, tested, or designed as public constructors. Users were suppose to use ssl.wrap_socket() or SSLContext.

Remove the tri-state parameter "hash", and add the boolean "unsafe_hash". If unsafe_hash is True, add a __hash__ function, but if a __hash__ exists, raise TypeError. If unsafe_hash is False, add a __hash__ based on the values of eq= and frozen=. The unsafe_hash=False behavior is the same as the old hash=None behavior. unsafe_hash=False is the default, just as hash=None used to be.

Add OP_ENABLE_MIDDLEBOX_COMPAT and test workaround for TLSv1.3 for future compatibility with OpenSSL 1.1.1.

Document the interaction between frozen executables and the spawn and forkserver start methods in multiprocessing.

The ssl module now detects missing NPN support in LibreSSL.

dbm.open() now encodes filename with the filesystem encoding rather than default encoding.

Free unused arenas in multiprocessing.heap.

In `os.dup2`, don't check every call whether the `dup3` syscall exists or not.

nt._getfinalpathname, nt._getvolumepathname and nt._getdiskusage now correctly convert from bytes.

Rewrite confusing message from setup.py upload from "No dist file created in earlier command" to the more helpful "Must create and upload files in one command".

In :mod:`tkinter`, `after_cancel(None)` now raises a :exc:`ValueError` instead of canceling the first scheduled function. Patch by Cheryl Sabella.

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\(cpython-main) (Misc) (NEWS.d)3.8.0a1.rst`, line 5743); *backlink*
>
> Unknown interpreted text role "mod".

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\(cpython-main) (Misc) (NEWS.d)3.8.0a1.rst`, line 5743); *backlink*
>
> Unknown interpreted text role "exc".

Make sure sys.argv remains as a list when running trace.

_abc module is added. It is a speedup module with C implementations for various functions and methods in `abc`. Creating an ABC subclass and calling `isinstance` or `issubclass` with an ABC subclass are up to 1.5x faster. In addition, this makes Python start-up up to 10% faster.

Note that the new implementation hides internal registry and caches, previously accessible via private attributes _abc_registry, _abc_cache, and _abc_negative_cache. There are three debugging helper methods that can be used instead _dump_registry,

`_abc_registry_clear`, and `_abc_caches_clear`.

Fixed *asyncio.Condition* issue which silently ignored cancellation after notifying and cancelling a conditional lock. Patch by Bar Harel.

ssl.match_hostname() has been simplified and no longer depends on re and ipaddress module for wildcard and IP addresses. Error reporting for invalid wildcards has been improved.

`multiprocessing.`Pool no longer leaks processes if its initialization fails.

socket: Remove TCP_FASTOPEN,TCP_KEEPCNT,TCP_KEEPIDLE,TCP_KEEPINTVL flags on older version Windows during run-time.

Fixed refleaks of `__init__()` methods in various modules. (Contributed by Oren Milman)

Fixed guessing quote and delimiter in csv.Sniffer.sniff() when only the last field is quoted. Patch by Jake Davis.

Added support of `\N{name}` escapes in regular expressions. Based on patch by Jonathan Eunice.

collections.ChainMap() preserves the order of the underlying mappings.

:func:`fnmatch.translate()` no longer produces patterns which contain set operations. Sets starting with '[' or containing '--', '&&', '~~' or '||' will be interpreted differently in regular expressions in future versions. Currently they emit warnings. fnmatch.translate() now avoids producing patterns containing such sets by accident.

> **System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\(cpython-main)(Misc)(NEWS.d)3.8.0a1.rst`, line 5860);** *backlink*
>
> Unknown interpreted text role "func".

Implement native fast sendfile for Windows proactor event loop.

Fix a rare but potential pre-exec child process deadlock in subprocess on POSIX systems when marking file descriptors inheritable on exec in the child process. This bug appears to have been introduced in 3.4.

The ctypes module used to depend on indirect linking for dlopen. The shared extension is now explicitly linked against libdl on platforms with dl.

A :mod:`dbm.dumb` database opened with flags 'r' is now read-only. :func:`dbm.dumb.open` with flags 'r' and 'w' no longer creates a database if it does not exist.

> **System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\(cpython-main)(Misc)(NEWS.d)3.8.0a1.rst`, line 5903);** *backlink*
>
> Unknown interpreted text role "mod".

> **System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\(cpython-main)(Misc)(NEWS.d)3.8.0a1.rst`, line 5903);** *backlink*
>
> Unknown interpreted text role "func".

Implement `asyncio.TimerHandle.when()` method.

Use mod_spec.parent when running modules with pdb

Fixed `asyncio.Lock()` safety issue which allowed acquiring and locking the same lock multiple times, without it being free. Patch by Bar Harel.

Do not include name field in SMTP envelope from address. Patch by Stéphane Wirtel

Add TLSVersion constants and SSLContext.maximum_version / minimum_version attributes. The new API wraps OpenSSL 1.1 https://www.openssl.org/docs/man1.1.0/ssl/SSL_CTX_set_min_proto_version.html feature.

Internal implementation details of ssl module were cleaned up. The SSLSocket has one less layer of indirection. Owner and session information are now handled by the SSLSocket and SSLObject constructor. Channel binding implementation has been simplified.

Fix the error handling in Aifc_read.initfp() when the SSND chunk is not found. Patch by Zackery Spytz.

