Improve the :exc:`SyntaxError` message when using `True`, `None` or `False` as keywords in a function call. Patch by Pablo Galindo.

:data:`sys.stdlib_module_names` now contains the macOS-specific module :mod:`_scproxy`.

Clarify :exc:`ImportError` message when we try to explicitly import a frozen module but frozen modules are disabled.

Specialize simple calls to Python functions (no starargs, keyowrd dict, or closure)

Cases of sorting using tuples as keys may now be significantly faster in some cases. Patch by Tim Peters.

The order of the result may differ from earlier releases if the tuple elements don't define a total ordering (see :ref:`expressions-value-comparisons` for information on total ordering). It's generally true that the result of sorting simply isn't well-defined in the absence of a total ordering on list elements.

In obmalloc, set ADDRESS_BITS to not ignore any bits (ignored 16 before). That is safer in the case that the kernel gives user-space virtual addresses that span a range greater than 48 bits.

Fixed a crash in `issubclass()` from infinite recursion when searching pathological `__bases__` tuples.

Fix a bug in the obmalloc radix tree code. On 64-bit machines, the bug causes the tree to hold 46-bits of virtual addresses, rather than the intended 48-bits.

Fix parser crash when reporting errors involving invalid continuation characters. Patch by Pablo Galindo.

Python now fails to initialize if it finds an invalid :option:`-X` option in the command line. Patch by Pablo Galindo.

Object attributes are held in an array instead of a dictionary. An object's dictionary are created lazily, only when needed. Reduces the memory consumption of a typical Python object by about 30%. Patch by Mark Shannon.

Fix a crash in the parser when reporting tokenizer errors that occur at the same time unclosed parentheses are detected. Patch by Pablo Galindo.

Add SipHash13 for string hash algorithm and use it by default.

Fix reference leak from descr_check. Patch by Dong-hee Na.

Specialized the `BINARY_MULTIPLY` opcode to `BINARY_MULTIPLY_INT` and `BINARY_MULTIPLY_FLOAT` using the PEP 659 machinery.

Frozen stdlib modules now have `__file__` to the .py file they would otherwise be loaded from, if possible. For packages, `__path__` now has the correct entry instead of being an empty list, which allows unfrozen submodules to be imported. These are set only if the stdlib directory is known when the runtime is initialized. Note that the file at `__file__` is not guaranteed to exist. None of

this affects non-stdlib frozen modules nor, for now, frozen modules imported using `PyImport_ImportFrozenModule()`. Also, at the moment `co_filename` is not updated for the module.

For frozen stdlib modules, record the original module name as `module.__spec__.loader_state.origname`. If the value is different than `module.__spec__.name` then the module was defined as an alias in Tools/scripts/freeze_modules.py. If it is `None` then the module comes from a source file outside the stdlib.

In FrozenImporter.find_spec(), we now preserve the information needed in exec_module() to load the module. This change mostly impacts internal details, rather than changing the importer's behavior.

Implement PEP 654. Add :class:`ExceptionGroup` and :class:`BaseExceptionGroup`. Update traceback display code.

> **System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\[cpython-main][Misc][NEWS.d]3.11.0a2.rst`, line 205);** *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]3.11.0a2.rst`, line 205);** *backlink*
>
> Unknown interpreted text role "class".

Change to the implementation of split dictionaries. Classes where the instances differ either in the exact set of attributes, or in the order in which those attributes are set, can still share keys. This should have no observable effect on users of Python or the C-API. Patch by Mark Shannon.

Extensions that indicate they use global state (by setting `m_size` to -1) can again be used in multiple interpreters. This reverts to behavior of Python 3.8.

Setup initial specialization infrastructure for the `CALL_FUNCTION` opcode. Implemented initial specializations for C function calls:

- `CALL_FUNCTION_BUILTIN_O` for `METH_O` flag.
- `CALL_FUNCTION_BUILTIN_FAST` for `METH_FASTCALL` flag without keywords.
- `CALL_FUNCTION_LEN` for `len(o)`.
- `CALL_FUNCTION_ISINSTANCE` for `isinstance(o, t)`.

Improve the generated bytecode for class and mapping patterns.

Speed up calls to `enumerate()` by using the PEP 590 `vectorcall` calling convention. Patch by Dong-hee Na.

Fix caching of multi-value :data:`typing.Literal`. `Literal[True, 2]` is no longer equal to `Literal[1, 2]`.

