Restore behavior from 3.9 and earlier when sending non-None to newly started generator. In 3.9 this did not affect the state of the generator. In 3.10.0 and 3.10.1 `gen_func().send(0)` is equivalent to `gen_func().throw(TypeError(...))` which exhausts the generator. In 3.10.2 onward, the behavior has been reverted to that of 3.9.

Fix the :exc:`SyntaxError` location for errors involving for loops with invalid targets. Patch by Pablo Galindo

:c:func:`_PyErr_ChainStackItem` no longer normalizes `exc_info` (including setting the traceback on the exception instance) because `exc_info` is always normalized.

The `__note__` field was added to :exc:`BaseException`. It is `None` by default but can be set to a string which is added to the exception's traceback.

Place pointers to dict and values immediately before GC header. This reduces number of dependent memory loads to access either dict or values from 3 to 1.

`is_valid_fd` now uses faster `fcntl(fd, F_GETFD)` on Linux, macOS, and Windows.

Reverts a change to the `code.__new__` :ref:`audit event <audit-events>` from an earlier prerelease.

Fail the configure step if the selected compiler doesn't support memory sanitizer. Patch by Pablo Galindo

The three values of `exc_info` are now always consistent with each other. In particular, the `type` and `traceback` fields are now derived from the exception instance. This impacts the return values of :func:`sys.exc_info` and :c:func:`PyErr_GetExcInfo()` if the exception instance is modified while the exception is handled, as well as :c:func:`PyErr_SetExcInfo()`, which now ignores the `type` and `traceback` arguments provided to it.

Refine the custom syntax error that suggests that a comma may be missing to trigger only when the expressions are detected between parentheses or brackets. Patch by Pablo Galindo

Specialized the `COMPARE_OP` opcode using the PEP 659 machinery.

Allocate space for the interpreter frame in the frame object, to avoid an additional allocation when the frame object outlives the frame activation.

Fix :mod:`traceback` display for exceptions with invalid module name.

Fix crash when calling coro.cr_frame.clear() after coroutine has been freed.

Improve the tokenizer errors when encountering invisible control characters in the parser. Patch by Pablo Galindo

Allow the parser to obtain error lines directly from encoded files. Patch by Pablo Galindo

Restore behavior from 3.10 when tracing an exception raised within a with statement.

Adds new :opcode:`COPY_FREE_VARS` opcode, to make copying of free variables from function to frame explicit. Helps optimization of calls to Python function.

Specialize :opcode:`BINARY_SUBSCR` for classes with a `__getitem__` method implemented in Python

Fixed a crash when calling `.with_traceback(None)` on `NameError`. This occurs internally in `unittest.TestCase.assertRaises()`.

Fixed a bug in the parser that was causing it to not respect PEP 263 coding cookies when no flags are provided. Patch by Pablo Galindo

Fix a segfault when the parser fails without reading any input. Patch by Pablo Galindo

Simplify the implementation of :opcode:`BINARY_OP` by indexing into an array of function pointers (rather than switching on the oparg).

Fix crash when :func:`os.fork` is called with an active non-default memory allocator.

Fix computation of error location for invalid continuation characters in the parser. Patch by Pablo Galindo.

Remove an existing "fast path" for old-style string formatting, since it no longer appears to have any measurable impact.

Make recursion checks a bit more efficient by tracking amount of calls left before overflow.

Fix a compiler hang when attempting to optimize certain jump patterns.

The parser now gives a better error message when leaving out the opening parenthesis `(` after a `def`-statement:

```
>>> def f:
  File "<stdin>", line 1
    def f:
         ^
SyntaxError: expected '('
```

Specialized the `STORE_SUBSCR` opcode using the PEP 659 machinery.

Replace all numeric `BINARY_*` and `INPLACE_*` instructions with a single :opcode:`BINARY_OP` implementation.

Path calculation (known as `getpath`) has been reimplemented as a frozen Python module. This should have no visible impact, but may affect calculation of all paths referenced in :mod:`sys` and :mod:`sysconfig`.

Improve the syntax error message for parenthesized arguments. Patch by Pablo Galindo.