Add Ttk spinbox widget to :mod:`tkinter.ttk`. Patch by Alan D Moore.

> **System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\(cpython-main)(Misc)(NEWS.d)3.8.0a1.rst`, line 5986);** *backlink*
>
> Unknown interpreted text role "mod".

:mod:`profile` CLI accepts *-m module_name* as an alternative to script path.

help() on a type now displays builtin subclasses. This is intended primarily to help with notification of more specific exception subclasses.

Patch by Sanyam Khurana.

http.server now exposes a ThreadingHTTPServer class and uses it when the module is run with `-m` to cope with web browsers pre-opening sockets.

compileall: import ProcessPoolExecutor only when needed, preventing hangs on low resource platforms

Various functions returning tuple containing IPv6 addresses now omit `%scope` part since the same information is already encoded in *scopeid* tuple item. Especially this speeds up :func:`socket.recvfrom` when it receives multicast packet since useless resolving of network interface name is omitted.

:func:`binascii.unhexlify` is now up to 2 times faster. Patch by Sergey Fedoseev.

The TarFile class now recurses directories in a reproducible way.

The ZipFile class now recurses directories in a reproducible way.

Added :data:`curses.ncurses_version`.

Fix output of cover files for `trace` module command-line tool. Previously emitted cover files only when `--missing` option was used. Patch by Michael Selik.

Raise a `TypeError` instead of crashing if a `collections.deque` subclass returns a non-deque from `__new__`. Patch by Oren Milman.

Add support for sockets of the AF_QIPCRTR address family, supported by the Linux kernel. This is used to communicate with services, such as GPS or radio, running on Qualcomm devices. Patch by Bjorn Andersson.

Implemented unpickling instances of :class:`~datetime.datetime`, :class:`~datetime.date` and :class:`~datetime.time` pickled by Python 2. `encoding='latin1'` should be used for successful decoding.

:class:`sqlite3.Connection` now exposes a :class:`~sqlite3.Connection.backup` method, if the underlying SQLite library is at version 3.6.11 or higher. Patch by Lele Gaifax.

Support arrays >=2GiB in :mod:`ctypes`. Patch by Segev Finer.

Removed support of arguments in *tkinter.ttk.Treeview.selection*. It was deprecated in 3.6. Use specialized methods like *selection_set* for changing the selection.

Fix bugs in hangul normalization: u1176, u11a7 and u11c3

Document :func:`http.client.parse_headers`.

Improve example of iter() with 2nd sentinel argument.

Explicitly set master_doc variable in conf.py for compliance with Sphinx 2.0

Specified that profile.Profile class doesn't not support enable or disable methods. Also, elaborated that Profile object as a context manager is only supported in cProfile module.

Enhance the gettext docs. Patch by Ã‰ric Araujo

Remove mention of `typing.io` and `typing.re`. Their types should be imported from `typing` directly.

Fix the documentation about an unexisting *f_restricted* attribute in the frame object. Patch by StÃ©phane Wirtel

Replace PEP XYZ by the pep role and allow to use the direct links to the PEPs.

Fix the documentation with the role `exc` for the appropriated exception. Patch by StÃ©phane Wirtel

Rename documentation for :mod:`email.utils` to `email.utils.rst`.

Use app.add_object_type() instead of the deprecated Sphinx function app.description_unit()

Add documentation about the new command line interface of the gzip module.

chm document displays non-ASCII charaters properly on some MBCS Windows systems.

Create availability directive for documentation. Original patch by Georg Brandl.

Document how passing coroutines to asyncio.wait() can be confusing.

Make clear that == operator sometimes is equivalent to *is*. The <, <=, > and >= operators are only defined where they make sense.

Fixed info in the stdtypes docs concerning the types that support membership tests.

Migrate datetime.date.fromtimestamp to Argument Clinic. Patch by Tim Hoffmann.

Fix wrongly written basicConfig documentation markup syntax

replaced ellipsis with correct error codes in tutorial chapter 3.

Add '@' operator entry to index.

Clarified the relationship between PEP 538's PYTHONCOERCECLOCALE and PEP 540's PYTHONUTF8 mode.

Add versionadded tag to the documentation of ParameterKind.description

Improve the C-API doc for PyTypeObject. This includes adding several quick-reference tables and a lot of missing slot/typedef entries. The existing entries were also cleaned up with a slightly more consistent format.

Improve the documentation of :func:`asyncio.open_connection`, :func:`asyncio.start_server` and their UNIX socket counterparts.

Document *asyncio.wait()* does not cancel its futures on timeout.

Document PEP 567 changes to asyncio.

Update HMAC md5 default to a DeprecationWarning, bump removal to 3.8.

Document `getargspec`, `from_function` and `from_builtin` as deprecated in their respective docstring, and include version since deprecation in DeprecationWarning message.

Fix broken pypi link

Add missing documentation for `typing.AsyncContextManager`.

BZ2file now emit a DeprecationWarning when buffering=None is passed, the deprecation message and documentation also now explicitly state it is deprecated since 3.0.

Add Korean language switcher for https://docs.python.org/3/

Clarify that the `__path__` attribute on modules cannot be just any value.

Modernize documentation for writing C extension types.

