:mod:`tarfile` --- Read and write tar archive files

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

Unknown directive type "module".

```
.. module:: tarfile
   :synopsis: Read and write tar-format archive files.
```

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

Unknown directive type "moduleauthor".

```
.. moduleauthor:: Lars GustÃ¤bel <lars@gustaebel.de>
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]tarfile.rst, line 8)

Unknown directive type "sectionauthor".

```
.. sectionauthor:: Lars GustÃ¤bel <lars@gustaebel.de>
```

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

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]tarfile.rst, line 10); *backlink*

Unknown interpreted text role "source".

The :mod:`tarfile` module makes it possible to read and write tar archives, including those using gzip, bz2 and lzma compression. Use the :mod:`zipfile` module to read or write :file:`.zip` files, or the higher-level functions in :ref:`shutil <archiving-operations>`.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]tarfile.rst, line 14); *backlink*

Unknown interpreted text role "mod".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]tarfile.rst, line 14); *backlink*

Unknown interpreted text role "mod".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]tarfile.rst, line 14); *backlink*

Unknown interpreted text role "file".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]tarfile.rst, line 14); *backlink*

Unknown interpreted text role "ref".

Some facts and figures:

- reads and writes :mod:`gzip`, :mod:`bz2` and :mod:`lzma` compressed archives if the respective modules are available.

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

Unknown interpreted text role "mod".

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

Unknown interpreted text role "mod".

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

Unknown interpreted text role "mod".

- read/write support for the POSIX.1-1988 (ustar) format.
- read/write support for the GNU tar format including *longname* and *longlink* extensions, read-only support for all variants of the *sparse* extension including restoration of sparse files.
- read/write support for the POSIX.1-2001 (pax) format.
- handles directories, regular files, hardlinks, symbolic links, fifos, character devices and block devices and is able to acquire and restore file information like timestamp, access permissions and owner.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]tarfile.rst, line 36)

Unknown directive type "versionchanged".

```
.. versionchanged:: 3.3
   Added support for :mod:`lzma` compression.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]tarfile.rst, line 40)

