:mod:`tracemalloc` --- Trace memory allocations

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)tracemalloc.rst`, line 1); *backlink*

Unknown interpreted text role "mod".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)tracemalloc.rst`, line 4)

Unknown directive type "module".

```
.. module:: tracemalloc
   :synopsis: Trace memory allocations.
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)tracemalloc.rst`, line 7)

Unknown directive type "versionadded".

```
.. versionadded:: 3.4
```

Source code: :source:`Lib/tracemalloc.py`

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)tracemalloc.rst`, line 9); *backlink*

Unknown interpreted text role "source".

The tracemalloc module is a debug tool to trace memory blocks allocated by Python. It provides the following information:

- Traceback where an object was allocated
- Statistics on allocated memory blocks per filename and per line number: total size, number and average size of allocated memory blocks
- Compute the differences between two snapshots to detect memory leaks

To trace most memory blocks allocated by Python, the module should be started as early as possible by setting the :envvar:`PYTHONTRACEMALLOC` environment variable to `1`, or by using :option:`-X` `tracemalloc` command line option. The :func:`tracemalloc.start` function can be called at runtime to start tracing Python memory allocations.

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)tracemalloc.rst`, line 21); *backlink*

Unknown interpreted text role "envvar".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)tracemalloc.rst`, line 21); *backlink*

Unknown interpreted text role "option".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)tracemalloc.rst`, line 21); *backlink*

Unknown interpreted text role "func".

By default, a trace of an allocated memory block only stores the most recent frame (1 frame). To store 25 frames at startup: set the :envvar:`PYTHONTRACEMALLOC` environment variable to `25`, or use the :option:`-X` `tracemalloc=25` command line option.

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)tracemalloc.rst`, line 27); *backlink*

Unknown interpreted text role "envvar".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)tracemalloc.rst`, line 27); *backlink*

Unknown interpreted text role "option".

Examples

Display the top 10

Display the 10 files allocating the most memory:

```
import tracemalloc

tracemalloc.start()

# ... run your application ...

snapshot = tracemalloc.take_snapshot()
top_stats = snapshot.statistics('lineno')

print("[ Top 10 ]")
for stat in top_stats[:10]:
    print(stat)
```

Example of output of the Python test suite:

```
[ Top 10 ]
<frozen importlib._bootstrap>:716: size=4855 KiB, count=39328, average=126 B
<frozen importlib._bootstrap>:284: size=521 KiB, count=3199, average=167 B
/usr/lib/python3.4/collections/__init__.py:368: size=244 KiB, count=2315, average=108 B
/usr/lib/python3.4/unittest/case.py:381: size=185 KiB, count=779, average=243 B
/usr/lib/python3.4/unittest/case.py:402: size=154 KiB, count=378, average=416 B
/usr/lib/python3.4/abc.py:133: size=88.7 KiB, count=347, average=262 B
<frozen importlib._bootstrap>:1446: size=70.4 KiB, count=911, average=79 B
<frozen importlib._bootstrap>:1454: size=52.0 KiB, count=25, average=2131 B
<string>:5: size=49.7 KiB, count=148, average=344 B
/usr/lib/python3.4/sysconfig.py:411: size=48.0 KiB, count=1, average=48.0 KiB
```

We can see that Python loaded `4855 KiB` data (bytecode and constants) from modules and that the :mod:`collections` module allocated `244 KiB` to build :class:`~collections.namedtuple` types.

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)tracemalloc.rst`, line 69); *backlink*

Unknown interpreted text role "mod".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)tracemalloc.rst`, line 69); *backlink*

Unknown interpreted text role "class".

See :meth:`Snapshot.statistics` for more options.

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)tracemalloc.rst`, line 73); *backlink*

Unknown interpreted text role "meth".

Compute differences

Take two snapshots and display the differences:

```
import tracemalloc
tracemalloc.start()
# ... start your application ...