> **System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\[cpython-main][Misc][NEWS.d]3.11.0a2.rst`, line 275);** *backlink*
>
> Unknown interpreted text role "data".

Convert :mod:`sqlite3` to multi-phase initialisation (PEP 489). Patches by Erlend E. Aasland.

> **System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\[cpython-main][Misc][NEWS.d]3.11.0a2.rst`, line 285);** *backlink*
>
> Unknown interpreted text role "mod".

Fix typing.Signature string representation for generic builtin types.

:mod:`sqlite3` now sets :attr:`sqlite3.threadsafety` based on the default threading mode the underlying SQLite library has been compiled with. Patch by Erlend E. Aasland.

> **System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\[cpython-main][Misc][NEWS.d]3.11.0a2.rst`, line 304);** *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]3.11.0a2.rst`, line 304);** *backlink*
>
> Unknown interpreted text role "attr".

Fix warning about `print_escape` being unused.

:meth:`sqlite3.connect` now correctly raises :exc:`MemoryError` if the underlying SQLite API signals memory error. Patch by Erlend

E. Aasland.

pprint.pprint() now handles underscore_numbers correctly. Previously it was always setting it to False.

Add :func:`operator.call` to `operator.__all__`. Patch by Kreusada.

:meth:`shutil.get_terminal_size` now falls back to sane values if the column or line count are 0.

Improve the introspectability of the `__loader__` attribute for namespace packages. :class:`importlib.machinery.NamespaceLoader` is now public, and implements the :class:`importlib.abc.InspectLoader` interface. `_NamespaceLoader` is kept for backward compatibility.

Add references to :mod:`zoneinfo` in the :mod:`datetime` documentation, mostly replacing outdated references to `dateutil.tz`. Change by Paul Ganssle.

Reverted optimization of iterating :class:`gzip.GzipFile`, :class:`bz2.BZ2File`, and :class:`lzma.LZMAFile` (see bpo-43787) because it caused regression when user iterate them without having reference of them. Patch by Inada Naoki.

Unknown interpreted text role "class".

Update :class:`~typing.ForwardRef` to support | operator. Patch by Dong-hee Na.

Removed deprecated support for float arguments in *randrange()*.

Fix a regression in py_compile when reading filenames from standard input.

Fix incremental decoder and stream reader in the "raw-unicode-escape" codec. Previously they failed if the escape sequence was split.

Fix incremental decoder and stream reader in the "unicode-escape" codec. Previously they failed if the escape sequence was split.

Fixed :func:`email.utils.parsedate_tz` crashing with :exc:`UnboundLocalError` on certain invalid input instead of returning `None`. Patch by Ben Hoyt.

Fix quadratic behaviour in the enum module: Creation of enum classes with a lot of entries was quadratic.

Fix the behaviour of :func:`traceback.print_exc` when displaying the caret when the `end_offset` in the exception is set to 0. Patch by Pablo Galindo

Fix use of :class:`asyncio.Condition` with explicit :class:`asyncio.Lock` objects, which was a regression due to removal of explicit loop arguments. Patch by Joongi Kim.

Empty escapechar/quotechar is not allowed when initializing :class:`csv.Dialect`. Patch by Vajrasky Kok and Dong-hee Na.

Fix bug in the :mod:`doctest` module that caused it to fail if a docstring included an example with a `classmethod property`. Patch by Alex Waygood.

Make :func:`inspect.getmodule` catch `FileNotFoundError` raised by :'func:*inspect.getabsfile*, and return `None` to indicate that the module could not be determined.

Add extensions for files containing subtitles - .srt & .vtt - to the mimetypes.py module.

Migrated pydoc to HTML5 (without changing the look of it). Side effect is to update xmlrpc's `ServerHTMLDoc` which now uses the CSS too. cgitb now relies less on pydoc (as it can't use the CSS file).

Add support of null characters in :mod:`csv`.

Prevent use-after-free in asyncio. Make sure the cached running loop holder gets cleared on dealloc to prevent use-after-free in get_running_loop

Make :mod:`xmlrpc.client` more robust to C runtimes where the underlying C `strftime` function results in a `ValueError` when testing for year formatting options.

Improve error message of :class:`csv.Dialect` when initializing. Patch by Vajrasky Kok and Dong-hee Na.

Update bundled pip to 21.2.4 and setuptools to 58.1.0

Fixed :class:`http.client.HTTPConnection` to work properly in OSs that don't support the `TCP_NODELAY` socket option.