Unknown directive type "function".

```
.. function:: open(name=None, mode='r', fileobj=None, bufsize=10240, **kwargs)

   Return a :class:`TarFile` object for the pathname *name*. For detailed
   information on :class:`TarFile` objects and the keyword arguments that are
   allowed, see :ref:`tarfile-objects`.

   *mode* has to be a string of the form ``'filemode[:compression]'``, it defaults
   to ``'r'``. Here is a full list of mode combinations:

   +------------------+---------------------------------------------+
   | mode             | action                                      |
   +==================+=============================================+
   | ``'r' or 'r:*'`` | Open for reading with transparent           |
   |                  | compression (recommended).                  |
   +------------------+---------------------------------------------+
   | ``'r:'``         | Open for reading exclusively without        |
   |                  | compression.                                |
   +------------------+---------------------------------------------+
   | ``'r:gz'``       | Open for reading with gzip compression.     |
   +------------------+---------------------------------------------+
   | ``'r:bz2'``      | Open for reading with bzip2 compression.    |
   +------------------+---------------------------------------------+
   | ``'r:xz'``       | Open for reading with lzma compression.     |
   +------------------+---------------------------------------------+
   | ``'x'`` or       | Create a tarfile exclusively without        |
   | ``'x:'``         | compression.                                |
   |                  | Raise a :exc:`FileExistsError` exception    |
   |                  | if it already exists.                       |
   +------------------+---------------------------------------------+
   | ``'x:gz'``       | Create a tarfile with gzip compression.     |
   |                  | Raise a :exc:`FileExistsError` exception    |
   |                  | if it already exists.                       |
   +------------------+---------------------------------------------+
   | ``'x:bz2'``      | Create a tarfile with bzip2 compression.    |
   |                  | Raise a :exc:`FileExistsError` exception    |
   |                  | if it already exists.                       |
   +------------------+---------------------------------------------+
   | ``'x:xz'``       | Create a tarfile with lzma compression.     |
   |                  | Raise a :exc:`FileExistsError` exception    |
   |                  | if it already exists.                       |
   +------------------+---------------------------------------------+
   | ``'a' or 'a:'``  | Open for appending with no compression. The |
```

```
|                  | file is created if it does not exist.       |
+------------------+---------------------------------------------+
| ``'w' or 'w:'``  | Open for uncompressed writing.              |
+------------------+---------------------------------------------+
| ``'w:gz'``       | Open for gzip compressed writing.           |
+------------------+---------------------------------------------+
| ``'w:bz2'``      | Open for bzip2 compressed writing.          |
+------------------+---------------------------------------------+
| ``'w:xz'``       | Open for lzma compressed writing.           |
+------------------+---------------------------------------------+

Note that ``'a:gz'``, ``'a:bz2'`` or ``'a:xz'`` is not possible. If *mode*
is not suitable to open a certain (compressed) file for reading,
:exc:`ReadError` is raised. Use *mode* ``'r'`` to avoid this.  If a
compression method is not supported, :exc:`CompressionError` is raised.

If *fileobj* is specified, it is used as an alternative to a :term:`file object`
opened in binary mode for *name*. It is supposed to be at position 0.

For modes ``'w:gz'``, ``'r:gz'``, ``'w:bz2'``, ``'r:bz2'``, ``'x:gz'``,
``'x:bz2'``, :func:`tarfile.open` accepts the keyword argument
*compresslevel* (default ``9``) to specify the compression level of the file.

For modes ``'w:xz'`` and ``'x:xz'``, :func:`tarfile.open` accepts the
keyword argument *preset* to specify the compression level of the file.

For special purposes, there is a second format for *mode*:
``'filemode|[compression]'``.  :func:`tarfile.open` will return a :class:`TarFile`
object that processes its data as a stream of blocks.  No random seeking will
be done on the file. If given, *fileobj* may be any object that has a
:meth:`read` or :meth:`write` method (depending on the *mode*). *bufsize*
specifies the blocksize and defaults to ``20 * 512`` bytes. Use this variant
in combination with e.g. ``sys.stdin``, a socket :term:`file object` or a tape
device. However, such a :class:`TarFile` object is limited in that it does
not allow random access, see :ref:`tar-examples`.  The currently
possible modes:

+-------------+---------------------------------------------+
| Mode        | Action                                      |
+=============+=============================================+
| ``'r|*'``   | Open a *stream* of tar blocks for reading   |
|             | with transparent compression.               |
+-------------+---------------------------------------------+
| ``'r|'``    | Open a *stream* of uncompressed tar blocks  |
|             | for reading.                                |
+-------------+---------------------------------------------+
| ``'r|gz'``  | Open a gzip compressed *stream* for         |
|             | reading.                                    |
+-------------+---------------------------------------------+
| ``'r|bz2'`` | Open a bzip2 compressed *stream* for        |
|             | reading.                                    |
+-------------+---------------------------------------------+
| ``'r|xz'``  | Open an lzma compressed *stream* for        |
|             | reading.                                    |
+-------------+---------------------------------------------+
| ``'w|'``    | Open an uncompressed *stream* for writing.  |
+-------------+---------------------------------------------+
| ``'w|gz'``  | Open a gzip compressed *stream* for         |
|             | writing.                                    |
+-------------+---------------------------------------------+
| ``'w|bz2'`` | Open a bzip2 compressed *stream* for        |
|             | writing.                                    |
+-------------+---------------------------------------------+
| ``'w|xz'``  | Open an lzma compressed *stream* for        |
|             | writing.                                    |
+-------------+---------------------------------------------+

.. versionchanged:: 3.5
   The ``'x'`` (exclusive creation) mode was added.

.. versionchanged:: 3.6
   The *name* parameter accepts a :term:`path-like object`.
```

Class for reading and writing tar archives. Do not use this class directly: use :func:`tarfile.open` instead. See :ref:`tarfile-objects`.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]tarfile.rst, line 159); *backlink*

Unknown interpreted text role "func".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]tarfile.rst, line 159); *backlink*

Unknown interpreted text role "ref".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]tarfile.rst, line 163)

Unknown directive type "function".

```
.. function:: is_tarfile(name)

   Return :const:`True` if *name* is a tar archive file, that the :mod:`tarfile`
   module can read. *name* may be a :class:`str`, file, or file-like object.

   .. versionchanged:: 3.9
      Support for file and file-like objects.
```

The :mod:`tarfile` module defines the following exceptions:

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]tarfile.rst, line 172); *backlink*

Unknown interpreted text role "mod".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]tarfile.rst, line 175)

Unknown directive type "exception".

```
.. exception:: TarError

   Base class for all :mod:`tarfile` exceptions.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]tarfile.rst, line 180)

Unknown directive type "exception".

```
.. exception:: ReadError

   Is raised when a tar archive is opened, that either cannot be handled by the
   :mod:`tarfile` module or is somehow invalid.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]tarfile.rst, line 186)

Unknown directive type "exception".

```
.. exception:: CompressionError

   Is raised when a compression method is not supported or when the data cannot be
   decoded properly.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]tarfile.rst, line 192)

Unknown directive type "exception".

```
.. exception:: StreamError

   Is raised for the limitations that are typical for stream-like :class:`TarFile`
   objects.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]tarfile.rst, line 198)

Unknown directive type "exception".

```
.. exception:: ExtractError

   Is raised for *non-fatal* errors when using :meth:`TarFile.extract`, but only if
```

```
   :attr:`TarFile.errorlevel`\ ``== 2``.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]tarfile.rst, line 204)

Unknown directive type "exception".

