`gc.freeze()` is a new API that allows for moving all objects currently tracked by the garbage collector to a permanent generation, effectively removing them from future collection events. This can be used to protect those objects from having their PyGC_Head mutated. In effect, this enables great copy-on-write stability at fork().

Restored blocking "from package import module" by setting sys.modules["package.module"] to None.

Allow use of asynchronous generator expressions in synchronous functions.

Drop support of asynchronous __aiter__.

The -u option now makes the stdout and stderr streams unbuffered rather than line-buffered.

Fixed a ValueError when convert a string with large number of underscores to integer with binary base.

Fix an assertion failure in *zipimporter.get_source()* in case of a bad *zlib.decompress()*. Patch by Oren Milman.

Fixed an assertion failure in Python parser in case of a bad *unicodedata.normalize()*. Patch by Oren Milman.

Raise a *TypeError* with a helpful error message when class creation fails due to a metaclass with a bad `__prepare__()` method. Patch by Oren Milman.

Importlib was instrumented with two dtrace probes to profile import timing.

Fix an assertion failure in *_warnings.warn()* in case of a bad `__name__` global. Patch by Oren Milman.

Improved the error message logic for object.__new__ and object.__init__.

Fix an assertion failure in *json*, in case *_json.make_encoder()* received a bad *encoder()* argument. Patch by Oren Milman.

Fix assertion failures in case of failing to import from a module with a bad `__name__` attribute, and in case of failing to access an attribute of such a module. Patch by Oren Milman.

Fix an assertion failure in *_random.Random.seed()* in case the argument has a bad `__abs__()` method. Patch by Oren Milman.

Speed up class creation by 10-20% by reducing the overhead in the necessary special method lookups. Patch by Stefan Behnel.

Add `-X importtime` option to show how long each import takes. It can be used to optimize application's startup time. Support the :envvar:`PYTHONPROFILEIMPORTTIME` as an equivalent way to enable this.

> **System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\(cpython-main)(Misc)(NEWS.d)3.7.0a2.rst`, line 168);** *backlink*
>
> Unknown interpreted text role "envvar".

Optimized calling wrapper and classmethod descriptors.

[PEP 553](#) - Add a new built-in called `breakpoint()` which calls `sys.breakpointhook()`. By default this imports `pdb` and calls `pdb.set_trace()`, but users may override `sys.breakpointhook()` to call whatever debugger they want. The original value of the hook is saved in `sys.__breakpointhook__`.

Maintain a list of open buffered files, flush them before exiting the interpreter. Based on a patch from Armin Rigo.

Fix an assertion failure in imp.create_dynamic(), when spec.name is not a string. Patch by Oren Milman.

Fix a crash in the `__setstate__()` method of *ctypes._CData*, in case of a bad `__dict__`. Patch by Oren Milman.

Fix crashes in true division and multiplication of a timedelta object by a float with a bad as_integer_ratio() method. Patch by Oren Milman.

Fix an assertion failure in *warnings.warn_explicit*, when the return value of the received loader's get_source() has a bad splitlines() method. Patch by Oren Milman.

Make `async` and `await` proper keywords, as specified in [PEP 492](#).

Fixed buffer overflow in select.kqueue.control().

`idpattern` in `string.Template` matched some non-ASCII characters. Now it uses `-i` regular expression local flag to avoid non-ASCII characters.

On Windows, faulthandler.enable() now ignores MSC and COM exceptions.

Prevent crashes in *_elementtree* due to unsafe cleanup of *Element.text* and *Element.tail*. Patch by Oren Milman.

Now `re.compile()` converts passed RegexFlag to normal int object before compiling. bm_regex_compile benchmark shows 14% performance improvements.

The types of compiled regular objects and match objects are now exposed as *re.Pattern* and *re.Match*. This adds information in pydoc output for the re module.

Fixed memory leaks in Tkinter's methods splitlist() and split() when pass a string larger than 2 GiB.

Fixed typo in the name of Tkinter's method adderrorinfo().

Improvements to path predicates in ElementTree: Allow whitespace around predicate parts, i.e. "[a = 'text']" instead of requiring the

less readable "[a='text']". Add support for text comparison of the current node, like "[.='text']". Patch by Stefan Behnel.

