

25. Python 을 이용하여 zipfile handling 하는 방법

zipfile

- zip archive object 를 생성하고 개별 file 을 add

shutil

- 전체 directory 를 빠르게 compress

In [1]:

```
1 # sample file 2 개 생성
2 f = open('new_file_1.txt', 'w')
3 f.write('이것은 첫번째 테스트 file 입니다.')
4 f.close()
5
6 f = open('new_file_2.txt', 'w')
7 f.write('두번째 테스트 file 입니다')
8 f.close()
```

In [2]:

```
1 import zipfile
```

In [3]:

```
1 # empty zipfile object 생성
2 comp_file = zipfile.ZipFile('new_compressed_file.zip', 'w')
```

In [4]:

```
1 # empty object 에 압축할 file 추가
2 comp_file.write('new_file_1.txt')
3 comp_file.write('new_file_2.txt')
```

In [5]:

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

Extract from zipfile

In [6]:

```
1 # 읽어들이 zipfile object 생성
2 zip_object = zipfile.ZipFile('new_compressed_file.zip', 'r')
3
4 # 새로운 directory 에 모든 file 압축 해제
5 zip_object.extractall('unzipped_content')
```

In [7]:

```
1 # 개별 file 해제 (file name, 저장 directory)
2 zip_object.extract('new_file_1.txt', 'unzipped_content_2')
```

Out[7]:

'unzipped_content_2\\new_file_1.txt'

shutil Library

In [8]:

```
1 import shutil
```

In [9]:

```
1 # make_archive (압축 file name, format, source directory)
2 shutil.make_archive('zipped_content3', 'zip', 'unzipped_content')
```

Out[9]:

'C:\\Users\\trimu\\Desktop\\Algorithm 으로 배우는 Python\\jupyterNotebooks\\zipped_content3.zip'

In [10]:

```
1 # archived file 압축 해제 : unpack(압축해제할 filename, target directory, format)
2 shutil.unpack_archive('zipped_content3.zip', 'unzipped_content_3', 'zip')
```