Newline characters have been escaped when performing uu encoding to prevent them from overflowing into to content section of the encoded file. This prevents malicious or accidental modification of data during the decoding process.

Due to significant security concerns, the *reuse_address* parameter of :meth:`asyncio.loop.create_datagram_endpoint` is no longer supported. This is because of the behavior of `SO_REUSEADDR` in UDP. For more details, see the documentation for `loop.create_datagram_endpoint()`. (Contributed by Kyle Stanley, Antoine Pitrou, and Yury Selivanov in :issue:`37228`.)

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\[cpython-main][Misc][NEWS.d]3.9.0a2.rst`, line 19); *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]3.9.0a2.rst`, line 19); *backlink*
>
> Unknown interpreted text role "issue".

Fixes a ReDoS vulnerability in :mod:`http.cookiejar`. Patch by Ben Caller.

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\[cpython-main][Misc][NEWS.d]3.9.0a2.rst`, line 32); *backlink*
>
> Unknown interpreted text role "mod".

Slightly improve the speed of keyword argument parsing with many kwargs by strengthening the assumption that kwargs are interned strings.

Fix the value of *end_col_offset* for Starred Expression AST nodes when they are among the elements in the *args* attribute of Call AST nodes.

When parsing an "elif" node, lineno and col_offset of the node now point to the "elif" keyword and not to its condition, making it consistent with the "if" node. Patch by Lysandros Nikolaou.

In Python 3.9.0a1, sys.argv[0] was made an absolute path if a filename was specified on the command line. Revert this change, since most users expect sys.argv to be unmodified.

:c:func:`PySys_Audit` now requires `Py_ssize_t` to be used for size arguments in the format string, regardless of whether `PY_SSIZE_T_CLEAN` was defined at include time.

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\[cpython-main][Misc][NEWS.d]3.9.0a2.rst`, line 83); *backlink*
>
> Unknown interpreted text role "c:func".

In REPL mode, don't switch to PS2 if the line starts with comment or whitespace. Based on work by Batuhan TaÅŸkaya.

Calling `replace` on a code object now raises the `code.__new__` audit event.

Add audit hooks for when :func:`sys.excepthook` and :func:`sys.unraisablehook` are invoked.

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\[cpython-main][Misc][NEWS.d]3.9.0a2.rst`, line 114); *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]3.9.0a2.rst`, line 114); *backlink*
>
> Unknown interpreted text role "func".

Improve documentation for audit events table and functions.

Set the thread stack size to 8 Mb for debug builds on android platforms.

Each Python subinterpreter now has its own "small integer singletons": numbers in [-5; 257] range. It is no longer possible to change the number of small integers at build time by overriding `NSMALLNEGINTS` and `NSMALLPOSINTS` macros: macros should now be modified manually in `pycore_pystate.h` header file.

The garbage collector state becomes per interpreter (`PyInterpreterState.gc`), rather than being global (`_PyRuntimeState.gc`).

The `PyFPE_START_PROTECT()` and `PyFPE_END_PROTECT()` macros are empty: they have been doing nothing for the last year, so

stop using them.

Sped up the creation time of constant :class:`list` and :class:`set` displays. Patch by Brandt Bucher.

`MainThread.native_id` is now correctly reset in child processes spawned using :class:`multiprocessing.Process`, instead of retaining the parent's value.

Added `__floor__` and `__ceil__` methods to float object. Patch by Batuhan TaÅŸkaya.

int + int and int - int operators can now return small integer singletons. Patch by hongweipeng.

Provide a platform tag for AIX that is sufficient for PEP425 binary distribution identification. Patch by Michael Felt.

Ignore GeneratorExit exceptions when throwing an exception into the aclose coroutine of an asynchronous generator.

Removed WITH_CLEANUP_START, WITH_CLEANUP_FINISH, BEGIN_FINALLY, END_FINALLY, CALL_FINALLY and POP_FINALLY bytecodes. Replaced with RERAISE and WITH_EXCEPT_START bytecodes. The compiler now generates different code for exceptional and non-exceptional branches for 'with' and 'try-except' statements. For 'try-finally' statements the 'finally' block is replicated for each exit from the 'try' body.

Fix :exc:`NameError` in :mod:`zipimport`. Patch by Karthikeyan Singaravelan.

Update importlib.metadata to include improvements from importlib_metadata 1.3 including better serialization of EntryPoints and improved documentation for custom finders.

Fix asyncio when the ssl module is missing: only check for ssl.SSLSocket instance if the ssl module is available.

Fix a potential IndexError in email parser when parsing an empty msg-id.

Add a new `InvalidMessageID` token to email parser to represent invalid Message-ID headers. Also, add defects when there is remaining value after parsing the header.

Implement `__class_getitem__` for `os.PathLike`, `pathlib.Path`.

Return class from `ContextVar.__class_getitem__` to simplify subclassing.

Implement `__class_getitem__` on asyncio objects (Future, Task, Queue). Patch by Batuhan Taskaya.

:class:`array.array`: Remove `tostring()` and `fromstring()` methods. They were aliases to `tobytes()` and `frombytes()`, deprecated since Python 3.2.