Add :meth:`~sqlite3.Connection.setlimit` and :meth:`~sqlite3.Connection.getlimit` to :class:`sqlite3.Connection` for setting and getting SQLite limits by connection basis. Patch by Erlend E. Aasland.

Unknown interpreted text role "class".

Removed from the :mod:`inspect` module:

- the `getargspec` function, deprecated since Python 3.0;

  use :func:`inspect.signature` or :func:`inspect.getfullargspec` instead.

- the `formatargspec` function, deprecated since Python 3.5; use the :func:`inspect.signature` function and :class:`Signature` object directly.

- the undocumented `Signature.from_callable` and `Signature.from_function` functions, deprecated since Python 3.5; use the :meth:`Signature.from_callable() <inspect.Signature.from_callable>` method instead.

Patch by Hugo van Kemenade.

Fix the `tempfile._infer_return_type` function so that the `dir` argument of the :mod:`tempfile` functions accepts an object implementing the `os.PathLike` protocol.

Patch by Kyungmin Lee.

When tracing a tkinter variable used by a ttk OptionMenu, callbacks are no longer made twice.

Added non parallel-safe :func:`~contextlib.chdir` context manager to change the current working directory and then restore it on exit. Simple wrapper around :func:`~os.chdir`.

main\Misc\NEWS.d\[cpython-main][Misc][NEWS.d]3.11.0a2.rst, line 658); *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]3.11.0a2.rst`, line 658); *backlink*

Unknown interpreted text role "func".

Add support for SQLite extended result codes in :exc:`sqlite3.Error`. Patch by Erlend E. Aasland.

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\[cpython-main][Misc][NEWS.d]3.11.0a2.rst`, line 669); *backlink*

Unknown interpreted text role "exc".

Fixed an error raised in :mod:`argparse` help display when help for an option is set to 1+ blank spaces or when *choices* arg is an empty container.

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\[cpython-main][Misc][NEWS.d]3.11.0a2.rst`, line 679); *backlink*

Unknown interpreted text role "mod".

Implement `Fraction.__int__`, so that a :class:`fractions.Fraction` instance `f` passes an `isinstance(f, typing.SupportsInt)` check.

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\[cpython-main][Misc][NEWS.d]3.11.0a2.rst`, line 690); *backlink*

Unknown interpreted text role "class".

Adds support for HTTP 308 redirects to :mod:`urllib`. See RFC 7538 for details. Patch by Jochem Schulenklopper.

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\[cpython-main][Misc][NEWS.d]3.11.0a2.rst`, line 700); *backlink*

Unknown interpreted text role "mod".

Move :mod:`sqlite3` tests to `/Lib/test/test_sqlite3`. Patch by Erlend E. Aasland.

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\[cpython-main][Misc][NEWS.d]3.11.0a2.rst`, line 710); *backlink*

Unknown interpreted text role "mod".

Ensure that `socket.TCP_*` constants are exposed on Cygwin 3.1.6 and greater.

Add help flag to the base64 module's command line interface. Patch contributed by Robert Kuska.

Improve documentation for :func:`functools.singledispatch` and :class:`functools.singledispatchmethod`.

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\[cpython-main][Misc][NEWS.d]3.11.0a2.rst`, line 740); *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]3.11.0a2.rst`, line 740); *backlink*

Unknown interpreted text role "class".

Amend the docs on `GenericAlias` objects to clarify that non-container classes can also implement `__class_getitem__`. Patch contributed by Alex Waygood.

Update Sphinx version used to build the documentation to 4.2.0. Patch by Maciej Olko.

Add a new "relevant PEPs" section to the top of the documentation for the `typing` module. Patch by Alex Waygood.

Add `level` argument to `multiprocessing.log_to_stderr` function docs.

Add protocol description to the :class:`importlib.abc.TraversableResources` documentation.

Mention in the documentation of :ref:`Built-in Exceptions <bltin-exceptions>` that inheriting from multiple exception types in a single subclass is not recommended due to possible memory layout incompatibility.

Add note about PEP 585 in :mod:`collections.abc`.

Add protocol description to the :class:`importlib.abc.Traversable` documentation.

Add Programming FAQ entry explaining that int literal attribute access requires either a space after or parentheses around the literal.

Add tests for scenarios in which :class:`functools.singledispatchmethod` is stacked on top of a method that has already been wrapped by two other decorators. Patch by Alex Waygood.

