# GIS4207 arcpy.mapping Exercises

Download and follow the instructions in <https://github.com/viljoed/gis4x07/raw/master/GIS4x07_ExerciseSetup.docx> to create a Git repository for this exercise.

Create another folder parallel to “lab” called temp, i.e.

GIS4207\day06\lab\FredFandBarneyR\

GIS4207\day06\temp

This folder will be used for outputs generated by Mapping08.py. This temp folder should not be in the Git repositories.

## Setup

You have been provided with MappingEx.mxd that you will be using in your scripts. If you copy this into your exercise folder and have structured your folders properly, there should be no broken data sources.

## Mapping01.py

Use the slides to write a script to set the following MapDocument properties with the following values:

|  |  |
| --- | --- |
| Property | Value |
| title | arcpy.mapping module exercises |
| author | <your names> |
| credits | David Viljoen made me do it |
| summary | See Description |
| description | This document was used for practicing Python coding with the arcpy.mapping module. |
| tags | arcpy.mapping,python,gis4207 |

Copy and paste the values (except author) into your script to ensure they are precise.

NOTE: Use MapDocument("MappingEx.mxd") and run from PyScripter. Call the save method on MapDocument to save changes to the mxd. Open in ArcMap to see if the changes you made to the above properties worked (File > Map Document Properties). If you need to run the script again, create a new MXD first. This closes MappingEx.mxd and opens an empty MXD. Run the script in PyScripter and then open again in ArcMap. Repeat until you have set the MXD properties.

## Mapping02.py

This script will force the World DataFrame as the ActiveView. If the ActiveView is any other data frame, or is the Page Layout, make the World DataFrame as the ActiveView. Use MapDocument(“CURRENT”) as the MapDocument object and run from within the Python window in ArcMap (Use “Load …” and “Clear all” for testing). Do not call the save method on the MapDocument object.

NOTE: This exercise is as simple as it gets. If you start coding if blocks, etc, you are doing more work than required.

## SaveLayerVisibility.py

This script will save the layer name and its visibility for all layers in all data frames to a tab-delimited file called LayerVisibility.txt. Each line will contain the name of the DataFrame, name of the layer, and True or False for its visibility.

Use MapDocument("MappingEx.mxd") and run from PyScripter.

# LoadLayerVisibility.py

Edit the LayerVisibility.txt file to change the visibility of one or more layers. This script will read the contents of LayerVisibility.txt and set the visibility of the layers according the whether or not the layer name is associated with a True or False. Use MapDocument(“CURRENT”) as the MapDocument object and run from within the Python window in ArcMap. Make sure you call arcpy.RefreshActiveView() and arcpy.RefreshTOC() after making changes to layers.

# Mapping03.py

This script will print a report. Each tab-delimited row will contain the dataframe name, scale, and extent (XMin, Ymin, Xmax, Ymax). For example,

DataFrame Scale Extent

Canada 34028495 -122, 43, -46, 84

HINT: Use the properties of the Extent object to get the values you need.

Use MapDocument("MappingEx.mxd") and run from PyScripter.

## Mapping04.py

This script will go through all DataFrame’s and make all point layers visible. If the layer is in a group layer, the group layer must be made visible as well. All polygon and line layers will not be visible. Make sure you call arcpy.RefreshActiveView() and arcpy.RefreshTOC() after making changes to layers.  
  
Use MapDocument(“CURRENT”) as the MapDocument object and run from within the Python window in ArcMap. Do not save the MXD as part of, or after running, this script.

## Mapping05.py

This script will add the Continents.lyr and World Cities.lyr files to the Canada DataFrame using AUTO\_ARRANGE. Use MapDocument("CURRENT") for this script.

Use MapDocument(“CURRENT”) as the MapDocument object and run from within the Python window in ArcMap. Do not save the MXD as part of, or after running, this script.

## Mapping06.py

This script will remove Continents.lyr and World Cities.lyr layers from the Canada DataFrame.

Use MapDocument(“CURRENT”) as the MapDocument object and run from within the Python window in ArcMap. Do not save the MXD as part of, or after running, this script.

## Mapping07.py

Create a sub-folder from your scripts folder called mxdTemp. Make sure you have saved MappingEx.mxd. Use Explorer to **copy** and paste MappingEx.mxd into the mxdTemp folder. If you open this mxd, you will find all of the data sources are broken. In your scripts folder (one level up), write a script that will fix all of the broken data sources in the MapDocument. When you have this working, you can close the MappingEx.mxd in mxdTemp and then delete the mxdTemp folder and its contents. Make sure you don’t delete Mapping07.py Use MapDocument("MappingEx.mxd") and call the save() method on the MappingEx.mxd in the mxdTemp folder.

## Mapping08.py

This script will export each of the DataFrames to PDF in a folder called temp with the name of the DataFrame as the base name for the pdf. You will then use PDFDocumentCreate to create a PDF called AllMaps.pdf in temp. Use PDFDocument.appendPages() to add the DataFrame pdf’s to AllMaps.pdf.

Use MapDocument("MappingEx.mxd") and run in PyScripter.