Fix the string representation of a netrc object.

Add optional argument `compressed` to `zipapp.create_archive`, and add option `--compress` to the command line interface of `zipapp`.

Avoid venv activate failures with undefined variables

Avoid ctypes use (if possible) and improve import time for uuid.

The regular expression cache is no longer completely dumped when it is full.

Added pthread_getcpuclockid() to the time module

Make 2to3 accept a trailing comma in generator expressions. For example, `set(x for x in [],)` is now allowed.

Stop crashes when concurrently iterate over itertools.groupby() iterators.

An iterator produced by itertools.groupby() iterator now becomes exhausted after advancing the groupby iterator.

Cancel asyncio.wait_for future faster if timeout <= 0

Allow passing a context object in :class:`concurrent.futures.ProcessPoolExecutor` constructor. Also, free job resources in :class:`concurrent.futures.ProcessPoolExecutor` earlier to improve memory usage when a worker waits for new jobs.

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\(cpython-main)(Misc)(NEWS.d)3.7.0a2.rst`, **line 444**); *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)3.7.0a2.rst`, **line 444**); *backlink*
>
> Unknown interpreted text role "class".

`threading.current_thread()` should not return a dummy thread at shutdown.

In the sqlite module, require the sqlite3_prepare_v2 API. Thus, the sqlite module now requires sqlite version at least 3.3.9.

argparse subparsers are now required by default. This matches behaviour in Python 2. For optional subparsers, use the new parameter `add_subparsers(required=False)`. Patch by Anthony Sottile. (As of 3.7.0rc1, the default was changed to not required as had been the case since Python 3.3.)

Reprs of subclasses of some collection and iterator classes (*bytearray*, *array.array*, *collections.deque*, *collections.defaultdict*, *itertools.count*, *itertools.repeat*) now contain actual type name insteads of hardcoded name of the base class.

python -m ensurepip now exits with non-zero exit code if pip bootstrapping has failed.

`pdb.set_trace()` now takes an optional keyword-only argument `header`. If given, this is printed to the console just before debugging begins.

Fix incorrect usage of `get_history_length` in readline documentation example code. Patch by Brad Smith.

The operator functions without double underscores are preferred for clarity. The one with underscores are only kept for back-compatibility.

Improve compiler version information in :data:`sys.version` when Python is built with Clang.

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\(cpython-main)(Misc)(NEWS.d)3.7.0a2.rst`, **line 540**); *backlink*
>
> Unknown interpreted text role "data".

Stop using ranlib on static libraries. Instead, we assume ar supports the 's' flag.

Remove support for BSD/OS.

Prevent double substitution of prefix in python-config.sh.

Correct PCBuild/ case to PCbuild/ in build scripts and documentation.

Avoid wholesale rebuild after *make regen-all* if nothing changed.

Simplify the API of IDLE's Module Browser. Passing a widget instead of an flist with a root widget opens the option of creating a browser frame that is only part of a window. Passing a full file name instead of pieces assumed to come from a .py file opens the possibility of browsing python files that do not end in .py.

IDLE - Make _htest, _utest parameters keyword only.

Remove test order dependence in idle_test.test_browser.

Rename IDLE's module browser from Class Browser to Module Browser. The original module-level class and method browser became a module browser, with the addition of module-level functions, years ago. Nested classes and functions were added yesterday. For back-compatibility, the virtual event <<open-class-browser>>, which appears on the Keys tab of the Settings dialog, is not changed. Patch by Cheryl Sabella.

Default fonts now are scaled on HiDPI displays.

IDLE module browser now shows nested classes and functions. Original patches for code and tests by Guilherme Polo and Cheryl Sabella, respectively.

Make *PyMapping_Keys()*, *PyMapping_Values()* and *PyMapping_Items()* always return a *list* (rather than a *list* or a *tuple*). Patch by Oren Milman.

Fix memory corruption due to allocator mix in getpath.c between Py_GetPath() and Py_SetPath()

Implement PEP 539 for Thread Specific Storage (TSS) API: it is a new Thread Local Storage (TLS) API to CPython which would supersede use of the existing TLS API within the CPython interpreter, while deprecating the existing API. PEP written by Erik M. Bray, patch by Masayuki Yamamoto.