

Fixed optimization of property descriptor getter.

PEP 489: Multi-phase extension module initialization. Patch by Petr Viktorin.

Add `pyvenv.cfg` option to suppress registry/environment lookup for generating `sys.path` on Windows.

Fixed system error in the comparison of faked types. `SimpleNamespace`.

Fixed integer overflow in iterator object. Patch by Clement Rouault.

Fix a possible buffer overrun when deleting a slice from the front of a bytearray and then appending some other bytes data.

Fixed exception type checking in standard error handlers.

The UTF-32 encoder is now 3x to 7x faster.

Optimize `set_merge()` for cases where the target is empty. (Contributed by Serhiy Storchaka.)

PEP 448: Additional Unpacking Generalizations.

Make `warnings.warn_explicit` more robust against mutation of the `warnings.filters` list.

Avoid a crash when a delegated generator raises an unnormalized `StopIteration` exception. Patch by Stefan Behnel.

Optimize `property()` getter calls. Patch by Joe Jevnik.

Move path-based `importlib` bootstrap code to a separate frozen module.

Fix namespace package imports.

Fix tokenizer crash when processing undecodable source code.

Added a `hex()` method to `bytes`, `bytearray`, and `memoryview`.

PEP 479: Change `StopIteration` handling inside generators.

PEP 492: Coroutines with `async` and `await` syntax.

Added C implementation of `functools.lru_cache()`. Based on patches by Matt Joiner and Alexey Kachayev.

The `tempfile` module now accepts `bytes` for `prefix`, `suffix` and `dir` parameters and returns `bytes` in such situations (matching the `os` module APIs).

`collections.UserString` now supports `__getnewargs__()`, `__rmod__()`, `casefold()`, `format_map()`, `isprintable()`, and `maketrans()`. Patch by Joe Jevnik.

Prevents termination when an invalid format string is encountered on Windows in `strftime`.

PEP 484: Add the `typing` module.

The `collections.abc.Sequence()` abstract base class added `start` and `stop` parameters to the `index()` mixin. Patch by Devin Jeanpierre.

Replaced the `tkinter._fix` module used for setting up the Tcl/Tk environment on Windows with a private function in the `_tkinter` module that makes no permanent changes to the environment.

Fixed segmentation fault in `sqlite3.Row` constructor with faked cursor type.

`assertRaises()`, `assertRaisesRegex()`, `assertWarns()` and `assertWarnsRegex()` assertions now check the type of the first argument to prevent possible user error. Based on patch by Daniel Wagner-Hall.

Add missing method stubs to `_io.RawIOBase`. Patch by Laura Rupperecht.

`attrgetter`, `itemgetter` and `methodcaller` objects in the `operator` module now support pickling. Added readable and evaluable repr for these objects. Based on patch by Josh Rosenberg.

`tempfile.gettempdir()` and `tempfile.mkdtemp()` now try again when a directory with the chosen name already exists on Windows as well as on Unix. `tempfile.mkstemp()` now fails early if parent directory is not valid (not exists or is a file) on Windows.

Improved error message in `os.path.join()` with single argument.

Increased time precision and random number range in `email.utils.make_msgid()` to strengthen the uniqueness of the message ID.

Fixed various crashes in corner cases in C implementation of `ElementTree`.

`msilib.FCICreate()` now raises `TypeError` in the case of a bad argument instead of a `ValueError` with a bogus FCI error number. Patch by Jeffrey Armstrong.

`quote_via` argument added to `urllib.parse.urlencode`.

New `mangle_from_policy` option for email, default `True` for `compat32`, but `False` for all other policies.

The email library now supports RFC 6532: it can generate headers using `utf-8` instead of encoded words.

Added support for the LZMA compression in `distutils`.

`poplib` now supports RFC 6856 (UTF8).

Optimized `pprint` functions for builtin scalar types.

`smtpplib` now supports RFC 6531 (SMTPUTF8).

Random generator objects now consume 2x less memory on 64-bit.

`platform.dist()` and `platform.linux_distribution()` functions are now deprecated. Initial patch by Vajrasky Kok.

Added the `math.gcd()` function. The `fractions.gcd()` function now is deprecated. Based on patch by Mark Dickinson.

`Property()` docstrings are now writeable. (Patch by Berker Peksag.)

