What's New in Python

Release 3.8.0a0

A. M. Kuchling

October 25, 2018

Python Software Foundation
Email: docs@python.org

Contents

This article explains the new features in Python 3.8, compared to 3.7.

For full details, see the changelog.

Note: Prerelease users should be aware that this document is currently in draft form. It will be updated substantially as Python 3.8 moves towards release, so it's worth checking back even after reading earlier versions.

1 Summary – Release highlights

2 New Features

3 Other Language Changes

- A `continue` statement was illegal in the `finally` clause due to a problem with the implementation. In Python 3.8 this restriction was lifted. (Contributed by Serhiy Storchaka in bpo-32489.)
- Added support of `\N{name}` escapes in `regular expressions`. (Contributed by Jonathan Eunice and Serhiy Storchaka in bpo-30688.)

4 New Modules

- None yet.

5 Improved Modules

6 Optimizations

- `shutil.copyfile()`, `shutil.copy()`, `shutil.copy2()`, `shutil.copytree()` and `shutil.move()` use platform-specific "fast-copy" syscalls on Linux, OSX and Solaris in order to copy the file more efficiently. "fast-copy" means that the copying operation occurs within the kernel, avoiding the use of userspace buffers in Python as in "`outfd.write(infd.read())`". All other platforms not using such technique will rely on a faster `shutil.copyfile()` implementation using `memoryview()`, `bytearray` and `BufferedIOBase.readinto()`. Finally, `shutil.copyfile()` default buffer size on Windows was increased from 16KB to 1MB. The speedup for copying a 512MB file within the same partition is about +26% on Linux, +50% on OSX and +38% on Windows. Also, much less CPU cycles are consumed. (Contributed by Giampaolo Rodola' in bpo-25427.)
- The default protocol in the `pickle` module is now Protocol 4, first introduced in Python 3.4. It offers better performance and smaller size compared to Protocol 3 available since Python 3.0.

7 Build and C API Changes

8 Deprecated

9 Removed

- The `pyvenv` script has been removed in favor of `python3.8 -m venv` to help eliminate confusion as to what Python interpreter the `pyvenv` script is tied to. (Contributed by Brett Cannon in bpo-25427.)

10 Porting to Python 3.8

This section lists previously described changes and other bugfixes that may require changes to your code.

10.1 Changes in Python behavior

- Yield expressions (both `yield` and `yield from` clauses) are now disallowed in comprehensions and generator expressions (aside from the iterable expression in the leftmost `for` clause). (Contributed by Serhiy Storchaka in bpo-10544.)

10.2 Changes in the Python API

- The `selection()` method of the `tkinter.ttk.Treeview` class no longer takes arguments. Using it with arguments for changing the selection was deprecated in Python 3.6. Use specialized methods like `selection_set()` for changing the selection. (Contributed by Serhiy Storchaka in bpo-31508.)
- A `dbm.dumb` database opened with flags `'r'` is now read-only. `dbm.dumb.open()` with flags `'r'` and `'w'` no longer creates a database if it does not exist. (Contributed by Serhiy Storchaka in bpo-32749.)
- A `RuntimeError` is now raised when the custom metaclass doesn't provide the `__classcell__` entry in the namespace passed to `type.__new__`. A `DeprecationWarning` was emitted in Python 3.6–3.7. (Contributed by Serhiy Storchaka in bpo-23722.)
- The `cProfile.Profile` class can now be used as a context manager. (Contributed by Scott Sanderson in bpo-29235.)

10.3 CPython bytecode changes

- The interpreter loop has been simplified 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 `break`, `continue` and `return`.

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

  (Contributed by Mark Shannon, Antoine Pitrou and Serhiy Storchaka in bpo-17611.)
- Added new opcode `END_ASYNC_FOR` for handling exceptions raised when awaiting a next item in an `async for` loop. (Contributed by Serhiy Storchaka in bpo-33041.)