

Unzipping files in Python

Asked 13 years, 1 month ago Modified 1 year, 4 months ago Viewed 1.2m times



I read through the [zipfile documentation](#), but couldn't understand how to *unzip* a file, only how to zip a file. How do I unzip all the contents of a zip file into the same directory?

839



[python](#) [zip](#) [unzip](#) [python-zipfile](#)



Share Follow



edited May 1, 2022 at 12:27



Dharman ♦

31.1k ● 25 ● 86 ● 137

asked Aug 10, 2010 at 16:19



John Howard

61.1k ● 23 ● 50 ● 66

3 Related: unzipping files recursively [stackoverflow.com/questions/28339000/...](#) & [stackoverflow.com/questions/36285502/...](#) – Tung Feb 26, 2020 at 18:33

6 For a one line extraction, see [shutil.unpack_archive\(\)](#) . – phoenix Dec 23, 2020 at 13:05 ✎

1 @fonini answer is -- as of 2021 -- the right/best one: [stackoverflow.com/a/64110098/687896](#) – Brandt Jun 7, 2021 at 6:06

9 Answers

Sorted by:

Highest score (default) ▾



```
import zipfile
with zipfile.ZipFile(path_to_zip_file, 'r') as zip_ref:
    zip_ref.extractall(directory_to_extract_to)
```

1583



That's pretty much it!



Share Follow



edited Sep 24, 2019 at 20:09

answered Aug 10, 2010 at 16:23



Rahul

19.8k ● 1 ● 25 ● 29

11 what if the contents of the .zip archive are same, in all .zip archives? how to rename the content before extracting? example: 1.zip 2.zip.. all contain content.txt : extract all like 1content.txt 2content.txt? – Irtaza Apr 7, 2016 at 16:59

13 @iratzhash I typically create a new temporary directory for the contents using tempfile: [docs.python.org/3/library/tempfile.html](#) I unzip to the temporary directory and the move / organize the files from there. – Dave Forgac Jul 12, 2017 at 17:28

19 @3kstc I would `from zipfile import ZipFile` . When using it, you no longer need to use `zipfile.ZipFile` , and can use `ZipFile(zip_file_name)` . – Debug255 Feb 13, 2018 at 2:00

- 3 thnx, note: There is no zipfile library, no need to pip install, zipfile is already there... – [ntg](#) Apr 9, 2019 at 9:06
- 4 I'd add: `from tempfile import mkdtemp; directory_to_extract_to = mkdtemp()` – [Martin Thoma](#) Jul 31, 2019 at 15:23



If you are using *Python 3.2* or later:

411



```
import zipfile
with zipfile.ZipFile("file.zip", "r") as zip_ref:
    zip_ref.extractall("targetdir")
```



You dont need to use the *close* or *try/catch* with this as it uses the [context manager](#) construction.



Share Follow

edited Sep 13, 2018 at 17:32

answered Apr 16, 2016 at 10:11



[simhumileco](#)

32k ● 16 ● 137 ● 115



[user1741137](#)

4,949 ● 2 ● 19 ● 28

- 24 ZipFile also works as a context manager in 2.7 or later:
docs.python.org/2/library/zipfile.html#zipfile.ZipFile – [FelixEnescu](#) Jan 22, 2017 at 15:01

How to deal with docs.python.org/3.6/library/zipfile.html#zipfile.BadZipFile exception? Generally, what is the best practice to use try/except with context manager (with-statement)? – [SnowOnion](#) Mar 6, 2019 at 8:24

- 2 `zipfile` + `pathlib` = win. mind if i slightly update your answer? – [Umar.H](#) Jan 21, 2020 at 15:38
- 2 is there a reason to avoid `os.system(f'tar -xvzf {filename}')` and instead use zipfile (e.g. `zip = ZipFile('file.zip'); zip.extractall()`) or `shutil.unpack_archive(filename, extract_dir)`? – [Charlie Parker](#) Jan 21, 2021 at 20:09
- 3 @CharlieParker The main reason is portability. `system` calls are OS dependent. For example, `tar` would not be available on Windows. – [Fareanor](#) May 18, 2021 at 16:24



`zipfile` is a somewhat low-level library. Unless you need the specifics that it provides, you can get away with `shutil`'s higher-level functions `make_archive` and `unpack_archive`.

148



`make_archive` is already described in [this](#) answer. As for `unpack_archive`:

```
import shutil
shutil.unpack_archive(filename, extract_dir)
```



`unpack_archive` detects the compression format automatically from the "extension" of `filename` (`.zip`, `.tar.gz`, etc), and so does `make_archive`. Also, `filename` and `extract_dir` can be any [path-like](#) objects (e.g. [pathlib.Path](#) instances) since Python 3.7.

Share Follow

answered Sep 28, 2020 at 21:51



fonini

2,989 ● 3 ● 21 ● 39

- 2 This method doesn't work when the zip file has a custom extension, e.g. (.omt for OmegaT project packages). It gives `raise ReadError("Unknown archive format '{0}'".format(filename))`. – [msoutopico](#) Mar 3, 2021 at 21:00
- 1 @msoutopico you can specify thje format explicitly: `shutil.unpack_archive(filename, extract_dir, format)` – [fonini](#) Mar 4, 2021 at 19:42
- 1 what is wrong with `os.system(f'tar -xvzf {path2zip} -C {path2unzip}/')`? – [Charlie Parker](#) Aug 10, 2021 at 18:49
- 10 @CharlieParker you have already asked the same thing in a comment to another answer, and that comment was answered: [stackoverflow.com/questions/3451111/unzipping-files-in-python/...](https://stackoverflow.com/questions/3451111/unzipping-files-in-python/) `os.system` is not portable, opens up security issues, is harder to use correctly (e.g. your proposal fails when the paths have special characters), and is less readable. – [fonini](#) Aug 10, 2021 at 18:58 ✎
- This solution doesn't seem to maintain executable bits on the stuff inside the archive as it's unzipped? – [Cody Martin](#) Jan 26 at 0:51