```
.. exception:: HeaderError

   Is raised by :meth:`TarInfo.frombuf` if the buffer it gets is invalid.
```

The following constants are available at the module level:

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]tarfile.rst, line 211)

Unknown directive type "data".

```
.. data:: ENCODING

   The default character encoding: ``'utf-8'`` on Windows, the value returned by
   :func:`sys.getfilesystemencoding` otherwise.
```

Each of the following constants defines a tar archive format that the :mod:`tarfile` module is able to create. See section :ref:`tar-formats` for details.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]tarfile.rst, line 217); *backlink*

Unknown interpreted text role "mod".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]tarfile.rst, line 217); *backlink*

Unknown interpreted text role "ref".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]tarfile.rst, line 222)

Unknown directive type "data".

```
.. data:: USTAR_FORMAT

   POSIX.1-1988 (ustar) format.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]tarfile.rst, line 227)

Unknown directive type "data".

```
.. data:: GNU_FORMAT

   GNU tar format.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]tarfile.rst, line 232)

Unknown directive type "data".

```
.. data:: PAX_FORMAT

   POSIX.1-2001 (pax) format.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]tarfile.rst, line 237)

Unknown directive type "data".

```
.. data:: DEFAULT_FORMAT

   The default format for creating archives. This is currently :const:`PAX_FORMAT`.

   .. versionchanged:: 3.8
      The default format for new archives was changed to
      :const:`PAX_FORMAT` from :const:`GNU_FORMAT`.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]tarfile.rst, line 246)

Unknown directive type "seealso".

```
.. seealso::

   Module :mod:`zipfile`
      Documentation of the :mod:`zipfile` standard module.

   :ref:`archiving-operations`
      Documentation of the higher-level archiving facilities provided by the
      standard :mod:`shutil` module.

   `GNU tar manual, Basic Tar Format <https://www.gnu.org/software/tar/manual/html_node/Standard.html`
      Documentation for tar archive files, including GNU tar extensions.
```

TarFile Objects

The :class:`TarFile` object provides an interface to a tar archive. A tar archive is a sequence of blocks. An archive member (a stored file) is made up of a header block followed by data blocks. It is possible to store a file in a tar archive several times. Each archive member is represented by a :class:`TarInfo` object, see :ref:`tarinfo-objects` for details.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]tarfile.rst, line 264); *backlink*

Unknown interpreted text role "class".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]tarfile.rst, line 264); *backlink*

Unknown interpreted text role "class".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]tarfile.rst, line 264); *backlink*

Unknown interpreted text role "ref".

A :class:`TarFile` object can be used as a context manager in a :keyword:`with` statement. It will automatically be closed when the block is completed. Please note that in the event of an exception an archive opened for writing will not be finalized; only the internally used file object will be closed. See the :ref:`tar-examples` section for a use case.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]tarfile.rst, line 270); *backlink*

Unknown interpreted text role "class".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]tarfile.rst, line 270); *backlink*

Unknown interpreted text role "keyword".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]tarfile.rst, line 270); *backlink*

Unknown interpreted text role "ref".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]tarfile.rst, line 276)

Unknown directive type "versionadded".

