

## فشرده‌سازی و استخراج فایل‌ها

همانطور که احتمالاً می‌دانید، فایل‌ها به فرمت فشرده با فرمت zip قابل فشرده‌سازی هستند. اغلب افراد از برنامه‌های ویژه روی کامپیوتر خود برای باز کردن این فایل‌ها استفاده می‌کنند، اما خوشبختانه برای ما، پایتون با چند خط کد ساده می‌تواند همین کار را انجام دهد.

### ایجاد فایل‌ها برای فشرده‌سازی

In [1]:

```
1 # slashes may need to change for MacOS or Linux
2 f = open("new_file.txt", 'w+')
3 f.write("Here is some text")
4 f.close()
```

In [2]:

```
1 # slashes may need to change for MacOS or Linux
2 f = open("new_file2.txt", 'w+')
3 f.write("Here is some text")
4 f.close()
```

### فشرده‌سازی فایل‌ها

کتابخانه [zipfile \(https://docs.python.org/3/library/zipfile.html\)](https://docs.python.org/3/library/zipfile.html) در پایتون به صورت پیش‌فرض وجود دارد و می‌توان از آن برای فشرده‌سازی پوشه‌ها یا فایل‌ها استفاده کرد. برای فشرده‌سازی تمام فایل‌ها در یک پوشه، می‌توانید از روش `os.walk()` برای تکرار این فرایند برای تمام فایل‌ها در یک دایرکتوری استفاده کنید.

In [3]:

```
1 import zipfile
```

Create Zip file first , then write to it (the write step compresses the files.)

In [4]:

```
1 comp_file = zipfile.ZipFile('comp_file.zip', 'w')
```

In [5]:

```
1 comp_file.write("new_file.txt", compress_type=zipfile.ZIP_DEFLATED)
```

In [6]:

```
1 comp_file.write('new_file2.txt',compress_type=zipfile.ZIP_DEFLATED)
```

In [7]:

```
1 comp_file.close()
```

## استخراج از فایل‌های Zip

می‌توانیم به راحتی فایل‌ها را با استفاده از متد `extractall()` برای استخراج تمام فایل‌ها یا با استفاده از متد `extract()` برای استخراج فایل‌های انفرادی، استخراج کنیم.

In [8]:

```
1 zip_obj = zipfile.ZipFile('comp_file.zip','r')
```

In [9]:

```
1 zip_obj.extractall("extracted_content")
```

## استفاده از کتابخانه shutil

به طور معمول شما نمی‌خواهید فایل‌های انفرادی را از یک فایل zip استخراج یا آرشیو کنید، بلکه به جای آن، همه چیز را در یک بار آرشیو کنید. کتابخانه `shutil` که در پایتون به صورت پیش‌فرض وجود دارد، دستوراتی آسان برای این منظور دارد:

In [10]:

```
1 import shutil
```

کتابخانه `shutil` می‌تواند یک پارامتر فرمت بپذیرد. `format` فرمت آرشیو است: یکی از "zip"، "tar"، "gztar"، "bztar" و "xztar".

In [11]:

```
1 pwd
```

Out[11]:

```
'C:\\Users\\babak\\Python For Everyone\\13-Advanced Python Modules'
```

In [12]:

```
1 directory_to_zip='C:\\Users\\babak\\Python For Everyone\\13-Advanced Python Modules'
```

In [13]:

```
1 # Creating a zip archive
2 output_filename = 'example'
3 # Just fill in the output_filename and the directory to zip
4 # Note this won't run as is because the variable are undefined
5 shutil.make_archive(output_filename, 'zip', directory_to_zip)
```

Out[13]:

```
'C:\\Users\\babak\\Python For Everyone\\13-Advanced Python Modules\\example.zip'
```

In [ ]:

```
1 # Extracting a zip archive
2 # Notice how the parameter/argument order is slightly different here
3 shutil.unpack_archive(output_filename, dir_for_extract_result, 'zip')
```