Exercise 2: Working with the Python Window

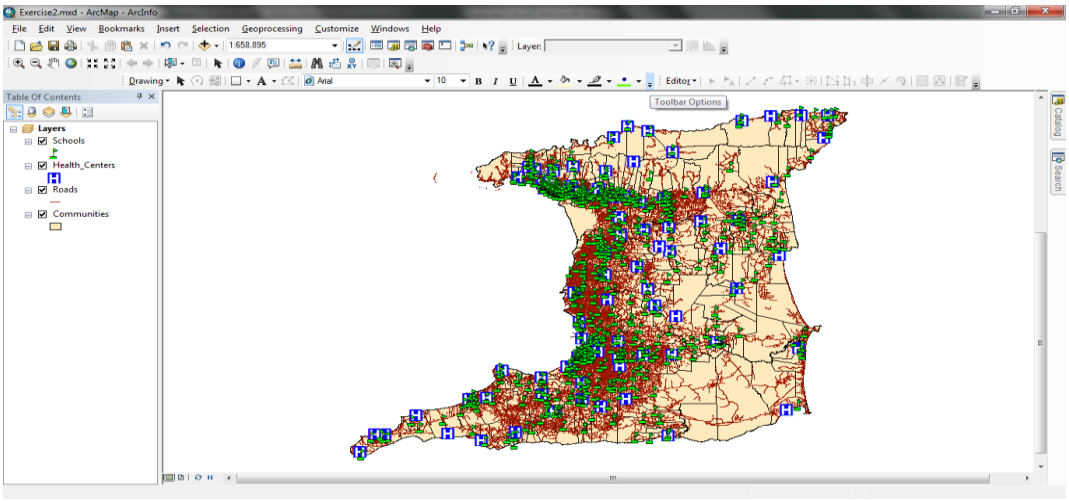
The Python Window is new to ArcGIS at version 10.0. It is an embedded, interactive Python window in ArcGIS that is great for testing small blocks of Python code, learning Python basics, building quick and easy workflows, and executing geoprocessing tools. The Python Window allows you to save the contents of the window to a Python script file on disk or load an existing Python script into the window. This makes you more efficient at performing geoprocessing tasks.

In this exercise you will learn some of the basic concepts of working with the Python Window.
At the end of this exercise you will have learned the following:

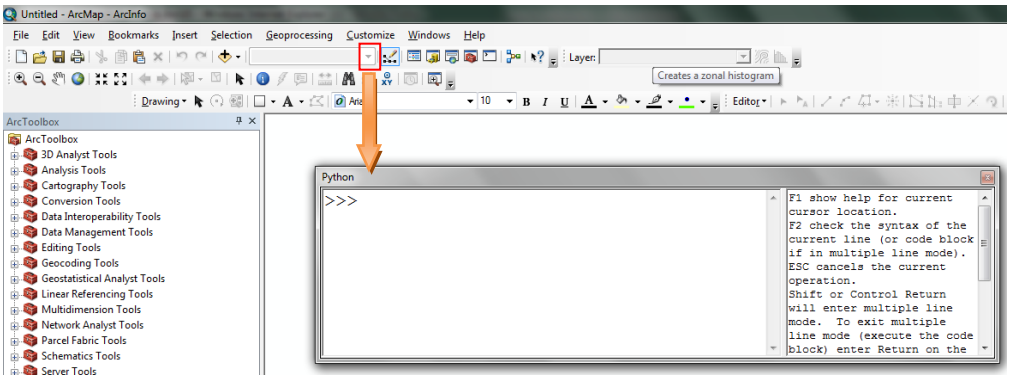
- How to open the Python Window in ArcGIS Desktop
- How to import the ArcPy site package
- Execute geoprocessing tools from the Python Window
- Use the code assist functionality provided by the window
- Save the contents of your code to a script file

Step 1: Start ArcMap and Load IntroPython.mxd File

- Start ArcMap and load Exercise2.mxd

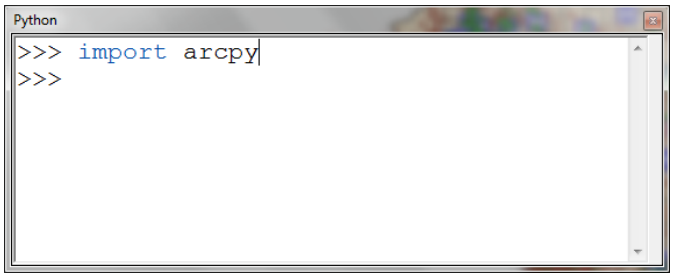


Step 2: Opening Python window

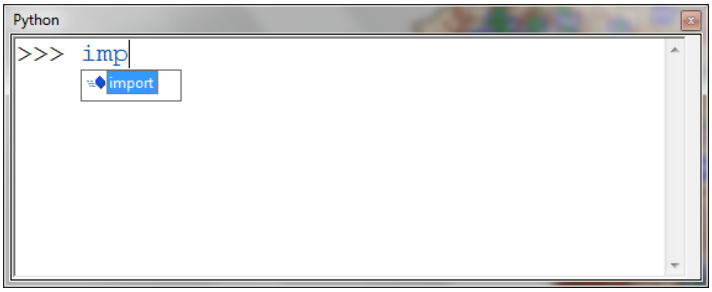


Step 3: Reference the ArcPy Module

- In the Python Window type the code you see below. This will reference the ArcPy Mapping module. You must always import the ArcPy site package before you do anything else.



