

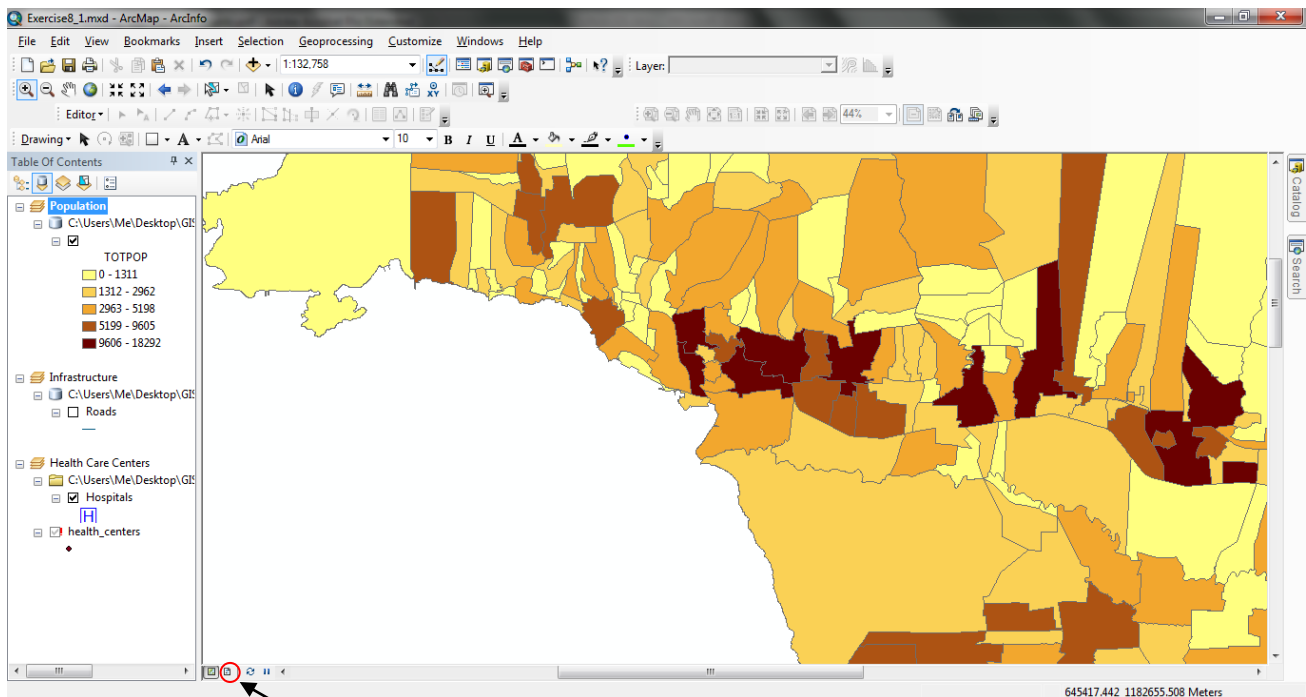
## Exercise 8: Working with Layout Elements

The ArcPy Mapping module, released with ArcGIS 10 provides a number of capabilities including the management of map documents and layers, printing, exporting, and ArcGIS Server publishing as well as map automation and the creation of PDF map books. This exercise will focus on accessing and manipulating layout elements. At the end of this exercise you will have learned the following:

- How to create lists of layout elements
- How to filter layout elements through layout class types
- Alter various layout element properties

### Step 1: Open the Crime.mxd Map Document file

- Start ArcMap and load Exercise 8\_1.mxd and switch to layout view



Click to switch to layout view

### Step 2: Open the Python Window

### Step 3: Reference the ArcPy Module

```
Python
>>> import arcpy.mapping
```

### Step 4: Reference the Current Map Document File

```
Python
>>> import arcpy.mapping
>>> mxdDoc = arcpy.mapping.MapDocument("CURRENT")
>>>
```

#### Step 5: Getting a List of Layout Elements

Now that you have a reference to the current map document file let's generate a list of layout elements and print them out to the screen.