Add tests for :func:`dis.distb`

Add tests to ensure that `functools.singledispatchmethod` correctly wraps the attributes of the target function.

PGO tests now pass when Python is built without test extension modules.

Add subtests for all `pickle` protocols in `test_zoneinfo`.

Fix `test_frozen_pickle` in `test_dataclasses` to check all `pickle` versions.

:mod:`test.libregrtest` now raises the soft resource limit for the maximum number of file descriptors when the default is too low for our test suite as was often the case on macOS.

Add more test cases for *@functools.singledispatchmethod* when combined with *@classmethod* or *@staticmethod*.

When libregrtest spawns a worker process, stderr is now written into stdout to keep messages order. Use a single pipe for stdout and stderr, rather than two pipes. Previously, messages were out of order which made analysis of buildbot logs harder Patch by Victor Stinner.

Fix test_tools.test_sundry() when Python is built out of tree: fix how the freeze_modules.py tool locates the _freeze_module program. Patch by Victor Stinner.

Fix test_sys.test_stdlib_dir() when Python is built outside the source tree: compare normalized paths. Patch by Victor Stinner.

Fix test_name_error_suggestions_do_not_trigger_for_too_many_locals() of test_exceptions if a directory name contains "a1" (like "Python-3.11.0a1"): use a stricter regular expression. Patch by Victor Stinner.

`setup.py` now uses values from configure script to build the `_uuid` extension module. Configure now detects util-linux's `libuuid`, too.

Fix warning of `swprintf` and `%s` usage in `_testembed.c`

`Modules/Setup` and `Modules/makesetup` have been improved. The `Setup` file now contains working rules for all extensions. Outdated comments have been removed. Rules defined by `makesetup` track dependencies correctly.

The :mod:`math` and :mod:`cmath` implementation now require a C99 compatible `libm` and no longer ship with workarounds for missing acosh, asinh, atanh, expm1, and log1p functions.

> **System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\[cpython-main][Misc][NEWS.d]3.11.0a2.rst`, line 995);** *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]3.11.0a2.rst`, line 995);** *backlink*
>
> Unknown interpreted text role "mod".

`setup.py` and `makesetup` now track build dependencies on all Python header files and module specific header files.

`Modules/Setup` now use `PY_CFLAGS_NODIST` instead of `PY_CFLAGS` to compile shared modules.

:mod:`pyexpat` and :mod:`_elementtree` no longer define obsolete macros `HAVE_EXPAT_CONFIG_H` and `USE_PYEXPAT_CAPI`. `XML_POOR_ENTROPY` is now defined in `expat_config.h`.

> **System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\[cpython-main][Misc][NEWS.d]3.11.0a2.rst`, line 1026);** *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]3.11.0a2.rst`, line 1026);** *backlink*
>
> Unknown interpreted text role "mod".

`setup.py` no longer defines `Py_BUILD_CORE_MODULE`. Instead every module, that uses the internal API, defines the macro.

Fill in missing entries in Modules/Setup.

Update :data:`sys.version` to use `main` as fallback information. Patch by Jeong YunWon.

> **System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\[cpython-main][Misc][NEWS.d]3.11.0a2.rst`, line 1056);** *backlink*
>
> Unknown interpreted text role "data".

The `configure` script now checks whether OpenSSL headers and libraries provide required APIs. Most common APIs are verified. The check detects outdated or missing OpenSSL. Failures do not stop configure.

Fixed regression in handling of `LDFLAGS` and `CPPFLAGS` options where :meth:`argparse.parse_known_args` could interpret an option as one of the built-in command line argument, for example `-h` for help.

> **System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\[cpython-main][Misc][NEWS.d]3.11.0a2.rst`, line 1077);** *backlink*
>
> Unknown interpreted text role "meth".

Building Python now requires a C99 `<math.h>` header file providing the following functions: `copysign()`, `hypot()`, `isfinite()`, `isinf()`, `isnan()`, `round()`. Patch by Victor Stinner.

Prevent `internal configure error` when running `configure` with recent versions of non-Apple clang. Patch by David

Bohman.

Avoid linking libpython with libcrypt.

Update Tcl/Tk to 8.6.11, actually this time. The previous update incorrectly included 8.6.10.

venv now warns when the created environment may need to be accessed at a different path, due to redirections, links or junctions. It also now correctly installs or upgrades components when the alternate path is required.