- Notice that as you begin to type the 'import' statement the Python code assist window will appear. You can double click the term or simply hit 'Enter' to complete the word. This is a handy tool that helps you become more efficient and less prone to make errors in your code.



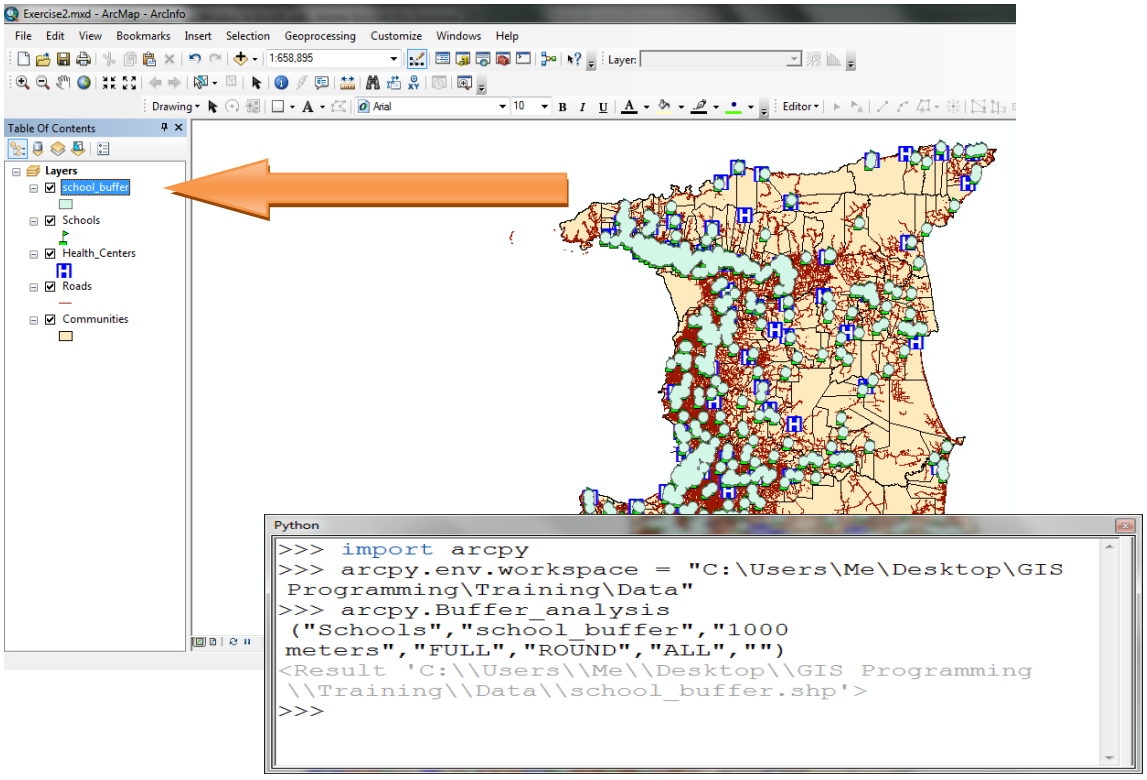
Step 4: Run a Geoprocessing Tool

In this step you will run the Buffer geoprocessing tool from the Python Window.

- First, set the current workspace as seen in the code below.
- Next we'll reference the Buffer tool found in the Analysis toolbox. Start by typing arcpy.Buffer_analysis and then an open parenthesis as seen in the code example below. Notice that the Python Window provides you with a help system that assists you in a couple different ways. First, notice in the help screen at the bottom that you are provided with help on what parameters need to be passed into the tool. The current parameter is always highlighted. What this is telling you is that the first parameter (in_features) is the input feature class that will be supplied to the Buffer tool. This brings us to the second way that code assist helps in calling a tool. Each parameter that you pass into the tool needs to be surrounded by double quotes. Here we want to specify that the Schools should be our input feature class. After you type the first double quote and then an 's', code assist will automatically find all feature classes in the table of contents that begin with 's'. You simply need to highlight and select the correct feature class. Select 'Schools'.
- Finish out the call to the Buffer tool as seen in the code below.

```
Python
>>> import arcpy
>>> arcpy.env.workspace = r"c:\Users\Me\Desktop\GIS
 Programming\Training\Data"
>>> arcpy.Buffer_analysis("Schools",schools_buffer","1000
 meters","FULL","ROUND","ALL","")
```

- Click Enter to execute the Buffer tool. Upon completion the new feature class will be added to the data frame and you will see a message in the help frame.



Step 5: Save Your Script

- The Python Window also gives you the ability to save your work to a Python script which can be run from an IDE such as PythonWin or re-imported to the Python Window at a later date to save time.
- Right click inside the Python Window and select “Save As…”. Navigate to where you’d like to save the script file and save.

- Similarly, you can load a pre-existing Python script by right clicking inside the Python Window and selecting “Load…”. Then you simply navigate to where the file was saved to open the code inside Python Window.