Fix possible crash when getting an attribute of class:*xml.etree.ElementTree.Element* simultaneously with replacing the `attrib` dict.

Make :mod:`asyncio` normalize exceptions as soon as they are captured with :c:func:`PyErr_Fetch`, and before they are stored as an exc_info triplet. This brings :mod:`asyncio` in line with the rest of the codebase, where an exc_info triplet is always normalized.

Replaced asserts with exceptions in asyncio, patch by Kumar Aditya.

:class:`unittest.TextTestResult` and :class:`unittest.TextTestRunner` flush now the output stream more often.

Added :func:`math.exp2`:, which returns 2 raised to the power of x.

Fix issue when on certain conditions `asyncio.wait_for()` may allow a coroutine to complete successfully, but fail to return the result, potentially causing memory leaks or other issues.

Improve the accuracy of stdev() and pstdev() in the statistics module. When the inputs are floats or fractions, the output is a correctly rounded float

Handle dataclass(slots=True) with a field that has default a default value, but for which init=False.

Added missing kw_only parameter to dataclasses.make_dataclass().

The :meth:`turtle.RawTurtle.settiltangle` is deprecated since Python 3.1, it now emits a deprecation warning and will be removed in Python 3.13.

Use :meth:`turtle.RawTurtle.tiltangle` instead.

:meth:`turtle.RawTurtle.tiltangle` was earlier incorrectly marked as deprecated, its docstring has been corrected.

Patch by Hugo van Kemenade.

:mod:`faulthandler` can now write ASCII-only strings (like filenames and function names) with a single write() syscall when dumping a traceback. It reduces the risk of getting an unreadable dump when two threads or two processes dump a traceback to the same file (like stderr) at the same time. Patch by Victor Stinner.

:mod:`sqlite` C callbacks now use unraisable exceptions if callback tracebacks are enabled. Patch by Erlend E. Aasland.

Fix thread lock in `zlib.Decompress.flush()` method before `PyObject_GetBuffer`.

Reverted an argparse bugfix that caused regression in the handling of default arguments for subparsers. This prevented leaf level arguments from taking precedence over root level arguments.

Fix a regression in Python 3.11a1 and 3.11a2 where :mod:`sqlite3` incorrectly would use `SQLITE_LIMIT_LENGTH` when checking SQL statement lengths. Now, `SQLITE_LIMIT_SQL_LENGTH` is used. Patch by Erlend E. Aasland.

Added *proportional* option to :meth:`statistics.linear_regression`.

In importlib.metadata, fix distribution discovery for an empty path.

Fix bug where :mod:`dis` produced an incorrect oparg when :opcode:`EXTENDED_ARG` is followed by an opcode that does not use its argument.

Unknown interpreted text role "mod".

In-place JSON file formatting using `python3 -m json.tool infile infile` now works correctly, previously it left the file empty. Patch by Chris Wesseling.

When a namespace package is imported before another module from the same namespace is created/installed in a different :data:`sys.path` location while the program is running, calling the :func:`importlib.invalidate_caches` function will now also guarantee the new module is noticed.

Improve output of `dir()` with Enums.

Fix :func:`types.resolve_bases` and :func:`types.new_class` for :class:`types.GenericAlias` instance as a base.

Fix :func:`dataclasses.is_dataclass` for dataclasses which are subclasses of :class:`types.GenericAlias`.

Fix the repr of :data:`dataclasses.InitVar` with a type alias to the built-in class, e.g. `InitVar[list[int]]`.

Launch GNOME web browsers via gio tool instead of obsolete gvfs-open

On Windows, :func:`time.sleep` now uses a waitable timer which supports high-resolution timers. Patch by Dong-hee Na and Eryk Sun.

Optimize :func:`math.comb` and :func:`math.perm`.

Deprecated legacy functions in :mod:`importlib.resources`.

Add tests for truncated/missing trailers in gzip.decompress implementation.

Implement PEP 585 for :class:`graphlib.TopologicalSorter`.

Add `max_tasks_per_child` to :class:`concurrent.futures.ProcessPoolExecutor`. This allows users to specify the maximum number of tasks a single process should execute before the process needs to be restarted.