snapshot1 = tracemalloc.take_snapshot()
# ... call the function leaking memory ...
snapshot2 = tracemalloc.take_snapshot()

top_stats = snapshot2.compare_to(snapshot1, 'lineno')

print("[ Top 10 differences ]")
for stat in top_stats[:10]:
    print(stat)
```

Example of output before/after running some tests of the Python test suite:

```
[ Top 10 differences ]
<frozen importlib._bootstrap>:716: size=8173 KiB (+4428 KiB), count=71332 (+39369), average=117 B
/usr/lib/python3.4/linecache.py:127: size=940 KiB (+940 KiB), count=8106 (+8106), average=119 B
/usr/lib/python3.4/unittest/case.py:571: size=298 KiB (+298 KiB), count=589 (+589), average=519 B
<frozen importlib._bootstrap>:284: size=1005 KiB (+166 KiB), count=7423 (+1526), average=139 B
/usr/lib/python3.4/mimetypes.py:217: size=112 KiB (+112 KiB), count=1334 (+1334), average=86 B
/usr/lib/python3.4/http/server.py:848: size=96.0 KiB (+96.0 KiB), count=1 (+1), average=96.0 KiB
/usr/lib/python3.4/inspect.py:1465: size=83.5 KiB (+83.5 KiB), count=109 (+109), average=784 B
/usr/lib/python3.4/unittest/mock.py:491: size=77.7 KiB (+77.7 KiB), count=143 (+143), average=557 B
/usr/lib/python3.4/urllib/parse.py:476: size=71.8 KiB (+71.8 KiB), count=969 (+969), average=76 B
/usr/lib/python3.4/contextlib.py:38: size=67.2 KiB (+67.2 KiB), count=126 (+126), average=546 B
```

We can see that Python has loaded `8173 KiB` of module data (bytecode and constants), and that this is `4428 KiB` more than had been loaded before the tests, when the previous snapshot was taken. Similarly, the :mod:`linecache` module has cached `940 KiB` of Python source code to format tracebacks, all of it since the previous snapshot.

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)tracemalloc.rst`, line 109); *backlink*

Unknown interpreted text role "mod".

If the system has little free memory, snapshots can be written on disk using the :meth:`Snapshot.dump` method to analyze the snapshot offline. Then use the :meth:`Snapshot.load` method reload the snapshot.

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)tracemalloc.rst`, line 115); *backlink*

Unknown interpreted text role "meth".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)tracemalloc.rst`, line 115); *backlink*

Unknown interpreted text role "meth".

Get the traceback of a memory block

Code to display the traceback of the biggest memory block:

```
import tracemalloc

# Store 25 frames
tracemalloc.start(25)

# ... run your application ...

snapshot = tracemalloc.take_snapshot()
top_stats = snapshot.statistics('traceback')

# pick the biggest memory block
stat = top_stats[0]
print("%s memory blocks: %.1f KiB" % (stat.count, stat.size / 1024))
for line in stat.traceback.format():
    print(line)
```

Example of output of the Python test suite (traceback limited to 25 frames):