Add the following code making sure that you add the colon at the end of the 'for' statement and indenting the print statement with the use of a tab. As you may recall from an earlier lesson, Python relies on indentation for code blocks.

```
Python
>>> import arcpy.mapping
>>> mxdDoc = arcpy.mapping.MapDocument("CURRENT")
>>> for elms in arcpy.mapping.ListLayoutElements(mxdDoc):
...     print elms.name
```

Python will print out the name of each element in the layout.

```
title
Scale Line
North Arrow
Legend
Population
```

#### Step 6: Filtering the Layout Elements by Type

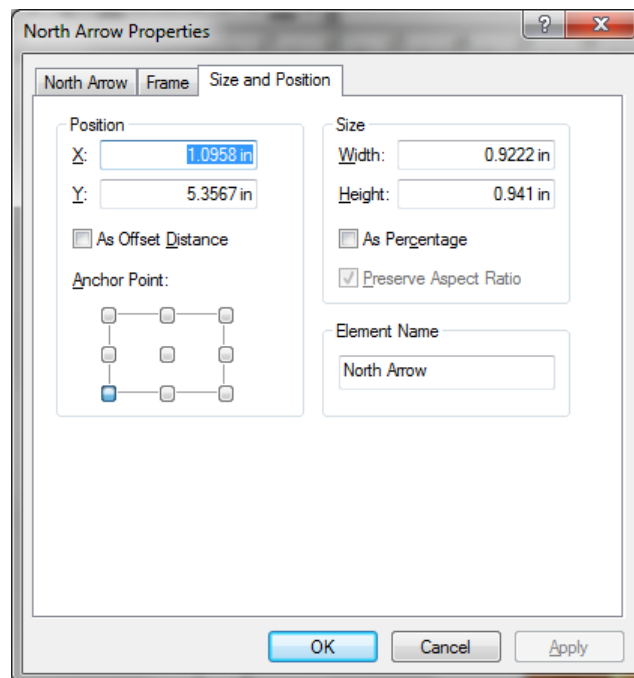
- There are many different types of layout elements including graphics, legends, pictures, text, maps, and pictures. When you return a list of layout elements you can restrict (filter) the types of elements that are returned. In this step you will alter the code you just wrote to return only legend elements.
- First, save the code you have just written to a file so that you don't have to re-type the entire code block. Right click inside the Python Window and select "Save As...". Navigate to a location on your computer of your choice and save the file as ListLayoutElements.py.
- Right click the Python Window again and select "Clear All"
- Right click the Python Window again and select "Load..." and open the ListLayoutElements.py file that you saved.
- Alter your code as follows:

```

Python
>>> import arcpy.mapping
... mxdDoc = arcpy.mapping.MapDocument("CURRENT")
... for elms in arcpy.mapping.ListLayoutElements(mxdDoc, "LEGEND_ELEMENT"):
...     print elms.name
...

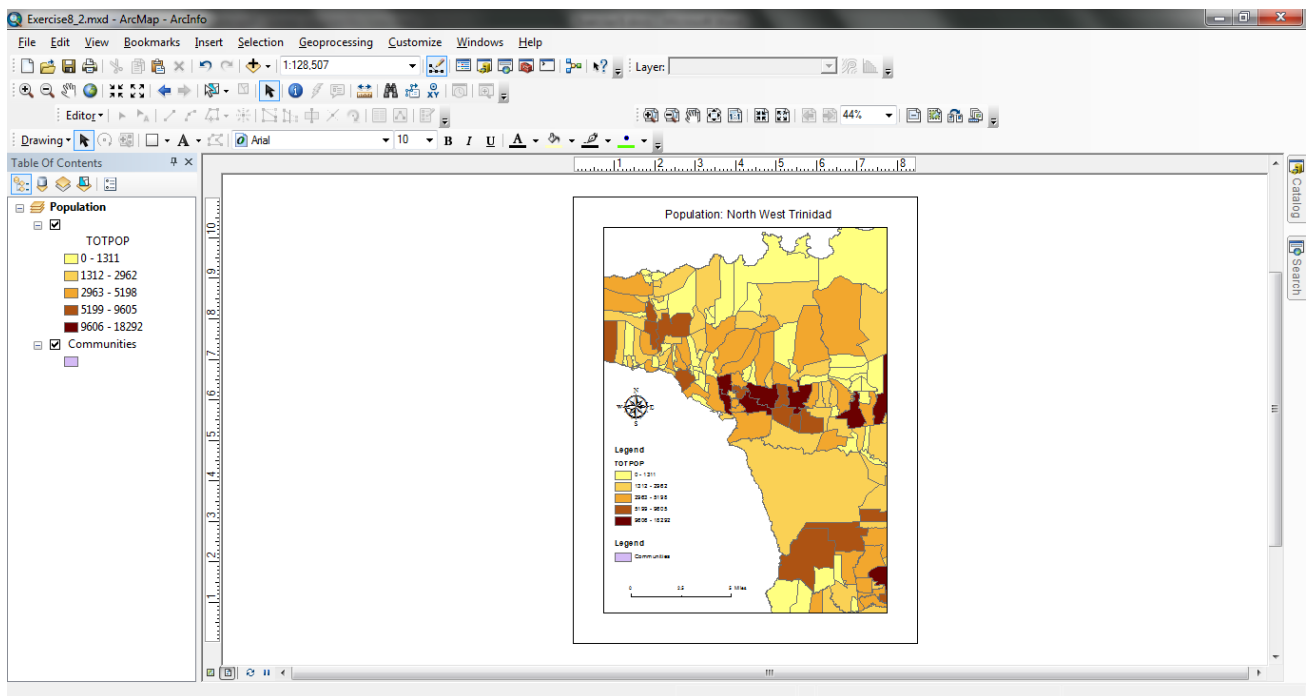
```

