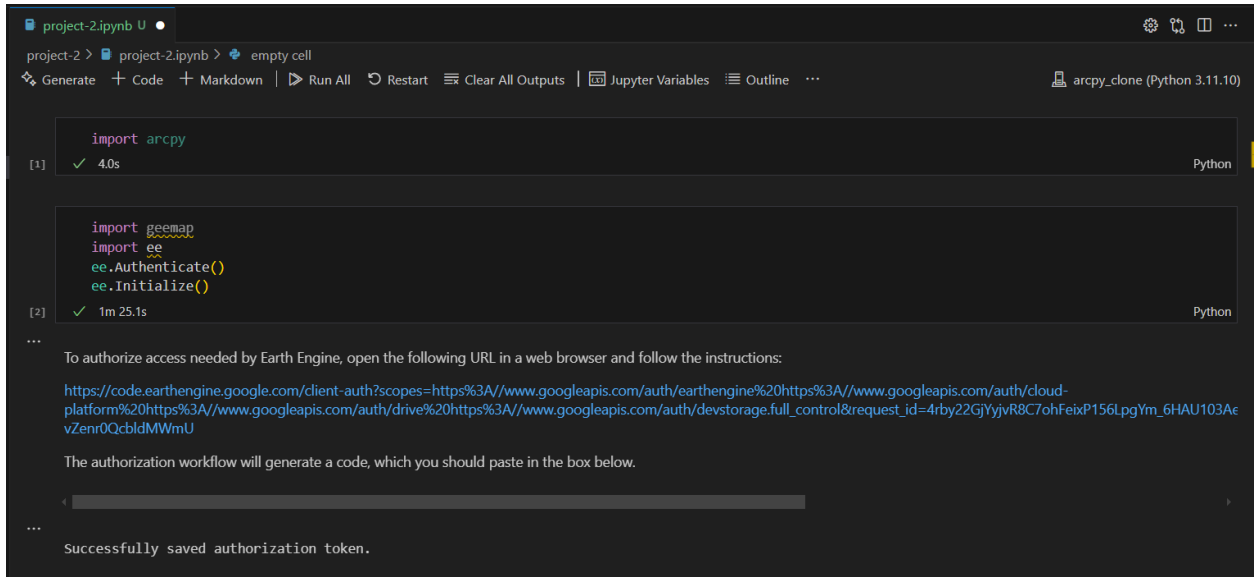


## Working with GEE

### 1. Ensure all libraries are installed in the correct environment



The screenshot shows a Jupyter Notebook titled "project-2.ipynb" with the following content:

```
[1] import arcpy
✓ 4.0s Python
```

```
[2] import geemap
import ee
ee.Authenticate()
ee.Initialize()
✓ 1m 25.1s Python
```

Below the code cells, there is a text block with the following instructions:

... To authorize access needed by Earth Engine, open the following URL in a web browser and follow the instructions:

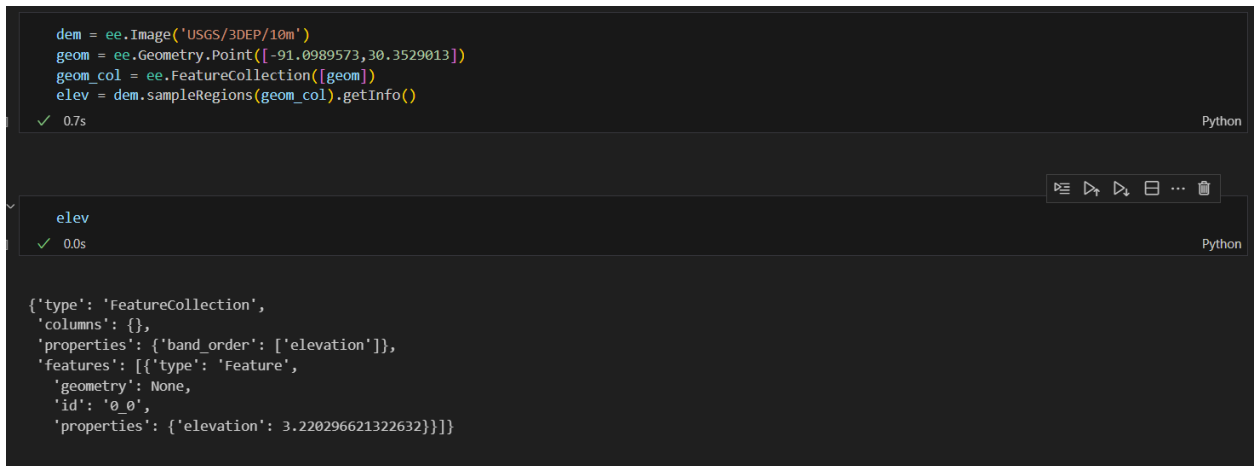
[https://code.earthengine.google.com/client-auth?scopes=https%3A/www.googleapis.com/auth/earthengine%20https%3A/www.googleapis.com/auth/cloud-platform%20https%3A/www.googleapis.com/auth/drive%20https%3A/www.googleapis.com/auth/devstorage.full\\_control&request\\_id=4rby22GjYjvR8C/ohFeixP156LpgYm\\_6HAU103AevZenr0QcblidMWmU](https://code.earthengine.google.com/client-auth?scopes=https%3A/www.googleapis.com/auth/earthengine%20https%3A/www.googleapis.com/auth/cloud-platform%20https%3A/www.googleapis.com/auth/drive%20https%3A/www.googleapis.com/auth/devstorage.full_control&request_id=4rby22GjYjvR8C/ohFeixP156LpgYm_6HAU103AevZenr0QcblidMWmU)

The authorization workflow will generate a code, which you should paste in the box below.

... [A text input box for the authorization code]

... Successfully saved authorization token.

### 2. Explore raster data in gee



The screenshot shows a Jupyter Notebook with the following code cell:

```
dem = ee.Image('USGS/3DEP/10m')
geom = ee.Geometry.Point([-91.0989573, 30.3529013])
geom_col = ee.FeatureCollection([geom])
elev = dem.sampleRegions(geom_col).getInfo()
```

Below the code cell, there is a text block with the following output:

```
✓ 0.7s Python
```

Below the text block, there is a table with the following content:

elev
3.220296621322632

Below the table, there is a text block with the following JSON output:

```
{'type': 'FeatureCollection',
 'columns': {},
 'properties': {'band_order': ['elevation']},
 'features': [{'type': 'Feature',
 'geometry': None,
 'id': '0_0',
 'properties': {'elevation': 3.220296621322632}}]}
```

- Open the provided csv and the raster file, convert the csv file to a spatial dataframe with spatial reference as that of the raster

```
import pandas as pd
table = pd.read_csv("D:\\LSU\\Sp_2025\\GEOG4057\\Ishman-GEOG4057-final-project\\project-2\\data\\boundary.csv")

ra1 = arcpy.Raster("D:\\LSU\\Sp_2025\\GEOG4057\\Ishman-GEOG4057-final-project\\project-2\\data\\flood_2class.tif")
import geopandas
gdf = geopandas.GeoDataFrame(table)
gdf.set_geometry(geopandas.points_from_xy(gdf['X'], gdf['Y']), inplace=True, crs=f'EPSG:{ra1.spatialReference.factoryCode}')

gdf
```

	col	row	X	Y	geometry
0	4871	174	699102.887792	186780.445813	POINT (699102.888 186780.446)
1	4871	174	699102.887792	186780.445813	POINT (699102.888 186780.446)
2	4872	174	699105.887419	186780.445813	POINT (699105.887 186780.446)
3	4870	175	699099.888166	186777.446186	POINT (699099.888 186777.446)
4	4873	174	699108.887046	186780.445813	POINT (699108.887 186780.446)
...	...	...	...	...	...
333	4825	232	698964.904966	186606.467466	POINT (698964.905 186606.467)
334	4826	234	698967.904593	186600.468213	POINT (698967.905 186600.468)
335	4825	233	698964.904966	186603.467840	POINT (698964.905 186603.468)
336	4825	234	698964.904966	186600.468213	POINT (698964.905 186600.468)
337	4825	234	698964.904966	186600.468213	POINT (698964.905 186600.468)