```
903 memory blocks: 870.1 KiB
  File "<frozen importlib._bootstrap>", line 716
  File "<frozen importlib._bootstrap>", line 1036
  File "<frozen importlib._bootstrap>", line 934
  File "<frozen importlib._bootstrap>", line 1068
  File "<frozen importlib._bootstrap>", line 619
  File "<frozen importlib._bootstrap>", line 1581
  File "<frozen importlib._bootstrap>", line 1614
  File "/usr/lib/python3.4/doctest.py", line 101
    import pdb
  File "<frozen importlib._bootstrap>", line 284
  File "<frozen importlib._bootstrap>", line 938
  File "<frozen importlib._bootstrap>", line 1068
  File "<frozen importlib._bootstrap>", line 619
  File "<frozen importlib._bootstrap>", line 1581
  File "<frozen importlib._bootstrap>", line 1614
  File "/usr/lib/python3.4/test/support/__init__.py", line 1728
    import doctest
  File "/usr/lib/python3.4/test/test_pickletools.py", line 21
    support.run_doctest(pickletools)
  File "/usr/lib/python3.4/test/regrtest.py", line 1276
    test_runner()
  File "/usr/lib/python3.4/test/regrtest.py", line 976
    display_failure=not verbose)
  File "/usr/lib/python3.4/test/regrtest.py", line 761
    match_tests=ns.match_tests)
  File "/usr/lib/python3.4/test/regrtest.py", line 1563
    main()
  File "/usr/lib/python3.4/test/__main__.py", line 3
    regrtest.main_in_temp_cwd()
  File "/usr/lib/python3.4/runpy.py", line 73
    exec(code, run_globals)
  File "/usr/lib/python3.4/runpy.py", line 160
    "__main__", fname, loader, pkg_name)
```

We can see that the most memory was allocated in the :mod:`importlib` module to load data (bytecode and constants) from modules: `870.1 KiB`. The traceback is where the :mod:`importlib` loaded data most recently: on the `import pdb` line of the :mod:`doctest` module. The traceback may change if a new module is loaded.

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)tracemalloc.rst`, line 178); ***backlink***

Unknown interpreted text role "mod".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)tracemalloc.rst`, line 178); ***backlink***

Unknown interpreted text role "mod".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)tracemalloc.rst`, line 178); ***backlink***

Unknown interpreted text role "mod".

Pretty top

Code to display the 10 lines allocating the most memory with a pretty output, ignoring `<frozen importlib._bootstrap>` and `<unknown>` files:

```
import linecache
import os
import tracemalloc

def display_top(snapshot, key_type='lineno', limit=10):
    snapshot = snapshot.filter_traces((
        tracemalloc.Filter(False, "<frozen importlib._bootstrap>"),
        tracemalloc.Filter(False, "<unknown>"),
    ))
    top_stats = snapshot.statistics(key_type)

    print("Top %s lines" % limit)
    for index, stat in enumerate(top_stats[:limit], 1):
        frame = stat.traceback[0]
        print("#%s: %s:%s: %.1f KiB"
              % (index, frame.filename, frame.lineno, stat.size / 1024))
        line = linecache.getline(frame.filename, frame.lineno).strip()
        if line:
            print('    %s' % line)

    other = top_stats[limit:]
    if other:
        size = sum(stat.size for stat in other)
        print("%s other: %.1f KiB" % (len(other), size / 1024))
    total = sum(stat.size for stat in top_stats)
    print("Total allocated size: %.1f KiB" % (total / 1024))

tracemalloc.start()

# ... run your application ...

snapshot = tracemalloc.take_snapshot()
display_top(snapshot)
```

Example of output of the Python test suite:

```
Top 10 lines
#1: Lib/base64.py:414: 419.8 KiB
    _b85chars2 = [(a + b) for a in _b85chars for b in _b85chars]
#2: Lib/base64.py:306: 419.8 KiB
    _a85chars2 = [(a + b) for a in _a85chars for b in _a85chars]
#3: collections/__init__.py:368: 293.6 KiB
    exec(class_definition, namespace)
#4: Lib/abc.py:133: 115.2 KiB
    cls = super().__new__(mcls, name, bases, namespace)
#5: unittest/case.py:574: 103.1 KiB
    testMethod()
#6: Lib/linecache.py:127: 95.4 KiB
    lines = fp.readlines()
#7: urllib/parse.py:476: 71.8 KiB
    for a in _hexdig for b in _hexdig}
#8: <string>:5: 62.0 KiB
#9: Lib/_weakrefset.py:37: 60.0 KiB
    self.data = set()
#10: Lib/base64.py:142: 59.8 KiB
    _b32tab2 = [a + b for a in _b32tab for b in _b32tab]
6220 other: 3602.8 KiB
Total allocated size: 5303.1 KiB
```

See :meth:`Snapshot.statistics` for more options.

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)tracemalloc.rst`, line 250); *backlink*

Unknown interpreted text role "meth".

Record the current and peak size of all traced memory blocks

The following code computes two sums like `0 + 1 + 2 + ...` inefficiently, by creating a list of those numbers. This list consumes a lot of memory temporarily. We can use :func:`get_traced_memory` and :func:`reset_peak` to observe the small memory usage after the sum is computed as well as the peak memory usage during the computations:

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)tracemalloc.rst`, line 255); *backlink*

Unknown interpreted text role "func".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)tracemalloc.rst`, line 255); *backlink*

Unknown interpreted text role "func".

