SECTION 14: ADVANCED PYTHON MODULES, 2 hours 23 minutes, 13 sections

11/13 Zipping and Unzipping files with Python

- -> a zipped file is a compressed file
- -> how we can do this with Python
- -> creating a zip file, compressing text files, inserting them into the zip file, then unzipping them in the file we created

- **creating txt files to compress**
  - -> using the zip file, and then the shell utilities method
  - -> first creating the files
  - -> he opens the file and stores it in the name of a variable
  - -> then he writes the file and closes it
  - -> he does this in a second cell
  - -> then import zipfile <- this compresses a file, and we can create an individual one and compress it
    - **-> there is an entire module just for zip files**
  - -> we are zipping the file and writing to it
  - -> then coming back to the folder we are working with
  - -> the txt files have been created and we are adding them back in
  - -> we are specifying the compression type which we want it to use
  - -> zip deflated
  - -> we are compressing the txt file, compressing it and writing it to another file
  - -> then repeating the process for the second file
  - -> this has created a zip file
  - -> then he has compressed the open zip file
  - -> and then repeated the process for the second file
  - -> this is done with the second compression method
  - -> then he closes the file

```
In [1]: f = open('fileone.txt','w+')
        f.write('ONE FILE')
        f.close()

In [2]: f = open('filetwo.txt','w+')
        f.write('TWO FILE')
        f.close()

In [3]: import zipfile

In [4]: comp_file = zipfile.ZipFile('comp_file.zip','w')
```

```
In [3]: import zipfile

In [4]: comp_file = zipfile.ZipFile('comp_file.zip','w')

In [5]: comp_file.write('fileone.txt',compress_type=zipfile.ZIP_DEFLATED)

In [6]: comp_file.write('filetwo.txt',compress_type=zipfile.ZIP_DEFLATED)

In [7]: comp_file.close()
```

- **extracting the items from the zip file**
  - -> extracting the items from the zip file
  - -> we are creating a variable
  - -> calling the zip file library
  - -> pointing to the zip file we are opening
  - -> then extracting things from it
  - -> when looking for one file, we can tell it which
  - -> else, we tell it to extract all
  - -> and then the path which we want to extract it to
  - -> we are compressing the entire archive
  - -> he also gives an output file name
  - -> we are then calling shell utility
  - -> one of the arguments for this is format
  - -> the output name is where we want to output the zip parameter
  - -> we are also choosing the directory to zip

```
In [11]: import shutil

In [12]: ourses\\Complete-Python-3-Bootcamp\\12-Advanced Python Modules\\extracted_content'

In [13]: output_filename = 'example'

In [14]: shutil.make_archive(output_filename,'zip',dir_to_zip)

Out[14]: 'C:\\Users\\Marcial\\Pierian-Data-Courses\\Complete-Python-3-Bootcamp\\12-Advanced Python Modules\\example.zip'
```

- -> then running the code creates the zip
- -> the shell utilities is being used to do this
- -> this enables us to zip an entire folder

- **to extract the contents from a zip file**
  - -> this is done with shell utilities
  - -> unpacking an archive
  - -> we are telling it the format of the file which we want
  - -> and then the directory to zip <- in this case, the content folder
  - -> we can call shell utilities
  - -> we can also unpack on archive
  - -> the final argument is the folder which it creates
  - -> and then the file type for this
  - -> if it takes example.zip
  - -> we can also use shell utilities