Project #2

GEOG 4057

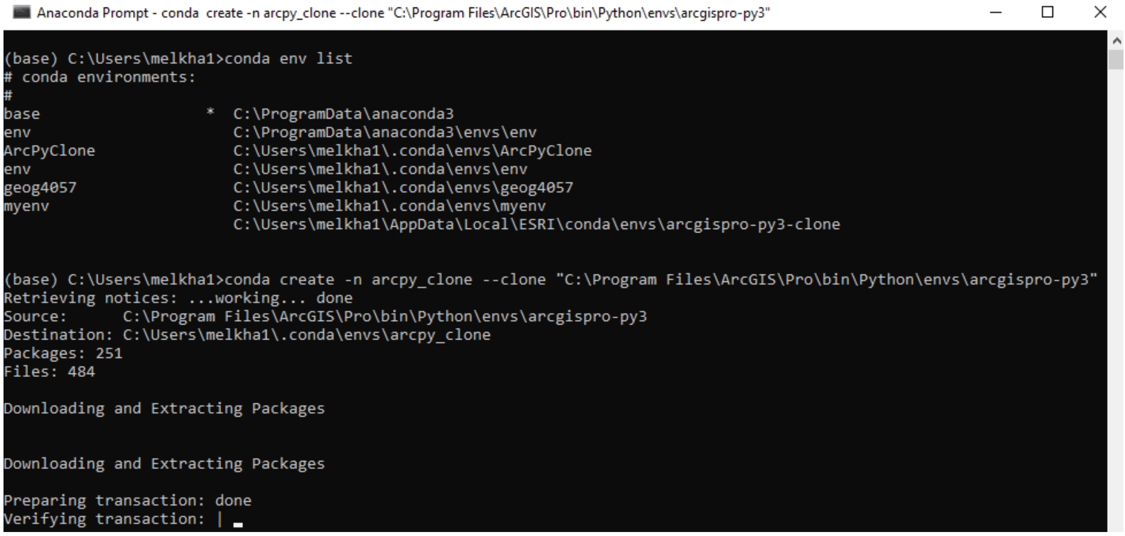
GIS programming

Prof. Lei Wang

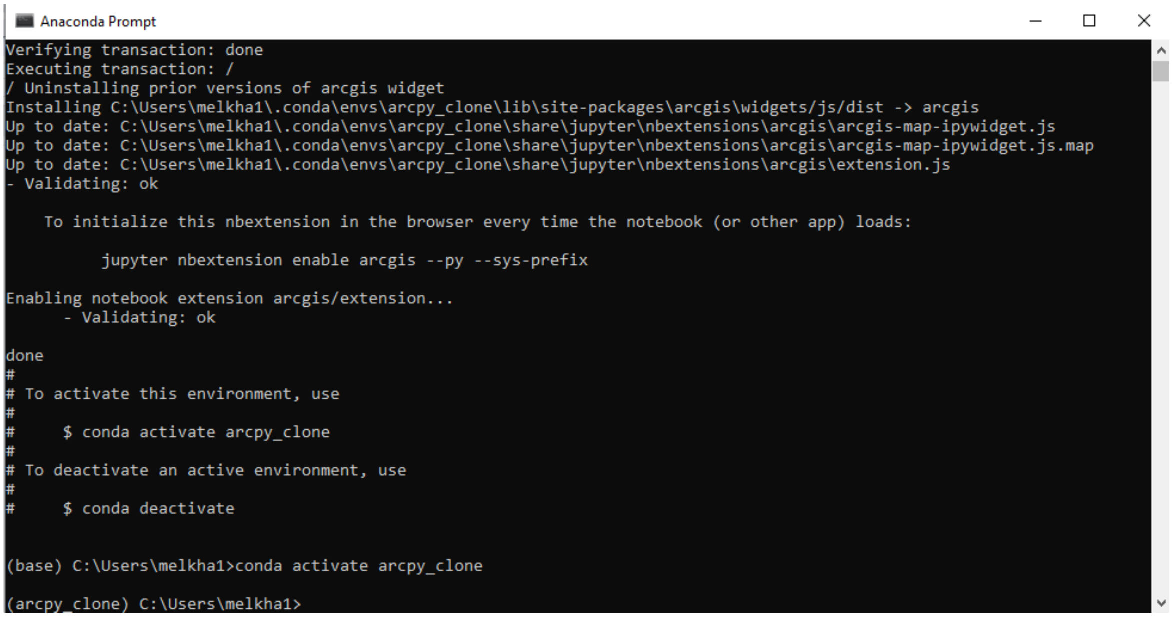
Prepared by/

Mohammed Elkharakany

Clone and activate the environment

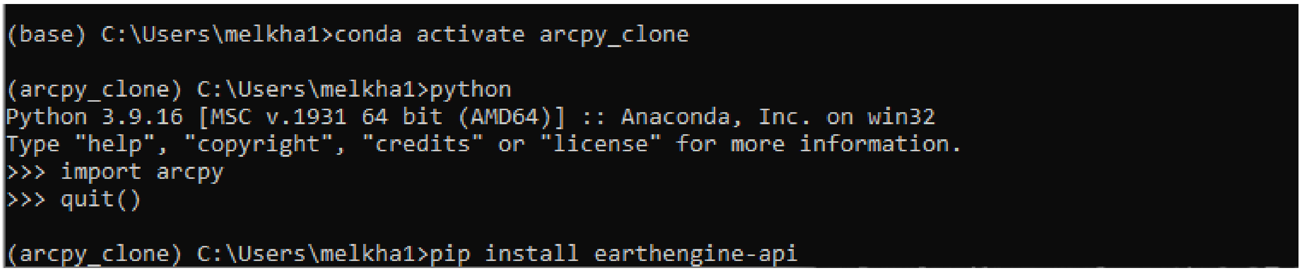

Anaconda Prompt - conda create -n arcpy_clone --clone "C:\Program Files\ArcGIS\Pro\bin\Python\envs\arcgispro-py3"

```
(base) C:\Users\melkha1>conda env list
# conda environments:
#
base                  *  C:\ProgramData\anaconda3
env                      C:\ProgramData\anaconda3\envs\env
ArcPyClone               C:\Users\melkha1\.conda\envs\ArcPyClone
env                      C:\Users\melkha1\.conda\envs\env
geog4057                 C:\Users\melkha1\.conda\envs\geog4057
myenv                    C:\Users\melkha1\.conda\envs\myenv
                         C:\Users\melkha1\AppData\Local\ESRI\conda\envs\arcgispro-py3-clone


(base) C:\Users\melkha1>conda create -n arcpy_clone --clone "C:\Program Files\ArcGIS\Pro\bin\Python\envs\arcgispro-py3"
Retrieving notices: ...working... done
Source:      C:\Program Files\ArcGIS\Pro\bin\Python\envs\arcgispro-py3
Destination: C:\Users\melkha1\.conda\envs\arcpy_clone
Packages: 251
Files: 484

Downloading and Extracting Packages


Downloading and Extracting Packages

Preparing transaction: done
Verifying transaction: |
```



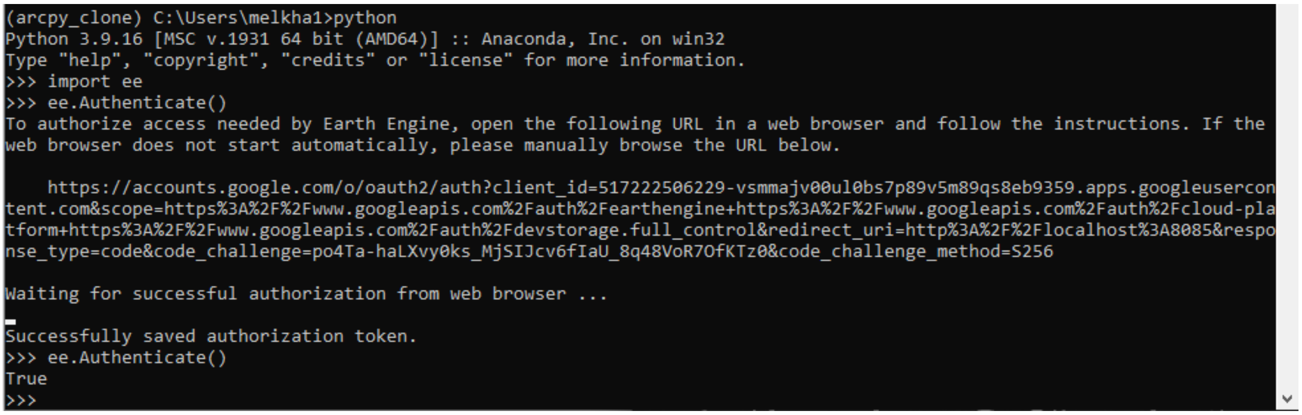
```
Verifying transaction: done
Executing transaction: /
/ Uninstalling prior versions of arcgis widget
Installing C:\Users\melkha1\.conda\envs\arcpy_clone\lib\site-packages\arcgis\widgets/js/dist -> arcgis
Up to date: C:\Users\melkha1\.conda\envs\arcpy_clone\share\jupyter\nbextensions\arcgis\arcgis-map-ipywidget.js
Up to date: C:\Users\melkha1\.conda\envs\arcpy_clone\share\jupyter\nbextensions\arcgis\arcgis-map-ipywidget.js.map
Up to date: C:\Users\melkha1\.conda\envs\arcpy_clone\share\jupyter\nbextensions\arcgis\extension.js
- Validating: ok

    To initialize this nbextension in the browser every time the notebook (or other app) loads:

          jupyter nbextension enable arcgis --py --sys-prefix

Enabling notebook extension arcgis/extension...
      - Validating: ok

done
#
# To activate this environment, use
#
#     $ conda activate arcpy_clone
#
# To deactivate an active environment, use
#
#     $ conda deactivate


(base) C:\Users\melkha1>conda activate arcpy_clone

(arcpy_clone) C:\Users\melkha1>
```

Installation of the Earth Engine



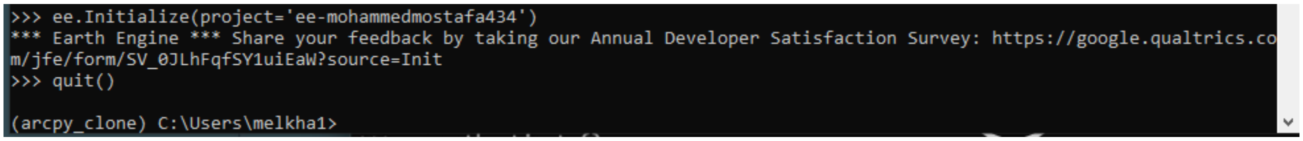
```
(base) C:\Users\melkha1>conda activate arcpy_clone

(arcpy_clone) C:\Users\melkha1>python
Python 3.9.16 [MSC v.1931 64 bit (AMD64)] :: Anaconda, Inc. on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> import arcpy
>>> quit()

(arcpy_clone) C:\Users\melkha1>pip install earthengine-api
```

Check and Authenticate



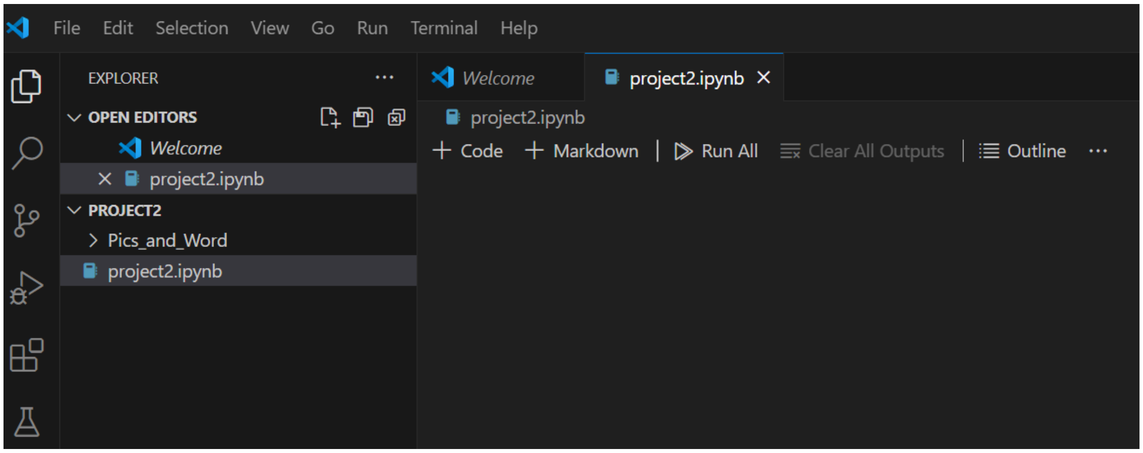
```
(arcpy_clone) C:\Users\melkha1>python
Python 3.9.16 [MSC v.1931 64 bit (AMD64)] :: Anaconda, Inc. on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> import ee
>>> ee.Authenticate()
To authorize access needed by Earth Engine, open the following URL in a web browser and follow the instructions. If the
web browser does not start automatically, please manually browse the URL below.

    https://accounts.google.com/o/oauth2/auth?client_id=517222506229-vsmmajv00ul0bs7p89v5m89qs8eb9359.apps.googleusercon
tent.com&scope=https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fearthengine+https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fcloud-pla
tform+https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fdevstorage.full_control&redirect_uri=http%3A%2F%2Flocalhost%3A8085&respo
nse_type=code&code_challenge=po4Ta-haLXvy0ks_MjSIJcv6fIaU_8q48VoR7OfKTz0&code_challenge_method=S256

Waiting for successful authorization from web browser ...

Successfully saved authorization token.
>>> ee.Authenticate()
True
>>>
```