Deprecate `Py_UNICODE` usage in `c-api/arg` document. `Py_UNICODE` related APIs are deprecated since Python 3.3, but it is missed in the document.

Document PyBuffer_ToContiguous().

Modify documentation for the :func:`islice` recipe to consume initial values up to the start index.

Update :mod:`zipapp` documentation to describe how to make standalone applications.

Documentation changes for ipaddress. Patch by Jon Foster and Berker Peksag.

Update documentation to clarify that `WindowsRegistryFinder` implements `MetaPathFinder`. (Patch by Himanshu Lakhara)

The ssl module function ssl.wrap_socket() has been de-emphasized and deprecated in favor of the more secure and efficient SSLContext.wrap_socket() method.

Clarify docs for -O and -OO. Patch by Terry Reedy.

Add documentation for the contextvars module (PEP 567).

Update link to w3c doc for xml default namespaces.

Update :mod:`test.support` documentation.

Update the faq/windows.html to use the py command from PEP 397 instead of python.

Document :meth:`__getattr__` behavior when property :meth:`get` method raises :exc:`AttributeError`.

Modify RE examples in documentation to use raw strings to prevent :exc:`DeprecationWarning` and add text to REGEX HOWTO to highlight the deprecation.

Remove the paragraph where we explain that os.utime() does not support a directory as path under Windows. Patch by Jan-Philip Gehrcke

Remove the bad example in the tutorial of the Generator Expression. Patch by StÃ©phane Wirtel

Improve docstrings for *pathlib.PurePath* subclasses.

Use the externalized `python-docs-theme` package when building the documentation.

Add a note about curses.addch and curses.addstr exception behavior when writing outside a window, or pad.

Update documentation related with `dict` order.

Document `AF_PACKET` in the :mod:`socket` module.

Clarify meaning of CERT_NONE, CERT_OPTIONAL, and CERT_REQUIRED flags for ssl.SSLContext.verify_mode.

Fix sparse file tests of test_tarfile on ppc64 with the tmpfs filesystem. Fix the function testing if the filesystem supports sparse files: create a file which contains data and "holes", instead of creating a file which contains no data. tmpfs effective block size is a page size (tmpfs lives in the page cache). RHEL uses 64 KiB pages on aarch64, ppc64, ppc64le, only s390x and x86_64 use 4 KiB pages, whereas the test punch holes of 4 KiB.

Make ssl tests less strict and also accept TLSv1 as system default. The changes unbreaks test_min_max_version on Fedora 29.

`test_asyncio/test_sendfile.py` now resets the event loop policy using :func:`tearDownModule` as done in other tests, to prevent a warning when running tests on Windows.

test.pythoninfo now logs information of all clocks, not only time.time() and time.perf_counter().

Add a test to pathlib's Path.match() to verify it does not support glob-style ** recursive pattern matching.

Fix a race condition in `check_interrupted_write()` of test_io: create directly the thread with SIGALRM signal blocked, rather than blocking the signal later from the thread. Previously, it was possible that the thread gets the signal before the signal is blocked.

Fix test_multiprocessing_main_handling: use :class:`multiprocessing.Pool` with a context manager and then explicitly join the pool.

Unknown interpreted text role "class".

Rename :mod:`test.bisect` module to :mod:`test.bisect_cmd` to avoid conflict with :mod:`bisect` module when running directly a test like ./python Lib/test/test_xmlrpc.py.

Replace :func:`time.time` with :func:`time.monotonic` in tests to measure time delta.

:func:`test.support.run_unittest` no longer raise :exc:`TestDidNotRun` if the test result contains skipped tests. The exception is now only raised if no test have been run and no test have been skipped.

Add testcase to test_future4: check unicode literal.

Added test demonstrating double-patching of an instance method. Patch by Anthony Sottile.

test_multiprocessing_fork may crash on recent versions of macOS. Until the issue is resolved, skip the test on macOS.

Modify test_asyncio to use the certificate set from the test directory.

Fix mktime() overflow error in test_email: run test_localtime_daylight_true_dst_true() and test_localtime_daylight_false_dst_true() with a specific timezone.

After several reports that test_gdb does not work properly on macOS and since gdb is not shipped by default anymore, test_gdb is now skipped on macOS when LLVM Clang has been used to compile Python. Patch by Lysandros Nikolaou

regrtest issue a warning when no tests have been executed in a particular test file. Also, a new final result state is issued if no test have been executed across all test files. Patch by Pablo Galindo.

make docstest in Doc now passes., and is enforced in CI

Use argparse for the command line of the gzip module. Patch by Antony Lee

Fix test_gdb.test_strings() when LC_ALL=C and GDB was compiled with Python 3.6 or earlier.

test_socket: Remove RDSTest.testCongestion(). The test tries to fill the receiver's socket buffer and expects an error. But the RDS

protocol doesn't require that. Moreover, the Linux implementation of RDS expects that the producer of the messages reduces its rate, it's not the role of the receiver to trigger an error. The test fails on Fedora 28 by design, so just remove it.

Fix test_shutil if unzip doesn't support -t.

Fixed non-deterministic flakiness of test_pkg by not using the scary test.support.module_cleanup() logic to save and restore sys.modules contents between test cases.

The experimental PEP 554 data channels now correctly pass negative PyLong objects between subinterpreters on 32-bit systems. Patch by Michael Felt.