```
.. versionadded:: 3.2
   Added support for the context management protocol.
```

All following arguments are optional and can be accessed as instance attributes as well.

name is the pathname of the archive. *name* may be a :term:`path-like object`. It can be omitted if *fileobj* is given. In this case, the file object's :attr:`name` attribute is used if it exists.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]tarfile.rst, line 284); *backlink*

Unknown interpreted text role "term".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]tarfile.rst, line 284); *backlink*

Unknown interpreted text role "attr".

mode is either `'r'` to read from an existing archive, `'a'` to append data to an existing file, `'w'` to create a new file overwriting an existing one, or `'x'` to create a new file only if it does not already exist.

If *fileobj* is given, it is used for reading or writing data. If it can be determined, *mode* is overridden by *fileobj*'s mode. *fileobj* will be used from position 0.

Note

fileobj is not closed, when :class:`TarFile` is closed.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]tarfile.rst, line 298); *backlink*

Unknown interpreted text role "class".

format controls the archive format for writing. It must be one of the constants :const:`USTAR_FORMAT`, :const:`GNU_FORMAT` or :const:`PAX_FORMAT` that are defined at module level. When reading, format will be automatically detected, even if different formats are present in a single archive.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]tarfile.rst, line 300); *backlink*

Unknown interpreted text role "const".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]tarfile.rst, line 300); *backlink*

Unknown interpreted text role "const".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]tarfile.rst, line 300); *backlink*

Unknown interpreted text role "const".

The *tarinfo* argument can be used to replace the default :class:`TarInfo` class with a different one.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]tarfile.rst, line 305); *backlink*

Unknown interpreted text role "class".

If *dereference* is :const:`False`, add symbolic and hard links to the archive. If it is :const:`True`, add the content of the target files to the archive. This has no effect on systems that do not support symbolic links.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]tarfile.rst, line 308); *backlink*

Unknown interpreted text role "const".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]tarfile.rst, line 308); *backlink*

Unknown interpreted text role "const".

If *ignore_zeros* is :const:`False`, treat an empty block as the end of the archive. If it is :const:`True`, skip empty (and invalid) blocks and try to get as many members as possible. This is only useful for reading concatenated or damaged archives.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]tarfile.rst, line 312); *backlink*

Unknown interpreted text role "const".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]tarfile.rst, line 312); *backlink*

Unknown interpreted text role "const".

debug can be set from `0` (no debug messages) up to `3` (all debug messages). The messages are written to `sys.stderr`.

If *errorlevel* is `0`, all errors are ignored when using :meth:`TarFile.extract`. Nevertheless, they appear as error messages in the debug output, when debugging is enabled. If `1`, all *fatal* errors are raised as :exc:`OSError` exceptions. If `2`, all *non-fatal* errors are raised as :exc:`TarError` exceptions as well.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]tarfile.rst, line 319); *backlink*

Unknown interpreted text role "meth".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]tarfile.rst, line 319); *backlink*

Unknown interpreted text role "exc".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]tarfile.rst, line 319); *backlink*

Unknown interpreted text role "exc".

The *encoding* and *errors* arguments define the character encoding to be used for reading or writing the archive and how conversion errors are going to be handled. The default settings will work for most users. See section :ref:`tar-unicode` for in-depth information.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]tarfile.rst, line 325); *backlink*

Unknown interpreted text role "ref".

The *pax_headers* argument is an optional dictionary of strings which will be added as a pax global header if *format* is :const:`PAX_FORMAT`.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]tarfile.rst, line 330); *backlink*

Unknown interpreted text role "const".

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

Unknown directive type "versionchanged".

```
.. versionchanged:: 3.2
   Use ``'surrogateescape'`` as the default for the *errors* argument.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]tarfile.rst, line 336)

Unknown directive type "versionchanged".

```
.. versionchanged:: 3.5
   The ``'x'`` (exclusive creation) mode was added.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]tarfile.rst, line 339)

Unknown directive type "versionchanged".

```
.. versionchanged:: 3.6
   The *name* parameter accepts a :term:`path-like object`.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]tarfile.rst, line 343)

Unknown directive type "classmethod".

```
.. classmethod:: TarFile.open(...)

   Alternative constructor. The :func:`tarfile.open` function is actually a
   shortcut to this classmethod.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]tarfile.rst, line 349)

Unknown directive type "method".

```
.. method:: TarFile.getmember(name)

   Return a :class:`TarInfo` object for member *name*. If *name* can not be found
   in the archive, :exc:`KeyError` is raised.

   .. note::

      If a member occurs more than once in the archive, its last occurrence is assumed
      to be the most up-to-date version.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]tarfile.rst, line 360)

Unknown directive type "method".

```
.. method:: TarFile.getmembers()

   Return the members of the archive as a list of :class:`TarInfo` objects. The
   list has the same order as the members in the archive.
```

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

Unknown directive type "method".

```
.. method:: TarFile.getnames()

   Return the members as a list of their names. It has the same order as the list
   returned by :meth:`getmembers`.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]tarfile.rst, line 372)

Unknown directive type "method".

```
.. method:: TarFile.list(verbose=True, *, members=None)

   Print a table of contents to ``sys.stdout``. If *verbose* is :const:`False`,
   only the names of the members are printed. If it is :const:`True`, output
   similar to that of :program:`ls -l` is produced. If optional *members* is
   given, it must be a subset of the list returned by :meth:`getmembers`.

   .. versionchanged:: 3.5
      Added the *members* parameter.
```

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

main\Doc\library\[cpython-main][Doc][library]tarfile.rst, line 383)

Unknown directive type "method".

```
.. method:: TarFile.next()

   Return the next member of the archive as a :class:`TarInfo` object, when
   :class:`TarFile` is opened for reading. Return :const:`None` if there is no more
   available.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]tarfile.rst, line 390)

Unknown directive type "method".

```
.. method:: TarFile.extractall(path=".", members=None, *, numeric_owner=False)

   Extract all members from the archive to the current working directory or
   directory *path*. If optional *members* is given, it must be a subset of the
   list returned by :meth:`getmembers`. Directory information like owner,
   modification time and permissions are set after all members have been extracted.
   This is done to work around two problems: A directory's modification time is
   reset each time a file is created in it. And, if a directory's permissions do
   not allow writing, extracting files to it will fail.

   If *numeric_owner* is :const:`True`, the uid and gid numbers from the tarfile
   are used to set the owner/group for the extracted files. Otherwise, the named
   values from the tarfile are used.

   .. warning::

      Never extract archives from untrusted sources without prior inspection.
      It is possible that files are created outside of *path*, e.g. members
      that have absolute filenames starting with ``"/"`` or filenames with two
      dots ``".."``.

   .. versionchanged:: 3.5
      Added the *numeric_owner* parameter.

   .. versionchanged:: 3.6
      The *path* parameter accepts a :term:`path-like object`.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]tarfile.rst, line 418)

Unknown directive type "method".

```
.. method:: TarFile.extract(member, path="", set_attrs=True, *, numeric_owner=False)

   Extract a member from the archive to the current working directory, using its
   full name. Its file information is extracted as accurately as possible. *member*
   may be a filename or a :class:`TarInfo` object. You can specify a different
   directory using *path*. *path* may be a :term:`path-like object`.
   File attributes (owner, mtime, mode) are set unless *set_attrs* is false.

   If *numeric_owner* is :const:`True`, the uid and gid numbers from the tarfile
   are used to set the owner/group for the extracted files. Otherwise, the named
   values from the tarfile are used.

   .. note::

      The :meth:`extract` method does not take care of several extraction issues.
      In most cases you should consider using the :meth:`extractall` method.

   .. warning::

      See the warning for :meth:`extractall`.

   .. versionchanged:: 3.2
      Added the *set_attrs* parameter.

   .. versionchanged:: 3.5
      Added the *numeric_owner* parameter.

   .. versionchanged:: 3.6
      The *path* parameter accepts a :term:`path-like object`.
```

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

main\Doc\library\[cpython-main][Doc][library]tarfile.rst, line 449)

Unknown directive type "method".

```
.. method:: TarFile.extractfile(member)

   Extract a member from the archive as a file object. *member* may be
   a filename or a :class:`TarInfo` object. If *member* is a regular file or
   a link, an :class:`io.BufferedReader` object is returned. For all other
   existing members, :const:`None` is returned. If *member* does not appear
   in the archive, :exc:`KeyError` is raised.

   .. versionchanged:: 3.3
      Return an :class:`io.BufferedReader` object.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]tarfile.rst, line 461)

Unknown directive type "method".

```
.. method:: TarFile.add(name, arcname=None, recursive=True, *, filter=None)

   Add the file *name* to the archive. *name* may be any type of file
   (directory, fifo, symbolic link, etc.). If given, *arcname* specifies an
   alternative name for the file in the archive. Directories are added
   recursively by default. This can be avoided by setting *recursive* to
   :const:`False`. Recursion adds entries in sorted order.
   If *filter* is given, it
   should be a function that takes a :class:`TarInfo` object argument and
   returns the changed :class:`TarInfo` object. If it instead returns
   :const:`None` the :class:`TarInfo` object will be excluded from the
   archive. See :ref:`tar-examples` for an example.

   .. versionchanged:: 3.2
      Added the *filter* parameter.

   .. versionchanged:: 3.7
      Recursion adds entries in sorted order.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]tarfile.rst, line 481)

Unknown directive type "method".

```
.. method:: TarFile.addfile(tarinfo, fileobj=None)

   Add the :class:`TarInfo` object *tarinfo* to the archive. If *fileobj* is given,
   it should be a :term:`binary file`, and
   ``tarinfo.size`` bytes are read from it and added to the archive.  You can
   create :class:`TarInfo` objects directly, or by using :meth:`gettarinfo`.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]tarfile.rst, line 489)

Unknown directive type "method".

```
.. method:: TarFile.gettarinfo(name=None, arcname=None, fileobj=None)

   Create a :class:`TarInfo` object from the result of :func:`os.stat` or
   equivalent on an existing file.  The file is either named by *name*, or
   specified as a :term:`file object` *fileobj* with a file descriptor.
   *name* may be a :term:`path-like object`.  If
   given, *arcname* specifies an alternative name for the file in the
   archive, otherwise, the name is taken from *fileobj*â€™s
   :attr:`~io.FileIO.name` attribute, or the *name* argument.  The name
   should be a text string.

   You can modify
   some of the :class:`TarInfo`â€™s attributes before you add it using :meth:`addfile`.
   If the file object is not an ordinary file object positioned at the
   beginning of the file, attributes such as :attr:`~TarInfo.size` may need
   modifying.  This is the case for objects such as :class:`~gzip.GzipFile`.
   The :attr:`~TarInfo.name` may also be modified, in which case *arcname*
   could be a dummy string.

   .. versionchanged:: 3.6
      The *name* parameter accepts a :term:`path-like object`.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]tarfile.rst, line 512)

Unknown directive type "method".

```
.. method:: TarFile.close()

   Close the :class:`TarFile`. In write mode, two finishing zero blocks are
   appended to the archive.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]tarfile.rst, line 518)

Unknown directive type "attribute".

```
.. attribute:: TarFile.pax_headers

   A dictionary containing key-value pairs of pax global headers.
```

TarInfo Objects

A :class:`TarInfo` object represents one member in a :class:`TarFile`. Aside from storing all required attributes of a file (like file type, size, time, permissions, owner etc.), it provides some useful methods to determine its type. It does *not* contain the file's data itself.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]tarfile.rst, line 529); *backlink*

Unknown interpreted text role "class".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]tarfile.rst, line 529); *backlink*

Unknown interpreted text role "class".

:class:`TarInfo` objects are returned by :class:`TarFile`'s methods :meth:`getmember`, :meth:`getmembers` and :meth:`gettarinfo`.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]tarfile.rst, line 534); *backlink*

Unknown interpreted text role "class".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]tarfile.rst, line 534); *backlink*

Unknown interpreted text role "class".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]tarfile.rst, line 534); *backlink*

Unknown interpreted text role "meth".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]tarfile.rst, line 534); *backlink*

Unknown interpreted text role "meth".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]tarfile.rst, line 534); *backlink*

Unknown interpreted text role "meth".

Create a :class:`TarInfo` object.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]tarfile.rst, line 540); *backlink*

Unknown interpreted text role "class".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]tarfile.rst, line 543)

Unknown directive type "classmethod".

```
.. classmethod:: TarInfo.frombuf(buf, encoding, errors)

   Create and return a :class:`TarInfo` object from string buffer *buf*.

   Raises :exc:`HeaderError` if the buffer is invalid.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]tarfile.rst, line 550)

Unknown directive type "classmethod".

```
.. classmethod:: TarInfo.fromtarfile(tarfile)

   Read the next member from the :class:`TarFile` object *tarfile* and return it as
   a :class:`TarInfo` object.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]tarfile.rst, line 556)

Unknown directive type "method".

```
.. method:: TarInfo.tobuf(format=DEFAULT_FORMAT, encoding=ENCODING, errors='surrogateescape')

   Create a string buffer from a :class:`TarInfo` object. For information on the
   arguments see the constructor of the :class:`TarFile` class.

   .. versionchanged:: 3.2
      Use ``'surrogateescape'`` as the default for the *errors* argument.
```

A `TarInfo` object has the following public data attributes:

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]tarfile.rst, line 568)

Unknown directive type "attribute".

```
.. attribute:: TarInfo.name

   Name of the archive member.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]tarfile.rst, line 573)

Unknown directive type "attribute".

```
.. attribute:: TarInfo.size

   Size in bytes.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]tarfile.rst, line 578)

Unknown directive type "attribute".

```
.. attribute:: TarInfo.mtime

   Time of last modification.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]tarfile.rst, line 583)

Unknown directive type "attribute".

```
.. attribute:: TarInfo.mode

   Permission bits.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]tarfile.rst, line 588)

Unknown directive type "attribute".

```
.. attribute:: TarInfo.type

   File type.  *type* is usually one of these constants: :const:`REGTYPE`,
   :const:`AREGTYPE`, :const:`LNKTYPE`, :const:`SYMTYPE`, :const:`DIRTYPE`,
   :const:`FIFOTYPE`, :const:`CONTTYPE`, :const:`CHRTYPE`, :const:`BLKTYPE`,
   :const:`GNUTYPE_SPARSE`.  To determine the type of a :class:`TarInfo` object
   more conveniently, use the ``is*()`` methods below.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]tarfile.rst, line 597)

Unknown directive type "attribute".

```
.. attribute:: TarInfo.linkname

   Name of the target file name, which is only present in :class:`TarInfo` objects
   of type :const:`LNKTYPE` and :const:`SYMTYPE`.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]tarfile.rst, line 603)

Unknown directive type "attribute".

```
.. attribute:: TarInfo.uid

   User ID of the user who originally stored this member.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]tarfile.rst, line 608)

Unknown directive type "attribute".

```
.. attribute:: TarInfo.gid

   Group ID of the user who originally stored this member.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]tarfile.rst, line 613)

Unknown directive type "attribute".

```
.. attribute:: TarInfo.uname

   User name.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]tarfile.rst, line 618)

Unknown directive type "attribute".

```
.. attribute:: TarInfo.gname

   Group name.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]tarfile.rst, line 623)

Unknown directive type "attribute".

```
.. attribute:: TarInfo.pax_headers

   A dictionary containing key-value pairs of an associated pax extended header.
```

A :class:`TarInfo` object also provides some convenient query methods:

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]tarfile.rst, line 628); *backlink*

Unknown interpreted text role "class".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]tarfile.rst, line 631)

Unknown directive type "method".

```
.. method:: TarInfo.isfile()

   Return :const:`True` if the :class:`Tarinfo` object is a regular file.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]tarfile.rst, line 636)

Unknown directive type "method".

```
.. method:: TarInfo.isreg()

   Same as :meth:`isfile`.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]tarfile.rst, line 641)

Unknown directive type "method".

```
.. method:: TarInfo.isdir()

   Return :const:`True` if it is a directory.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]tarfile.rst, line 646)

Unknown directive type "method".

```
.. method:: TarInfo.issym()

   Return :const:`True` if it is a symbolic link.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]tarfile.rst, line 651)

Unknown directive type "method".

```
.. method:: TarInfo.islnk()

   Return :const:`True` if it is a hard link.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]tarfile.rst, line 656)

Unknown directive type "method".

```
.. method:: TarInfo.ischr()

   Return :const:`True` if it is a character device.
```

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

Unknown directive type "method".

```
.. method:: TarInfo.isblk()

   Return :const:`True` if it is a block device.
```

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

Unknown directive type "method".

```
.. method:: TarInfo.isfifo()

   Return :const:`True` if it is a FIFO.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]tarfile.rst, line 671)

Unknown directive type "method".

```
.. method:: TarInfo.isdev()

   Return :const:`True` if it is one of character device, block device or FIFO.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]tarfile.rst, line 677)

Unknown directive type "program".

```
.. program:: tarfile
```

Command-Line Interface

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]tarfile.rst, line 682)

Unknown directive type "versionadded".

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

The :mod:`tarfile` module provides a simple command-line interface to interact with tar archives.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]tarfile.rst, line 684); *backlink*

Unknown interpreted text role "mod".

If you want to create a new tar archive, specify its name after the :option:`-c` option and then list the filename(s) that should be included:

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]tarfile.rst, line 687); *backlink*

Unknown interpreted text role "option".

```
$ python -m tarfile -c monty.tar  spam.txt eggs.txt
```

Passing a directory is also acceptable:

```
$ python -m tarfile -c monty.tar life-of-brian_1979/
```

If you want to extract a tar archive into the current directory, use the :option:`-e` option:

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]tarfile.rst, line 700); *backlink*

Unknown interpreted text role "option".

```
$ python -m tarfile -e monty.tar
```

You can also extract a tar archive into a different directory by passing the directory's name:

```
$ python -m tarfile -e monty.tar  other-dir/
```

For a list of the files in a tar archive, use the :option:`-l` option:

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]tarfile.rst, line 714); *backlink*

Unknown interpreted text role "option".

```
$ python -m tarfile -l monty.tar
```

Command-line options

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]tarfile.rst, line 724)

Unknown directive type "cmdoption".

```
.. cmdoption:: -l <tarfile>
               --list <tarfile>

   List files in a tarfile.
```

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

Unknown directive type "cmdoption".

```
.. cmdoption:: -c <tarfile> <source1> ... <sourceN>
               --create <tarfile> <source1> ... <sourceN>

   Create tarfile from source files.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]tarfile.rst, line 734)

Unknown directive type "cmdoption".

```
.. cmdoption:: -e <tarfile> [<output_dir>]
               --extract <tarfile> [<output_dir>]

   Extract tarfile into the current directory if *output_dir* is not specified.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]tarfile.rst, line 739)

Unknown directive type "cmdoption".

```
.. cmdoption:: -t <tarfile>
               --test <tarfile>

   Test whether the tarfile is valid or not.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]tarfile.rst, line 744)

Unknown directive type "cmdoption".

```
.. cmdoption:: -v, --verbose

   Verbose output.
```

Examples

How to extract an entire tar archive to the current working directory:

```
import tarfile
tar = tarfile.open("sample.tar.gz")
tar.extractall()
tar.close()
```

How to extract a subset of a tar archive with :meth:`TarFile.extractall` using a generator function instead of a list:

> **System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]tarfile.rst, line 760); *backlink***
>
> Unknown interpreted text role "meth".

```
import os
import tarfile

def py_files(members):
    for tarinfo in members:
        if os.path.splitext(tarinfo.name)[1] == ".py":
            yield tarinfo

tar = tarfile.open("sample.tar.gz")
tar.extractall(members=py_files(tar))
tar.close()
```

How to create an uncompressed tar archive from a list of filenames:

```
import tarfile
tar = tarfile.open("sample.tar", "w")
for name in ["foo", "bar", "quux"]:
    tar.add(name)
tar.close()
```

The same example using the :keyword:`with` statement:

> **System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]tarfile.rst, line 783); *backlink***
>
> Unknown interpreted text role "keyword".

```
import tarfile
with tarfile.open("sample.tar", "w") as tar:
    for name in ["foo", "bar", "quux"]:
        tar.add(name)
```

How to read a gzip compressed tar archive and display some member information:

```
import tarfile
tar = tarfile.open("sample.tar.gz", "r:gz")
for tarinfo in tar:
    print(tarinfo.name, "is", tarinfo.size, "bytes in size and is ", end="")
    if tarinfo.isreg():
        print("a regular file.")
    elif tarinfo.isdir():
        print("a directory.")
    else:
        print("something else.")
tar.close()
```