```
import tracemalloc

tracemalloc.start()

# Example code: compute a sum with a large temporary list
large_sum = sum(list(range(100000)))

first_size, first_peak = tracemalloc.get_traced_memory()

tracemalloc.reset_peak()

# Example code: compute a sum with a small temporary list
small_sum = sum(list(range(1000)))

second_size, second_peak = tracemalloc.get_traced_memory()

print(f"{first_size=}, {first_peak=}")
print(f"{second_size=}, {second_peak=}")
```

Output:

```
first_size=664, first_peak=3592984
second_size=804, second_peak=29704
```

Using :func:`reset_peak` ensured we could accurately record the peak during the computation of `small_sum`, even though it is much smaller than the overall peak size of memory blocks since the :func:`start` call. Without the call to :func:`reset_peak`, `second_peak` would still be the peak from the computation `large_sum` (that is, equal to `first_peak`). In this case, both peaks are much higher than the final memory usage, and which suggests we could optimise (by removing the unnecessary call to :class:`list`, and writing sum(range(...))).

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)tracemalloc.rst, line 285); *backlink*

Unknown interpreted text role "func".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)tracemalloc.rst, line 285); *backlink*

Unknown interpreted text role "func".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)tracemalloc.rst, line 285); *backlink*

Unknown interpreted text role "func".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)tracemalloc.rst, line 285); *backlink*

Unknown interpreted text role "class".

API

Functions

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)tracemalloc.rst, line 300)

Unknown directive type "function".

```
.. function:: clear_traces()

   Clear traces of memory blocks allocated by Python.

   See also :func:`stop`.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)tracemalloc.rst, line 307)

Unknown directive type "function".

```
.. function:: get_object_traceback(obj)

   Get the traceback where the Python object *obj* was allocated.
   Return a :class:`Traceback` instance, or ``None`` if the :mod:`tracemalloc`
   module is not tracing memory allocations or did not trace the allocation of
   the object.

   See also :func:`gc.get_referrers` and :func:`sys.getsizeof` functions.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)tracemalloc.rst, line 317)

Unknown directive type "function".

```
.. function:: get_traceback_limit()

   Get the maximum number of frames stored in the traceback of a trace.

   The :mod:`tracemalloc` module must be tracing memory allocations to
   get the limit, otherwise an exception is raised.

   The limit is set by the :func:`start` function.
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)tracemalloc.rst`, line 327)

Unknown directive type "function".

```
.. function:: get_traced_memory()

   Get the current size and peak size of memory blocks traced by the
   :mod:`tracemalloc` module as a tuple: ``(current: int, peak: int)``.
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)tracemalloc.rst`, line 333)

Unknown directive type "function".

```
.. function:: reset_peak()

   Set the peak size of memory blocks traced by the :mod:`tracemalloc` module
   to the current size.

   Do nothing if the :mod:`tracemalloc` module is not tracing memory
   allocations.

   This function only modifies the recorded peak size, and does not modify or
   clear any traces, unlike :func:`clear_traces`. Snapshots taken with
   :func:`take_snapshot` before a call to :func:`reset_peak` can be
   meaningfully compared to snapshots taken after the call.

   See also :func:`get_traced_memory`.

   .. versionadded:: 3.9
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)tracemalloc.rst`, line 351)

Unknown directive type "function".

```
.. function:: get_tracemalloc_memory()

   Get the memory usage in bytes of the :mod:`tracemalloc` module used to store
   traces of memory blocks.
   Return an :class:`int`.
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)tracemalloc.rst`, line 358)

Unknown directive type "function".

```
.. function:: is_tracing()

    ``True`` if the :mod:`tracemalloc` module is tracing Python memory
    allocations, ``False`` otherwise.