Fix usage of hardcoded `errno` values in the tests.

Fix test_embed for AIX Patch by Michael Felt

Use 3072 RSA keys and SHA-256 signature for test certs and keys.

Remove special condition for AIX in *test_subprocess.test_undecodable_env*

Fix *test_utf8_mode.test_cmd_line* for AIX

On AIX with AF_UNIX family sockets getsockname() does not provide 'sockname', so skip calls to transport.get_extra_info('sockname')

Fix ftplib test for TLS 1.3 by reading from data socket.

Fix *test_socket* on AIX 6.1 and later IPv6 zone id supports only supported by inet_pton6_zone() Switch to runtime-based platform.system() to establish current platform rather than build-time based sys.platform()

Update all RSA keys and DH params to use at least 2048 bits.

Fix `test_mktime` and `test_pthread_getcpuclickid` tests for AIX Add range checking for `_PyTime_localtime` for AIX Patch by Michael Felt

Skip the distutils test 'test_search_cpp' when using XLC as compiler patch by aixtools (Michael Felt)

Improved an error message when mock assert_has_calls fails.

Fix test_unittest when run in verbose mode.

Fix test_dbm_gnu on macOS with gdbm 1.15: add a larger value to make sure that the file size changes.

Fix a bug in `regrtest` that caused an extra test to run if --huntrleaks/-R was used. Exit with error in case that invalid parameters are specified to --huntrleaks/-R (at least one warmup run and one repetition must be used).

Check that a global asyncio event loop policy is not left behind by any tests.

Ignore test_posix_fallocate failures on BSD platforms that might be due to running on ZFS.

Fixed test_gdb when Python is compiled with flags -mcet -fcf-protection -O0.

Fix `test_embed.test_pre_initialization_sys_options()` when the interpreter is built with `--enable-shared`.

Avoid regrtest compatibility issue with namespace packages.

Fix failing `test_asyncio` on macOS 10.12.2+ due to transport of `KqueueSelector` loop was not being closed.

Making sure the *SMTPUTF8SimTests* class of tests gets run in test_smtplib.py.

Test_C test case needs "signed short" bitfields, but the IBM XLC compiler (on AIX) does not support this Skip the code and test when AIX and XLC are used

Applicable to Python2-2.7 and later

Add test_bdb.py.

Add tests to verify connection with secp ECDH curves.

The _contextvars module is now built into the core Python library on Windows.

Improved Azure Pipelines build steps and now verifying layouts correctly

Remove asynciomodule.c from pythoncore.vcxproj

Fix incorrect Solaris #ifdef checks to look for __sun && __SVR4 instead of sun when compiling.

`make profile-opt` no longer replaces `CFLAGS_NODIST` with `CFLAGS`. It now adds profile-guided optimization (PGO) flags to `CFLAGS_NODIST`: existing `CFLAGS_NODIST` flags are kept.

Avoid leaking the linker flags from Link Time Optimizations (LTO) into distutils when compiling C extensions.

When building Python with clang and LTO, LTO flags are no longer passed into CFLAGS to build third-party C extensions through distutils.

Fix a compiler error when statically linking *pyexpat* in *Modules/Setup*.

PCbuild: Set InlineFunctionExpansion to OnlyExplicitInline ("/Ob1" option) in pyproject.props in Debug mode to expand functions marked as inline. This change should make Python compiled in Debug mode a little bit faster on Windows.

Restores the use of pyexpatns.h to isolate our embedded copy of the expat C library so that its symbols do not conflict at link or dynamic loading time with an embedding application or other extension modules with their own version of libexpat.

Have --with-lto works correctly with clang.

Update the outdated install-sh file to the latest revision from automake v1.16.1

Check for floating-point byte order in configure.ac using compilation tests instead of executing code, so that these checks work in cross-compiled builds.

Fixed SSL module build with OpenSSL & pedantic CFLAGS.

Add JUnit XML output for regression tests and update Azure DevOps builds.

Make Sphinx warnings as errors in the Docs Makefile.

Fix for case where it was not possible to have both `HAVE_LINUX_VM_SOCKETS_H` and `HAVE_SOCKADDR_ALG` be undefined.

Fix an undefined behaviour in the pthread implementation of :c:func:`PyThread_start_new_thread`: add a function wrapper to always return `NULL`.

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\(cpython-main) (Misc) (NEWS.d) 3.8.0a1.rst`, line 7385); *backlink*
>
> Unknown interpreted text role "c:func".

The Python shared library is now installed with write permission (mode 0755), which is the standard way of installing such libraries.

Fix detection of C11 atomic support on clang.

Rename Modules/Setup.dist to Modules/Setup, and remove the necessity to copy the former manually to the latter when updating the local source tree.

Add -g to LDFLAGS when compiling with LTO to get debug symbols.

Move `-Wstrict-prototypes` option to `CFLAGS_NODIST` from `OPT`. This option emitted annoying warnings when building extension modules written in C++.

Ensures module definition files for the stable ABI on Windows are correctly regenerated.

The --with-c-locale-warning configuration flag has been removed. It has had no effect for about a year.

Enable CI builds on Visual Studio Team Services at https://python.visualstudio.com/cpython

configure's check for "long double" has been simplified

C compiler is now correctly detected from the standard environment variables. --without-gcc and --with-icc options have been removed.