Build SQLite `SQLITE_OMIT_AUTOINIT` on Windows. Patch by Erlend E. Aasland.

Avoid tkinter file dialog failure on macOS 12 Monterey when using the Tk 8.6.11 provided by python.org macOS installers. Patch by Marc Culler of the Tk project.

Add context keywords 'case' and 'match' to completions list.

:c:func:`PyType_FromSpec* <PyType_FromModuleAndSpec>` now copies the class name from the spec to a buffer owned by the class, so the original can be safely deallocated. Patch by Petr Viktorin.

> **System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\[cpython-main][Misc][NEWS.d]3.11.0a2.rst`, line 1170); *backlink***
>
> Unknown interpreted text role "c:func".

The internal freelists for frame, float, list, dict, async generators, and context objects can now be disabled.

Exclude :c:func:`PyWeakref_GET_OBJECT` from the limited C API. It never worked since the :c:type:`PyWeakReference` structure is opaque in the limited C API.

> **System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\[cpython-main][Misc][NEWS.d]3.11.0a2.rst`, line 1191); *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]3.11.0a2.rst`, line 1191); *backlink***
>
> Unknown interpreted text role "c:type".

Move the `interpreteridobject.h` header file from `Include/` to `Include/internal/`. It only provides private functions. Patch by Victor Stinner.

The non-limited API files `cellobject.h`, `classobject.h`, `context.h`, `funcobject.h`, `genobject.h` and `longintrepr.h` have been moved to the `Include/cpython` directory. Moreover, the `eval.h` header file was removed. These files must not be included directly, as they are already included in `Python.h`: :ref:`Include Files <api-includes>`. If they have been included directly, consider including `Python.h` instead. Patch by Victor Stinner.

> **System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\[cpython-main][Misc][NEWS.d]3.11.0a2.rst`, line 1213); *backlink***
>
> Unknown interpreted text role "ref".

The following items are no longer available when `Py_LIMITED_API` is defined:

- :c:func:`PyMarshal_WriteLongToFile`

  > **System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\[cpython-main][Misc][NEWS.d]3.11.0a2.rst`, line 1231); *backlink***
  >
  > Unknown interpreted text role "c:func".

- :c:func:`PyMarshal_WriteObjectToFile`

  > **System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\[cpython-main][Misc][NEWS.d]3.11.0a2.rst`, line 1232); *backlink***
  >
  > Unknown interpreted text role "c:func".

- :c:func:`PyMarshal_ReadObjectFromString`

- :c:func:`PyMarshal_WriteObjectToString`

- the `Py_MARSHAL_VERSION` macro

These are not part of the :ref:`limited API <stable-abi-list>`.

Patch by Victor Stinner.

Remove the `pystrhex.h` header file. It only contains private functions. C extensions should only include the main `<Python.h>` header file. Patch by Victor Stinner.

Remove the `Py_FORCE_DOUBLE()` macro. It was used by the `Py_IS_INFINITY()` macro. Patch by Victor Stinner.

`<Python.h>` no longer includes the header files `<stdlib.h>`, `<stdio.h>`, `<errno.h>` and `<string.h>` when the `Py_LIMITED_API` macro is set to `0x030b0000` (Python 3.11) or higher. C extensions should explicitly include the header files after `#include <Python.h>`. Patch by Victor Stinner.

Remove `Py_UNICODE_COPY()` and `Py_UNICODE_FILL()` macros, deprecated since Python 3.3. Use `PyUnicode_CopyCharacters()` or `memcpy()` (`wchar_t*` string), and `PyUnicode_Fill()` functions instead. Patch by Victor Stinner.

Remove the following math macros using the `errno` variable:

- `Py_ADJUST_ERANGE1()`
- `Py_ADJUST_ERANGE2()`
- `Py_OVERFLOWED()`
- `Py_SET_ERANGE_IF_OVERFLOW()`
- `Py_SET_ERRNO_ON_MATH_ERROR()`

Patch by Victor Stinner.

Custom frozen modules (the array set to `PyImport_FrozenModules`) are now treated as additions, rather than replacing all the default frozen modules. Frozen stdlib modules can still be disabled by setting the "code" field of the custom array entry to NULL.

Add new :c:func:`PyThreadState_EnterTracing`, and :c:func:`PyThreadState_LeaveTracing` functions to the limited C API to suspend and resume tracing and profiling. Patch by Victor Stinner.

:c:var:`PyStructSequence_UnnamedField` is added to the Stable ABI.