Unzipping and Zipping File

1. Files can be compressed into a zip format, as you are undoubtedly aware. People frequently utilize specialized computer programs to unzip these files; fortunately for us, Python can accomplish the same thing with a few straightforward lines of code.
2. In this lab, we are going to learn how to zip and unzip files. We start by creating two files and writing some content in them.

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

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

3. Now we are going to import the zipfile feature in our Jupyter notebook to zip the files we just created.
4. As you can see below, the format and the code to zip the files. Once you run these commands, you will see that you have a zipped folder with the name comp_file.zip, but it does not contain any files; it is just a zip folder.
5. So, we will compress the two files that we created and send them to the compressed folder.
6. When you will run cell number 10 and 12 you will see that both files have been compressed and they have been sent to the compressed folder.

[6]:
```
import zipfile
```

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

[8]:
```
comp_file = zipfile.ZipFile('comp_file.zip','w')
```

[10]:
```
comp_file.write("new_file.txt",compress_type=zipfile.ZIP_DEFLATED)
```

[12]:
```
comp_file.write('new_file2.txt',compress_type=zipfile.ZIP_DEFLATED)
```

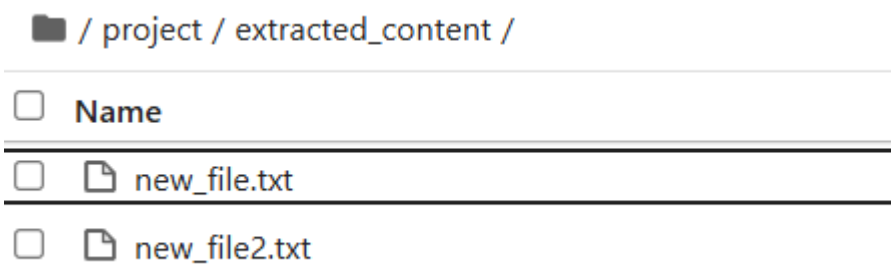
[14]:
```
comp_file.close()
```

7. Now, to unzip the files, first we create a variable, then pass the command and give the name of our compressed file then we will extract all the files.

```
[16]: zip_obj = zipfile.ZipFile('comp_file.zip','r')

[18]: zip_obj.extractall("extracted_content")
```

8. You can see your files in the extracted folder.



9. We can also use the shell utility to zip and unzip the files. Now we are going to take an entire folder that is the extracted_content folder and zip that.
10. First, we need to import the shell util, then we will check our current working directory and then we will run the command to zip the whole directory or folder.

```
[20]: import shutil

[22]: pwd

[22]: 'C:\\Users\\PULKIT\\project'

[24]: directory_to_zip='C:\\Users\\PULKIT\\project\\extracted_content'
```

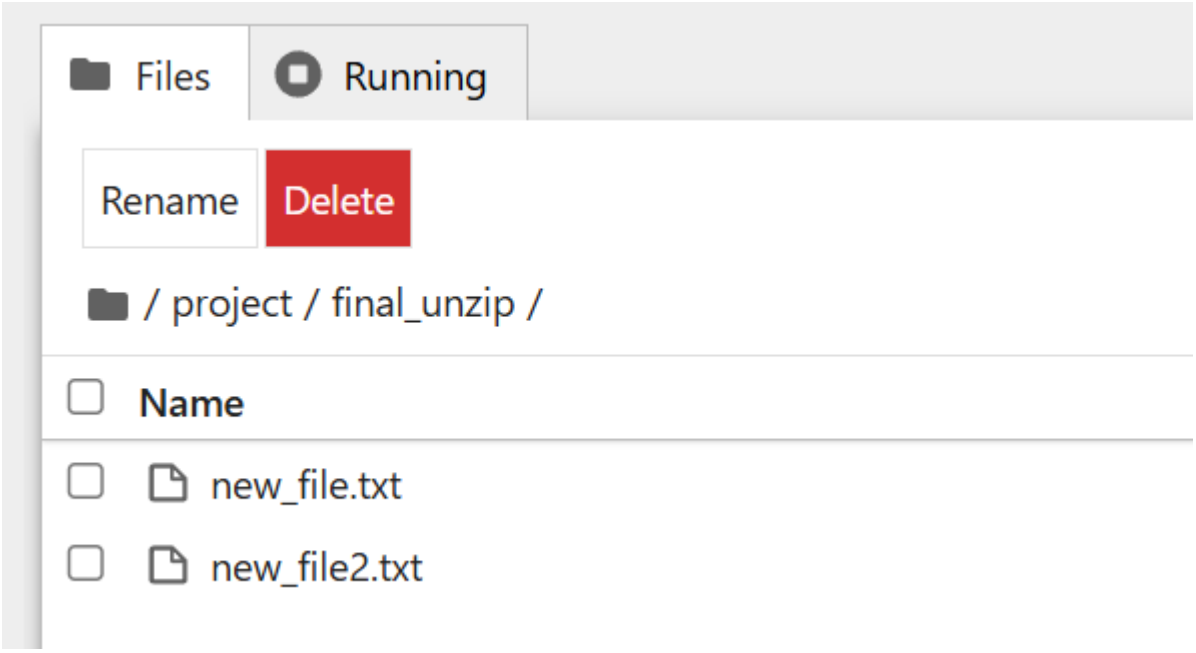
11. Now we will give the output filename and then we will use the shell util to archive it.

```
[26]: # Creating a zip archive
      output_filename = 'example'
      # Just fill in the output_filename and the directory to zip
      # Note this won't run as is because the variable are undefined
      shutil.make_archive(output_filename,'zip',directory_to_zip)

[26]: 'C:\\Users\\PULKIT\\project\\example.zip'
```

12. In the end, we are using shell utility to unpack our archive, or you can say are unzipping the example directory.

```
[28]: # Extracting a zip archive
      # Notice how the parameter/argument order is slightly different here
      shutil.unpack_archive('example.zip','final_unzip','zip')
```


Files
Running
Rename
Delete
/ project / final_unzip /
Name
new_file.txt
new_file2.txt