338 rows x 5 columns

- Extracting the elevation values from the dem to the points shapefile by adding another attribute called “elevation” [full code has been provided in the github repository as project-2.ipynb]

```
import os
arcpy.env.workspace = "D:\\LSU\\Sp_2025\\GEOG4057\\Ishman-GEOG4057-final-project\\project-2\\data"

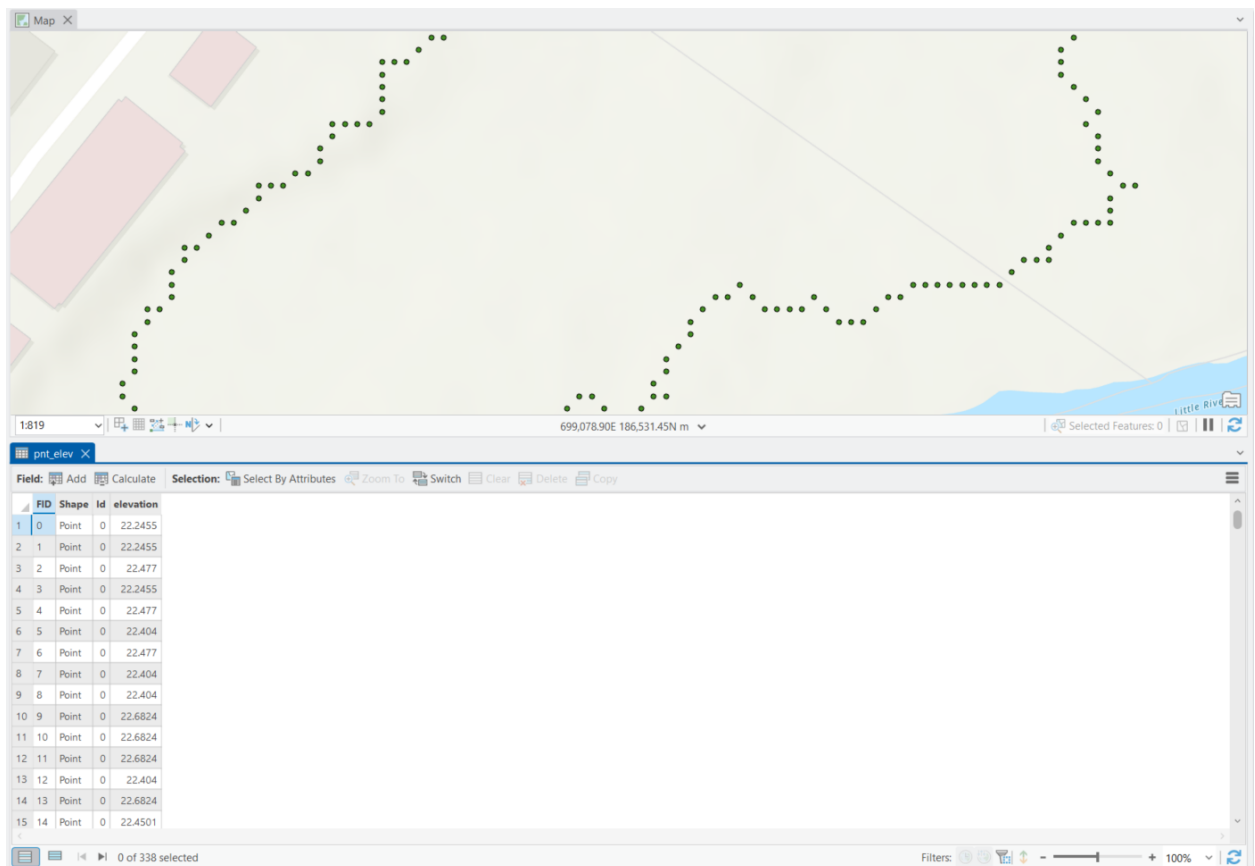
fcname=os.path.join(arcpy.env.workspace,'pnt_elev.shp')
if arcpy.Exists(fcname):
    arcpy.management.Delete(fcname)
arcpy.management.CreateFeatureclass(arcpy.env.workspace,'pnt_elev.shp',geometry_type='POINT',spatial_reference=32119)

arcpy.management.AddField(fcname, field_name='elevation', field_type='FLOAT')
```

