

▼

PART ONE

```
In [27]: 1 #Build the ETL to access LiDAR data files
2
3 from ftplib import FTP
4 ftp = FTP ('ftp.lmic.state.mn.us')
5 ftp.login()
6 ftp.cwd("/pub/data/elevation/lidar/projects/central_lakes/block_1/laz/")
7 ftp.retrlines('LIST')

-rw-rw-r-- 4 12102 5070 35913178 Jan 30 2013 2726-06-27.laz
-rw-rw-r-- 3 12102 5070 40089261 Jan 30 2013 2726-06-28.laz
-rw-rw-r-- 3 12102 5070 37649506 Jan 30 2013 2726-06-29.laz
-rw-rw-r-- 3 12102 5070 37009402 Jan 30 2013 2726-06-30.laz
-rw-rw-r-- 3 12102 5070 35358516 Jan 30 2013 2726-06-31.laz
-rw-rw-r-- 3 12102 5070 35367160 Jan 30 2013 2726-06-32.laz
-rw-rw-r-- 3 12102 5070 22222990 Jan 30 2013 2726-06-33.laz
-rw-rw-r-- 3 12102 5070 25603211 Jan 30 2013 2726-06-34.laz
-rw-rw-r-- 3 12102 5070 28469881 Jan 30 2013 2726-06-35.laz
-rw-rw-r-- 3 12102 5070 33647349 Jan 30 2013 2726-06-36.laz
-rw-rw-r-- 3 12102 5070 46969666 Jan 30 2013 2726-06-37.laz
-rw-rw-r-- 3 12102 5070 41528097 Jan 30 2013 2726-06-38.laz
-rw-rw-r-- 3 12102 5070 43995686 Jan 30 2013 2726-06-39.laz
-rw-rw-r-- 3 12102 5070 57981689 Jan 30 2013 2726-06-40.laz
-rw-rw-r-- 3 12102 5070 40102565 Jan 30 2013 2726-06-41.laz
-rw-rw-r-- 3 12102 5070 33700629 Jan 30 2013 2726-06-42.laz
-rw-rw-r-- 4 12102 5070 32318621 Jan 30 2013 2726-06-43.laz
-rw-rw-r-- 3 12102 5070 39511966 Jan 30 2013 2726-06-44.laz
-rw-rw-r-- 3 12102 5070 43747802 Jan 30 2013 2726-06-45.laz
-rw-rw-r-- 3 12102 5070 32318621 Jan 30 2013 2726-06-46.laz
-rw-rw-r-- 3 12102 5070 32318621 Jan 30 2013 2726-06-47.laz
-rw-rw-r-- 3 12102 5070 32318621 Jan 30 2013 2726-06-48.laz
-rw-rw-r-- 3 12102 5070 32318621 Jan 30 2013 2726-06-49.laz
-rw-rw-r-- 3 12102 5070 32318621 Jan 30 2013 2726-06-50.laz
```

```
In [28]: 1 #select a file from the list and download in laz format
2
3 with open('2726-06-33.laz', 'wb') as fp:
4     ftp.retrbinary('RETR 2726-06-33.laz', fp.write)
```

```
In [29]: 1 ftp.quit

Out[29]: <bound method FTP.quit of <ftplib.FTP object at 0x0000020BE8CC4B48>>
```

```
In [ ]: 1 #use convert LAS to convert the LAZ to an LAS file
2
3 with arcpy.EnvManager(scratchWorkspace=r"C:\Users\celia\Documents\ArcGIS\Projects\Lab2_GIS5572\Lab2_GIS55
4     arcpy.conversion.ConvertLas(r"C:\Users\celia\Documents\ArcGIS\Projects\Lab2_GIS5572\2726-06-33.laz",
5
6
```

```
In [35]: 1 #now the LAS can be converted to a TIN
2
3 arcpy.env.workspace = r"C:\Users\celia\Documents\ArcGIS\Projects\Lab2_GIS5572"
4
5 in_las_dataset = 'Lab2_GIS5572_LAS.las'
6
7 out_tin = 'TIN_Lab2'
```

```
In [36]: 1 arcpy.ddd.LasDatasetToTin(
2     in_las_dataset,
3     out_tin,
4     'RANDOM',
5     'PERCENT',
6     50)
```

Out[36]: **Output**
C:\Users\celia\Documents\ArcGIS\Projects\Lab2_GIS5572\TIN_Lab2

