

Exercise 3: Chapter 3

Creating Python script tools

Exercise data

Exercise data for this book can be downloaded from <https://go.esri.com/PythonProAdv>.

This is a link to the ArcGIS Online group named Advanced Python Scripting for ArcGIS Pro (Esri Press). The data for exercise 3 is posted as a ZIP file named PythonProAdv_Ex03.zip. Download this file and extract it to a folder of your choice. The instructions use the folder C:\PythonProAdv, but you can use a different folder provided you update any paths.

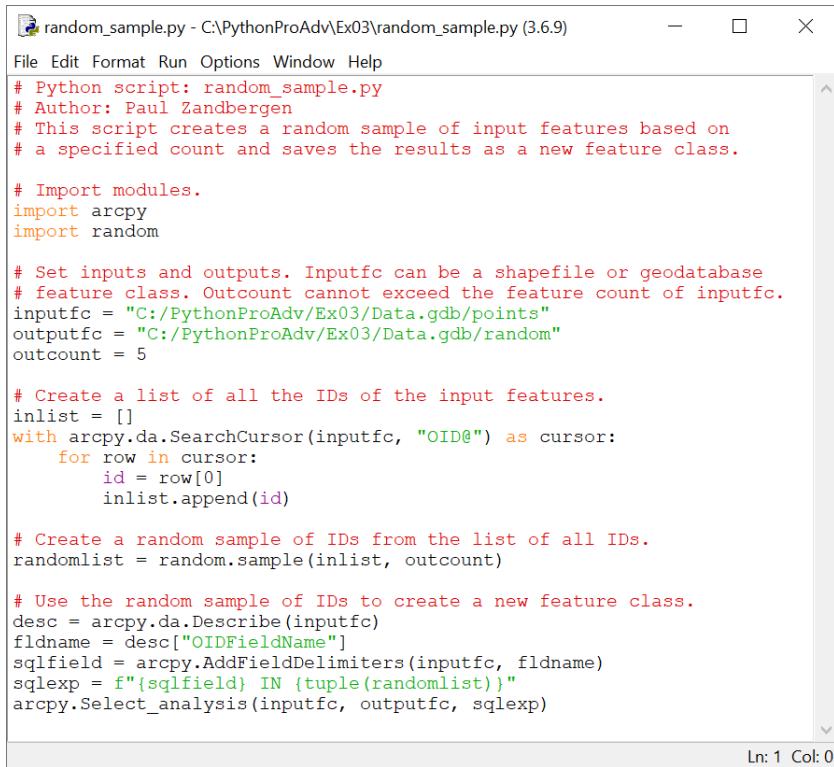
Review and test a Python script

In this exercise, you will create a Python script tool, consisting of a tool dialog box and the associated Python script. An existing stand-alone script is provided. You will examine and test this script before developing the tool dialog box.

- 1. Start IDLE.**

- 2. Click File > Open, navigate to the C:\PythonProAdv\Ex03 folder, and open the script random_sample.py.**

Take time to review the contents of the script. The script takes a random sample of input features on the basis of a specified count and saves the results as a new feature class.



The screenshot shows a Python script titled "random_sample.py" open in a code editor. The script uses ArcPy to create a random sample of input features. It includes comments explaining the purpose, imports, and the process of creating a list of IDs from the input feature class, selecting a random sample, and using it to create a new feature class.

```
# Python script: random_sample.py
# