

## Navigation

- [index](#)
- [modules](#) |
- [next](#) |
- [previous](#) |
- [Python v2.6.4 documentation](#) »
- [The Python Standard Library](#) »
- [13. Data Compression and Archiving](#) »

## 13.2. gzip — Support for gzip files¶

This module provides a simple interface to compress and decompress files just like the GNU programs **gzip** and **gunzip** would.

The data compression is provided by the [zlib](#) module.

The `gzip` module provides the [GzipFile](#) class which is modeled after Python's File Object. The [GzipFile](#) class reads and writes **gzip**-format files, automatically compressing or decompressing the data so that it looks like an ordinary file object.

Note that additional file formats which can be decompressed by the **gzip** and **gunzip** programs, such as those produced by **compress** and **pack**, are not supported by this module.

For other archive formats, see the [bz2](#), [zipfile](#), and [tarfile](#) modules.

The module defines the following items:

```
class gzip.GzipFile([filename[, mode[, compresslevel[, fileobj]]]])¶
```

Constructor for the [GzipFile](#) class, which simulates most of the methods of a file object, with the exception of the `readinto()` and `truncate()` methods. At least one of `fileobj` and `filename` must be given a non-trivial value.

The new class instance is based on `fileobj`, which can be a regular file, a `StringIO` object, or any other object which simulates a file. It defaults to `None`, in which case `filename` is opened to provide a file object.

When `fileobj` is not `None`, the `filename` argument is only used to be included in the **gzip** file header, which may include the original filename of the uncompressed file. It defaults to the filename of `fileobj`, if discernible; otherwise, it defaults to the empty string, and in this case the original filename is not included in the header.

The `mode` argument can be any of `'r'`, `'rb'`, `'a'`, `'ab'`, `'w'`, or `'wb'`, depending on whether the file will be read or written. The default is the mode of `fileobj` if discernible; otherwise, the default is `'rb'`. If not given, the `'b'` flag will be added to the mode to ensure the file is opened in binary mode for cross-platform portability.

The `compresslevel` argument is an integer from 1 to 9 controlling the level of compression; 1 is fastest and produces the least compression, and 9 is slowest and produces the most compression. The default is 9.

Calling a [GzipFile](#) object's `close()` method does not close `fileobj`, since you might wish to append more material after the compressed data. This also allows you to pass a `StringIO` object opened for writing as `fileobj`, and retrieve the resulting memory buffer using the `StringIO` object's `getvalue()` method.

```
gzip.open(filename[, mode[, compresslevel]])¶
```

This is a shorthand for `GzipFile(filename, mode, compresslevel)`. The `filename` argument is required; `mode` defaults to `'rb'` and `compresslevel` defaults to 9.

### 13.2.1. Examples of usage¶

Example of how to read a compressed file:

```
import gzip
f = gzip.open('/home/joe/file.txt.gz', 'rb')
file_content = f.read()
f.close()
```

Example of how to create a compressed GZIP file:

```
import gzip
content = "Lots of content here"
f = gzip.open('/home/joe/file.txt.gz', 'wb')
f.write(content)
f.close()
```

Example of how to GZIP compress an existing file:

```
import gzip
f_in = open('/home/joe/file.txt', 'rb')
f_out = gzip.open('/home/joe/file.txt.gz', 'wb')
f_out.writelines(f_in)
f_out.close()
f_in.close()
```

See also

Module [zlib](#)

The basic data compression module needed to support the **gzip** file format.

## [Table Of Contents](#)

[13.2. gzip — Support for gzip files](#)

- [13.2.1. Examples of usage](#)

### Previous topic

[13.1. zlib — Compression compatible with gzip](#)

### Next topic

[13.3. bz2 — Compression compatible with bzip2](#)

### This Page

- [Show Source](#)

### Navigation

- [index](#)
- [modules](#) |
- [next](#) |
- [previous](#) |
- [Python v2.6.4 documentation](#) »
- [The Python Standard Library](#) »
- [13. Data Compression and Archiving](#) »

© [Copyright](#) 1990-2010, Python Software Foundation.

The Python Software Foundation is a non-profit corporation. [Please donate.](#)

Last updated on Feb 26, 2010. Created using [Sphinx](#) 0.6.3.