Make repr of C accelerated TaskWakeupMethWrapper the same as of pure Python version.

Fix asyncio `PidfdChildWatcher`: handle `waitpid()` error. If `waitpid()` is called elsewhere, `waitpid()` call fails with

:exc:`ChildProcessError`: use return code 255 in this case, and log a warning. It ensures that the pidfd file descriptor is closed if this error occurs.

Drop too noisy asyncio warning about deletion of a stream without explicit `.close()` call.

Added ability to pass through `ensure_ascii` options to json.dumps in the `json.tool` command-line interface.

The :mod:`readline` module now detects if Python is linked to libedit at runtime on all platforms. Previously, the check was only done on macOS.

Fix `json.tool` failed to read a JSON file with non-ASCII characters when locale encoding is not UTF-8.

Prevent UnboundLocalError to pop up in parse_message_id.

parse_message_id() was improperly using a token defined inside an exception handler, which was raising *UnboundLocalError* on parsing an invalid value. Patch by Claudiu Popa.

Use `python -m pip` instead of `pip` to upgrade dependencies in venv.

Fix `SpooledTemporaryFile.rollover()` might corrupt the file when it is in text mode. Patch by Serhiy Storchaka.

random.choices() now raises a ValueError when all the weights are zero.

Raise pickle.UnpicklingError when loading an item from memo for invalid input.

The previous code was raising a *KeyError* for both the Python and C implementation. This was caused by the specified index of an invalid input which did not exist in the memo structure, where the pickle stores what objects it has seen. The malformed input would have caused either a *BINGET* or *LONG_BINGET* load from the memo, leading to a *KeyError* as the determined index was bogus. Patch by Claudiu Popa

Calling func:*shutil.copytree* to copy a directory tree from one directory to another subdirectory resulted in an endless loop and a RecursionError. A fix was added to consume an iterator and create the list of the entries to be copied, avoiding the recursion for newly created directories. Patch by Bruno P. Kinoshita.

Improve :func:`is_cgi` function in :mod:`http.server`, which enables processing the case that cgi directory is a child of another directory other than root.

:meth:`typing.get_type_hints` properly handles functions decorated with :meth:`functools.wraps`.

Expose :func:`ast.unparse` as a function of the :mod:`ast` module that can be used to unparse an :class:`ast.AST` object and produce a string with code that would produce an equivalent :class:`ast.AST` object when parsed. Patch by Pablo Galindo and Batuhan Taskaya.

AsyncMock now returns StopAsyncIteration on the exhaustion of a side_effects iterable. Since PEP-479 its Impossible to raise a StopIteration exception from a coroutine.

AsyncMock fix for return values that are awaitable types. This also covers side_effect iterable values that happened to be awaitable, and wraps callables that return an awaitable type. Before these awaitables were being awaited instead of being returned as is.

:class:`typing.TypedDict` subclasses now track which keys are optional using the `__required_keys__` and `__optional_keys__` attributes, to enable runtime validation by downstream projects. Patch by Zac Hatfield-Dodds.

Fix unhandled exceptions in :mod:`argparse` when internationalizing error messages for arguments with `nargs` set to special (non-integer) values. Patch by Federico Bond.

Make Python compatible with OpenSSL 3.0.0. :func:`ssl.SSLSocket.getpeercert` no longer returns IPv6 addresses with a trailing new line.

Fix an unhandled exception in :mod:`pathlib` when :meth:`os.link` is missing. Patch by Toke HÃ¸iland-JÃ¸rgensen.

Added support for multiple `qop` values in :class:`urllib.request.AbstractDigestAuthHandler`.

Unknown interpreted text role "class".

Add the Linux-specific :func:`signal.pidfd_send_signal` function, which allows sending a signal to a process identified by a file descriptor rather than a pid.

Add `-i` and `--indent` (indentation level), and `--no-type-comments` (type comments) command line options to ast parsing tool.

Change :class:`zipfile.ZipExtFile` to raise `ValueError` when trying to access the underlying file object after it has been closed. This new behavior is consistent with how accessing closed files is handled in other parts of Python.

Improve the performance of :func:`enum_decompose` in :mod:`enum`. Patch by hongweipeng.

Break cycle generated when saving an exception in socket.py, codeop.py and dyld.py as they keep alive not only the exception but user objects through the `__traceback__` attribute. Patch by Mario Corchero.

Handle namespace packages in :mod:`doctest`. Patch by Karthikeyan Singaravelan.

Fix dataclasses to support forward references in type annotations

ElementTree supports recursive XInclude processing. Patch by Stefan Behnel.

Add whitespace options for formatting JSON with the `json.tool` CLI. The following mutually exclusive options are now supported: `--indent` for setting the indent level in spaces; `--tab` for indenting with tabs; `--no-indent` for suppressing newlines; and `--compact` for suppressing all whitespace. The default behavior remains the same as `--indent=4`.

Correct when venv's `upgrade_dependencies()` and `--upgrade-deps` are added.

Update documentation to state that to activate virtual environments under fish one should use *source*, not . as documented at https://fishshell.com/docs/current/commands.html#source.

