Tips for Writing Python Scripts

Transcript

Hello, I'm Jeff Bigos, an instructor with the ESRI Educational Services group here in Redlands, California. I teach our *Introduction to Geoprocessing Scripts Using Python* course. Today I want to highlight five topics that can be useful to you when writing your geoprocessing scripts with Python. This discussion is for anyone who has an interest in writing geoprocessing scripts with Python.

Our first topic is deleting variables that hold an insert or an update cursor. When using an update cursor, or an insert cursor within our scripts to calculate values within a table or a feature class, it is a good practice to delete the variable that created the cursor. For example, if we wanted to update a name field on a rivers feature class within a script, we can use an update cursor. To establish that update cursor, we might set a variable in our script called *cur* and set it equal to gp.UpdateCursor. From there, we might put the update operation in a WHILE loop to calculate the values for us. When we are done with updating the values within our feature class, it is a good practice to delete the *cur* variable. To do that, Python has a statement called *del*. This statement allows us to delete variables from memory that we have used within our script. Make sure the *del* statement is in all lowercase letters, and if you're writing your script from within Python Win, that *del* statement should show up in blue. Within our script we would see *del cur*. This removes the lock on the feature class that Python needed to update the field on the input feature class.

The second topic is a ValidateTableName method. This is a method on the geoprocessor that will return a proper table name based on name restrictions within the output workspace. We can use this in our scripts to verify a user-defined input table name. If we wrote a script that requires a user to type in the name of an output feature class, there's a chance that they could use a character that is invalid with the output workspace. This method has two parameters.

- The first is the table name. This could be a new table or a feature class that you want to generate within your script and put into a database.
- The second parameter is a workspace. This is the workspace, or folder, that you are validating your table with. For example, if a user tries to name an output feature class *test** we could validate the user input with the workspace and verify whether we could use *test** as a feature class name. In this case, the asterisk is invalid for the database we're working with, so the ValidateTableName method will turn the asterisk into an underscore. Our result will be the name *test_*.

Our third topic is the Windows Scheduler. With the Windows Scheduler, you can set up your Python scripts to run at various times as a scheduled task. On most Windows operating systems, the Windows Scheduler can be found in the following location: Start > Programs > Accessories > System Tools > Scheduled Tasks. Then choose the option for Add Scheduled Task. This will take you into the Schedule Task Wizard, where you can specify what script to run and when.

Our fourth topic is the Make Feature Layer tool. In ArcToolbox, in the Data Management Tools toolbox, there is a toolset called Layers and Table Views. In this toolset you will find these Select Layer by Location and Select Layer by Attribute tools. If you look at the Help for these tools, the first parameter you have to supply is a feature layer, and not a feature class. To help you run these tools within a Python script, there is a tool called Make Feature Layer. This tool is also found in the Layers and Table Views toolset. This tool will take an input feature class and create a layer from it. The output of the Make Feature Layer tool can then be used as input to the Select by Location and Select Layer by Attributes tools. In addition, there is a tool in there called Make Table View. The output of the Make Table View tool can be used as input to the Select Layer by Attribute tool as well.

Finally, the Get Count tool. This tool is a personal favorite of mine to use within Python scripts. This tool will return how many records are within a table or how many features are within a feature class. It's a simple yet powerful tool.

To review our topics, if you're using an update or an insert cursor to update values within a feature class or a table, it is a good practice to delete the variable, holding onto that cursor when you're finished. The ValidateTableName method will validate the name of a table with a workspace. Use the Windows Scheduler to run your scripts at specific dates and times. The Make Feature Layer tool will create a feature layer from a feature class that you can use with other tools. And finally, the Get Count tool will tell you the number of records in a table, or features in a feature class.

For further resources, check out our instructor-led course *Introduction to Writing Geoprocessing Scripts Using Python*. Thank you for tuning into this session of our ESRI Instructional Series podcast. Stay tuned for future broadcasts.