Messages
Start Time: Sunday, February 14, 2021 6:43:40 PM
Succeeded at Sunday, February 14, 2021 6:43:59 PM (Elapsed Time: 19.52 seconds)

```
In [37]: 1 #do the same for a DEM
2
3 arcpy.conversion.LasDatasetToRaster(in_las_dataset, "DEM_Lab2", "ELEVATION", "BINNING AVERAGE LINEAR", "F
4
5
6
```

Out[37]: **Output**
C:\Users\celia\Documents\ArcGIS\Projects\Lab2_GIS5572\DEM_Lab2

Messages
Start Time: Sunday, February 14, 2021 6:50:09 PM
Succeeded at Sunday, February 14, 2021 6:50:11 PM (Elapsed Time: 1.91 seconds)

```
In [44]: 1 #First use the GUI to make two layouts, one of the TIN and one of the DEM
2 #then use Python to export a PDF of the DEM
3 aprx = arcpy.mp.ArcGISProject(r"C:\Users\celia\Documents\ArcGIS\Projects\Lab2_GIS5572\Lab2_GIS5572.aprx")
4 lyr = aprx.listLayouts("DEM_Layout")[0]
5 lyr.exportToPDF(r"C:\Users\celia\Documents\ArcGIS\Projects\Lab2_GIS5572\Lab2_DEM.pdf", resolution = 300)
6
```

Out[44]: 'C:\Users\celia\Documents\ArcGIS\Projects\Lab2_GIS5572\Lab2_DEM.pdf'

```
In [45]: 1 #end of the TIN
2
3 aprx = arcpy.mp.ArcGISProject(r"C:\Users\celia\Documents\ArcGIS\Projects\Lab2_GIS5572\Lab2_GIS5572.aprx")
4 lyr = aprx.listLayouts("TIN_Layout")[0]
5 lyr.exportToPDF(r"C:\Users\celia\Documents\ArcGIS\Projects\Lab2_GIS5572\Lab2_TIN.pdf", resolution = 300)
6
```

Out[45]: 'C:\Users\celia\Documents\ArcGIS\Projects\Lab2_GIS5572\Lab2_TIN.pdf'

▼

PART THREE

```
In [8]: 1 #build the ETL to download the BIL files
2 #first create the URL
3
4 base = "http://services.nacse.org/prism/data/public/normals/4km/ppt/"
5
6 #resolution = '4km'
7 #element = 'ppt'
8 date = '_all'
9
10
11 r.get(base + date)
12
13 #URL.json()
```

Out[8]: <Response [200]>

```
In [15]: 1 #then unzip and save
2
3 from zipfile import ZipFile
4
5 # Create a ZipFile Object and load sample.zip in it
6 with ZipFile('PRISM_ppt_30yr_normal_4km2_all_bil.zip', 'r') as zipObj:
7     # Extract all the contents of zip file in current directory
8     zipObj.extractall()
9
10 input = PRISM_ppt_30yr_normal_4km2_all_bil.bil
```

```
In [ ]: 1 #to create the timeseries, first project the bil files so they don't cause problems later
2 with arcpy.EnvManager(scratchWorkspace=r"C:\Users\celia\Documents\ArcGIS\Projects\Lab2_GIS5572\Lab2_GIS55
3     arcpy.Tmp.BatchProjectRaster(r"C:\Users\celia\Documents\ArcGIS\Projects\Lab2_GIS5572\PRISM_ppt_30yr_n
```

```
In [21]: 1 #next convert bil to tiff since Arc does not like bil files
2 #I don't know why there's an error, this did work
3
4 arcpy.conversion.RasterToOtherFormat(r"C:\Users\celia\Documents\ArcGIS\Projects\Lab2_GIS5572\PRISM_ppt_30
```

NameError Traceback (most recent call last)
In [21]: conversion.arcpy.RasterToOtherFormat("PRISM_ppt_30yr_normal_4km2_annual_bil.bil", r"C:\Users\ce
Line 3: conversion.arcpy.RasterToOtherFormat(r"C:\Users\celia\Documents\ArcGIS\Projects\Lab2_GIS5572", "TIFF")

NameError: name 'conversion' is not defined

```
In [3]: 1 #Create an empty mosaic dataset
2 #Accidentally used my practice mosaic set, Gaudi, later on, so I changed it here to match
3
4 arcpy.management.CreateMosaicDataset(r"C:\Users\celia\Documents\ArcGIS\Projects\Lab2_GIS5572\Lab2_GIS5572
5     "gaudi", "PROJCS['NAD_1983_UTM_Zone_18N', GEOGCS['GCS_North_American_
6
7
```

Out[3]: **Output**
C:\Users\celia\Documents\ArcGIS\Projects\Lab2_GIS5572\Lab2_GIS5572.gdb\gaudi