Project Initialization

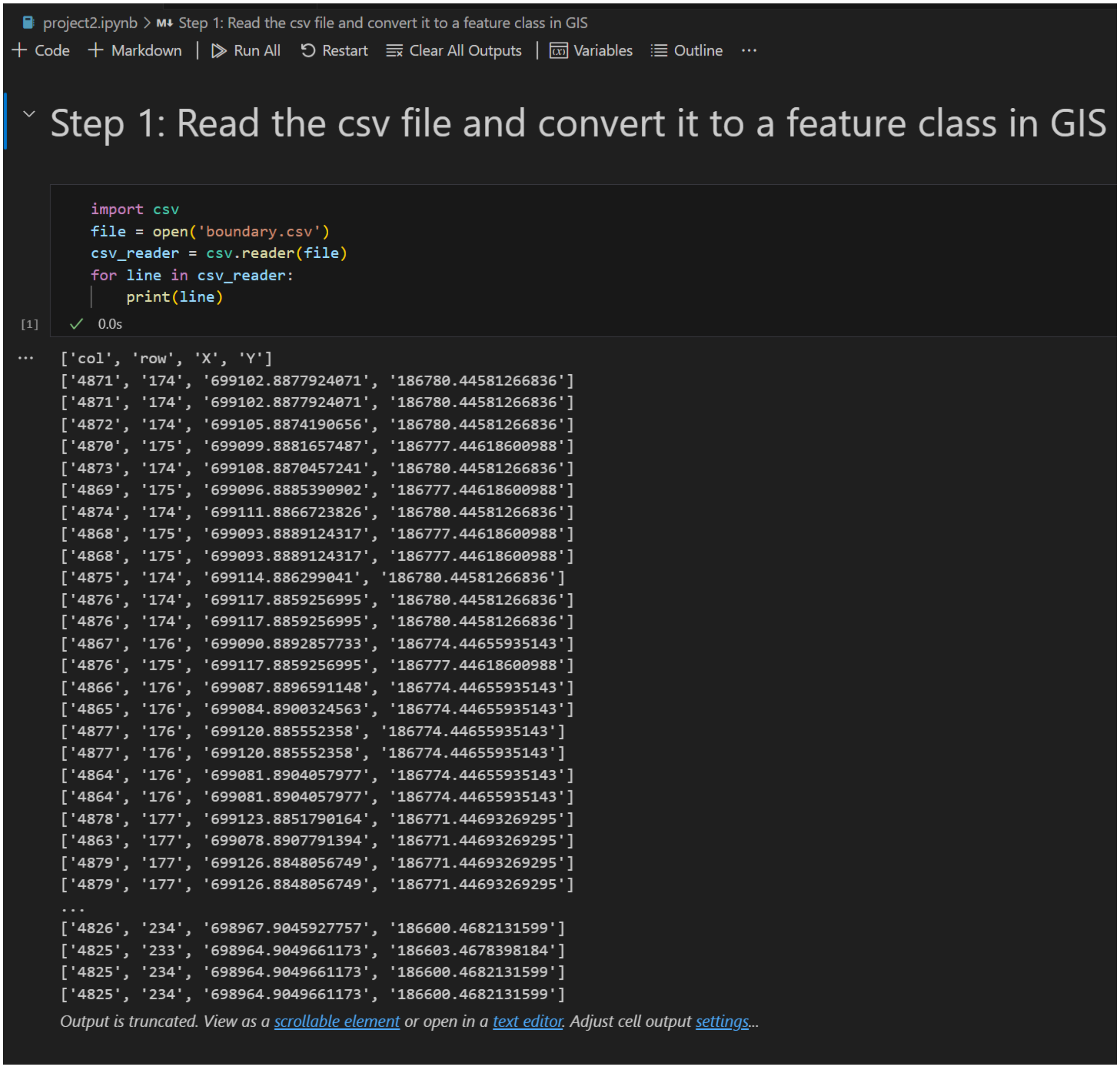


```
>>> ee.Initialize(project='ee-mohammedmostafa434')
*** Earth Engine *** Share your feedback by taking our Annual Developer Satisfaction Survey: https://google.qualtrics.co
m/jfe/form/SV_0JLhFqfSY1uiEaW?source=Init
>>> quit()

(arcpy_clone) C:\Users\melkha1>
```

Set workspace in VS code



Retrieve information of the csv and tif files

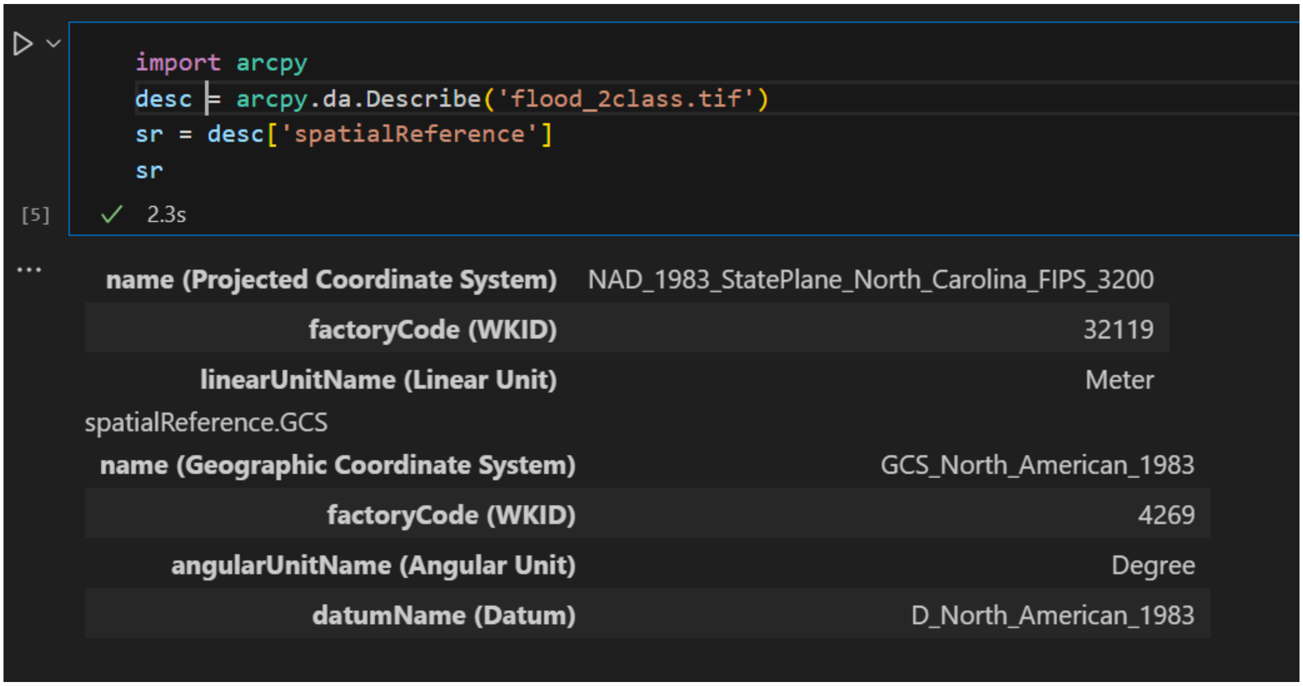

project2.ipynb > M↓ Step 1: Read the csv file and convert it to a feature class in GIS
+ Code
+ Markdown
Run All
Restart
Clear All Outputs
Variables
Outline
···
[1]
✓ 0.0s

Step 1: Read the csv file and convert it to a feature class in GIS

```python
import csv
file = open('boundary.csv')
csv_reader = csv.reader(file)
for line in csv_reader:
    print(line)
```

```
['col', 'row', 'X', 'Y']
['4871', '174', '699102.8877924071', '186780.44581266836']
['4871', '174', '699102.8877924071', '186780.44581266836']
['4872', '174', '699105.8874190656', '186780.44581266836']
['4870', '175', '699099.8881657487', '186777.44618600988']
['4873', '174', '699108.8870457241', '186780.44581266836']
['4869', '175', '699096.8885390902', '186777.44618600988']
['4874', '174', '699111.8866723826', '186780.44581266836']
['4868', '175', '699093.8889124317', '186777.44618600988']
['4868', '175', '699093.8889124317', '186777.44618600988']
['4875', '174', '699114.886299041', '186780.44581266836']
['4876', '174', '699117.8859256995', '186780.44581266836']
['4876', '174', '699117.8859256995', '186780.44581266836']
['4867', '176', '699090.8892857733', '186774.44655935143']
['4876', '175', '699117.8859256995', '186777.44618600988']
['4866', '176', '699087.8896591148', '186774.44655935143']
['4865', '176', '699084.8900324563', '186774.44655935143']
['4877', '176', '699120.885552358', '186774.44655935143']
['4877', '176', '699120.885552358', '186774.44655935143']
['4864', '176', '699081.8904057977', '186774.44655935143']
['4864', '176', '699081.8904057977', '186774.44655935143']
['4878', '177', '699123.8851790164', '186771.44693269295']
['4863', '177', '699078.8907791394', '186771.44693269295']
['4879', '177', '699126.8848056749', '186771.44693269295']
['4879', '177', '699126.8848056749', '186771.44693269295']
...
['4826', '234', '698967.9045927757', '186600.4682131599']
['4825', '233', '698964.9049661173', '186603.4678398184']
['4825', '234', '698964.9049661173', '186600.4682131599']
['4825', '234', '698964.9049661173', '186600.4682131599']
```

Output is truncated. View as a scrollable element or open in a text editor. Adjust cell output settings...

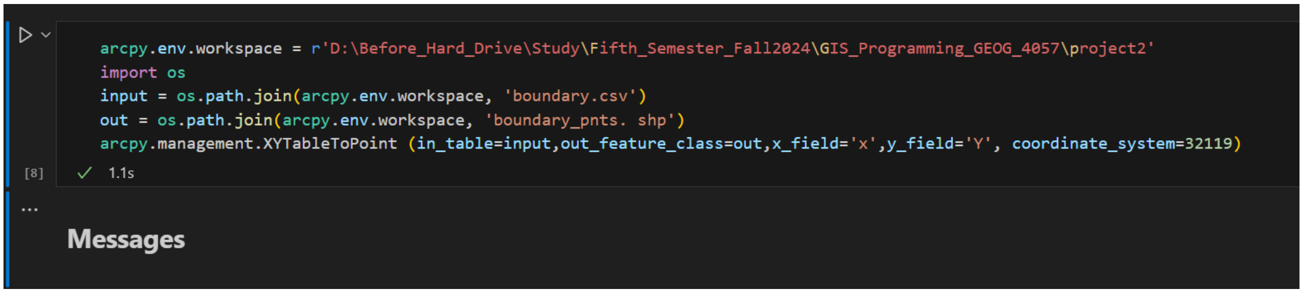


```
import arcpy
desc = arcpy.da.Describe('flood_2class.tif')
sr = desc['spatialReference']
sr
```

[5] ✓ 2.3s

```
name (Projected Coordinate System)    NAD_1983_StatePlane_North_Carolina_FIPS_3200
               factoryCode (WKID)                                            32119
       linearUnitName (Linear Unit)                                           Meter
spatialReference.GCS
name (Geographic Coordinate System)                       GCS_North_American_1983
                factoryCode (WKID)                                             4269
     angularUnitName (Angular Unit)                                          Degree
                  datumName (Datum)                             D_North_American_1983
```

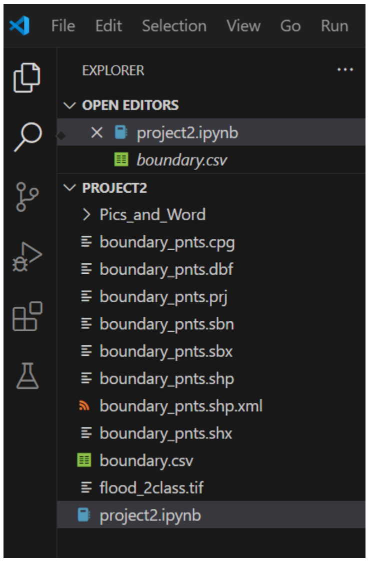
Creating feature class using arcpy module



```
arcpy.env.workspace = r'D:\Before_Hard_Drive\Study\Fifth_Semester_Fall2024\GIS_Programming_GEOG_4057\project2'
import os
input = os.path.join(arcpy.env.workspace, 'boundary.csv')
out = os.path.join(arcpy.env.workspace, 'boundary_pnts. shp')
arcpy.management.XYTableToPoint (in_table=input,out_feature_class=out,x_field='x',y_field='Y', coordinate_system=32119)
```

[8] ✓ 1.1s

Messages



OPEN EDITORS

× project2.ipynb

boundary.csv

PROJECT2

Pics_and_Word

boundary_pnts.cpg

boundary_pnts.dbf

boundary_pnts.prj

boundary_pnts.sbn

boundary_pnts.sbx

boundary_pnts.shp

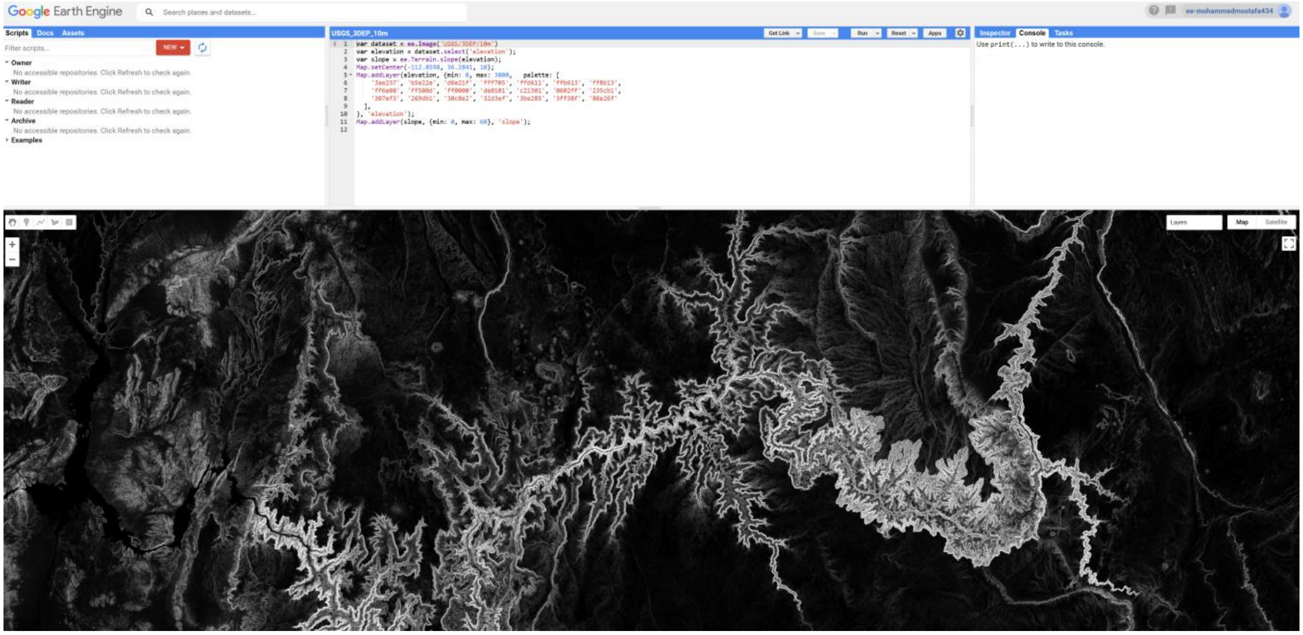
boundary_pnts.shp.xml

boundary_pnts.shx

boundary.csv

flood_2class.tif

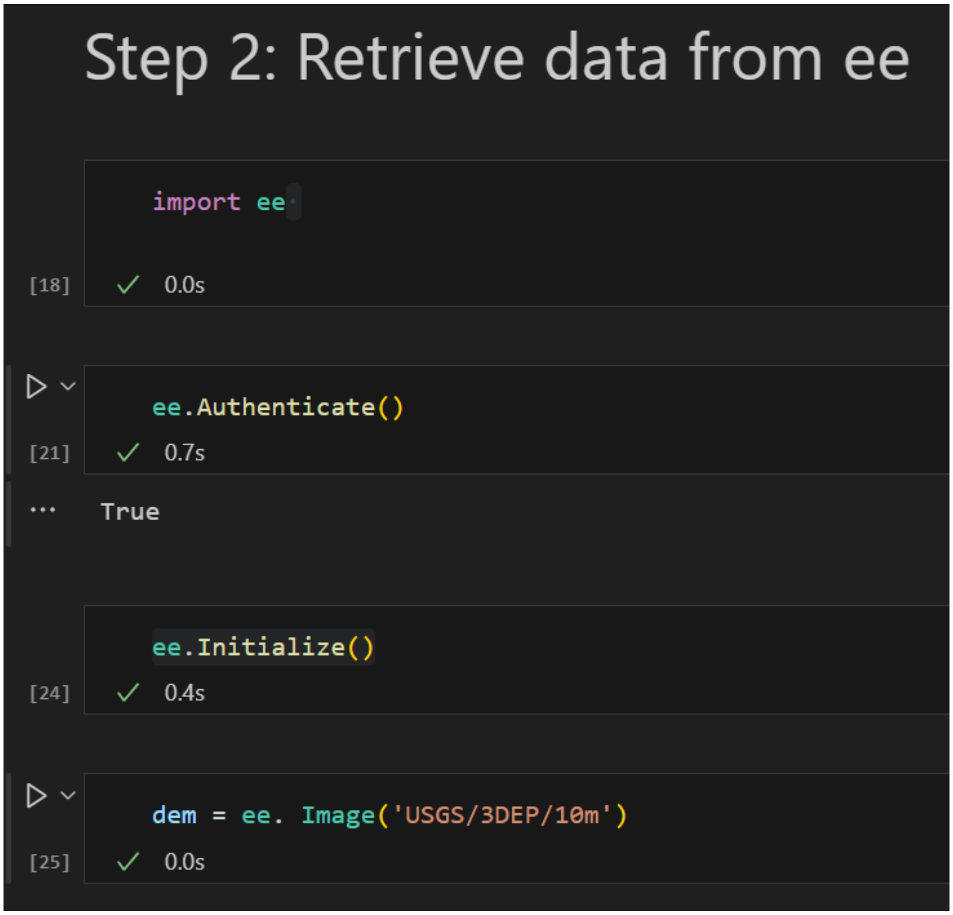
project2.ipynb


Google Earth Engine
Search places and datasets...
Scripts Docs Assets
Filter scripts...
NEW
Owner
No accessible repositories. Click Refresh to check again.
Writer
No accessible repositories. Click Refresh to check again.
Reader
No accessible repositories. Click Refresh to check again.
Archive
No accessible repositories. Click Refresh to check again.
Examples
USGS_3DEP_10m
Get Link
Save
Run
Reset
Apps
Inspector Console Tasks
Layers
Map Satellite

```
var dataset = ee.Image('USGS/3DEP/10m')
var elevation = dataset.select('elevation');
var slope = ee.Terrain.slope(elevation);
Map.setCenter(-112.8598, 36.2841, 10);
Map.addLayer(elevation, {min: 0, max: 3000,   palette: [
    '3ae237', 'b5e22e', 'd6e21f', 'fff705', 'ffd611', 'ffb613', 'ff8b13',
    'ff6e08', 'ff500d', 'ff0000', 'de0101', 'c21301', '0602ff', '235cb1',
    '307ef3', '269db1', '30c8e2', '32d3ef', '3be285', '3ff38f', '86e26f'
  ],
}, 'elevation');
Map.addLayer(slope, {min: 0, max: 60}, 'slope');
```

Use print(...) to write to this console.

Retrieve data from ee



Step 2: Retrieve data from ee

```
import ee
```

[18] ✓ 0.0s

```
ee.Authenticate()
```

[21] ✓ 0.7s

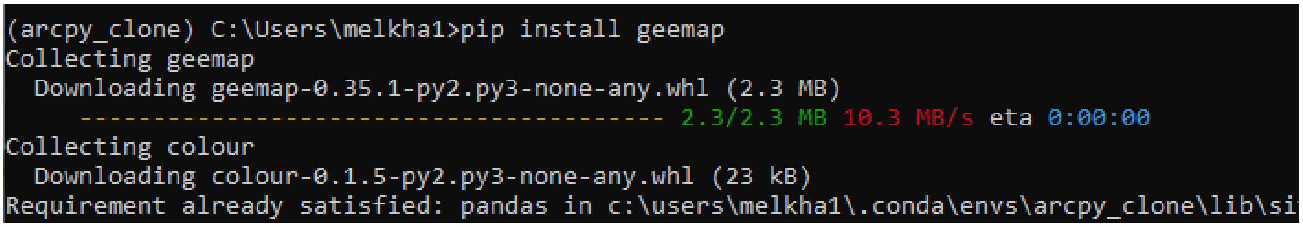
```
True
```

```
ee.Initialize()
```

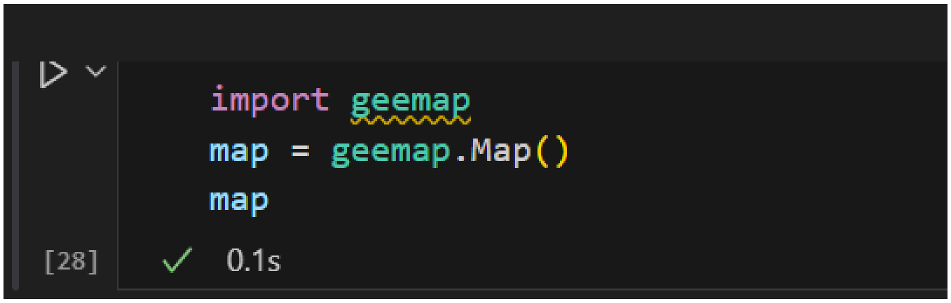
[24] ✓ 0.4s

```
dem = ee. Image('USGS/3DEP/10m')
```

[25] ✓ 0.0s

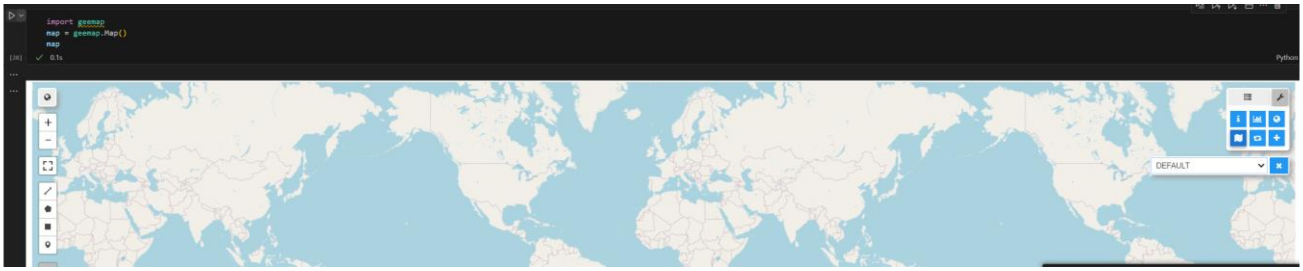


```
(arcpy_clone) C:\Users\melkha1>pip install geemap
Collecting geemap
  Downloading geemap-0.35.1-py2.py3-none-any.whl (2.3 MB)
     ---------------------------------------- 2.3/2.3 MB 10.3 MB/s eta 0:00:00
Collecting colour
  Downloading colour-0.1.5-py2.py3-none-any.whl (23 kB)
Requirement already satisfied: pandas in c:\users\melkha1\.conda\envs\arcpy_clone\lib\si
```



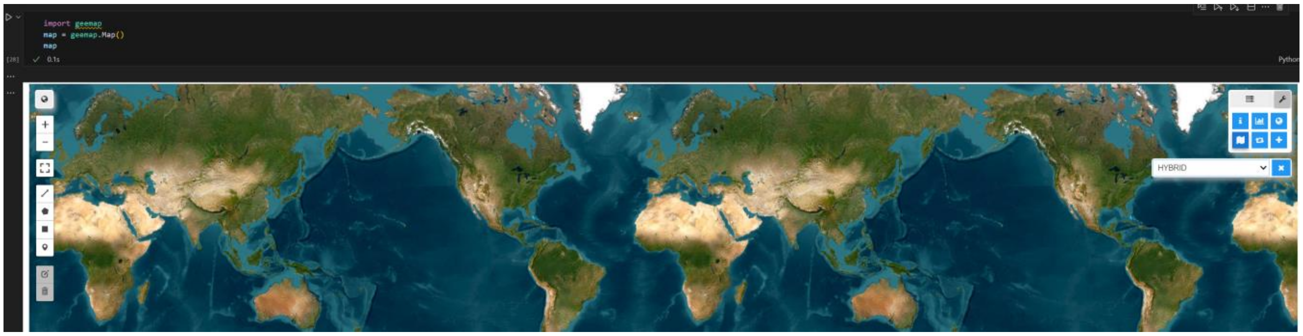
```
import geemap
map = geemap.Map()
map
```

[28] ✓ 0.1s



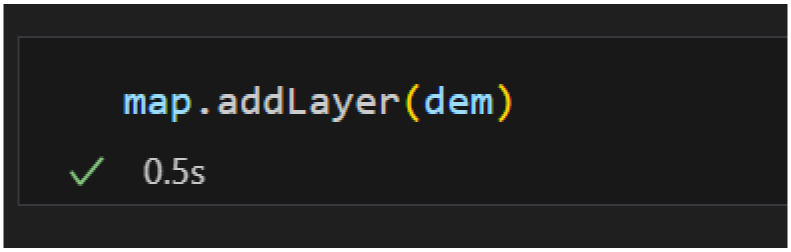
```
import geemap
map = geemap.Map()
map
```

[28] ✓ 0.1s

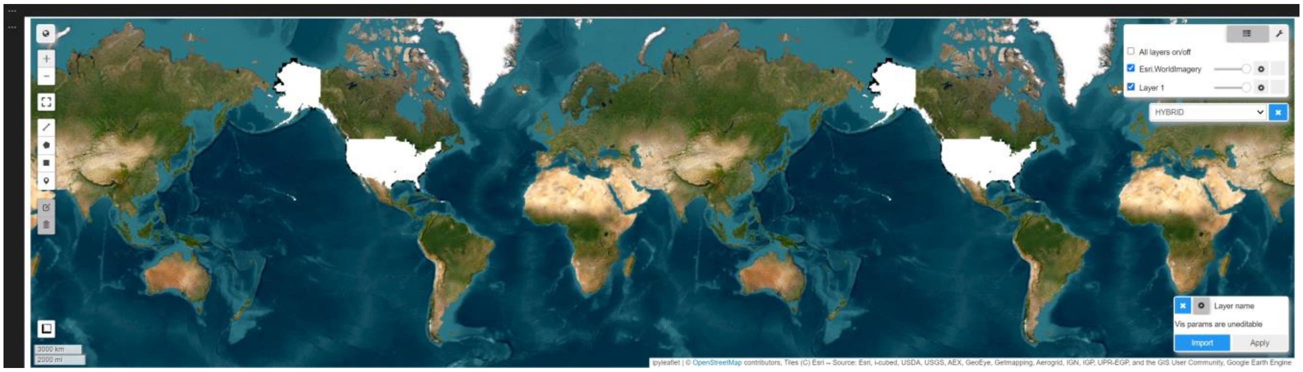


```
import geemap
map = geemap.Map()
map
```

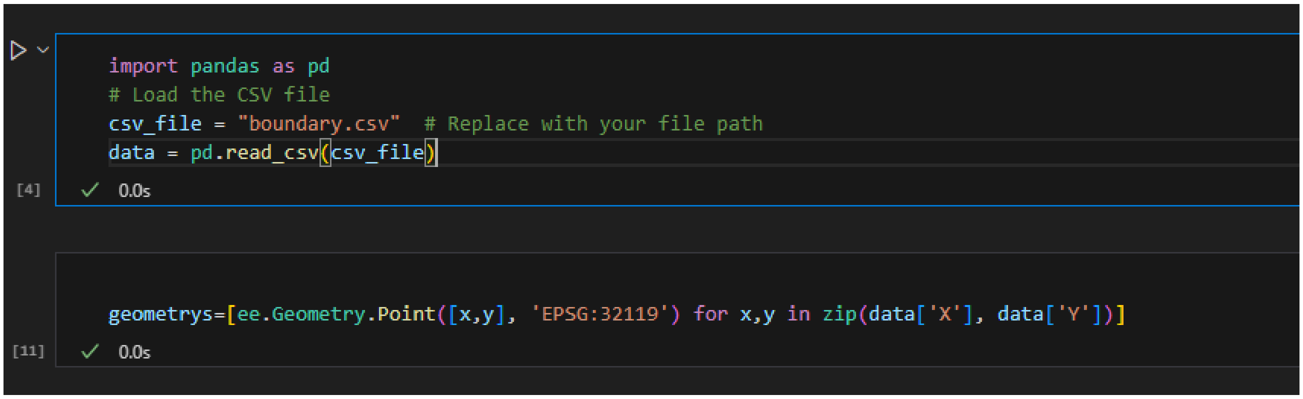
[28] ✓ 0.1s



```
map.addLayer(dem)
✓ 0.5s
```



ipyleaflet | © OpenStreetMap contributors, Tiles (C) Esri -- Source: Esri, i-cubed, USDA, USGS, AEX, GeoEye, Getmapping, Aerogrid, IGN, IGP, UPR-EGP, and the GIS User Community, Google Earth Engine