Improves documentation of the values that :meth:`datetime.datetime.strptime` accepts for `%Z`. Patch by Karl Dubost.

Fix test_ressources_gced_in_workers() of test_concurrent_futures: explicitly stop the manager to prevent leaking a child process running in the background after the test completes.

Multiprocessing and concurrent.futures tests now stop the resource tracker process when tests complete.

Replace hardcoded timeout constants in tests with new :mod:`test.support` constants: :data:`~test.support.LOOPBACK_TIMEOUT`, :data:`~test.support.INTERNET_TIMEOUT`, :data:`~test.support.SHORT_TIMEOUT` and :data:`~test.support.LONG_TIMEOUT`. It becomes easier to adjust these four timeout constants for all tests at once, rather than having to adjust every single test file.

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\[cpython-main][Misc][NEWS.d]3.9.0a2.rst`, **line 729**); *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]3.9.0a2.rst`, **line 729**); *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]3.9.0a2.rst`, **line 729**); *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]3.9.0a2.rst`, **line 729**); *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]3.9.0a2.rst`, **line 729**); *backlink*
>
> Unknown interpreted text role "data".

Fix test_pty: if the process is the session leader, closing the master file descriptor raises a SIGHUP signal: simply ignore SIGHUP when running the tests.

Fix a test for :func:`math.fsum` that was failing due to constant folding.

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\[cpython-main][Misc][NEWS.d]3.9.0a2.rst`, **line 754**); *backlink*
>
> Unknown interpreted text role "func".

:mod:`test.support` :func:`~test.support.run_python_until_end`, :func:`~test.support.assert_python_ok` and :func:`~test.support.assert_python_failure` functions no longer strip whitespaces from stderr. Remove `test.support.strip_python_stderr()` function.

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\[cpython-main][Misc][NEWS.d]3.9.0a2.rst`, **line 763**); *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]3.9.0a2.rst`, **line 763**); *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]3.9.0a2.rst`, **line 763**); *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]3.9.0a2.rst`, **line 763**); *backlink*
>
> Unknown interpreted text role "func".

Fix test_faulthandler on GCC 10. Use the "volatile" keyword in `faulthandler._stack_overflow()` to prevent tail call optimization on any compiler, rather than relying on compiler specific pragma.

test_capi: trashcan tests now require the test "cpu" resource.

Skip asyncio test_create_datagram_endpoint_existing_sock_unix on platforms lacking a functional bind() for named unix domain

sockets.

Skip the test_posix.test_pidfd_open() test if `os.pidfd_open()` fails with a :exc:`PermissionError`. This situation can happen in a Linux sandbox using a syscall whitelist which doesn't allow the `pidfd_open()` syscall yet.

> **System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\[cpython-main][Misc][NEWS.d]3.9.0a2.rst`, line 806); *backlink***
>
> Unknown interpreted text role "exc".

Fix some unused functions in tests. Patch by Adam Johnson.

Raise :exc:`TypeError` when passing target as a string with :meth:`unittest.mock.patch.object`.

> **System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\[cpython-main][Misc][NEWS.d]3.9.0a2.rst`, line 826); *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]3.9.0a2.rst`, line 826); *backlink***
>
> Unknown interpreted text role "meth".

test.regrtest now can receive a list of test patterns to ignore (using the -i/--ignore argument) or a file with a list of patterns to ignore (using the --ignore-file argument). Patch by Pablo Galindo.

:mod:`asyncio` now raises :exc:`TyperError` when calling incompatible methods with an :class:`ssl.SSLSocket` socket. Patch by Ido Michael.

> **System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\[cpython-main][Misc][NEWS.d]3.9.0a2.rst`, line 847); *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]3.9.0a2.rst`, line 847); *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]3.9.0a2.rst`, line 847); *backlink***
>
> Unknown interpreted text role "class".

Added an optional "regen" project to the Visual Studio solution that will regenerate all grammar, tokens, and opcodes.

Add auditing events to functions in :mod:`winreg`.

> **System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\[cpython-main][Misc][NEWS.d]3.9.0a2.rst`, line 867); *backlink***
>
> Unknown interpreted text role "mod".

Add support for building and releasing Windows ARM64 packages.

Fixed a crash on OSX dynamic builds that occurred when re-initializing the posix module after a Py_Finalize if the environment had changed since the previous *import posix*. Patch by Benoît Hudson.

Escape key now closes IDLE completion windows. Patch by Johnny Najera.

Fix IDLE autocomplete windows not always appearing on some systems. Patch by Johnny Najera.

'Strip Trailing Whitespace' on the Format menu removes extra newlines at the end of non-shell files.

Fix IDLE Format menu tab toggle and file indent width. These functions (default shortcuts Alt-T and Alt-U) were mistakenly disabled in 3.7.5 and 3.8.0.

Remove `PyUnicode_ClearFreeList()` function: the Unicode free list has been removed in Python 3.3.

Remove `PyMethod_ClearFreeList()` and `PyCFunction_ClearFreeList()` functions: the free lists of bound method objects have been removed.

Exclude `PyFPE_START_PROTECT()` and `PyFPE_END_PROTECT()` macros of `pyfpe.h` from `Py_LIMITED_API` (stable API).