Enable the verbose build for extension modules, when GNU make is passed macros on the command line.

Update config.guess and config.sub files.

Add new triplets for mips r6 and riscv variants (used in extension suffixes).

By default, modules configured in *Modules/Setup* are no longer built with *-DPy_BUILD_CORE*. Instead, modules that specifically need that preprocessor definition include it in their individual entries.

The embedding tests can once again be built with clang 6.0

Upgrade pip to 9.0.3 and setuptools to v39.0.1.

gcc 8 has added a new warning heuristic to detect invalid function casts and a stock python build seems to hit that warning quite often. The most common is the cast of a METH_NOARGS function (that uses just one argument) to a PyCFunction. Fix this by adding a dummy argument to all functions that implement METH_NOARGS.

Fix the python debug build when using COUNT_ALLOCS.

Replace optparse with argparse in setup.py

Fix API calling consistency of GetVersionEx and wcstok.

The `py` launcher now forwards its `STARTUPINFO` structure to child processes.

Fix EnvBuilder and --symlinks in venv on Windows

Avoid propagating venv settings when launching via py.exe

Fix default executable used by the multiprocessing module

Allow building on ARM with MSVC.

Fix handle leaks in os.stat on Windows.

Use unchecked PYCs for the embeddable distro to avoid zipimport restrictions.

Fix vcruntime140.dll being added to embeddable distro multiple times.

Update Windows build to use Tcl and Tk 8.6.9

Updates Windows build to OpenSSL 1.1.0j

venv on Windows will now use a python.exe redirector rather than copying the actual binaries from the base environment.

Adds support for building a Windows App Store package

Remove _distutils_findvs module and use vswhere.exe instead.

Allow shutil.disk_usage to take a file path on Windows

Fix a possible null pointer dereference in pyshellext.cpp.

Fix returning structs from functions produced by MSVC

Guard MSVC-specific code in socketmodule.c with `#ifdef _MSC_VER`.

Fixes exit code of list version arguments for py.exe.

Fixed the '--list' and '--list-paths' arguments for the py.exe launcher

Ensure INCLUDE and LIB directories do not end with a backslash.

A suite of code has been changed which copied across DLLs and init.tcl from the running Python location into a venv being created. These copies are needed only when running from a Python source build, and the copying code is now only run when that is the case, rather than whenever a venv is created.

Revert line length limit for Windows help docs. The line-length limit is not needed because the pages appear in a separate app rather than on a browser tab. It can also interact badly with the DPI setting.

Restore running PyOS_InputHook while waiting for user input at the prompt. The restores integration of interactive GUI windows (such as Matplotlib figures) with the prompt on Windows.

Output error when ReadConsole is canceled by CancelSynchronousIo instead of crashing.

GIL is released while calling functions that acquire Windows loader lock.

Reduces maximum marshal recursion depth on release builds.

Fix bug where :meth:`datetime.fromtimestamp` erroneously throws an :exc:`OSError` on Windows for values between 0 and 86400. Patch by Ammar Askar.

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\(cpython-main)(Misc)(NEWS.d)3.8.0a1.rst`, line 7837); *backlink*
>
> Unknown interpreted text role "meth".

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\(cpython-main)(Misc)(NEWS.d)3.8.0a1.rst`, line 7837); *backlink*
>
> Unknown interpreted text role "exc".

PyThread_release_lock always fails

Update Windows installer to use OpenSSL 1.1.0h.

Fix usage of GetLastError() instead of errno in os.execve() and os.truncate().

Fix potential use of uninitialized memory in nt._getfinalpathname

Fix a memory leak in os.chdir() on Windows if the current directory is set to a UNC path.

Update Tcl and Tk versions to 8.6.8

Fixed WindowsConsoleIO.write() for writing empty data.

Ensures activate.bat can handle Unicode contents.

Improves handling of denormalized executable path when launching Python.

Use the correct encoding for ipconfig output in the uuid module. Patch by Segev Finer.

Fix :func:`os.readlink` on Windows, which was mistakenly treating the `PrintNameOffset` field of the reparse data buffer as a number of characters instead of bytes. Patch by Craig Holmquist and SSE4.

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\(cpython-main)(Misc)(NEWS.d)3.8.0a1.rst`, line 7941); *backlink*
>
> Unknown interpreted text role "func".

Correctly handle string length in `msilib.SummaryInfo.GetProperty()` to prevent it from truncating the last character.

Update macOS installer to use OpenSSL 1.1.0j.

Properly guard the use of the `CLOCK_GETTIME` et al. macros in `timemodule` on macOS.

On macOS, fix reading from and writing into a file with a size larger than 2 GiB.

Update to OpenSSL 1.1.0i for macOS installer builds.

In macOS stat on some file descriptors (/dev/fd/3 f.e) will result in bad file descriptor OSError. Guard against this exception was added in is_dir, is_file and similar methods. DirEntry.is_dir can also throw this exception so _RecursiveWildcardSelector._iterate_directories was also extended with the same error ignoring pattern.

The .editrc file in user's home directory is now processed correctly during the readline initialization through editline emulation on macOS.

Update macOS installer build to use OpenSSL 1.1.0h.