Run the code and you should see 1 Legend object returned. If there was another Legend object then 2 legend objects would have been returned and so forth. Try editing the script to Filter out the **scale bar**. So what if we had 2 legend elements with the same name? All elements in the ArcMap page layout have a 'name' property. This 'name' property can be accessed by selecting the element in the layout as seen below and then right clicking the element and selecting Properties. Right clicking the element and selecting Properties brings up the Properties dialog for the element. Notice that in the Size and Position tab there is an Element name text box. It is good practice to give each of your layout elements a unique name if you plan to access the elements through your Python scripts.



If an element does not have a default name as is the case with Text elements the 'name' property will be empty and when running code similar to the above it will appear as though an element does not exist. If two elements have the same name there is no way to specifically point the element of interest.

- Open Exercise8\_2.mxd



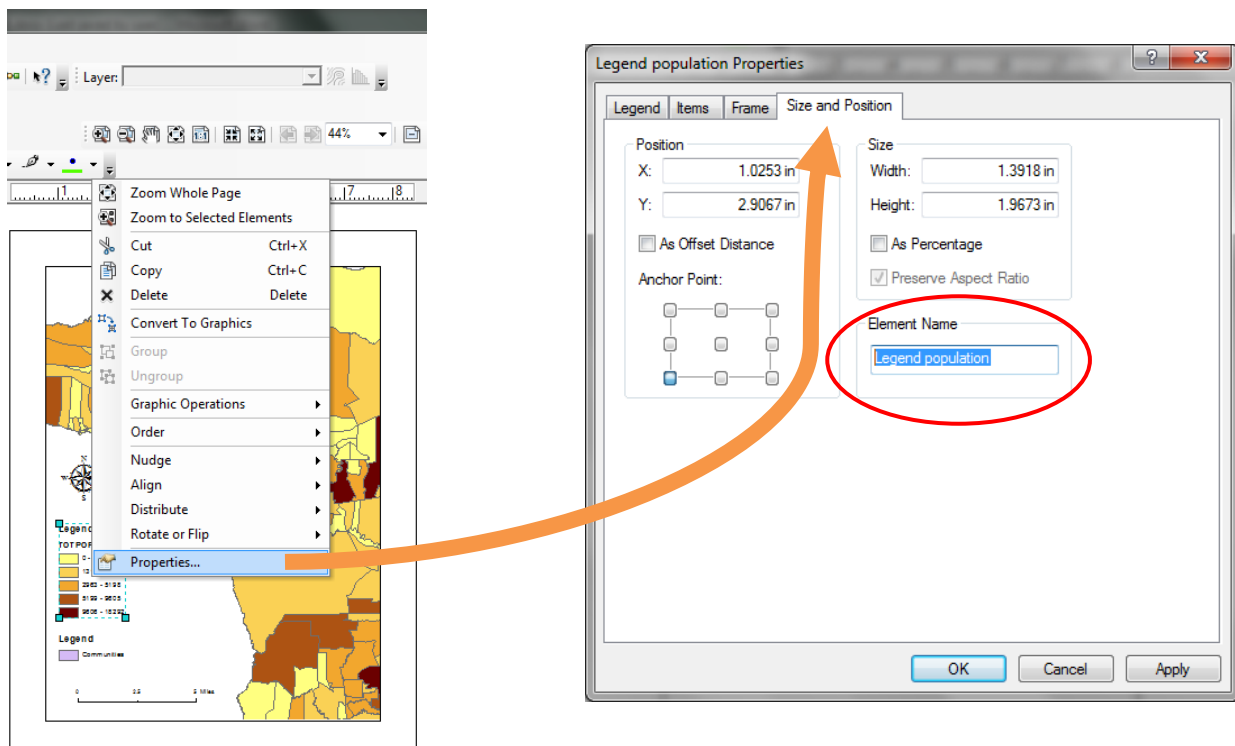
- Open the Python Window and type in the following

```
Python
>>> import arcpy.mapping
... mxdDoc = arcpy.mapping.MapDocument("CURRENT")
... for elms in arcpy.mapping.ListLayoutElements(mxdDoc, "LEGEND_ELEMENT") :
...     print elms.name
...
```

