This script is designed to convert a JSON file (no\_tax.json) containing spatial and attribute data into a shapefile (notax\_fc1.shp). It uses Python's json library for reading and processing the JSON file and the arcpy library from ArcGIS Pro to create and manage the shapefile.

1. Load the JSON File

* The json.load method reads the content of the no\_tax.json file and loads it into a Python dictionary called tax\_json.
* The JSON structure typically includes metadata (describing the fields) and data (rows of attributes and geometries).
* Key components:
  + - tax\_json['meta']['view']['columns']: Describes the fields in the dataset.
    - tax\_json['data']: Contains the rows of data, including geometry (spatial information) and attributes.

A computer screen shot of a program code

Description automatically generated

1. Display Field Names for Verification

* Prints the field names from the JSON file for verification.
* This step ensures you correctly identify field names and their structure before creating corresponding fields in the shapefile.

A computer screen shot of text

Description automatically generated

1. Convert Geometry Data

* Spatial data in the JSON is often stored in Well-Known Text (WKT) format, which is a standard textual representation of geometric shapes.
* The arcpy.FromWKT function converts these WKT strings into ArcPy geometry objects that can be written into a shapefile.

The geometry data is located in the 8th column (row[8]) of the JSON's data rows. Modify this index if your JSON structure differs.

A computer code on a black background

Description automatically generated

1. Define Workspace and Create Shapefile

* workspace is the folder where the shapefile will be created.
* fcname is the name of the shapefile.
* The script checks if the shapefile already exists using arcpy.Exists. If it does, it deletes it to avoid errors.
* A new shapefile is created using arcpy.management.CreateFeatureclass with:
  + Geometry type: POLYGON (can be adjusted based on your data).
  + Spatial reference: EPSG 4236 (WGS 84).

A black screen with white text

Description automatically generated

1. Inspect Existing Fields

Lists the existing fields in the newly created shapefile to ensure no conflicts arise when adding new fields.

A black screen with white text

Description automatically generated

1. Prepare Field Names and Types

* Filters out the the\_geom field (likely redundant, as geometry is handled separately).
* Sets field types (e.g., TEXT, LONG) for each field.
* Truncates field names to 10 characters to comply with shapefile conventions.
* Replaces spaces and periods in field names with underscores to ensure compatibility

A black screen with text

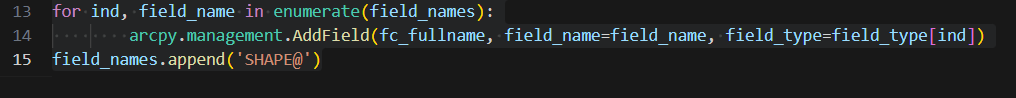
Description automatically generated

A screen shot of a computer code

Description automatically generated

1. Add Fields to the Shapefile

* Iterates through the processed field\_names and field\_type.
* Adds each field to the shapefile using arcpy.management.AddField.



1. Populate the shapefile

* Inserts rows of data into the shapefile.
* Add SHAPE@ to field\_names for handling geometry.
* For each row in the JSON:
  + Exclude the geometry column during attribute processing.
  + Append the geometry object to the row.
  + Insert the row into the shapefile using arcpy.da.InsertCursor.

A screen shot of a computer program

Description automatically generated

1. Finally developing a toolbox and use it in ArcPro

A screenshot of a computer map

Description automatically generated