Build and link with private copy of Tcl/Tk 8.6 for the macOS 10.6+ installer. The 10.9+ installer variant already does this. This means that the Python 3.7 provided by the python.org macOS installers no longer need or use any external versions of Tcl/Tk, either system-provided or user-installed, such as ActiveTcl.

Update macOS 10.9+ installer to Tcl/Tk 8.6.8.

In :mod:`_scproxy`, drop the GIL when calling into `SystemConfiguration` to avoid deadlocks.

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\(cpython-main) (Misc) (NEWS.d)3.8.0a1.rst`, line 8054); *backlink*
>
> Unknown interpreted text role "mod".

IDLE macosx deletes Options => Configure IDLE. It previously deleted Window => Zoom Height by mistake. (Zoom Height is now on the Options menu). On Mac, the settings dialog is accessed via Preferences on the IDLE menu.

Change IDLE's new file name from 'Untitled' to 'untitled'

Fix imports in idlelib.window.

Proper format *calltip* when the function has no docstring.

Use ttk Frame for ttk widgets.

Fix erroneous 'smart' indents and newlines in IDLE Shell.

Find Selection now works when selection not found.

Speed up squeezer line counting.

Update config_key: use PEP 8 names and ttk widgets, make some objects global, and add tests.

Add Previous/Next History entries to Shell menu.

Squeezer now properly counts wrapped lines before newlines.

Gray out Code Context menu entry when it's not applicable.

Document the IDLE editor code context feature. Add some internal references within the IDLE doc.

The Code Context menu label now toggles between Show/Hide Code Context. The Zoom Height menu now toggles between Zoom/Restore Height. Zoom Height has moved from the Window menu to the Options menu.

Where appropriate, use 'macOS' in idlelib.

On macOS, warn if the system preference "Prefer tabs when opening documents" is set to "Always".

Document two IDLE on MacOS issues. The System Preferences Dock "prefer tabs always" setting disables some IDLE features. Menus are a bit different than as described for Windows and Linux.

Remove unused imports from lib/idlelib

Document that IDLE's shell has no line limit. A program that runs indefinitely can overfill memory.

Explain how IDLE's Shell displays output.

Improve the doc about IDLE running user code. The section is renamed from "IDLE -- console differences" is renamed "Running user code". It mostly covers the implications of using custom sys.stdxxx objects.

Add IDLE doc subsection explaining editor windows. Topics include opening, title and status bar, .py* extension, and running.

Document the IDLE document viewer in the IDLE doc. Add a paragraph in "Help and preferences", "Help sources" subsection.

Update idlelib.help.copy_string docstring. We now use git and backporting instead of hg and forward merging.

Update idlelib help files for the current doc build. The main change is the elimination of chapter-section numbers.

Use configured color theme for read-only text views.

Enable "squeezing" of long outputs in the shell, to avoid performance degradation and to clean up the history without losing it. Squeezed outputs may be copied, viewed in a separate window, and "unsqueezed".

Fixed mousewheel scrolling direction on macOS.

Make IDLE calltips always visible on Mac. Some MacOS-tk combinations need .update_idletasks(). Patch by Kevin Walzer.

Fix unresponsiveness after closing certain windows and dialogs.

Avoid small type when running htests. Since part of the purpose of human-viewed tests is to determine that widgets look right, it is important that they look the same for testing as when running IDLE.

Add test for idlelib.stackview.StackBrowser.

Change mainmenu.menudefs key 'windows' to 'window'. Every other menudef key is lowercase version of main menu entry.

Rename idlelib.windows as window Match Window on the main menu and remove last plural module name.

Fix and document idlelib/idle_test/template.py. The revised file compiles, runs, and tests OK. idle_test/README.txt explains how to use it to create new IDLE test files.

IDLE: In rstrip, rename class RstripExtension as Rstrip

For consistency and clarity, rename an IDLE module and classes. Module calltips and its class CallTips are now calltip and Calltip. In module calltip_w, class CallTip is now CalltipWindow.

Add "help" in the welcome message of IDLE

IDLE: refactor ToolTip and CallTip and add documentation and tests

Minimally test all IDLE modules. Add missing files, import module, instantiate classes, and check coverage. Check existing files.

On Windows, add API call saying that tk scales for DPI. On Windows 8.1+ or 10, with DPI compatibility properties of the Python binary unchanged, and a monitor resolution greater than 96 DPI, this should make text and lines sharper. It should otherwise have no effect.

Clicking on a context line moves that line to the top of the editor window.

IDLE: Use read-only text widget for code context instead of label widget.

Scroll IDLE editor text by lines. Previously, the mouse wheel and scrollbar slider moved text by a fixed number of pixels, resulting in partial lines at the top of the editor box. The change also applies to the shell and grep output windows, but not to read-only text views.

Enable theme-specific color configuration for Code Context. Use the Highlights tab to see the setting for built-in themes or add settings to custom themes.

Display up to maxlines non-blank lines for Code Context. If there is no current context, show a single blank line.

IDLE: Cleanup codecontext.py and its test.

IDLE's code context now recognizes async as a block opener.

Update word/identifier definition from ascii to unicode. In text and entry boxes, this affects selection by double-click, movement left/right by control-left/right, and deletion left/right by control-BACKSPACE/DEL.

IDLE: consistently color invalid string prefixes. A 'u' string prefix cannot be paired with either 'r' or 'f'. Consistently color as much of the prefix, starting at the right, as is valid. Revise and extend colorizer test.

