What's New in Python

Release 3.14.0a0

A. M. Kuchling

luglio 04, 2024

Python Software Foundation
Email: docs@python.org

Indice

Editor
TBD

This article explains the new features in Python 3.14, compared to 3.13.

For full details, see the changelog.

Nota: Prerelease users should be aware that this document is currently in draft form. It will be updated substantially as Python 3.14 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

4 New Modules

- None yet.

5 Improved Modules

5.1 ast

Added `ast.compare()` for comparing two ASTs. (Contributed by Batuhan Taskaya and Jeremy Hylton in bpo-15987.)

5.2 os

- Added the `os.environ.refresh()` method to update `os.environ` with changes to the environment made by `os.putenv()`, by `os.unsetenv()`, or made outside Python in the same process. (Contributed by Victor Stinner in gh-120057.)

5.3 pathlib

- Add `pathlib.Path.copy()`, which copies the content of one file to another, like `shutil.copyfile()`. (Contributed by Barney Gale in gh-73991.)
- Add `pathlib.Path.copytree()`, which copies one directory tree to another. (Contributed by Barney Gale in gh-73991.)

5.4 symtable

- Expose the following `symtable.Symbol` methods:
  - `is_free_class()`
  - `is_comp_iter()`
  - `is_comp_cell()`

  (Contributed by Bénédikt Tran in gh-120029.)

6 Optimizations

6.1 asyncio

- `asyncio` now uses double linked list implementation for native tasks which speeds up execution by 10% on standard pyperformance benchmarks and reduces memory usage. (Contributed by Kumar Aditya in gh-107803.)

7 Deprecated

- Passing a complex number as the *real* or *imag* argument in the `complex()` constructor is now deprecated; it should only be passed as a single positional argument. (Contributed by Serhiy Storchaka in gh-109218.)
- Soft deprecate `os.popen()` and `os.spawn*` functions. They should no longer be used to write new code. The `subprocess` module is recommended instead. (Contributed by Victor Stinner in gh-120743.)

8 Removed

8.1 argparse

- Remove the *type*, *choices*, and *metavar* parameters of `argparse.BooleanOptionalAction`. They were deprecated since 3.12.

8.2 ast

- Remove the following classes. They were all deprecated since Python 3.8, and have emitted deprecation warnings since Python 3.12:
  - `ast.Num`
  - `ast.Str`
  - `ast.Bytes`
  - `ast.NameConstant`
  - `ast.Ellipsis`

  Use `ast.Constant` instead. As a consequence of these removals, user-defined `visit_Num`, `visit_Str`, `visit_Bytes`, `visit_NameConstant` and `visit_Ellipsis` methods on custom `ast.NodeVisitor` subclasses will no longer be called when the `NodeVisitor` subclass is visiting an AST. Define a `visit_Constant` method instead.

  Also, remove the following deprecated properties on `ast.Constant`, which were present for compatibility with the now-removed AST classes:
  - `ast.Constant.n`
  - `ast.Constant.s`

  Use `ast.Constant.value` instead.

  (Contributed by Alex Waygood in gh-119562.)

8.3 asyncio

- Remove the following classes and functions. They were all deprecated and emitted deprecation warnings since Python 3.12:
  - `asyncio.AbstractChildWatcher`
  - `asyncio.SafeChildWatcher`
  - `asyncio.MultiLoopChildWatcher`
  - `asyncio.FastChildWatcher`
  - `asyncio.ThreadedChildWatcher`
  - `asyncio.PidfdChildWatcher`
  - `asyncio.AbstractEventLoopPolicy.get_child_watcher()`
  - `asyncio.AbstractEventLoopPolicy.set_child_watcher()`
  - `asyncio.get_child_watcher()`
  - `asyncio.set_child_watcher()`

  (Contributed by Kumar Aditya in gh-120804.)

8.4 collections.abc

- Remove `collections.abc.ByteString`. It had previously raised a `DeprecationWarning` since Python 3.12.

8.5 email

- Remove the *isdst* parameter from `email.utils.localtime()`. (Contributed by Hugo van Kemenade in gh-118798.)