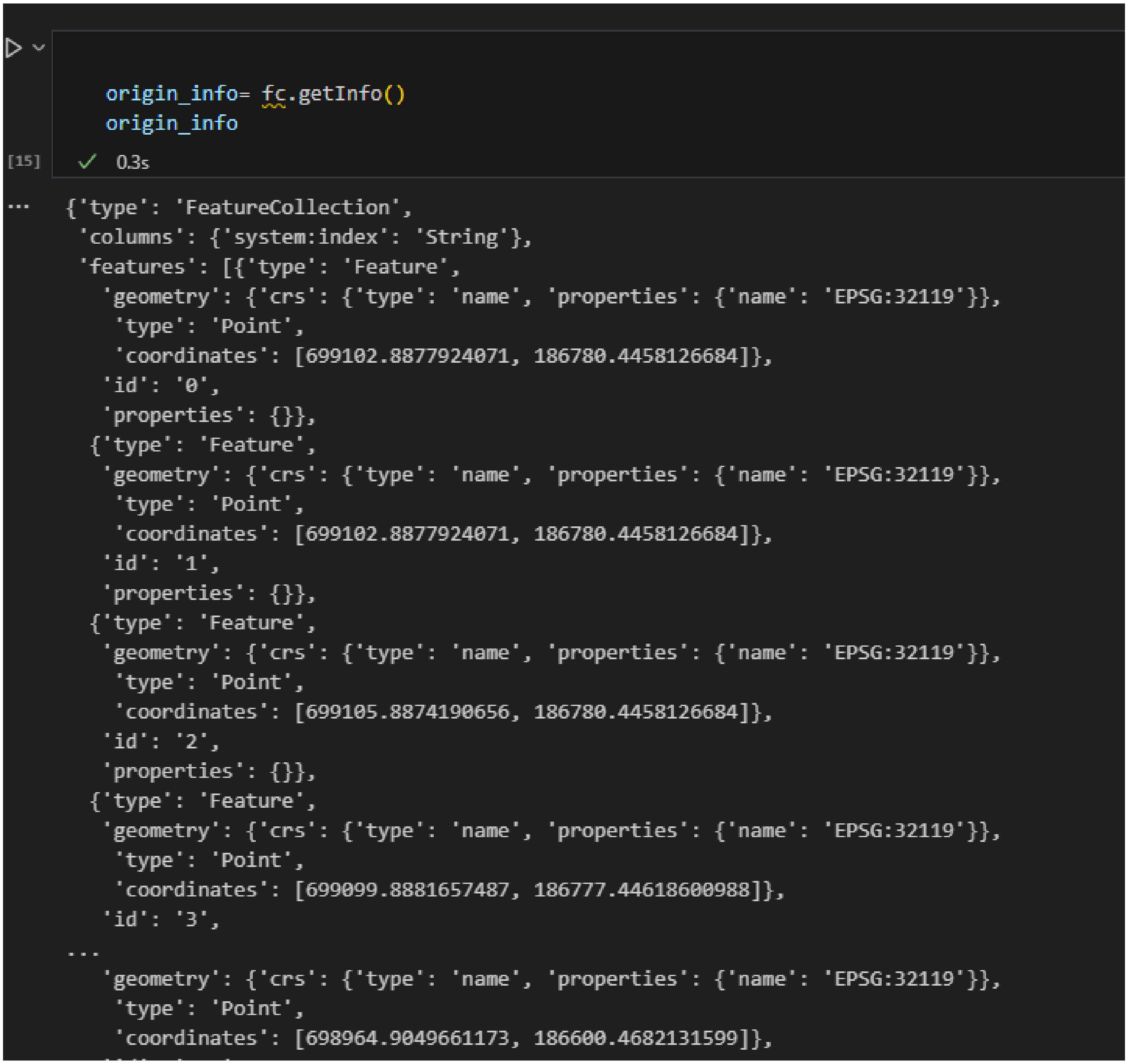


```
import pandas as pd
# Load the CSV file
csv_file = "boundary.csv"  # Replace with your file path
data = pd.read_csv(csv_file)
```

[4] ✓ 0.0s

```
geometrys=[ee.Geometry.Point([x,y], 'EPSG:32119') for x,y in zip(data['X'], data['Y'])]
```

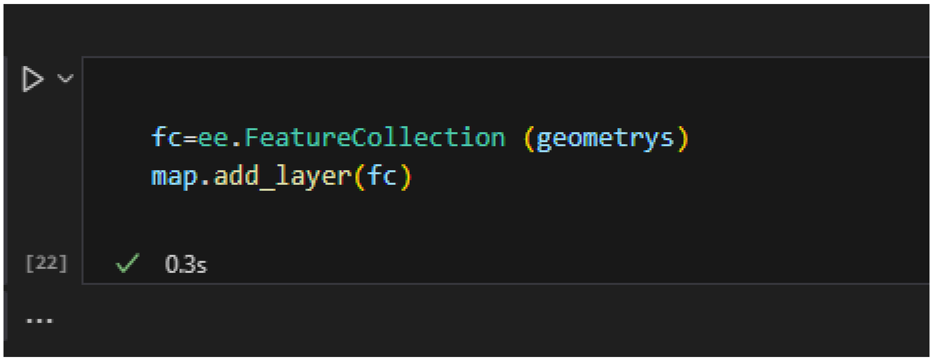
[11] ✓ 0.0s



```
origin_info= fc.getInfo()
origin_info
```

[15] ✓ 0.3s

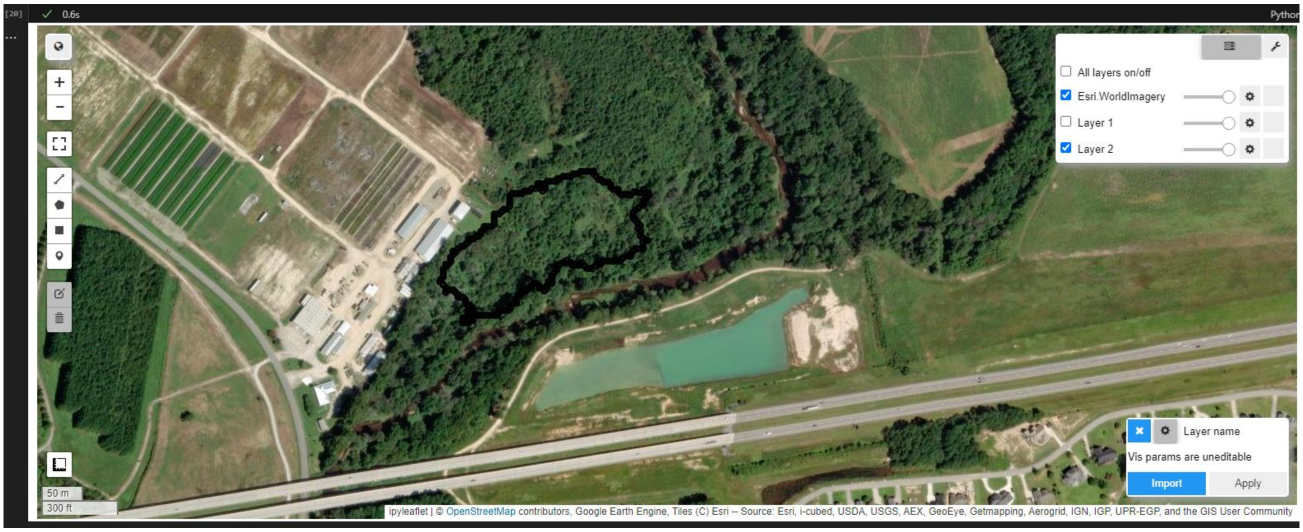
```
... {'type': 'FeatureCollection',
     'columns': {'system:index': 'String'},
     'features': [{'type': 'Feature',
       'geometry': {'crs': {'type': 'name', 'properties': {'name': 'EPSG:32119'}},
        'type': 'Point',
        'coordinates': [699102.8877924071, 186780.4458126684]},
       'id': '0',
       'properties': {}},
      {'type': 'Feature',
       'geometry': {'crs': {'type': 'name', 'properties': {'name': 'EPSG:32119'}},
        'type': 'Point',
        'coordinates': [699102.8877924071, 186780.4458126684]},
       'id': '1',
       'properties': {}},
      {'type': 'Feature',
       'geometry': {'crs': {'type': 'name', 'properties': {'name': 'EPSG:32119'}},
        'type': 'Point',
        'coordinates': [699105.8874190656, 186780.4458126684]},
       'id': '2',
       'properties': {}},
      {'type': 'Feature',
       'geometry': {'crs': {'type': 'name', 'properties': {'name': 'EPSG:32119'}},
        'type': 'Point',
        'coordinates': [699099.8881657487, 186777.44618600988]},
       'id': '3',
    ...
       'geometry': {'crs': {'type': 'name', 'properties': {'name': 'EPSG:32119'}},
        'type': 'Point',
        'coordinates': [698964.9049661173, 186600.4682131599]},
```

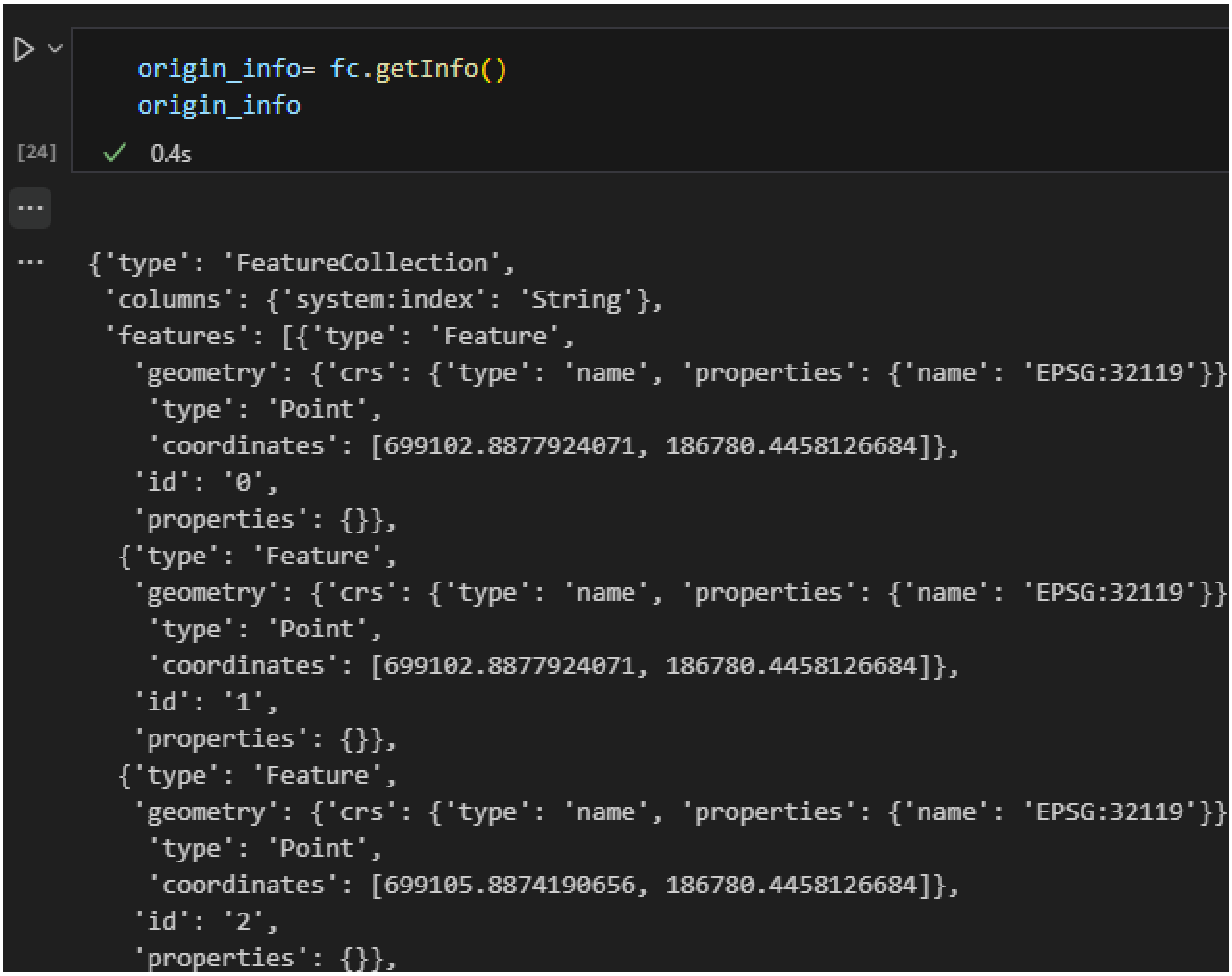


```
fc=ee.FeatureCollection (geometrys)
map.add_layer(fc)
```

[22] ✓ 0.3s

...

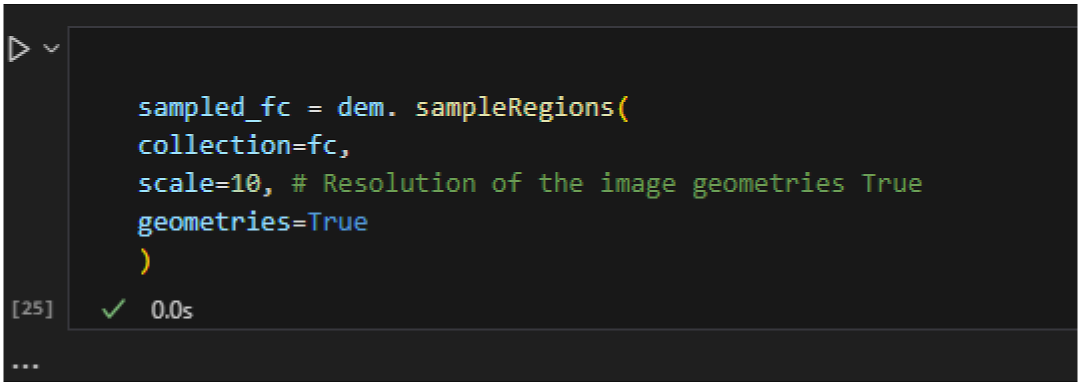

[20]
✓ 0.6s
Python
All layers on/off
Esri.WorldImagery
Layer 1
Layer 2
Layer name
Vis params are uneditable
Import
Apply
50 m
300 ft
ipyleaflet | © OpenStreetMap contributors, Google Earth Engine, Tiles (C) Esri -- Source: Esri, i-cubed, USDA, USGS, AEX, GeoEye, Getmapping, Aerogrid, IGN, IGP, UPR-EGP, and the GIS User Community



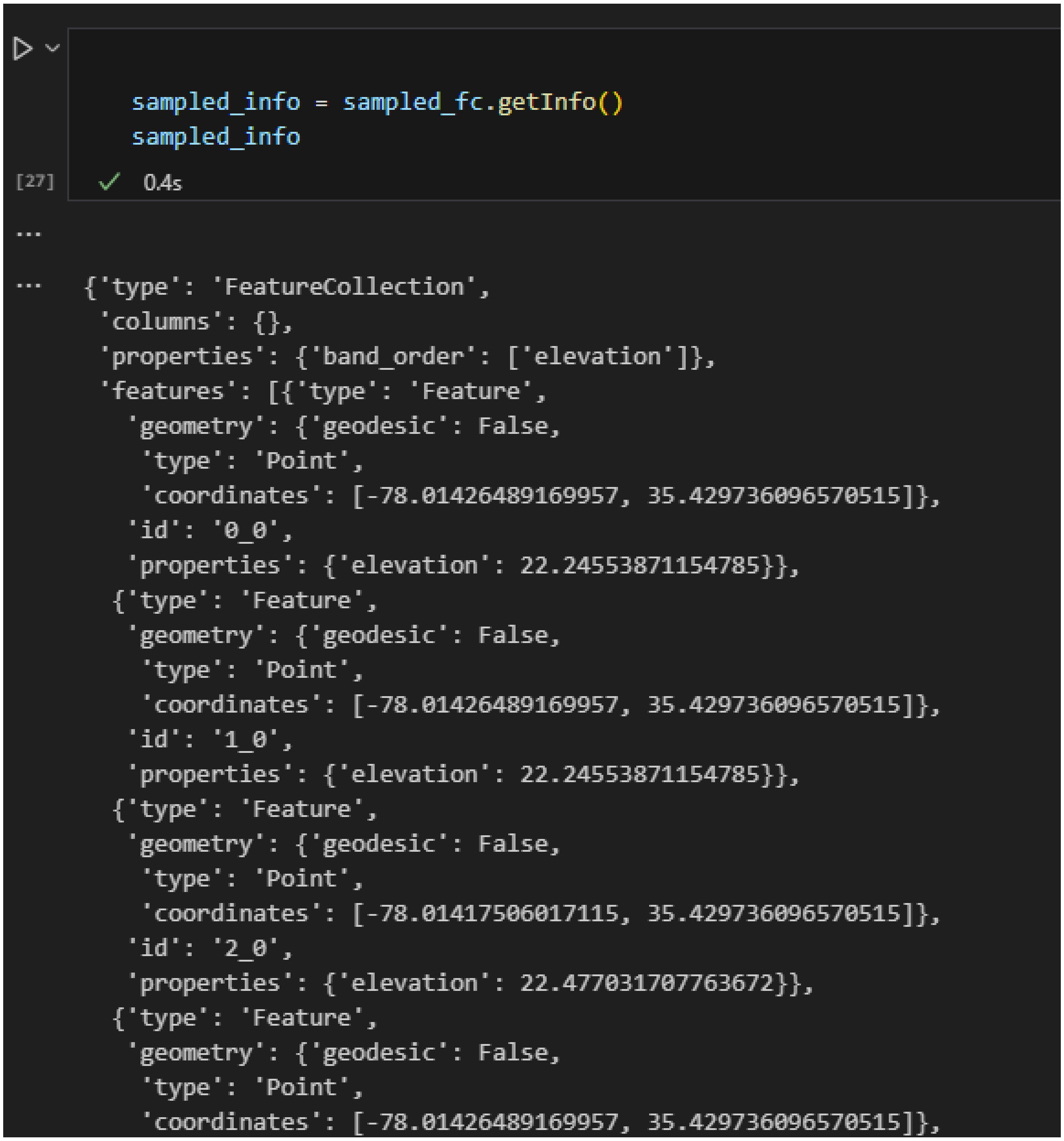
```
origin_info= fc.getInfo()
origin_info
```