Set `__file__` while running a startup file. Like Python, IDLE optionally runs one startup file in the Shell window before presenting the first interactive input prompt. For IDLE, -s runs a file named in environmental variable :envvar:`IDLESTARTUP` or :envvar:`PYTHONSTARTUP`; -r file runs file. Python sets `__file__` to the startup file name before running the file and unsets it before the first prompt. IDLE now does the same when run normally, without the -n option.

> **System Message: ERROR/3** `(D:\onboarding-resources\sample-onboarding-resources\cpython-`
> `main\Misc\NEWS.d\(cpython-main)(Misc)(NEWS.d)3.8.0a1.rst,` **line 8555);** *backlink*
>
> Unknown interpreted text role "envvar".

> **System Message: ERROR/3** `(D:\onboarding-resources\sample-onboarding-resources\cpython-`
> `main\Misc\NEWS.d\(cpython-main)(Misc)(NEWS.d)3.8.0a1.rst,` **line 8555);** *backlink*
>
> Unknown interpreted text role "envvar".

Simplify and rename StringTranslatePseudoMapping in pyparse.

Change `str` to `code` in pyparse.

Remove unused code in pyparse module.

Add tests for pyparse.

Using the system and place-dependent default encoding for open() is a bad idea for IDLE's system and location-independent files.

Add "encoding=utf-8" to open() in IDLE's test_help_about. GUI test test_file_buttons() only looks at initial ascii-only lines, but failed on systems where open() defaults to 'ascii' because readline() internally reads and decodes far enough ahead to encounter a non-ascii character in CREDITS.txt.

Add docstrings and tests for codecontext.

Update configdialog General tab docstring to add new widgets to the widget list.

Add a benchmark script for timing various ways to access variables: `Tools/scripts/var_access_benchmark.py`.

python-gdb.py now handles errors on computing the line number of a Python frame.

Argument Clinic now has non-bitwise unsigned int converters.

python-gdb now catches `UnicodeDecodeError` exceptions when calling `string()`.

python-gdb now catches ValueError on read_var(): when Python has no debug symbols for example.

:program:`pygettext.py` now recognizes only literal strings as docstrings and translatable strings, and rejects bytes literals and f-string expressions.

> **System Message: ERROR/3 (**`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\(cpython-main) (Misc) (NEWS.d)3.8.0a1.rst`**, line 8697);** *backlink*
>
> Unknown interpreted text role "program".

Fixed handling directories as arguments in the `pygettext` script. Based on patch by Oleg Krasnikov.

Fix pystackv and pystack gdbinit macros.

Remove the pyvenv script in favor of `python3 -m venv` in order to lower confusion as to what Python interpreter a virtual environment will be created for.

Add an `-n` flag for `Tools/scripts/pathfix.py` to disable automatic backup creation (files with ~ suffix).

Fix pygettext not extracting docstrings for functions with type annotated arguments. Patch by Toby Harradine.

Fix 2to3 for using with --add-suffix option but without --output-dir option for relative path to files in current directory.

The :c:func:`PyByteArray_Init` and :c:func:`PyByteArray_Fini` functions have been removed. They did nothing since Python 2.7.4 and Python 3.2.0, were excluded from the limited API (stable ABI), and were not documented.

> **System Message: ERROR/3 (**`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\(cpython-main) (Misc) (NEWS.d)3.8.0a1.rst`**, line 8768);** *backlink*
>
> Unknown interpreted text role "c:func".

> **System Message: ERROR/3 (**`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\(cpython-main) (Misc) (NEWS.d)3.8.0a1.rst`**, line 8768);** *backlink*
>
> Unknown interpreted text role "c:func".

Fixed :c:func:`_PyBytes_Resize` for empty bytes objects.

> **System Message: ERROR/3 (**`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\(cpython-main) (Misc) (NEWS.d)3.8.0a1.rst`**, line 8779);** *backlink*
>
> Unknown interpreted text role "c:func".

Fix memory leak in :c:func:`PyUnicode_EncodeLocale` and :c:func:`PyUnicode_EncodeFSDefault` on error handling.

> **System Message: ERROR/3 (**`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\(cpython-main) (Misc) (NEWS.d)3.8.0a1.rst`**, line 8788);** *backlink*
>
> Unknown interpreted text role "c:func".

> **System Message: ERROR/3 (**`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\(cpython-main) (Misc) (NEWS.d)3.8.0a1.rst`**, line 8788);** *backlink*
>
> Unknown interpreted text role "c:func".

The following C macros have been converted to static inline functions: :c:func:`Py_INCREF`, :c:func:`Py_DECREF`, :c:func:`Py_XINCREF`, :c:func:`Py_XDECREF`, :c:func:`PyObject_INIT`, :c:func:`PyObject_INIT_VAR`.

> **System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\(cpython-main) (Misc) (NEWS.d) 3.8.0a1.rst, line 8798);** *backlink*
>
> Unknown interpreted text role "c:func".

> **System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\(cpython-main) (Misc) (NEWS.d) 3.8.0a1.rst, line 8798);** *backlink*
>
> Unknown interpreted text role "c:func".

> **System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\(cpython-main) (Misc) (NEWS.d) 3.8.0a1.rst, line 8798);** *backlink*
>
> Unknown interpreted text role "c:func".