    See also :func:`start` and :func:`stop` functions.
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)tracemalloc.rst`, line 366)

Unknown directive type "function".

```
.. function:: start(nframe: int=1)

   Start tracing Python memory allocations: install hooks on Python memory
   allocators. Collected tracebacks of traces will be limited to *nframe*
   frames. By default, a trace of a memory block only stores the most recent
   frame: the limit is ``1``. *nframe* must be greater or equal to ``1``.

   You can still read the original number of total frames that composed the
   traceback by looking at the :attr:`Traceback.total_nframe` attribute.
```

```
    Storing more than ``1`` frame is only useful to compute statistics grouped
    by ``'traceback'`` or to compute cumulative statistics: see the
    :meth:`Snapshot.compare_to` and :meth:`Snapshot.statistics` methods.

    Storing more frames increases the memory and CPU overhead of the
    :mod:`tracemalloc` module. Use the :func:`get_tracemalloc_memory` function
    to measure how much memory is used by the :mod:`tracemalloc` module.

    The :envvar:`PYTHONTRACEMALLOC` environment variable
    (``PYTHONTRACEMALLOC=NFRAME``) and the :option:`-X` ``tracemalloc=NFRAME``
    command line option can be used to start tracing at startup.

    See also :func:`stop`, :func:`is_tracing` and :func:`get_traceback_limit`
    functions.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)tracemalloc.rst, line 392)

Unknown directive type "function".

```
.. function:: stop()

   Stop tracing Python memory allocations: uninstall hooks on Python memory
   allocators. Also clears all previously collected traces of memory blocks
   allocated by Python.

   Call :func:`take_snapshot` function to take a snapshot of traces before
   clearing them.

   See also :func:`start`, :func:`is_tracing` and :func:`clear_traces`
   functions.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)tracemalloc.rst, line 405)

Unknown directive type "function".

```
.. function:: take_snapshot()

   Take a snapshot of traces of memory blocks allocated by Python. Return a new
   :class:`Snapshot` instance.

   The snapshot does not include memory blocks allocated before the
   :mod:`tracemalloc` module started to trace memory allocations.

   Tracebacks of traces are limited to :func:`get_traceback_limit` frames. Use
   the *nframe* parameter of the :func:`start` function to store more frames.

   The :mod:`tracemalloc` module must be tracing memory allocations to take a
   snapshot, see the :func:`start` function.

   See also the :func:`get_object_traceback` function.
```

DomainFilter

Filter traces of memory blocks by their address space (domain).

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)tracemalloc.rst, line 429)

Unknown directive type "versionadded".

```
.. versionadded:: 3.6
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)tracemalloc.rst, line 431)

Unknown directive type "attribute".

```
.. attribute:: inclusive

   If *inclusive* is ``True`` (include), match memory blocks allocated
   in the address space :attr:`domain`.

   If *inclusive* is ``False`` (exclude), match memory blocks not allocated
   in the address space :attr:`domain`.
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)tracemalloc.rst`, line 439)

Unknown directive type "attribute".

```
.. attribute:: domain

   Address space of a memory block (``int``). Read-only property.
```

Filter

Filter on traces of memory blocks.

See the :func:`fnmatch.fnmatch` function for the syntax of *filename_pattern*. The `'.pyc'` file extension is replaced with `'.py'`.

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)tracemalloc.rst`, line 451); *backlink*

Unknown interpreted text role "func".

Examples:

- `Filter(True, subprocess.__file__)` only includes traces of the :mod:`subprocess` module

  System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)tracemalloc.rst`, line 457); *backlink*

  Unknown interpreted text role "mod".

- `Filter(False, tracemalloc.__file__)` excludes traces of the :mod:`tracemalloc` module

  System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)tracemalloc.rst`, line 459); *backlink*

  Unknown interpreted text role "mod".

- `Filter(False, "<unknown>")` excludes empty tracebacks

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)tracemalloc.rst`, line 464)

Unknown directive type "versionchanged".

```
.. versionchanged:: 3.5
   The ``'.pyo'`` file extension is no longer replaced with ``'.py'``.
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)tracemalloc.rst`, line 467)

Unknown directive type "versionchanged".