[24] ✓ 0.4s

```
{'type': 'FeatureCollection',
 'columns': {'system:index': 'String'},
 'features': [{'type': 'Feature',
   'geometry': {'crs': {'type': 'name', 'properties': {'name': 'EPSG:32119'}}
    'type': 'Point',
    'coordinates': [699102.8877924071, 186780.4458126684]},
   'id': '0',
   'properties': {}},
  {'type': 'Feature',
   'geometry': {'crs': {'type': 'name', 'properties': {'name': 'EPSG:32119'}}
    'type': 'Point',
    'coordinates': [699102.8877924071, 186780.4458126684]},
   'id': '1',
   'properties': {}},
  {'type': 'Feature',
   'geometry': {'crs': {'type': 'name', 'properties': {'name': 'EPSG:32119'}}
    'type': 'Point',
    'coordinates': [699105.8874190656, 186780.4458126684]},
   'id': '2',
   'properties': {}},
```

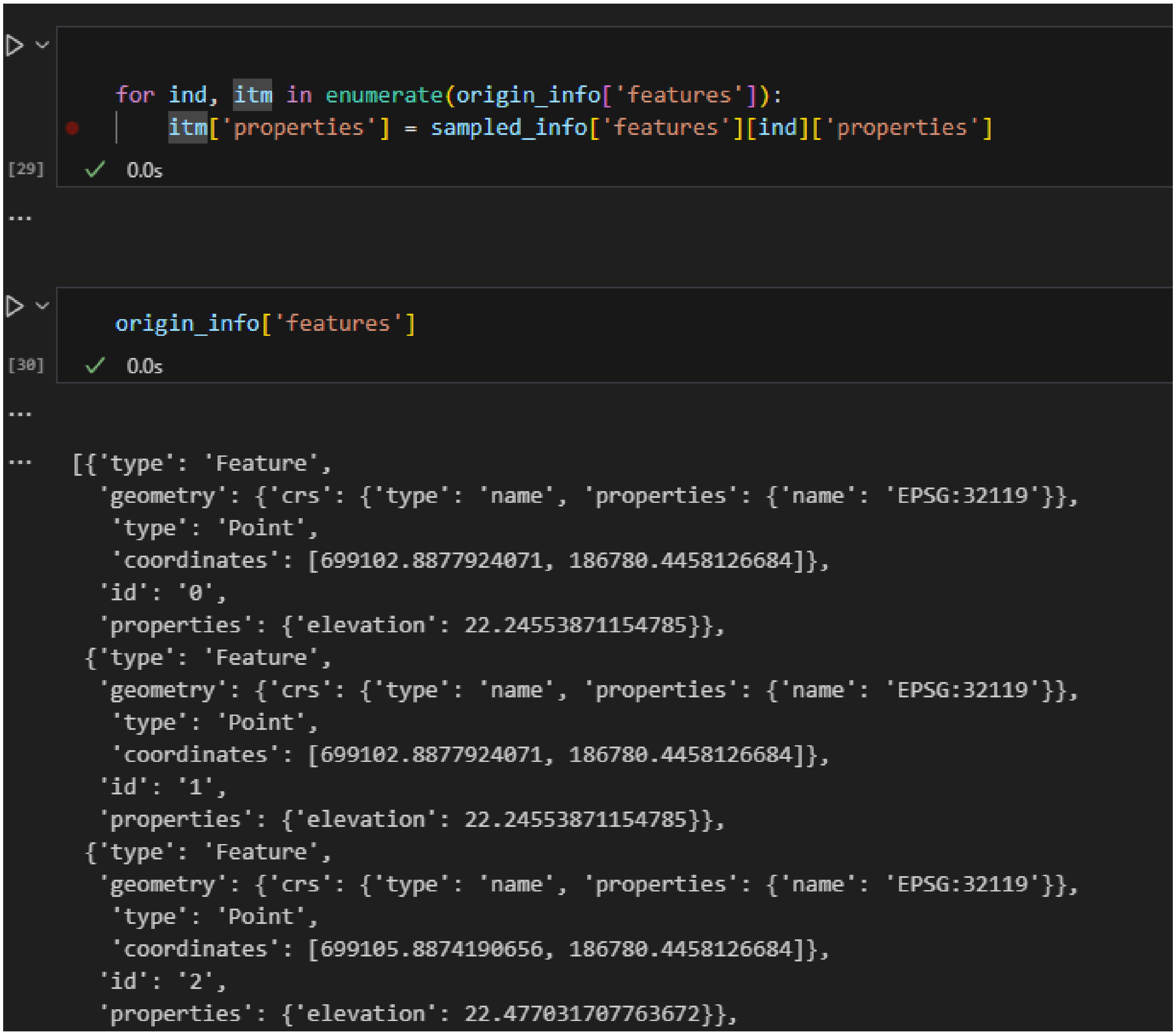


```
sampled_fc = dem. sampleRegions(
collection=fc,
scale=10, # Resolution of the image geometries True
geometries=True
)
```



```
sampled_info = sampled_fc.getInfo()
sampled_info
```

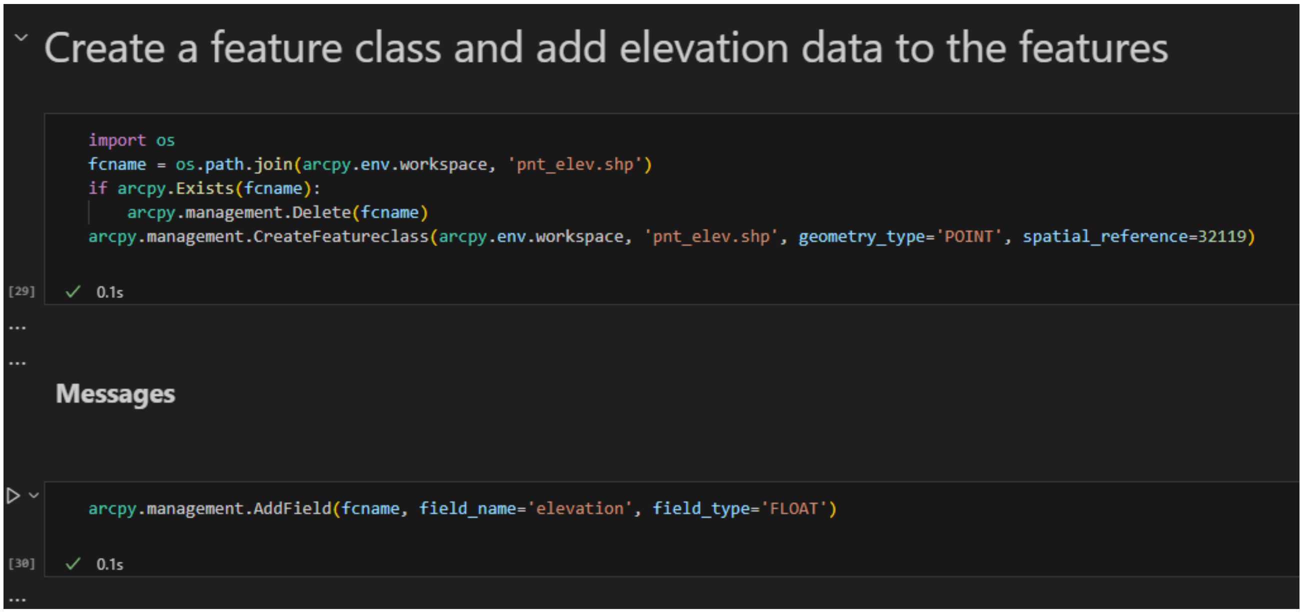
```
... {'type': 'FeatureCollection',
     'columns': {},
     'properties': {'band_order': ['elevation']},
     'features': [{'type': 'Feature',
       'geometry': {'geodesic': False,
        'type': 'Point',
        'coordinates': [-78.01426489169957, 35.429736096570515]},
       'id': '0_0',
       'properties': {'elevation': 22.24553871154785}},
      {'type': 'Feature',
       'geometry': {'geodesic': False,
        'type': 'Point',
        'coordinates': [-78.01426489169957, 35.429736096570515]},
       'id': '1_0',
       'properties': {'elevation': 22.24553871154785}},
      {'type': 'Feature',
       'geometry': {'geodesic': False,
        'type': 'Point',
        'coordinates': [-78.01417506017115, 35.429736096570515]},
       'id': '2_0',
       'properties': {'elevation': 22.477031707763672}},
      {'type': 'Feature',
       'geometry': {'geodesic': False,
        'type': 'Point',
        'coordinates': [-78.01426489169957, 35.429736096570515]},
```



```python
for ind, itm in enumerate(origin_info['features']):
    itm['properties'] = sampled_info['features'][ind]['properties']
```

```python
origin_info['features']
```

```
[{'type': 'Feature',
  'geometry': {'crs': {'type': 'name', 'properties': {'name': 'EPSG:32119'}},
   'type': 'Point',
   'coordinates': [699102.8877924071, 186780.4458126684]},
  'id': '0',
  'properties': {'elevation': 22.24553871154785}},
 {'type': 'Feature',
  'geometry': {'crs': {'type': 'name', 'properties': {'name': 'EPSG:32119'}},
   'type': 'Point',
   'coordinates': [699102.8877924071, 186780.4458126684]},
  'id': '1',
  'properties': {'elevation': 22.24553871154785}},
 {'type': 'Feature',
  'geometry': {'crs': {'type': 'name', 'properties': {'name': 'EPSG:32119'}},
   'type': 'Point',
   'coordinates': [699105.8874190656, 186780.4458126684]},
  'id': '2',
  'properties': {'elevation': 22.477031707763672}},
```



Create a feature class and add elevation data to the features

```python
import os
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)
```

[29] ✓ 0.1s

...

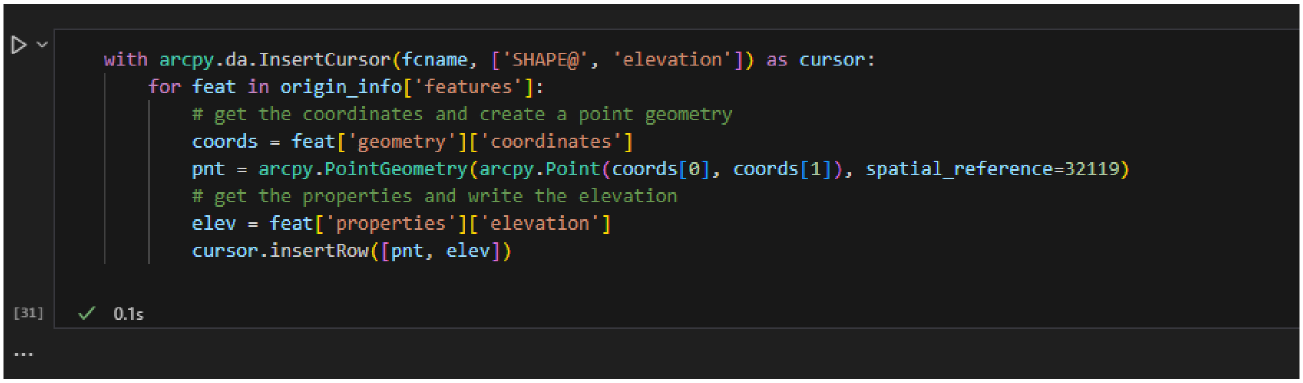
...

Messages

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

[30] ✓ 0.1s

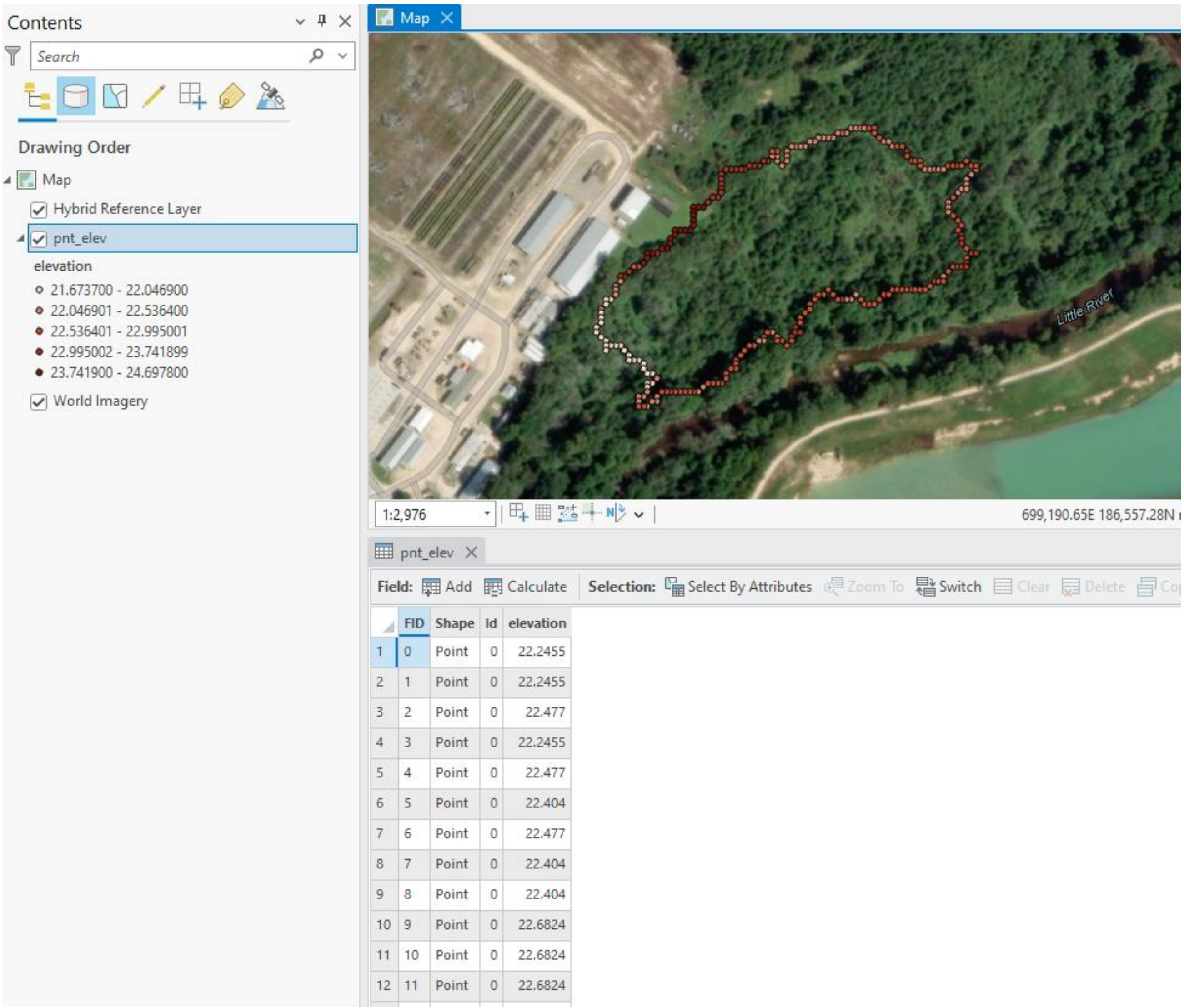
...