> **System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\(cpython-main) (Misc) (NEWS.d) 3.8.0a1.rst, line 8798);** *backlink*
>
> Unknown interpreted text role "c:func".

> **System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\(cpython-main) (Misc) (NEWS.d) 3.8.0a1.rst, line 8798);** *backlink*
>
> Unknown interpreted text role "c:func".

> **System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\(cpython-main) (Misc) (NEWS.d) 3.8.0a1.rst, line 8798);** *backlink*
>
> Unknown interpreted text role "c:func".

`make install` now also installs the internal API: `Include/internal/*.h` header files.

Internal APIs surrounded by `#ifdef Py_BUILD_CORE` have been moved from `Include/*.h` headers to new header files `Include/internal/pycore_*.h`.

Conditionally declare :c:func:`Py_FinalizeEx()` (new in 3.6) based on Py_LIMITED_API. Patch by Arthur Neufeld.

> **System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\(cpython-main) (Misc) (NEWS.d) 3.8.0a1.rst, line 8829);** *backlink*
>
> Unknown interpreted text role "c:func".

The :c:func:`_PyObject_GC_TRACK` and :c:func:`_PyObject_GC_UNTRACK` macros have been removed from the public C API.

> **System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\(cpython-main) (Misc) (NEWS.d) 3.8.0a1.rst, line 8839);** *backlink*
>
> Unknown interpreted text role "c:func".

> **System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\(cpython-main) (Misc) (NEWS.d) 3.8.0a1.rst, line 8839);** *backlink*
>
> Unknown interpreted text role "c:func".

Creation of a new `Include/cpython/` subdirectory.

Adds _Py_SetProgramFullPath so embedders may override sys.executable

Ensure that :c:func:`PyObject_Print` always returns `-1` on error. Patch by Zackery Spytz.

> **System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\(cpython-main) (Misc) (NEWS.d) 3.8.0a1.rst, line 8867);** *backlink*

Unknown interpreted text role "c:func".

Py_DecodeLocale() and Py_EncodeLocale() now use the UTF-8 encoding on Windows if Py_LegacyWindowsFSEncodingFlag is zero.

Fix pluralization in TypeError messages in getargs.c and typeobject.c: '1 argument' instead of '1 arguments' and '1 element' instead of '1 elements'.

Return grammatically correct error message based on argument count. Patch by Karthikeyan Singaravelan.

Fixed :exc:`SystemError` in :c:func:`PyArg_ParseTupleAndKeywords` when the `w*` format unit is used for optional parameter.

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\(cpython-main) (Misc) (NEWS.d) 3.8.0a1.rst`, **line 8907**); *backlink*
>
> Unknown interpreted text role "exc".

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\(cpython-main) (Misc) (NEWS.d) 3.8.0a1.rst`, **line 8907**); *backlink*
>
> Unknown interpreted text role "c:func".

Added :c:func:`PyCompile_OpcodeStackEffectWithJump`.

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\(cpython-main) (Misc) (NEWS.d) 3.8.0a1.rst`, **line 8917**); *backlink*
>
> Unknown interpreted text role "c:func".

Py_Main() can again be called after Py_Initialize(), as in Python 3.6.

Fixed error messages for :c:func:`PySequence_Size`, :c:func:`PySequence_GetItem`, :c:func:`PySequence_SetItem` and :c:func:`PySequence_DelItem` called with a mapping and :c:func:`PyMapping_Size` called with a sequence.

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\(cpython-main) (Misc) (NEWS.d) 3.8.0a1.rst`, **line 8935**); *backlink*
>
> Unknown interpreted text role "c:func".

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\(cpython-main) (Misc) (NEWS.d) 3.8.0a1.rst`, **line 8935**); *backlink*
>
> Unknown interpreted text role "c:func".

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\(cpython-main) (Misc) (NEWS.d) 3.8.0a1.rst`, **line 8935**); *backlink*
>
> Unknown interpreted text role "c:func".

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\(cpython-main) (Misc) (NEWS.d) 3.8.0a1.rst`, **line 8935**); *backlink*
>
> Unknown interpreted text role "c:func".

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\(cpython-main) (Misc) (NEWS.d) 3.8.0a1.rst`, **line 8935**); *backlink*
>
> Unknown interpreted text role "c:func".

:c:func:`PyExceptionClass_Name` will now return `const char *` instead of `char *`.

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\(cpython-main) (Misc) (NEWS.d) 3.8.0a1.rst`, **line 8947**); *backlink*
>
> Unknown interpreted text role "c:func".

Embedding applications may once again call PySys_ResetWarnOptions, PySys_AddWarnOption, and PySys_AddXOption prior to

calling Py_Initialize.

Document that m_traverse for multi-phase initialized modules can be called with m_state=NULL, and add a sanity check

:c:func:`PyUnicode_AsWideChar` and :c:func:`PyUnicode_AsWideCharString` no longer cache the `wchar_t*` representation of string objects.

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\(cpython-main) (Misc) (NEWS.d) 3.8.0a1.rst`, **line 8977**); *backlink*

Unknown interpreted text role "c:func".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\(cpython-main) (Misc) (NEWS.d) 3.8.0a1.rst`, **line 8977**); *backlink*

Unknown interpreted text role "c:func".