8.6 importlib

- Remove deprecated `importlib.abc` classes:
  - `importlib.abc.ResourceReader`
  - `importlib.abc.Traversable`
  - `importlib.abc.TraversableResources`

  Use `importlib.resources.abc` classes instead:
  - `importlib.resources.abc.Traversable`
  - `importlib.resources.abc.TraversableResources`

  (Contributed by Jason R. Coombs and Hugo van Kemenade in gh-93963.)

8.7 itertools

- Remove `itertools` support for copy, deepcopy, and pickle operations. These had previously raised a `DeprecationWarning` since Python 3.12. (Contributed by Raymond Hettinger in gh-101588.)

8.8 pathlib

- Remove support for passing additional keyword arguments to `pathlib.Path`. In previous versions, any such arguments are ignored.
- Remove support for passing additional positional arguments to `pathlib.PurePath.relative_to()` and `is_relative_to()`. In previous versions, any such arguments are joined onto *other*.

pty

- Remove deprecated `pty.master_open()` and `pty.slave_open()`. They had previously raised a `DeprecationWarning` since Python 3.12. Use `pty.openpty()` instead. (Contributed by Nikita Sobolev in gh-118824.)

8.9 sqlite3

- Remove `version` and `version_info` from `sqlite3`. (Contributed by Hugo van Kemenade in gh-118924.)
- Disallow using a sequence of parameters with named placeholders. This had previously raised a `DeprecationWarning` since Python 3.12; it will now raise a `sqlite3.ProgrammingError`. (Contributed by Erlend E. Aasland in gh-118928 and gh-101693.)

8.10 typing

- Remove `typing.ByteString`. It had previously raised a `DeprecationWarning` since Python 3.12.

8.11 urllib

- Remove deprecated `Quoter` class from `urllib.parse`. It had previously raised a `DeprecationWarning` since Python 3.11. (Contributed by Nikita Sobolev in gh-118827.)

8.12 Others

- Using `NotImplemented` in a boolean context will now raise a `TypeError`. It had previously raised a `DeprecationWarning` since Python 3.9. (Contributed by Jelle Zijlstra in gh-118767.)
- The `int()` built-in no longer delegates to `__trunc__()`. Classes that want to support conversion to integer must implement either `__int__()` or `__index__()`. (Contributed by Mark Dickinson in gh-119743.)

9 Porting to Python 3.14

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

10 Build Changes

11 C API Changes

11.1 New Features

- Add `PyLong_GetSign()` function to get the sign of `int` objects. (Contributed by Sergey B Kirpichev in gh-116560.)
- Add a new `PyUnicodeWriter` API to create a Python `str` object:
  - `PyUnicodeWriter_Create()`.
  - `PyUnicodeWriter_Discard()`.
  - `PyUnicodeWriter_Finish()`.
  - `PyUnicodeWriter_WriteChar()`.
  - `PyUnicodeWriter_WriteUTF8()`.
  - `PyUnicodeWriter_WriteUCS4()`.
  - `PyUnicodeWriter_WriteWideChar()`.
  - `PyUnicodeWriter_WriteStr()`.

- `PyUnicodeWriter_WriteRepr()`.
- `PyUnicodeWriter_WriteSubstring()`.
- `PyUnicodeWriter_Format()`.
- `PyUnicodeWriter_DecodeUTF8Stateful()`.

(Contributed by Victor Stinner in gh-119182.)

11.2 Porting to Python 3.14

- In the limited C API 3.14 and newer, `Py_TYPE()` is now implemented as an opaque function call to hide implementation details. (Contributed by Victor Stinner in gh-120600.)

11.3 Deprecated

- Macros `Py_IS_NAN`, `Py_IS_INFINITY` and `Py_IS_FINITE` are soft deprecated, use instead `isnan`, `isinf` and `isfinite` available from `math.h` since C99. (Contributed by Sergey B Kirpichev in gh-119613.)

11.4 Removed

- Creating `immutable types` with mutable bases was deprecated since 3.12 and now raises a `TypeError`.