```
.. versionchanged:: 3.6
   Added the :attr:`domain` attribute.
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)tracemalloc.rst`, line 471)

Unknown directive type "attribute".

```
.. attribute:: domain

   Address space of a memory block (``int`` or ``None``).

   tracemalloc uses the domain ``0`` to trace memory allocations made by
   Python. C extensions can use other domains to trace other resources.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library) tracemalloc.rst, line 478)

Unknown directive type "attribute".

```
.. attribute:: inclusive

   If *inclusive* is ``True`` (include), only match memory blocks allocated
   in a file with a name matching :attr:`filename_pattern` at line number
   :attr:`lineno`.

   If *inclusive* is ``False`` (exclude), ignore memory blocks allocated in
   a file with a name matching :attr:`filename_pattern` at line number
   :attr:`lineno`.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library) tracemalloc.rst, line 488)

Unknown directive type "attribute".

```
.. attribute:: lineno

   Line number (``int``) of the filter. If *lineno* is ``None``, the filter
   matches any line number.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library) tracemalloc.rst, line 493)

Unknown directive type "attribute".

```
.. attribute:: filename_pattern

   Filename pattern of the filter (``str``). Read-only property.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library) tracemalloc.rst, line 497)

Unknown directive type "attribute".

```
.. attribute:: all_frames

   If *all_frames* is ``True``, all frames of the traceback are checked. If
   *all_frames* is ``False``, only the most recent frame is checked.

   This attribute has no effect if the traceback limit is ``1``.  See the
   :func:`get_traceback_limit` function and :attr:`Snapshot.traceback_limit`
   attribute.
```

Frame

Frame of a traceback.

The :class:`Traceback` class is a sequence of :class:`Frame` instances.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library) tracemalloc.rst, line 514); *backlink*

Unknown interpreted text role "class".

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)tracemalloc.rst, line 514); *backlink*

Unknown interpreted text role "class".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)tracemalloc.rst, line 516)

Unknown directive type "attribute".

```
.. attribute:: filename

   Filename (``str``).
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)tracemalloc.rst, line 520)

Unknown directive type "attribute".

```
.. attribute:: lineno

   Line number (``int``).
```

Snapshot

Snapshot of traces of memory blocks allocated by Python.

The :func:`take_snapshot` function creates a snapshot instance.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)tracemalloc.rst, line 532); *backlink*

Unknown interpreted text role "func".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)tracemalloc.rst, line 534)

Unknown directive type "method".

```
.. method:: compare_to(old_snapshot: Snapshot, key_type: str, cumulative: bool=False)

   Compute the differences with an old snapshot. Get statistics as a sorted
   list of :class:`StatisticDiff` instances grouped by *key_type*.

   See the :meth:`Snapshot.statistics` method for *key_type* and *cumulative*
   parameters.

   The result is sorted from the biggest to the smallest by: absolute value
   of :attr:`StatisticDiff.size_diff`, :attr:`StatisticDiff.size`, absolute
   value of :attr:`StatisticDiff.count_diff`, :attr:`Statistic.count` and
   then by :attr:`StatisticDiff.traceback`.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)tracemalloc.rst, line 548)

Unknown directive type "method".

```
.. method:: dump(filename)

   Write the snapshot into a file.

   Use :meth:`load` to reload the snapshot.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)tracemalloc.rst, line 555)

Unknown directive type "method".

```
.. method:: filter_traces(filters)
```

```
      Create a new :class:`Snapshot` instance with a filtered :attr:`traces`
      sequence, *filters* is a list of :class:`DomainFilter` and
      :class:`Filter` instances.  If *filters* is an empty list, return a new
      :class:`Snapshot` instance with a copy of the traces.

      All inclusive filters are applied at once, a trace is ignored if no
      inclusive filters match it. A trace is ignored if at least one exclusive
      filter matches it.