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

[31] ✓ 0.1s

...

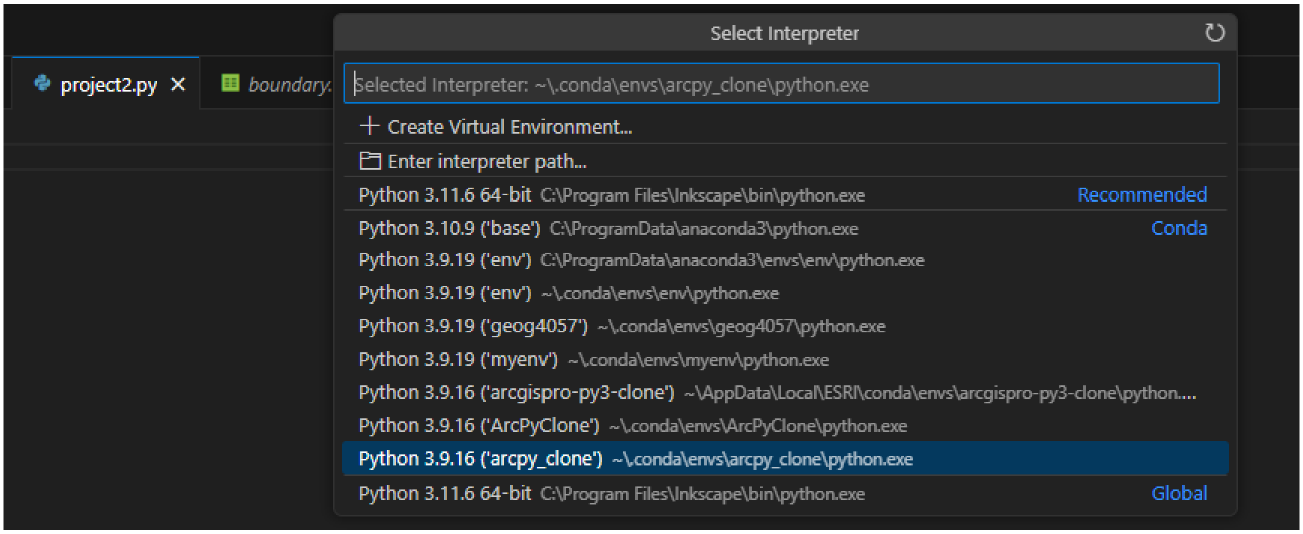


Drawing Order

- Map
  - ☑ Hybrid Reference Layer
  - ☑ pnt_elev
    - elevation
      - ● 21.673700 - 22.046900
      - ● 22.046901 - 22.536400
      - ● 22.536401 - 22.995001
      - ● 22.995002 - 23.741899
      - ● 23.741900 - 24.697800
  - ☑ World Imagery

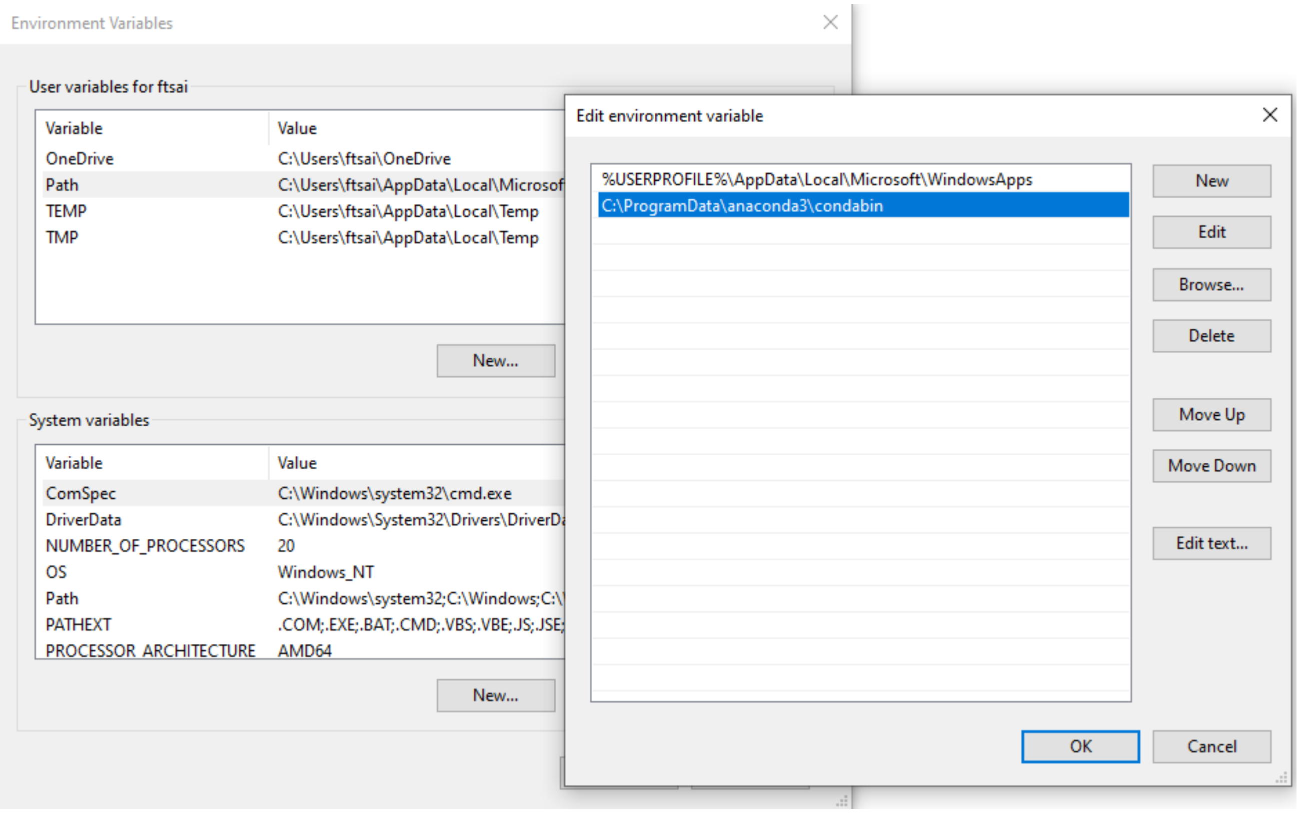
	FID	Shape	Id	elevation
1	0	Point	0	22.2455
2	1	Point	0	22.2455
3	2	Point	0	22.477
4	3	Point	0	22.2455
5	4	Point	0	22.477
6	5	Point	0	22.404
7	6	Point	0	22.477
8	7	Point	0	22.404
9	8	Point	0	22.404
10	9	Point	0	22.6824
11	10	Point	0	22.6824
12	11	Point	0	22.6824

Working with Python file project2.py



Selected Interpreter: ~\.conda\envs\arcpy_clone\python.exe

Python 3.11.6 64-bit	C:\Program Files\Inkscape\bin\python.exe	Recommended
Python 3.10.9 ('base')	C:\ProgramData\anaconda3\python.exe	Conda
Python 3.9.19 ('env')	C:\ProgramData\anaconda3\envs\env\python.exe	
Python 3.9.19 ('env')	~\.conda\envs\env\python.exe	
Python 3.9.19 ('geog4057')	~\.conda\envs\geog4057\python.exe	
Python 3.9.19 ('myenv')	~\.conda\envs\myenv\python.exe	
Python 3.9.16 ('arcgispro-py3-clone')	~\AppData\Local\ESRI\conda\envs\arcgispro-py3-clone\python....	
Python 3.9.16 ('ArcPyClone')	~\.conda\envs\ArcPyClone\python.exe	
Python 3.9.16 ('arcpy_clone')	~\.conda\envs\arcpy_clone\python.exe	
Python 3.11.6 64-bit	C:\Program Files\Inkscape\bin\python.exe	Global

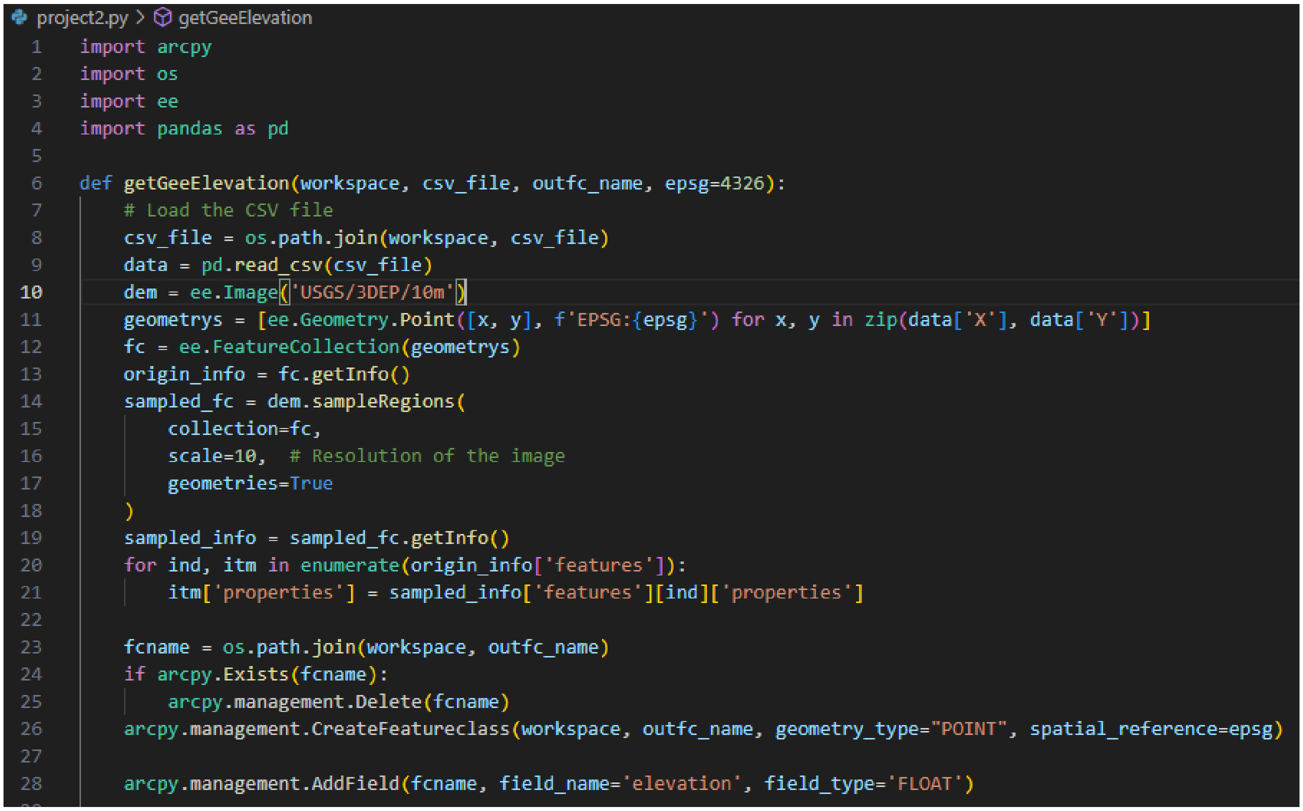


Variable	Value
OneDrive	C:\Users\ftsai\OneDrive
Path	C:\Users\ftsai\AppData\Local\Microsof
TEMP	C:\Users\ftsai\AppData\Local\Temp
TMP	C:\Users\ftsai\AppData\Local\Temp

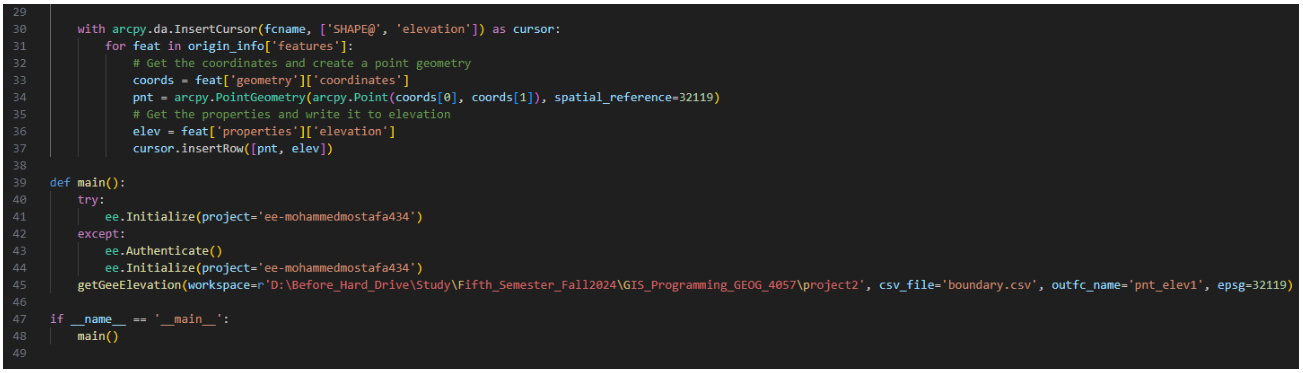
Variable	Value
ComSpec	C:\Windows\system32\cmd.exe
DriverData	C:\Windows\System32\Drivers\DriverDa
NUMBER_OF_PROCESSORS	20
OS	Windows_NT
Path	C:\Windows\system32;C:\Windows;C:\
PATHEXT	.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;
PROCESSOR_ARCHITECTURE	AMD64

```
%USERPROFILE%\AppData\Local\Microsoft\WindowsApps
C:\ProgramData\anaconda3\condabin
```

With hardcoding:

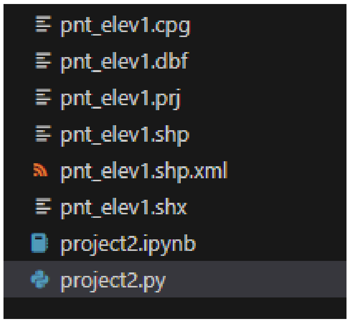


```python
import arcpy
import os
import ee
import pandas as pd

def getGeeElevation(workspace, csv_file, outfc_name, epsg=4326):
    # Load the CSV file
    csv_file = os.path.join(workspace, csv_file)
    data = pd.read_csv(csv_file)
    dem = ee.Image('USGS/3DEP/10m')
    geometrys = [ee.Geometry.Point([x, y], f'EPSG:{epsg}') for x, y in zip(data['X'], data['Y'])]
    fc = ee.FeatureCollection(geometrys)
    origin_info = fc.getInfo()
    sampled_fc = dem.sampleRegions(
        collection=fc,
        scale=10,  # Resolution of the image
        geometries=True
    )
    sampled_info = sampled_fc.getInfo()
    for ind, itm in enumerate(origin_info['features']):
        itm['properties'] = sampled_info['features'][ind]['properties']

    fcname = os.path.join(workspace, outfc_name)
    if arcpy.Exists(fcname):
        arcpy.management.Delete(fcname)
    arcpy.management.CreateFeatureclass(workspace, outfc_name, geometry_type="POINT", spatial_reference=epsg)

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



```python

    with arcpy.da.InsertCursor(fcname, ['SHAPE@', 'elevation']) as cursor:
        for feat in origin_info['features']:
            # Get the coordinates and create a point geometry
            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])

def main():
    try:
        ee.Initialize(project='ee-mohammedmostafa434')
    except:
        ee.Authenticate()
        ee.Initialize(project='ee-mohammedmostafa434')
    getGeeElevation(workspace=r'D:\Before_Hard_Drive\Study\Fifth_Semester_Fall2024\GIS_Programming_GEOG_4057\project2', csv_file='boundary.csv', outfc_name='pnt_elev1', epsg=32119)

if __name__ == '__main__':
    main()

```

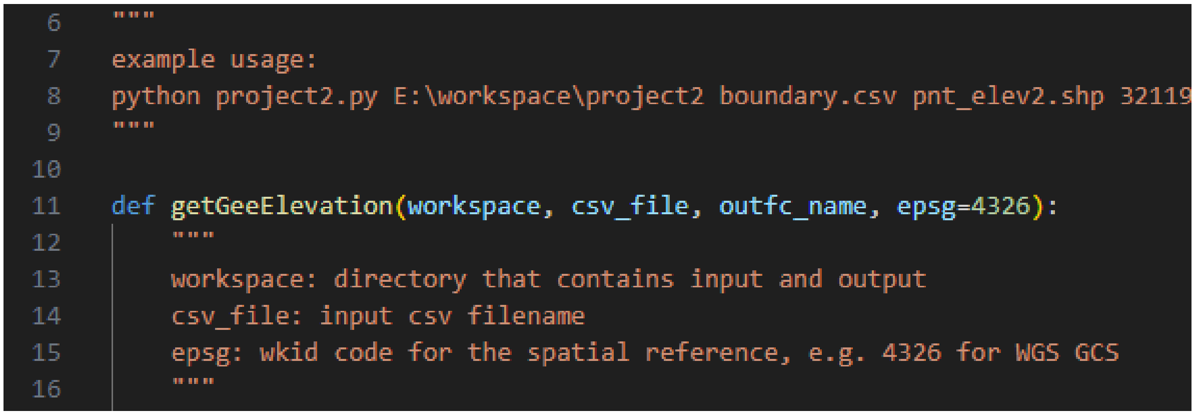
Output:



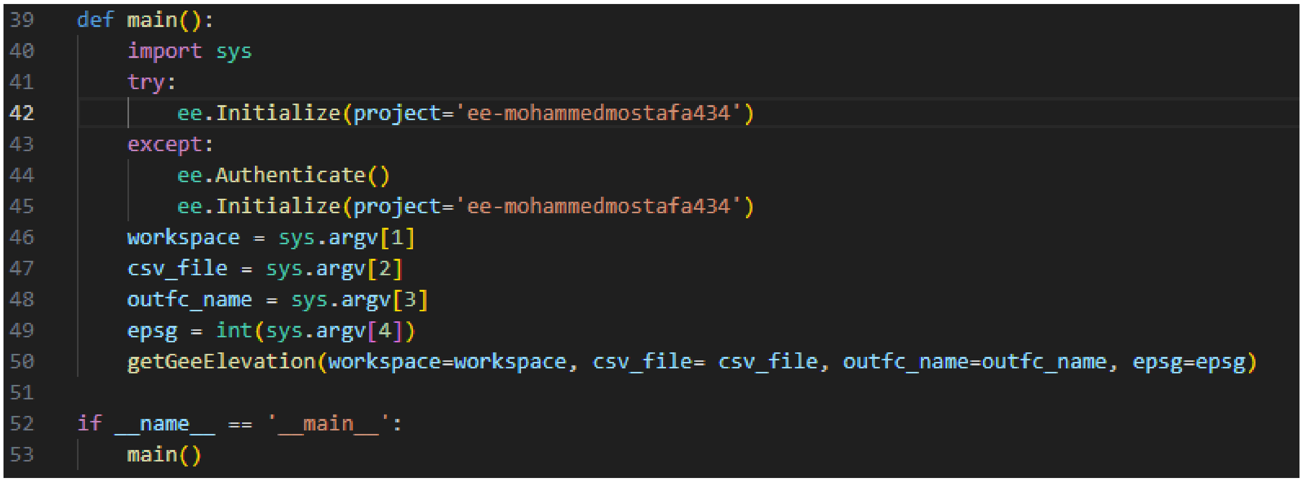
```
pnt_elev1.cpg
pnt_elev1.dbf
pnt_elev1.prj
pnt_elev1.shp
pnt_elev1.shp.xml
pnt_elev1.shx
project2.ipynb
project2.py
```

Without hardcoding (allowing the user to control inputs)

This is the changed part of the code:

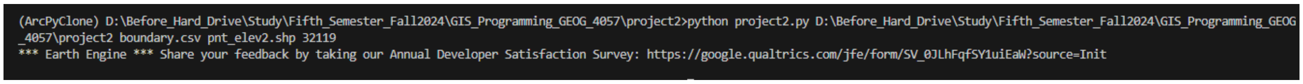


```python
"""
example usage:
python project2.py E:\workspace\project2 boundary.csv pnt_elev2.shp 32119
"""

def getGeeElevation(workspace, csv_file, outfc_name, epsg=4326):
    """
    workspace: directory that contains input and output
    csv_file: input csv filename
    epsg: wkid code for the spatial reference, e.g. 4326 for WGS GCS
    """
```



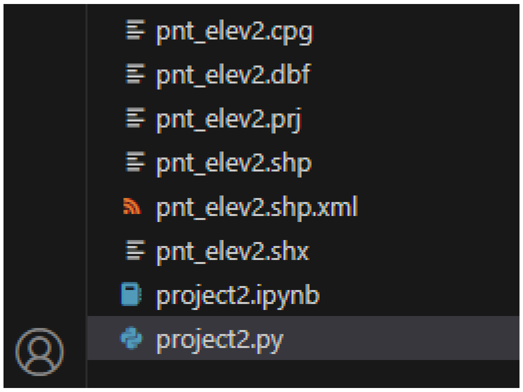
```python
def main():
    import sys
    try:
        ee.Initialize(project='ee-mohammedmostafa434')
    except:
        ee.Authenticate()
        ee.Initialize(project='ee-mohammedmostafa434')
    workspace = sys.argv[1]
    csv_file = sys.argv[2]
    outfc_name = sys.argv[3]
    epsg = int(sys.argv[4])
    getGeeElevation(workspace=workspace, csv_file= csv_file, outfc_name=outfc_name, epsg=epsg)

if __name__ == '__main__':
    main()
```

Run the code:



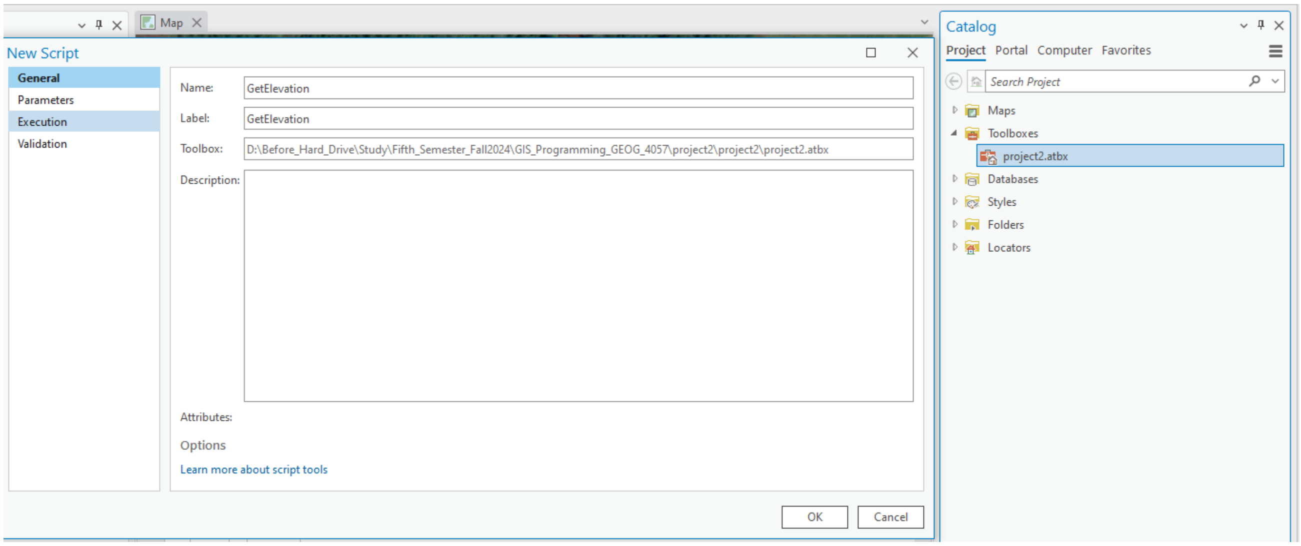
```
(ArcPyClone) D:\Before_Hard_Drive\Study\Fifth_Semester_Fall2024\GIS_Programming_GEOG_4057\project2>python project2.py D:\Before_Hard_Drive\Study\Fifth_Semester_Fall2024\GIS_Programming_GEOG
_4057\project2 boundary.csv pnt_elev2.shp 32119
*** Earth Engine *** Share your feedback by taking our Annual Developer Satisfaction Survey: https://google.qualtrics.com/jfe/form/SV_0JLhFqfSY1uiEaW?source=Init
```

Output(s):