- Run the script. You should now see 2 layout legend elements with the name Legend.

#### Step 7: Filtering the Layout Elements by Type and name

- Change the legend element name of the element showing the population to "Legend population". Refer to Step 6 above if you forgot how to do this.



- Return to the Python Window and Clear the screen.
- Load the ListLayoutElements.py file again and alter your code as follows to include a wildcard that searches only for the Population Legend.

```
Python
>>> import arcpy.mapping
... mxdDoc = arcpy.mapping.MapDocument("CURRENT")
... for elms in arcpy.mapping.ListLayoutElements(mxdDoc,"LEGEND_ELEMENT","Legend population"):
...     print elms.name
...
```

This will return only a single legend element. Thus, the ListLayoutElements function provides you with two ways to filter the elements returned.

```
Python
>>> import arcpy.mapping
... mxdDoc = arcpy.mapping.MapDocument("CURRENT")
... for elms in arcpy.mapping.ListLayoutElements(mxdDoc,"LEGEND_ELEMENT","Legend population"):
...     print elms.name
...
Legend population
>>>
```

### Step 8: Alter the Properties of an Element

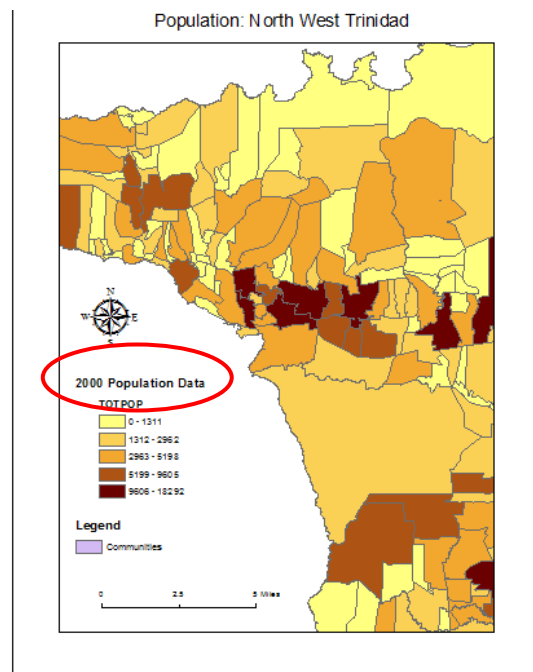
Each element in the page layout is returned as an instance of one of the element objects. In our case we have returned a [Legend](#) object. Each of these objects has various properties and methods that can be set through your Python code. We're going to change the title of our Population legend in this step.

- Clear the Python window and add the following code block.

```
import arcpy.mapping
mxdDoc = arcpy.mapping.MapDocument("CURRENT")
elms = arcpy.mapping.ListLayoutElements(mxdDoc, "LEGEND_ELEMENT", "Legend population")[0]
elms.title = "2000 Population Data"
```

- Run the code

What we've done here is create a new instance of LegendElement by referencing the first (and only) element returned by our call to ListLayoutElements. We then use the 'title' property on LegendElement to give our legend a new title.



Here we've made only a simple change to the Legend title. However, LegendElement provides other properties that enable it to be repositioned on the page layout as well as getting legend items and the parent data frame.

See <http://help.arcgis.com/en/arcgisdesktop/10.0/help/index.html#/LegendElement/00s300000041000000/> for more options for the LegendElement