      .. versionchanged:: 3.6
         :class:`DomainFilter` instances are now also accepted in *filters*.
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)tracemalloc.rst`, line 570)

Unknown directive type "classmethod".

```
   .. classmethod:: load(filename)

      Load a snapshot from a file.

      See also :meth:`dump`.
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)tracemalloc.rst`, line 577)

Unknown directive type "method".

```
   .. method:: statistics(key_type: str, cumulative: bool=False)

      Get statistics as a sorted list of :class:`Statistic` instances grouped
      by *key_type*:

      =====================  ========================
      key_type               description
      =====================  ========================
      ``'filename'``         filename
      ``'lineno'``           filename and line number
      ``'traceback'``        traceback
      =====================  ========================

      If *cumulative* is ``True``, cumulate size and count of memory blocks of
      all frames of the traceback of a trace, not only the most recent frame.
      The cumulative mode can only be used with *key_type* equals to
      ``'filename'`` and ``'lineno'``.

      The result is sorted from the biggest to the smallest by:
      :attr:`Statistic.size`, :attr:`Statistic.count` and then by
      :attr:`Statistic.traceback`.
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)tracemalloc.rst`, line 600)

Unknown directive type "attribute".

```
   .. attribute:: traceback_limit

      Maximum number of frames stored in the traceback of :attr:`traces`:
      result of the :func:`get_traceback_limit` when the snapshot was taken.
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)tracemalloc.rst`, line 605)

Unknown directive type "attribute".

```
   .. attribute:: traces

      Traces of all memory blocks allocated by Python: sequence of
      :class:`Trace` instances.
```

```
    The sequence has an undefined order. Use the :meth:`Snapshot.statistics`
    method to get a sorted list of statistics.
```

Statistic

Statistic on memory allocations.

:func:`Snapshot.statistics` returns a list of :class:`Statistic` instances.

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)tracemalloc.rst`, line 621); *backlink*

Unknown interpreted text role "func".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)tracemalloc.rst`, line 621); *backlink*

Unknown interpreted text role "class".

See also the :class:`StatisticDiff` class.

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)tracemalloc.rst`, line 623); *backlink*

Unknown interpreted text role "class".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)tracemalloc.rst`, line 625)

Unknown directive type "attribute".

```
.. attribute:: count

   Number of memory blocks (``int``).
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)tracemalloc.rst`, line 629)

Unknown directive type "attribute".

```
.. attribute:: size

   Total size of memory blocks in bytes (``int``).
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)tracemalloc.rst`, line 633)

Unknown directive type "attribute".

```
.. attribute:: traceback

   Traceback where the memory block was allocated, :class:`Traceback`
   instance.
```

StatisticDiff

Statistic difference on memory allocations between an old and a new :class:`Snapshot` instance.

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)tracemalloc.rst`, line 644); *backlink*

Unknown interpreted text role "class".

:func:`Snapshot.compare_to` returns a list of :class:`StatisticDiff` instances. See also the :class:`Statistic` class.

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

main\Doc\library\(cpython-main) (Doc) (library)tracemalloc.rst, line 647); *backlink*

Unknown interpreted text role "func".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)tracemalloc.rst, line 647); *backlink*

Unknown interpreted text role "class".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)tracemalloc.rst, line 647); *backlink*

Unknown interpreted text role "class".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)tracemalloc.rst, line 650)

Unknown directive type "attribute".

```
.. attribute:: count

   Number of memory blocks in the new snapshot (``int``): ``0`` if
   the memory blocks have been released in the new snapshot.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)tracemalloc.rst, line 655)

Unknown directive type "attribute".

```
.. attribute:: count_diff

   Difference of number of memory blocks between the old and the new
   snapshots (``int``): ``0`` if the memory blocks have been allocated in
   the new snapshot.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)tracemalloc.rst, line 661)

Unknown directive type "attribute".

```
.. attribute:: size

   Total size of memory blocks in bytes in the new snapshot (``int``):
   ``0`` if the memory blocks have been released in the new snapshot.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)tracemalloc.rst, line 666)

Unknown directive type "attribute".

```
.. attribute:: size_diff

   Difference of total size of memory blocks in bytes between the old and
   the new snapshots (``int``): ``0`` if the memory blocks have been
   allocated in the new snapshot.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)tracemalloc.rst, line 672)

Unknown directive type "attribute".