```
pnt_elev2.cpg
pnt_elev2.dbf
pnt_elev2.prj
pnt_elev2.shp
pnt_elev2.shp.xml
pnt_elev2.shx
project2.ipynb
project2.py
```

Creating a tool:

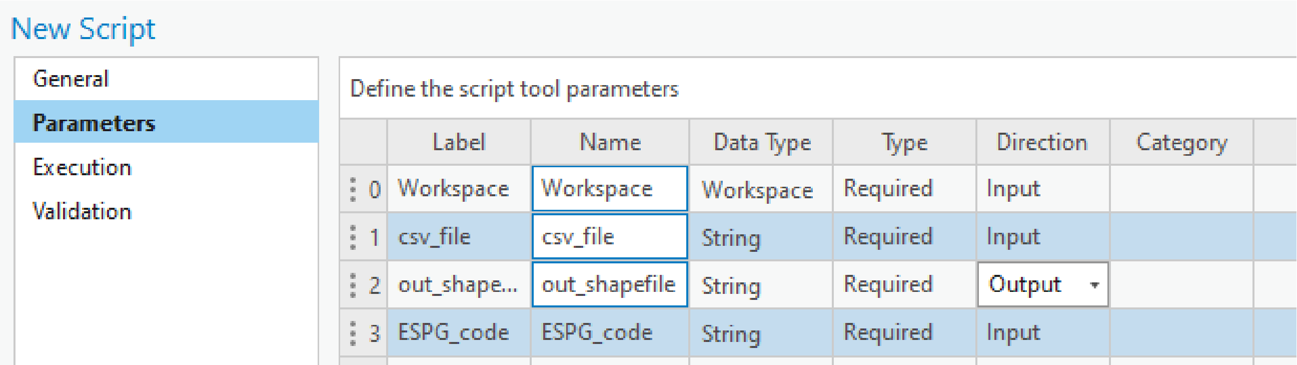


Name: GetElevation

Label: GetElevation

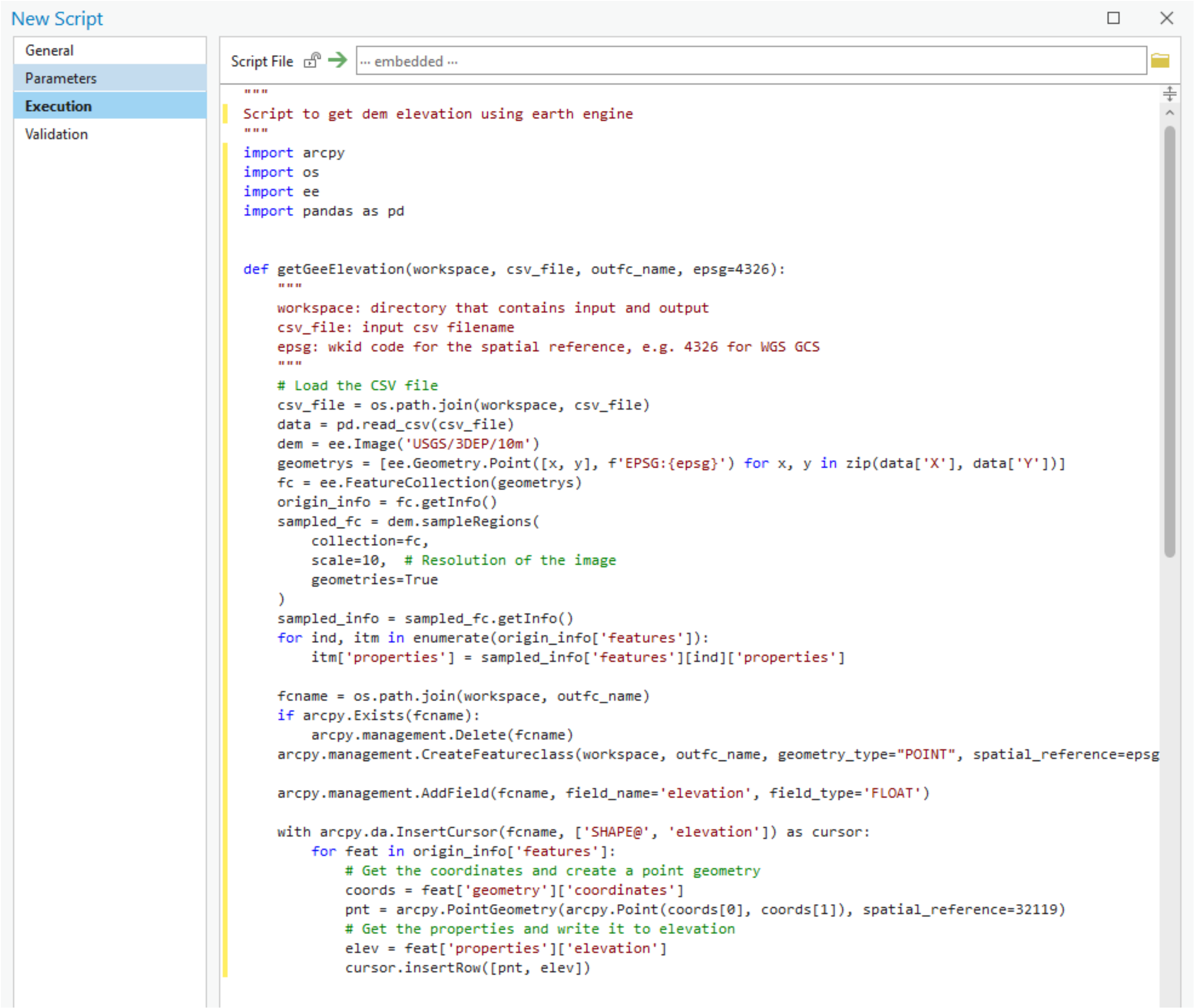
Toolbox: D:\Before_Hard_Drive\Study\Fifth_Semester_Fall2024\GIS_Programming_GEOG_4057\project2\project2\project2.atbx

Description:

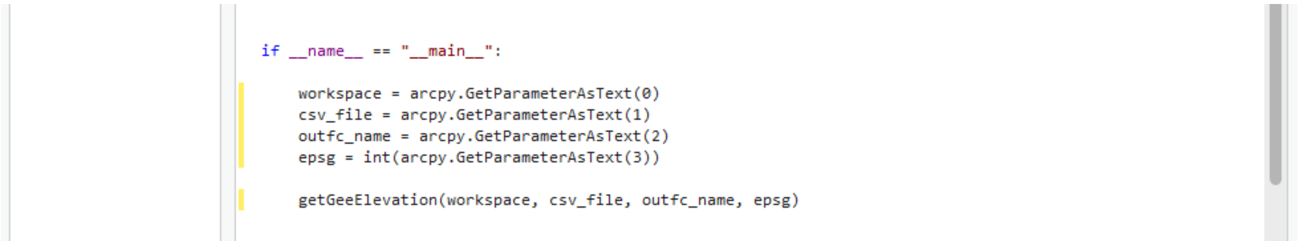


Define the script tool parameters

	Label	Name	Data Type	Type	Direction	Category
0	Workspace	Workspace	Workspace	Required	Input	
1	csv_file	csv_file	String	Required	Input	
2	out_shape...	out_shapefile	String	Required	Output	
3	ESPG_code	ESPG_code	String	Required	Input	

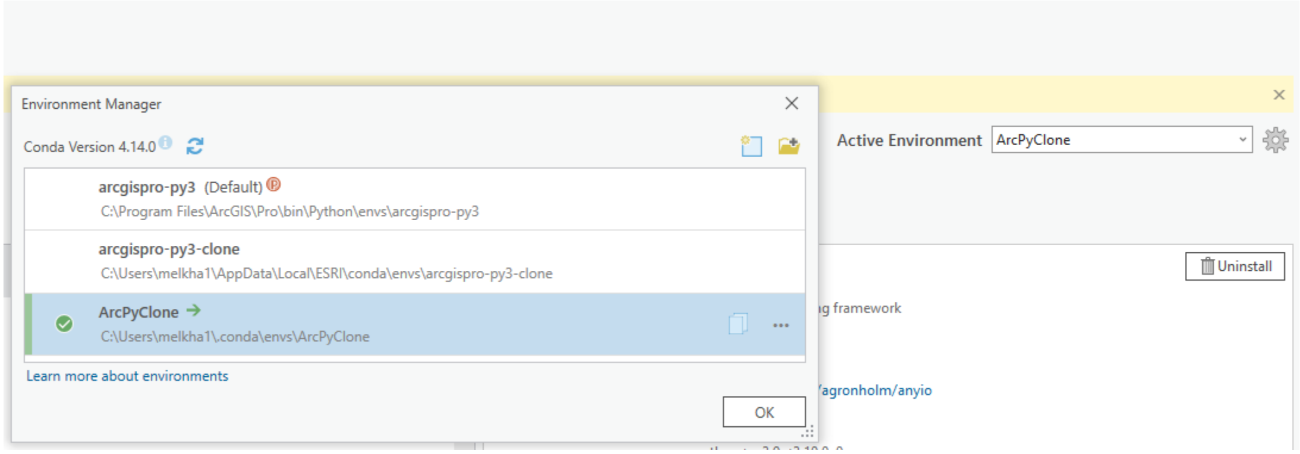


```
"""
Script to get dem elevation using earth engine
"""
import arcpy
import os
import ee
import pandas as pd


def getGeeElevation(workspace, csv_file, outfc_name, epsg=4326):
    """
    workspace: directory that contains input and output
    csv_file: input csv filename
    epsg: wkid code for the spatial reference, e.g. 4326 for WGS GCS
    """
    # Load the CSV file
    csv_file = os.path.join(workspace, csv_file)
    data = pd.read_csv(csv_file)
    dem = ee.Image('USGS/3DEP/10m')
    geometrys = [ee.Geometry.Point([x, y], f'EPSG:{epsg}') for x, y in zip(data['X'], data['Y'])]
    fc = ee.FeatureCollection(geometrys)
    origin_info = fc.getInfo()
    sampled_fc = dem.sampleRegions(
        collection=fc,
        scale=10,  # Resolution of the image
        geometries=True
    )
    sampled_info = sampled_fc.getInfo()
    for ind, itm in enumerate(origin_info['features']):
        itm['properties'] = sampled_info['features'][ind]['properties']

    fcname = os.path.join(workspace, outfc_name)
    if arcpy.Exists(fcname):
        arcpy.management.Delete(fcname)
    arcpy.management.CreateFeatureclass(workspace, outfc_name, geometry_type="POINT", spatial_reference=epsg

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

    with arcpy.da.InsertCursor(fcname, ['SHAPE@', 'elevation']) as cursor:
        for feat in origin_info['features']:
            # Get the coordinates and create a point geometry
            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])
```



```
if __name__ == "__main__":

    workspace = arcpy.GetParameterAsText(0)
    csv_file = arcpy.GetParameterAsText(1)
    outfc_name = arcpy.GetParameterAsText(2)
    epsg = int(arcpy.GetParameterAsText(3))

    getGeeElevation(workspace, csv_file, outfc_name, epsg)
```

Activate the cloned environment:



Environment Manager

Conda Version 4.14.0

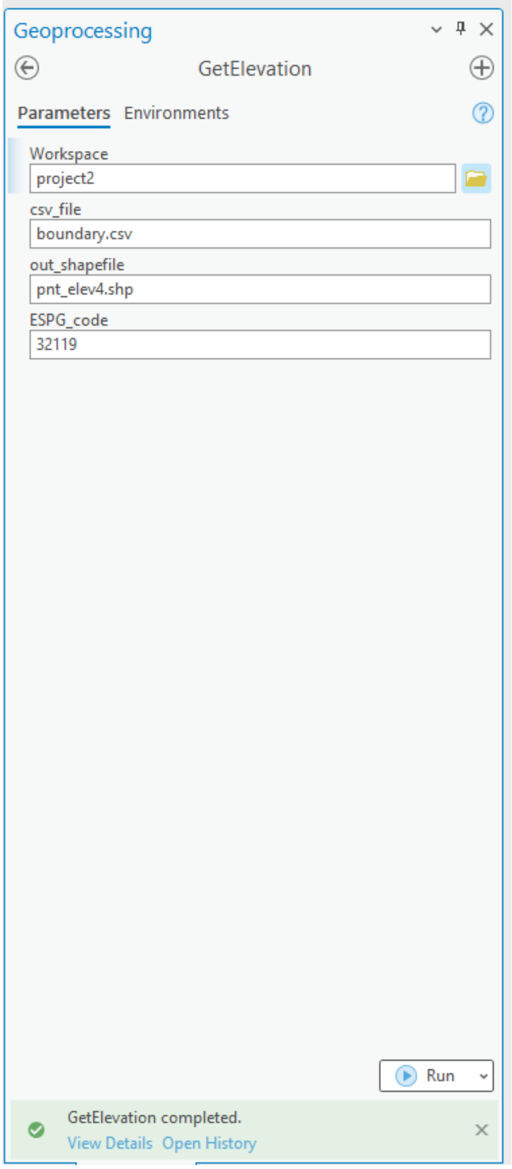
arcgispro-py3 (Default)
C:\Program Files\ArcGIS\Pro\bin\Python\envs\arcgispro-py3

arcgispro-py3-clone
C:\Users\melkha1\AppData\Local\ESRI\conda\envs\arcgispro-py3-clone

ArcPyClone
C:\Users\melkha1\.conda\envs\ArcPyClone

Learn more about environments

Active Environment: ArcPyClone



Geoprocessing

GetElevation

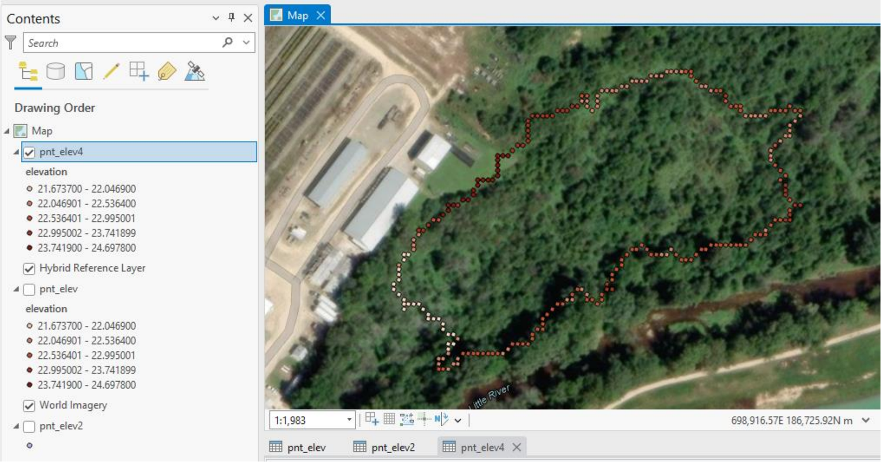
Workspace: project2

csv_file: boundary.csv

out_shapefile: pnt_elev4.shp

ESPG_code: 32119

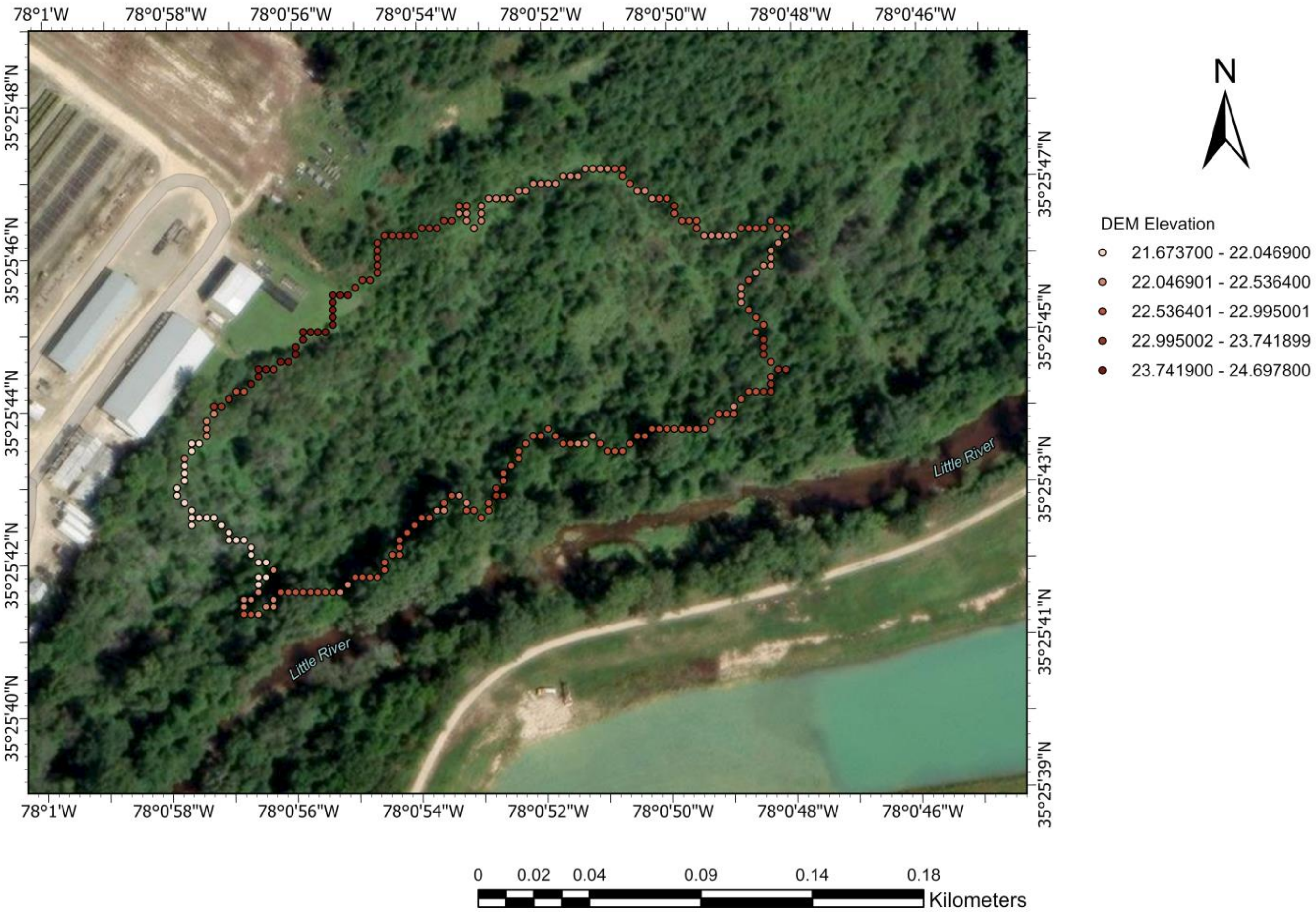
GetElevation completed.



Drawing Order

- Map
  - ☑ pnt_elev4
    - elevation
      - ○ 21.673700 - 22.046900
      - ● 22.046901 - 22.536400
      - ● 22.536401 - 22.995001
      - ● 22.995002 - 23.741899
      - ● 23.741900 - 24.697800
  - ☑ Hybrid Reference Layer
  - □ pnt_elev
    - elevation
      - ○ 21.673700 - 22.046900
      - ● 22.046901 - 22.536400
      - ● 22.536401 - 22.995001
      - ● 22.995002 - 23.741899
      - ● 23.741900 - 24.697800
  - ☑ World Imagery
  - □ pnt_elev2
    - ○

Layout



GitHub link

https://github.com/Mohammed-Elkharakany/project2_GEOG4057.git