Messages

Start Time: Thursday, May 8, 2025 2:24:47 AM  
Succeeded at Thursday, May 8, 2025 2:24:47 AM (Elapsed Time: 0.12 seconds)

```
arcpy.da.InsertCursor(fcname, ['SHAPE@', 'elevation']) as cursor:
    for feat in origin_info['features']:
        # get the coordinates and create a pointgeometry
        coords = feat['geometry']['coordinates']
        pnt = arcpy.PointGeometry(arcpy.Point(coords[0], coords[1]), spatial_reference=32119)
        # get the properties and write it to elevation
        elev = feat['properties']['elevation']
        cursor.insertRow([pnt, elev])
```

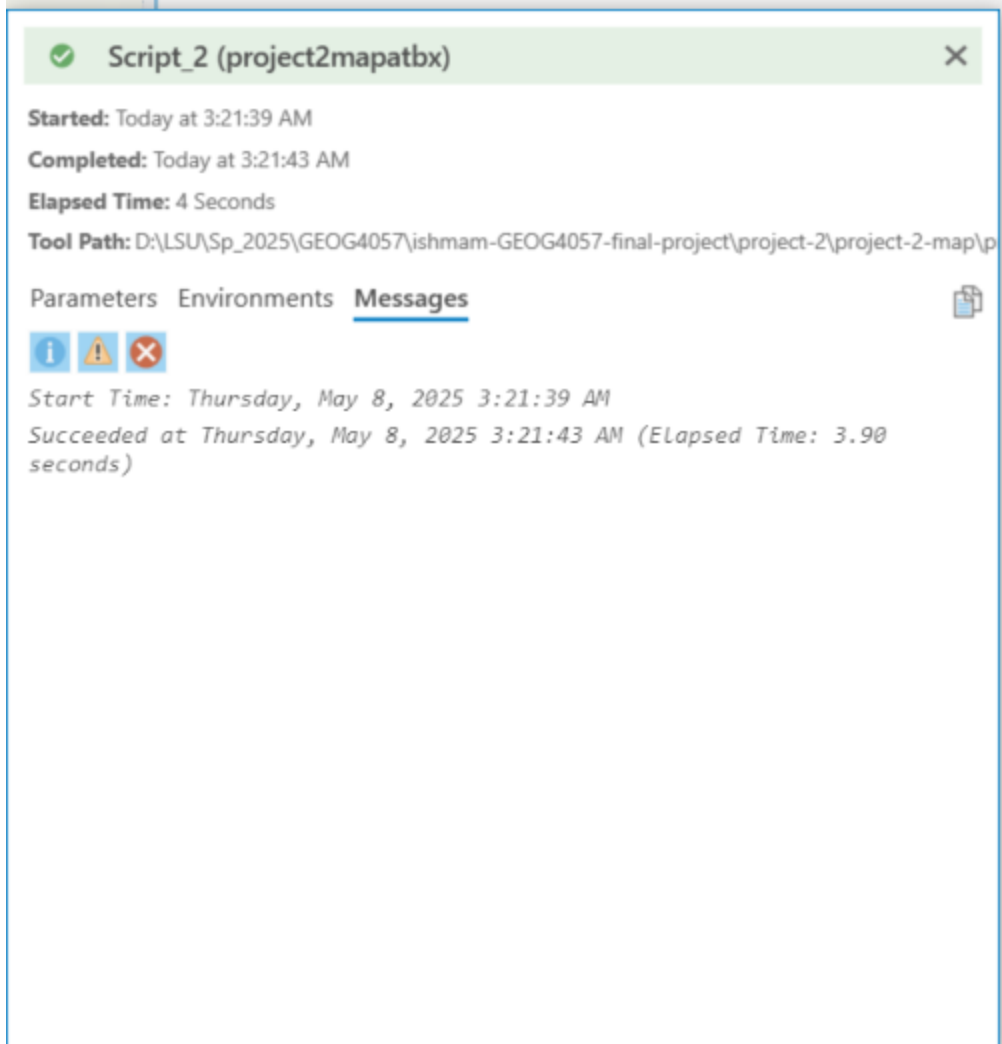
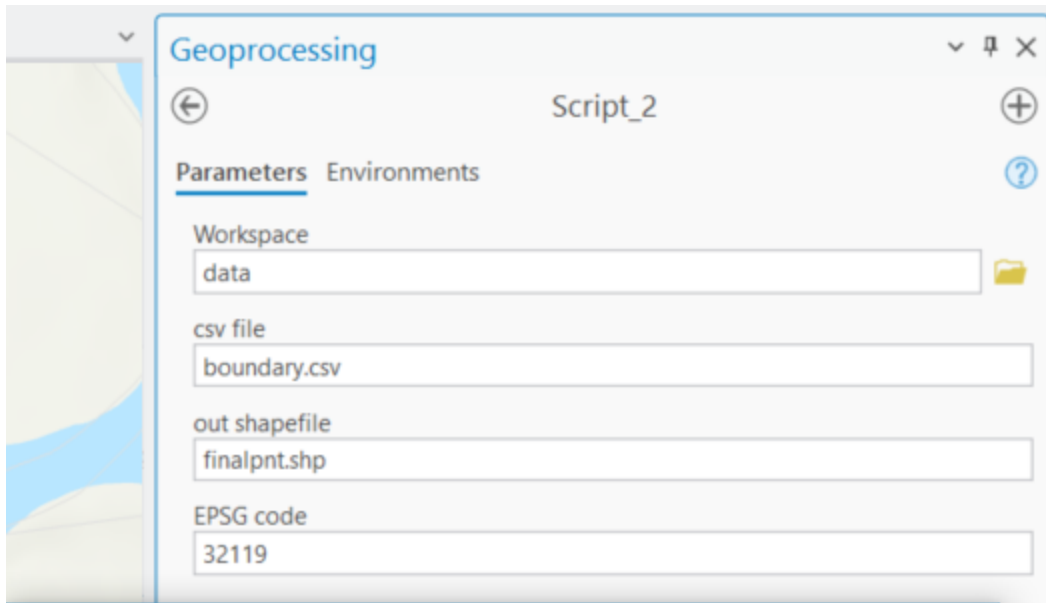


## 5. Achieving the same objective using script tool in ArcGIS Pro

The screenshot shows the 'New Script' dialog box in ArcGIS Pro. The 'Parameters' tab is selected, displaying a table of script parameters. The table has 12 columns: Label, Name, Data Type, Type, Direction, Description, Category, Filter, Dependency, Default, Environment, and Symbol. The parameters are as follows:

Label	Name	Data Type	Type	Direction	Description	Category	Filter	Dependency	Default	Environment	Symbol
0	Workspace	Workspace	Required	Input							
1	csv file	csv_file	String	Required	Input						
2	out shapef...	out_shapefile	String	Required	Output						
3	EPSG code	EPSG_code	String	Required	Input						
*			String	Required	Input						

At the bottom of the dialog, there is a 'Learn more about script tools' link and 'OK' and 'Cancel' buttons.



Github repo:

<https://github.com/ishmamshahid/ishmam-GEOG4057-final-project/tree/main/project-2>