Improve netrc library. netrc file no longer needs to contain all tokens. And if the login name is anonymous, security check is no longer need.

Avoid a possible *"RuntimeError: dictionary changed size during iteration"* when adjusting the process count of :class:`ProcessPoolExecutor`.

Add MIME types for N-quads, N-triples, Notation3 and TriG to `mimetypes`.

Add :func:`inspect.getmembers_static` , it return all members without triggering dynamic lookup via the descriptor protocol. Patch by Weipeng Hong.

`make -C Doc suspicious` will be removed soon in favor of `make -C Doc check`, mark it as deprecated.

Improve cross-references in the documentation for the data model.

Properly marked-up grammar tokens in the documentation are now clickable and take you to the definition of a given piece of grammar. Patch by Arthur Milchior.

Link doc for sys.prefix to sysconfig doc on installation paths.

`socket.socket` documentation is corrected to a class from a function.

Update the docstring of the :class:`type` built-in to remove a redundant line and to mention keyword arguments for the constructor.

Update the documentation to note that CPython does not consistently require iterators to define `__iter__`.

In the extending chapter of the extending doc, update a paragraph about the global variables containing exception information.

Expanded :func:`~dataclasses.astuple` and :func:`~dataclasses.asdict` docs, warning about deepcopy being applied and providing a workaround.

Out-of-tree builds with a read-only source directory are now tested by CI.

Add new Test for `Lib/email/mime/nonmultipart.py::MIMENonMultipart`.

Fix race condition in test_queue tests with multiple "feeder" threads.

The test for the freeze tool now handles file moves and deletions.

Remove the `--findleaks` command line option of regrtest: use the `--fail-env-changed` option instead. Since Python 3.7, it was a deprecated alias to the `--fail-env-changed` option.

Add tests with `tuple` type with :func:`functools.lru_cache` to `test_functools`.

CI now verifies that autoconf files have been regenerated with a current and unpatched autoconf package.

The build system now uses a :program:`_bootstrap_python` interpreter for freezing and deepfreezing again. To speed up build process the build tools :program:`_bootstrap_python` and :program:`_freeze_module` are no longer build with LTO.

The :program:`configure` script now accepts `--with-build-python` and `--with-freeze-module` options to make cross compiling easier.

Emscripten platform now uses `.wasm` suffix by default.

Disable unusable core extension modules on WASM/Emscripten targets.

`configure` now checks for socket `shutdown` function. The check makes it possible to disable `SYS_shutdown` with `ac_cv_func_shutdown=no` in CONFIG_SITE.

`configure` now checks for functions `fork1`, `getegid`, `geteuid`, `getgid`, `getppid`, `getuid`, `opendir`, `pipe`, `system`, `wait`, `ttyname`.

Update `config.guess` to 2021-06-03 and `config.sub` to 2021-08-14. `Makefile` now has an `update-config` target to make updating more convenient.

`make regen-all` now produces the same output when run from a directory other than the source tree: when building Python out of the source tree. pegen now strips directory of the "generated by pygen from <FILENAME>" header Patch by Victor Stinner.

`configure` now accepts machine `wasm32` or `wasm64` and OS `wasi` or `emscripten` for cross building, e.g. `wasm32-unknown-emscripten`, `wasm32-wasi`, or `wasm32-unknown-wasi`.

Python now compiles on platforms without `sigset_t`. Several functions in :mod:`signal` are not available when `sigset_t` is missing.

> **System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\(cpython-main)(Misc)(NEWS.d)3.11.0a3.rst`, line 965);** *backlink*
>
> Unknown interpreted text role "mod".

Based on patch by Roman Yurchak for pyodide.

`setup.py` now uses `CC` from environment first to discover multiarch and cross compile paths.

The `_freeze_module` program path can now be overridden on the command line, e.g. `make FREEZE_MODULE=../x86_64/Program/_freeze_module`.

Get rid of the `_bootstrap_python` build step. The deepfreeze.py script is now run using `$(PYTHON_FOR_REGEN)` which can be Python 3.7 or newer (on Windows, 3.8 or newer).

Port builtin hashlib extensions to `PY_STDLIB_MOD` macro and `addext()`.