49



Use the `extractall` method, if you're using Python 2.6+

```
zip = ZipFile('file.zip')
zip.extractall()
```

Share Follow

answered Aug 10, 2010 at 16:23



Dan Breen

12.6k ● 4 ● 38 ● 49

Don't you have to specify a destination (`zip.extractall(destination)`)? – [asonnenschein](#) Oct 24, 2013 at 18:19 ✎

- 4 Not if you're just extracting into the same directory as the zipfile – [Dan Gayle](#) Dec 4, 2013 at 20:01
- 17 @DanGayle this appears to be extracting the zip file into the current working directory, NOT the location of the zip file – [Brian Leishman](#) Jun 9, 2017 at 14:41
- 5 for me, `ZipFile()` didn't work but `zipfile.ZipFile()` did - after `import zipfile` – [Agile Bean](#) Sep 29, 2018 at 7:34
- 4 You need to `zip.close()` at the end if you don't use a `with` statement like the other answers suggest. – [Boris Verkhovskiy](#) Jun 22, 2020 at 17:52 ✎



23



You can also import only `ZipFile`:

```
from zipfile import ZipFile
zf = ZipFile('path_to_file/file.zip', 'r')
zf.extractall('path_to_extract_folder')
zf.close()
```

Works in *Python 2* and *Python 3*.



Share Follow

edited Sep 13, 2018 at 16:40

answered Sep 13, 2018 at 16:28



simhumileco

32k ● 16 ● 137 ● 115

- 1 Thank you for your attention @MylesHollowed However, this is not a copy from the accepted answer. I agree that they are similar to each other, but they are different. This is also indicated by your comment, because the accepted one is definitely better for you than mine. If it was a copy, it would be the same... For someone my answer may be valuable because it is perhaps more readable and as you noticed import less code... It is because of these differences that I decided to put my answer to give an alternative. Is not that why we can put other answers after accepting one? All the best – [simhumileco](#) Oct 30, 2018 at 8:22

What's wrong with this answer? Why did someone give her a negative point? After all, it is the answer to the question and is distinguished by its simplicity compared to other answers, which may be important for some people who are looking for an answer. Isn't it? – [simhumileco](#) Jan 28, 2020 at 14:09

- 1 @MylesHollowed `import zipfile.ZipFile` generates `ModuleNotFoundError: No module named 'zipfile.ZipFile'; 'zipfile' is not a package` in 3.6.5. I am open to it being operator error on my part, but I don't know what it is. – [MikeF](#) Feb 26, 2020 at 16:38 ✎
- 1 @MikeF I had the same problem with Python 3.8.5 but the workaround was to use `from zipfile import ZipFile`. Hope this helps. – [Ben Dalling](#) Oct 15, 2020 at 9:53

try this :

14

```
import zipfile
def un_zipFiles(path):
    files=os.listdir(path)
    for file in files:
        if file.endswith('.zip'):
            filePath=path+'/'+file
            zip_file = zipfile.ZipFile(filePath)
            for names in zip_file.namelist():
                zip_file.extract(names,path)
            zip_file.close()
```

path : unzip file's path

Share Follow

answered Jul 3, 2019 at 12:53



Done Jin

157 ● 1 ● 4

```
from zipfile import ZipFile
ZipFile("YOURZIP.zip").extractall("YOUR_DESTINATION_DIRECTORY")
```

12

The directory where you will extract your files doesn't need to exist before, you name it at this moment



YOURZIP.zip is the name of the zip if your project is in the same directory. If not, use the PATH i.e : C://....//YOURZIP.zip

Think to escape the `/` by an other `/` in the PATH If you have a `permission denied` try to launch your ide (i.e: Anaconda) as administrator

YOUR_DESTINATION_DIRECTORY will be created in the same directory than your project

Share Follow

edited Jul 2, 2020 at 14:31

answered Jul 2, 2020 at 14:06



Nicoolasens

2,881 ● 17 ● 22

1 to me it's strictly similar – Nicoolasens Jan 25, 2021 at 12:00



12



If you want to do it in shell, instead of writing code.

```
python3 -m zipfile -e myfiles.zip myfiles/
```



Share Follow

answered Aug 5, 2020 at 0:20



Indika K

1,332 ● 17 ● 27

2 why not just do `tar -xvzf path_file` in your case? – Charlie Parker Jan 21, 2021 at 19:39

3 ZIP files are not tar files. Unless you have a special version of `tar` which handles ZIP archives, your command wont work at all. `tar` with the `-z` option processes gzipped tar archives (generally files with extensions `.tgz` or `.tar.gz`) – Perry Jan 22, 2021 at 19:00



2



```
import os
zip_file_path = "C:\\AA\\BB"
file_list = os.listdir(path)
abs_path = []
for a in file_list:
    x = zip_file_path+'\\'+a
    print x
    abs_path.append(x)
for f in abs_path:
    zip=zipfile.ZipFile(f)
    zip.extractall(zip_file_path)
```

This does not contain validation for the file if its not zip. If the folder contains non .zip file it will fail.

Share Follow

answered Apr 13, 2018 at 10:40



[user3911901](#)

39 ● 3



Highly active question. Earn 10 reputation (not counting the [association bonus](#)) in order to answer this question. The reputation requirement helps protect this question from spam and non-answer activity.