How to create an archive and reset the user information using the *filter* parameter in :meth:`TarFile.add`:

> **System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]tarfile.rst, line 804); *backlink***
>
> Unknown interpreted text role "meth".

```
import tarfile
def reset(tarinfo):
    tarinfo.uid = tarinfo.gid = 0
    tarinfo.uname = tarinfo.gname = "root"
    return tarinfo
tar = tarfile.open("sample.tar.gz", "w:gz")
tar.add("foo", filter=reset)
tar.close()
```

Supported tar formats

There are three tar formats that can be created with the :mod:`tarfile` module:

> **System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]tarfile.rst, line 822); *backlink*
>
> Unknown interpreted text role "mod".

- The POSIX.1-1988 ustar format (:const:`USTAR_FORMAT`). It supports filenames up to a length of at best 256 characters and linknames up to 100 characters. The maximum file size is 8 GiB. This is an old and limited but widely supported format.

  > **System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]tarfile.rst, line 824); *backlink*
  >
  > Unknown interpreted text role "const".

- The GNU tar format (:const:`GNU_FORMAT`). It supports long filenames and linknames, files bigger than 8 GiB and sparse files. It is the de facto standard on GNU/Linux systems. :mod:`tarfile` fully supports the GNU tar extensions for long names, sparse file support is read-only.

  > **System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]tarfile.rst, line 829); *backlink*
  >
  > Unknown interpreted text role "const".

  > **System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]tarfile.rst, line 829); *backlink*
  >
  > Unknown interpreted text role "mod".

- The POSIX.1-2001 pax format (:const:`PAX_FORMAT`). It is the most flexible format with virtually no limits. It supports long filenames and linknames, large files and stores pathnames in a portable way. Modern tar implementations, including GNU tar, bsdtar/libarchive and star, fully support extended *pax* features; some old or unmaintained libraries may not, but should treat *pax* archives as if they were in the universally-supported *ustar* format. It is the current default format for new archives.

  > **System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]tarfile.rst, line 834); *backlink*
  >
  > Unknown interpreted text role "const".

  It extends the existing *ustar* format with extra headers for information that cannot be stored otherwise. There are two flavours of pax headers: Extended headers only affect the subsequent file header, global headers are valid for the complete archive and affect all following files. All the data in a pax header is encoded in *UTF-8* for portability reasons.

There are some more variants of the tar format which can be read, but not created:

- The ancient V7 format. This is the first tar format from Unix Seventh Edition, storing only regular files and directories. Names must not be longer than 100 characters, there is no user/group name information. Some archives have miscalculated header checksums in case of fields with non-ASCII characters.
- The SunOS tar extended format. This format is a variant of the POSIX.1-2001 pax format, but is not compatible.

Unicode issues

The tar format was originally conceived to make backups on tape drives with the main focus on preserving file system information. Nowadays tar archives are commonly used for file distribution and exchanging archives over networks. One problem of the original format (which is the basis of all other formats) is that there is no concept of supporting different character encodings. For example, an ordinary tar archive created on a *UTF-8* system cannot be read correctly on a *Latin-1* system if it contains non-*ASCII* characters. Textual metadata (like filenames, linknames, user/group names) will appear damaged. Unfortunately, there is no way to autodetect the encoding of an archive. The pax format was designed to solve this problem. It stores non-ASCII metadata using the universal character encoding *UTF-8*.

The details of character conversion in :mod:`tarfile` are controlled by the *encoding* and *errors* keyword arguments of the :class:`TarFile` class.

> **System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]tarfile.rst, line 876); *backlink*

Unknown interpreted text role "mod".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]tarfile.rst, line 876); *backlink*

Unknown interpreted text role "class".

encoding defines the character encoding to use for the metadata in the archive. The default value is :func:`sys.getfilesystemencoding` or `'ascii'` as a fallback. Depending on whether the archive is read or written, the metadata must be either decoded or encoded. If *encoding* is not set appropriately, this conversion may fail.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]tarfile.rst, line 879); *backlink*

Unknown interpreted text role "func".

The *errors* argument defines how characters are treated that cannot be converted. Possible values are listed in section :ref:`error-handlers`. The default scheme is `'surrogateescape'` which Python also uses for its file system calls, see :ref:`os-filenames`.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]tarfile.rst, line 885); *backlink*

Unknown interpreted text role "ref".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]tarfile.rst, line 885); *backlink*

Unknown interpreted text role "ref".

For :const:`PAX_FORMAT` archives (the default), *encoding* is generally not needed because all the metadata is stored using *UTF-8*. *encoding* is only used in the rare cases when binary pax headers are decoded or when strings with surrogate characters are stored.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]tarfile.rst, line 890); *backlink*

Unknown interpreted text role "const".