

Reading Attribute Data

Working with GIS files

1

This video will describe how to read data in GIS attribute tables using arcpy cursors.

Working with GIS data

- Python cannot work with GIS files directly...
 - arcpy provides this capability
- Cursor tools provide access (read/edit) to feature and row-level data...
 - works with feature classes, rasters, dbase and info tables, etc.
 - access to attribute data
 - access to spatial data

2

Arcpy provides Python with the capability of reading, modifying, and creating GIS datasets.

Arcpy's cursor tools allow feature or tabular data to be read, modified, or created. The cursors work with attribute data as well as the spatial data that define features such as polygons, lines, and points. The cursors can read the attribute table for raster datasets but they cannot read or modify the pixels.

Cursor tools

- Use to retrieve the cursor object for GIS files
- Three types of cursors:
 - Search Cursor: read-only access to features / rows
 - Update Cursor: read/edit access to features / rows
 - Insert Cursor: create new features / rows
- Search and update cursors work with existing row objects.
- Insert cursors create new row objects
- Cursors only work on existing GIS files.

4

We can use one of arcpy's cursor tools to get the cursor object for a file.

There are 3 types of cursors. The **search cursor** can only read data from GIS files; the **update cursor** can read and modify data in GIS files; the **insert cursor** can only create and add data to a GIS file.

The search and update cursors work with existing row objects.

The insert cursor can only create new row objects.

All cursors must work with existing GIS files – the cursors do not create new files.

Creating search and update cursors



- Create the basic search cursor with:

```
rows = arcpy.SearchCursor("towns.shp")
```

↙ cursor variable ↘ include extension ↗



- Create the basic update cursor with:

```
rows = arcpy.UpdateCursor("towns.shp")
```



Pencil image at: <http://www.ciu.edu/~naggora/pencil.png>

5

The search cursor is created using arcpy's **SearchCursor** method. The file name specified must include the extension and the file type can be any feature class or table, including a raster attribute table.

The update cursor is created using arcpy's **UpdateCursor** method. The cursor objects in ArcGIS documentation are usually named **rows**.

The search cursor cannot modify data but the update cursor can.

Retrieving row objects



- Search and Update cursors retrieve existing row objects.
- With either cursor, retrieve row object with:

```
row = rows.next()
```

↑ ↓
row cursor
variable variable

6

If we continue with our file cabinet analogy, creating the cursor object is like opening the drawer in the file cabinet that contains the dataset.

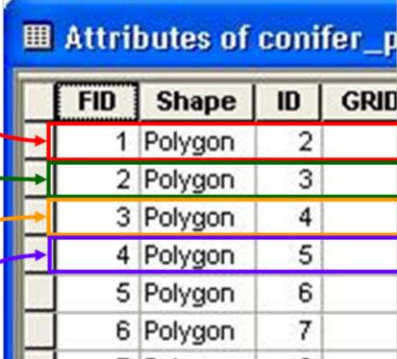
We can then use the **next** method of the cursor object to get the data for a single feature – in other words, the next method removes a folder from the file drawer. The data for a single feature is called the **row object**.

Retrieving row objects - illustration

```
rows = arcpy.SearchCursor("towns.shp")
```

- Iteratively use the next statement to retrieve successive rows.
- Rows retrieved in order (unless sorted by cursor).

```
row = rows.next()  
row = rows.next()  
row = rows.next()  
row = rows.next()
```



FID	Shape	ID	GRID
1	Polygon	2	
2	Polygon	3	
3	Polygon	4	
4	Polygon	5	
5	Polygon	6	
6	Polygon	7	

7

A new row object is retrieved each time the **next** method is repeated – this is the same as for the **readline** method for a text file. Rows are retrieved in the order that they are listed in the attribute table unless they are sorted by the cursor.

This example illustrates how the next method iteratively moves through the rows in a table.

Retrieving data from a row object



- Retrieve data from a given field with:

```
ID = row.getValue("FID")
```


name of field

8

The row object provides access to all the data that corresponds to a single feature.

The row object's **getValue** method will get the value, in the specified field, for the feature.

Retrieving row objects with a while loop

- Works in ArcGIS 9.x and ArcGIS 10...

```
rows = arcpy.SearchCursor("towns.shp")
row = rows.next() ← get 1st row

while row: ← continue until row is null
    print row.getValue("FID") ← get info from row
    row = rows.next() ← get next row
```

9

A while loop can be used to iteratively retrieve the row objects for each feature or row in a dataset. The set up of the loop is similar to one that is used to read through a text file.

After the cursor object (i.e. rows) is created, the cursor object's next method is used to get the 1st feature from the cursor.

The while loop condition is the row object – as long as the row object corresponds to data, then the loop will continue.

Inside the loop, the row object's getValue method can be used to retrieve values corresponding to specific fields in the attribute table.

At the end of the loop, the next method is used to get the next feature or row in the dataset.

Retrieving row objects with a *for* loop

- In ArcGIS 10, a *for* loop can iteratively retrieve row objects...

```
rows = arcpy.SearchCursor(table)
for row in rows:
    print row.getValue("FID")
```

create cursor

row variable

get info from row

10

A *for* loop can also be used to iteratively process a feature class or table. The *for* loop is usually preferable to a *while* loop when it can perform an equivalent operation because the *for* loop is more concise and cannot get stuck in an endless loop.

The cursor object is created using the `SearchCursor` or `UpdateCursor`.

The *for* loop header line includes the row variable and the cursor object.

Inside the loop, the feature can be processed as needed.

Removing cursor file locks

- Creating a cursor locks a file.
 - prevents accessing the file in ArcMap
- Delete cursor and row objects to remove file lock...
 - for the search and update cursors...

```
del rows, row
```

- for the insert cursor...

```
del rows, newRow
```

11

A cursor locks a file and can prevent the file from being viewed in ArcMap.

Use the **del** function to delete cursor and row objects when done with them – this will release any associated file locks.

Example script: reading data

```
# create search cursor...
rows = arcpy.SearchCursor("towns.shp")

# for each row...
for row in rows:

    # get ID value from field...
    ID = row.getValue("FID")

    # print ID...
    print ID

# delete cursor and row objects...
del rows, row
```

12

This slide shows an example script for reading a shapefile. This script will print each of the FID values associated with the features in the “towns.shp” file.

The geometry object

- Contains all spatial data for features...
 - area
 - length / perimeter
 - coordinates of center
 - coordinates of vertices
 - extent
- Geometry object stored under “Shape” field
- Retrieve with...

```
geom = row.getValue("Shape")
```



13

Each feature has a geometry object associated with it which contains all the spatial information for the feature. The spatial information includes:

- The area and perimeter of polygon features
- The length of line features
- The coordinates of the feature centroid
- The coordinates of vertices that define the feature’s shape
- The feature extent which includes the minimum and maximum X and Y coordinates of the feature vertices.

The geometry object is stored in the **Shape** field of the attribute table and is retrieved using the row object’s **getValue** method on the “Shape” field.

Geometry object properties

- Get area with...

geom.area ← number

- Get length / perimeter with...

geom.length ← number

- Get centroid point object with...

pnt = geom.centroid → x = pnt.X
y = pnt.Y

- Get extent object with...

extentObj = geom.extent → minX = extentObj.XMin
minY = extentObj.YMin
maxX = extentObj.XMax
maxY = extentObj.YMax



14

Certain spatial information can be retrieved from the geometry object's properties. The area property can be retrieved for polygon features.

The length property can be retrieved for polygon and line features. For polygons features, the length property gives the perimeter.

The centroid property returns the point object corresponding to the center of a polygon, the midpoint of a line, or the location of a point. The X and Y coordinates can be retrieved as properties of the point object.

The extent property of the geometry object returns an extent object. The extent object contains a set of coordinates that define the smallest rectangle that fully contains a feature. These coordinates are retrieved as properties of the extent object.