Add `autoconf` helpers for saving and restoring environment variables:

- `SAVE_ENV`: Save `$CFLAGS`, `$LDFLAGS`, `$LIBS`, and `$CPPFLAGS`.
- `RESTORE_ENV`: Restore `$CFLAGS`, `$LDFLAGS`, `$LIBS`, and `$CPPFLAGS`.
- `WITH_SAVE_ENV([SCRIPT])`: Run `SCRIPT` wrapped with `SAVE_ENV` and `RESTORE_ENV`.

Patch by Erlend E. Aasland.

Mandatory core modules, that are required to bootstrap Python, are now in `Modules/Setup.bootstrap`.

`configure` now creates `Modules/Setup.stdlib` with conditionally enabled/disabled extension module lines. The file is not used, yet.

`configure` now uses a unified format to set state, compiler flags, and linker flags in Makefile. The new macro `PY_STDLIB_MOD` sets three variables that are consumed by `Modules/Setup` and `setup.py`.

Python now supports building with Visual Studio 2022 (MSVC v143, VS Version 17.0). Patch by Jeremiah Vivian.

Settings for :mod:`pyexpat` C extension are now detected by `configure`. The bundled `expat` library is built in `Makefile`.

> **System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\(cpython-main)(Misc)(NEWS.d)3.11.0a3.rst`, line 1076);** *backlink*
>
> Unknown interpreted text role "mod".

Settings for :mod:`decimal` internal C extension are now detected by `configure`. The bundled `libmpdec` library is built in `Makefile`.

> **System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\(cpython-main)(Misc)(NEWS.d)3.11.0a3.rst`, line 1086);** *backlink*
>
> Unknown interpreted text role "mod".

:program:`configure` has a new option `--with-pkg-config` to disable or require pkg-config.

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

The build dependencies for :mod:`sqlite3` are now detected by `configure` and `pkg-config`. Patch by Erlend E. Aasland.

The build dependencies for :mod:`zlib`, :mod:`bz2`, and :mod:`lzma` are now detected by `configure`.

gdbm and dbm build dependencies are now detected by `configure`.

On macOS, the build system no longer passes `search_paths_first` to the linker. The flag has been the default since Xcode 4 / macOS 10.6.

`configure.ac` is now compatible with autoconf 2.71. Deprecated checks `STDC_HEADERS` and `AC_HEADER_TIME` have been removed.

`configure` now prints a warning when pkg-config is missing.

`configure --enable-loadable-sqlite-extensions` is now handled by new `PY_SQLITE_ENABLE_LOAD_EXTENSION` macro instead of logic in setup.py.

configure.ac now uses custom helper macros and `AC_CACHE_CHECK` to simplify and speed up configure runs.

Skip the marshal step for frozen modules by generating C code that produces a set of ready-to-use code objects. This speeds up startup time by another 10% or more.

Run smelly.py tool from $(srcdir).

Fixed calculation of :data:`sys.path` in a venv on Windows.

When installed through the Microsoft Store and set as the default app for :file:`*.py` files, command line arguments will now be passed to Python when invoking a script without explicitly launching Python (that is, `script.py args` rather than `python script.py args`).

Fix Python Launcher's ability to distinguish between versions 3.1 and 3.10 when either one is explicitly requested. Previously, 3.1 would be used if 3.10 was requested but not installed, and 3.10 would be used if 3.1 was requested but 3.10 was installed.

Implement changes to build with deep-frozen modules on Windows. Note that we now require Python 3.10 as the "bootstrap" or "host" Python.

Updates bundled Tcl/Tk to 8.6.12.

Internal reference to :file:`shlwapi.dll` was dropped to help improve startup time. This DLL will no longer be loaded at the start of every Python process.

Update python.org macOS installer to use Tcl/Tk 8.6.12.

Fix Python.h to build C extensions with Xcode: remove a relative include from `Include/cpython/pystate.h`.

Internal reference to :file:`shlwapi.dll` was dropped to help improve startup time. This DLL will no longer be loaded at the start of every Python process.

Update python.org macOS installer to use Tcl/Tk 8.6.12.

Fix Python.h to build C extensions with Xcode: remove a relative include from `Include/cpython/pystate.h`.