```
.. attribute:: traceback

   Traceback where the memory blocks were allocated, :class:`Traceback`
   instance.
```

Trace

Trace of a memory block.

The :attr:`Snapshot.traces` attribute is a sequence of :class:`Trace` instances.

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)tracemalloc.rst`, line 685); ***backlink***

Unknown interpreted text role "attr".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)tracemalloc.rst`, line 685); ***backlink***

Unknown interpreted text role "class".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)tracemalloc.rst`, line 688)

Unknown directive type "versionchanged".

```
.. versionchanged:: 3.6
   Added the :attr:`domain` attribute.
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)tracemalloc.rst`, line 691)

Unknown directive type "attribute".

```
.. attribute:: domain

   Address space of a memory block (``int``). Read-only property.

   tracemalloc uses the domain ``0`` to trace memory allocations made by
   Python. C extensions can use other domains to trace other resources.
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)tracemalloc.rst`, line 698)

Unknown directive type "attribute".

```
.. attribute:: size

   Size of the memory block in bytes (``int``).
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)tracemalloc.rst`, line 702)

Unknown directive type "attribute".

```
.. attribute:: traceback

   Traceback where the memory block was allocated, :class:`Traceback`
   instance.
```

Traceback

Sequence of :class:`Frame` instances sorted from the oldest frame to the most recent frame.

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)tracemalloc.rst`, line 713); ***backlink***

Unknown interpreted text role "class".

A traceback contains at least `1` frame. If the `tracemalloc` module failed to get a frame, the filename `"<unknown>"` at line number `0` is used.

When a snapshot is taken, tracebacks of traces are limited to :func:`get_traceback_limit` frames. See the :func:`take_snapshot` function. The original number of frames of the traceback is stored in the :attr:`Traceback.total_nframe` attribute. That allows to know if a traceback has been truncated by the traceback limit.

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)tracemalloc.rst`, line 720); *backlink*

Unknown interpreted text role "func".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)tracemalloc.rst`, line 720); *backlink*

Unknown interpreted text role "func".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)tracemalloc.rst`, line 720); *backlink*

Unknown interpreted text role "attr".

The :attr:`Trace.traceback` attribute is an instance of :class:`Traceback` instance.

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)tracemalloc.rst`, line 726); *backlink*

Unknown interpreted text role "attr".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)tracemalloc.rst`, line 726); *backlink*

Unknown interpreted text role "class".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)tracemalloc.rst`, line 729)

Unknown directive type "versionchanged".

```
.. versionchanged:: 3.7
   Frames are now sorted from the oldest to the most recent, instead of most recent to oldest.
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)tracemalloc.rst`, line 732)

Unknown directive type "attribute".

```
.. attribute:: total_nframe

   Total number of frames that composed the traceback before truncation.
   This attribute can be set to ``None`` if the information is not
   available.
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)tracemalloc.rst`, line 738)

Unknown directive type "versionchanged".

```
.. versionchanged:: 3.9
   The :attr:`Traceback.total_nframe` attribute was added.
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)tracemalloc.rst`, line 741)

Unknown directive type "method".

```
.. method:: format(limit=None, most_recent_first=False)

   Format the traceback as a list of lines. Use the :mod:`linecache` module to
   retrieve lines from the source code. If *limit* is set, format the *limit*
   most recent frames if *limit* is positive. Otherwise, format the
   ``abs(limit)`` oldest frames. If *most_recent_first* is ``True``, the order
   of the formatted frames is reversed, returning the most recent frame first
   instead of last.
```

```
Similar to the :func:`traceback.format_tb` function, except that
:meth:`.format` does not include newlines.

Example::

    print("Traceback (most recent call first):")
    for line in traceback:
        print(line)

Output::

    Traceback (most recent call first):
      File "test.py", line 9
        obj = Object()
      File "test.py", line 12
        tb = tracemalloc.get_object_traceback(f())
```