Part 3: Carrying out specialized tasks

Notes

Contents

Chapter 10: Map scripting

Chapter 10 Overview

This is simply an overview of Chapter 10, which explains the arcpy.mapping module, which helps automate mapping, as well as printing and creating PDF map books.

I will only explain briefly the main concepts, often referring to the appropriate section and page number for further exploration into the topic, as needed.

~

It is possible to use the mapping module in a variety of ways, including:

- Finding a layer with a particular data source and replacing it with another data source.
- Modifying the display properties of a specific layer in multiple ArcMap documents.
- Generating reports that describe the properties of ArcMap documents, including data sources, layers with broken data links, information on the spatial reference of data frames and more.

A map document, or MXD (.mxd) can be opened using the MapDocument syntax, MapDocument(mxd_path).

There are a variety of map document properties that can be accessed. To see these options, refer to Section 10.4, p. 200.

One can list all of the data frames in a map by using the following command: ListDataFrame(map_document, {wild_card}).

One can do a variety of commands regarding the layers of a map, including deleting, moving, exporting, inserting, and more. To learn more, visit Section 10.6, pp. 203 - 207.

To learn more about fixing broken data sources, visit Section 10.7, pp. 208 - 213.

The following sections may be useful when you are trying to make a final map:

- 10.8 Working with page layout elements
- 10.9 Exporting maps
- 10.10 Printing maps
- 10.11 Working with PDFs
- 10.12 Creating map books

Chapter 11: Debugging and error handling

Chapter 11 Overview

This is simply an overview of Chapter 11, which explains debugging procedures and provides a review of the common Python errors.

I will only explain briefly the main concepts, often referring to the appropriate section and page number for further exploration into the topic, as needed.

~

Standard debugging procedures include:

- Carefully reviewing the content of error messages.
- Adding print statements to your script.
- Selectively commenting out code.
- Using a Python debugger.

All of these common procedures are outlined thoroughly in this chapter. As mentioned, this chapter is mostly a review, so I will leave this chapter open for reference, as needed.

Chapter 12: Creating Python functions and classes

This is simply an overview of Chapter 12, which explains how to create custom functions in Python that can be called elsewhere in the script or from another script.

Much of this is a review from the MITx 6.00.1x course. Therefore, I will only explain briefly the main concepts, often referring to the appropriate section and page number for further exploration into the topic, as needed.

~

- Functions are small blocks of code that perform a specific task
- There's a bunch of built-in functions that Python already contains, but you can make more as-needed
- If you want to call functions from other scripts, you have to import the module, and then do the following:

```
import <module>
<module>.<function>
```

- You can strategically organize code into modules, and then import modules as-needed. This can create a more organized system for your code, and can become especially useful if you will be referencing some functions in different documents, or your particular code is especially lengthy
- Classes are another way to store functions in a module, specifically if you want to store variables in a particular way
    - Calling functions in classes can get complicated
- Sometimes importing packages can make scripting easier and more efficient