This video will discuss how feature geometry data are stored in a GIS dataset and how these data can be read using arcpy's cursor tools.
A feature’s geometry can be thought of as the blueprints needed to draw the feature.

All features are defined by their vertices and all properties of a feature’s shape are calculated based on the vertices. When viewing features, GIS software simply connects the dots following a set of rules that is determined by the type of features in the dataset.

There are 3 basic types of features that can be included in GIS datasets: points, lines, and polygons.
Last week, we learned that the geometry object contains all the spatial information for a given feature – this includes the feature’s geometry.

Recall that the geometry object is retrieved from a row object by using the row object’s getValue method of the “Shape” field.

The vertices that define a feature’s shape can be retrieved by using the geometry object’s `getPart` method. This method will return an array, for polygons and lines features, and a point object for point features.
Arrays – just another list

- Arrays organize the data...
- Work just like Python lists – same indexing and methods apply.
- Stores other arrays or point objects.
- Feature arrays contain data for a feature.
- Part arrays contain data for a part of a feature.

An array is a type of data collection in Python.

Arrays are similar to lists in the way that items are retrieved and added.

Arrays returned by the `getPart` method will either contain other arrays or point objects.

A feature array contains data for a feature.

A part array contains the data for a single part of a feature.
The feature array contains all the vertices for a single feature.

Features can contain one or more parts. Multi-part features have multiple separate parts that all correspond to a single FID and row in the dataset’s attribute table.

The feature array doesn’t exist for single-part point features – these features contain a single vertex which is stored in a point object.

For multi-part point features, the feature array contains point objects.

For polygons and lines, the feature array contains one or more part arrays.

To get the feature array, use the geometry object’s `getPart` method.
The **part array** contains all the vertices for a single part of a feature.

The part array does not exist for any point features – the geometry of point features is not complex enough to need a part array.

For lines and polygons, the part array will contain point objects. These point objects correspond to the vertices of a single contiguous feature part.

The part array can be retrieved from the feature array by specifying the feature array name followed by the position of the part in the array.

The part array can also be retrieved directly from the geometry object’s `getPart` method.
The **point object** stores the coordinates of a single point.

For a **single part** point, the point object is retrieved using the `getPart` method.

For more complex features, the point objects are retrieved from either the feature array (multi-part points) or the part array (polygons and lines).

The X and Y coordinates of a point are retrieved as properties from the point object.
This example script reads the X and Y coordinates from features in a single-part point shapefile. The SearchCursor is used to access the geometry object for each feature. The geometry object’s getPart method is used to retrieve the feature’s point object and the X and Y coordinates are retrieved from the point object.
This script demonstrates how to print the X and Y coordinates of features in a multi-part point shapefile. For multi-part point features, there are multiple points that correspond to a single FID value. The getPart method returns the feature array. For multi-part point features, the feature array contains point objects. A for loop can be used to retrieve each point object from the feature array and the coordinates can be retrieved from the point object.