Messages
Start Time: Wednesday, March 3, 2021 7:00:33 PM
Succeeded at Wednesday, March 3, 2021 7:00:35 PM (Elapsed Time: 2.31 seconds)

```
In [ ]: 1 #now we can add our tiff files to the mosaic
2 arcpy.management.AddRastersToMosaicDataset("gaudi", "Raster Dataset", r"C:\Users\celia\Documents\ArcGIS\P
3
```

```
In [6]: 1 #create and populate the variable field as 'precipitation'
2
3 arcpy.management.CalculateField("gaudi", "variable", "'precipitation'", "PYTHON3", '', "TEXT")
4
```

Out[6]: **Output**
a Layer object

Messages
Start Time: Wednesday, March 3, 2021 7:09:30 PM
Succeeded at Wednesday, March 3, 2021 7:09:31 PM (Elapsed Time: 0.58 seconds)

```
In [7]: 1 #create and populate timestamp field
2
3 arcpy.management.AddField("gaudi", "timestamp", 'date')
4 arcpy.management.CalculateField("gaudi", "timestamp", ""DateAdd(Date(2010,0,1), $feature.OBJECTID-1, "mo
5     """, "ARCADE", '', "TEXT")
6
7
```

Out[7]: **Output**
a Layer object

Messages
Start Time: Wednesday, March 3, 2021 7:10:30 PM
Succeeded at Wednesday, March 3, 2021 7:10:31 PM (Elapsed Time: 0.40 seconds)

```
In [9]: 1 #now build the multidimensional info and make a multidimensional raster layer
2
3 arcpy.md.BuildMultidimensionalInfo("gaudi", "variable", "timestamp #", "variable #")
4 arcpy.md.MakeMultidimensionalRasterLayer("gaudi", "mosaic_Multi", "precipitation", "ALL", None, None, '',
```

Out[9]: **Output**
a Layer object

Messages
Start Time: Wednesday, March 3, 2021 7:13:20 PM
Succeeded at Wednesday, March 3, 2021 7:13:29 PM (Elapsed Time: 8.63 seconds)

```
In [ ]: 1 #finally, create the spacetime cube
2 arcpy.stpm.CreateSpaceTimeCubeHDRasterLayer("mosaic_Multi", r"C:\Users\celia\Documents\ArcGIS\Projects\La
```

```
In [ ]: 1 #but you can't see the result until you visualize it
2 arcpy.stpm.VisualizeSpaceTimeCube3D('PRISM_ppt_30yr_normal_4km2.nc', cube_variable, display_theme, outp
3
4 #it's beautiful!
```

```
In [ ]: 1 #I tried a lot of different ways to export the animation but none of them worked in the time I had
2 #export_map comes from the developer tools and it does not require you to use the GUI to make a layout fi
3 #so it could potentially be used above. However, you need a json of the map and I could not successfully m
4
5 #ExporttoGIF in the GUI does work
6 arcgis.mapping.export_map(web_map_as_json: str = None, format: str = 'GIF', layout_template: str = 'MAP_0
```

```
In [11]: 1 arcpy.server.ExportWebMap("mosaic_Multi.json", r"C:\Users\celia\Documents\ArcGIS\Projects\Lab2_GIS5572\An
```

ExecuteError Traceback (most recent call last)
In [11]: arcpy.server.ExportWebMap("mosaic_Multi.json", r"C:\Users\celia\Documents\ArcGIS\Projects\Lab2_G
Line 1: arcpy.server.ExportWebMap("mosaic_Multi.json", r"C:\Users\celia\Documents\ArcGIS\Projects\Lab2_G
IS5572\Animation.gif", "GIF", None, "MAP_ONLY")

File C:\Program Files\ArcGIS\Pro\Resources\ArcPy\arcpy\server.py, in ExportWebMap:
Line 1001: raise e

File C:\Program Files\ArcGIS\Pro\Resources\ArcPy\arcpy\server.py, in ExportWebMap:
Line 998: retval = convertArcObjectToPythonObject(gp.ExportWebMap_server(*gp_fixargs((Web_Map_as_JSON, Out
put_File, Format, Layout_Templates_Folder, Layout_Template), True)))

File C:\Program Files\ArcGIS\Pro\Resources\ArcPy\arcpy\geoprocessing_base.py, in <lambda>:
Line 511: return lambda *args: val(*gp_fixargs(args, True))

ExecuteError: Web map has invalid JSON.
Failed to execute (ExportWebMap).

```
In [ ]: 1
```