Added support for the `koi8_t` encoding.

Added support for the `kz1048` encoding.

`peek` and `read1` methods of `BufferedReader` now raise `ValueError` if they called on a closed object. Patch by John Hergenroeder.

`smtplib` now supports the 8BITMIME extension whenever the new `decode_data` constructor argument is set to `False`.

optimize `heapq.heapify()` for better cache performance when heapifying large lists.

`imaplib` now supports RFC 5161 (enable), RFC 6855 (utf8/internationalized email) and automatically encodes non-ASCII usernames and passwords to UTF8.

When calling a `_sqlite.Connection`, it now complains if passed any keyword arguments. Previously it silently ignored them.

Remove ignored and erroneous "kwargs" parameters from three `METH_VARARGS` methods on `_sqlite.Connection`.

`assertRaises()`, `assertRaisesRegex()`, `assertWarns()` and `assertWarnsRegex()` checks now emits a deprecation warning when callable is `None` or keyword arguments except `msg` is passed in the context manager mode.

Add a `collections.abc.Generator` abstract base class. Contributed by Stefan Behnel.

Tkinter's `getint()` and `getdouble()` now support `Tcl_Obj`. Tkinter's `getdouble()` now supports any numbers (in particular `int`).

Added negative limit support in the `traceback` module. Based on patch by Dmitry Kazakov.

Fix possible crash in `json.encode` with poorly behaved dict subclasses.

On POSIX, `os.getcwd()` now supports paths longer than 1025 bytes. Patch written by William Orr.

add `difflib.diff_bytes()` to support comparison of byte strings (fixes a regression from Python 2).

Fall back to sequential compilation when `ProcessPoolExecutor` doesn't exist. Patch by Claudiu Popa.

Fixed resolving attributes with boolean value is `False` in `pydoc`.

Fix `asyncio` issue 235: `LifoQueue` and `PriorityQueue`'s `put` didn't increment unfinished tasks (this bug was introduced when `JoinableQueue` was merged with `Queue`).

`os` functions now reject paths with embedded null character on Windows instead of silently truncating them.

`binascii.crc_hqx()` could return an integer outside of the range 0-0xffff for empty data.

`urllib.error.HTTPError` now has a proper `repr()` representation. Patch by Berker Peksag.

`asyncio`: New event loop APIs: `set_task_factory()` and `get_task_factory()`.

`asyncio`: `async()` function is deprecated in favour of `ensure_future()`.

`asyncio.Lock`, `Condition`, `Semaphore`, and `BoundedSemaphore` support new 'async with' syntax. Contributed by Yury Selivanov.

Support 'async for' for `asyncio.StreamReader`. Contributed by Yury Selivanov.

Add `AsyncIterator` and `AsyncIterable` ABCs to `collections.abc`. Contributed by Yury Selivanov.

Implement informative `__repr__` for `inspect.BoundArguments`. Contributed by Yury Selivanov.

Implement `inspect.BoundArgument.apply_defaults()` method. Contributed by Yury Selivanov.

Add 'follow\_wrapped' argument to `inspect.Signature.from_callable()` and `inspect.signature()`. Contributed by Yury Selivanov.

Deprecate `inspect.Signature.from_function()` and `inspect.Signature.from_builtin()`.

Fix `inspect.classify_class_attrs()` to support attributes with overloaded `__eq__` and `__bool__`. Patch by Mike Bayer.

Fix `inspect.signature()` to correctly unwrap wrappers around bound methods.

remove unused names and imports in `idlelib`. Initial patch by Al Sweigart.

`test_zipfile` no longer fails if the word 'bad' appears anywhere in the name of the current directory.

Move `script_helper` into the support package. Patch by Christie Wilson.


Add File Handlers subsection with `createfilehandler` to `tkinter` doc. Remove obsolete example from FAQ. Patch by Martin Panter.

Document the name binding behavior for submodule imports.

Fix typo in man page for `-I` command option: `-s`, not `-S`

Improved Argument Clinic's mapping of converters to legacy "format units". Updated the documentation to match.

Argument Clinic converters now use `accept={type}` instead of `types={type}` to specify the types the converter accepts.



h2py now supports arbitrary filenames in `#include`.  
make patchcheck now supports `git` checkouts, too.

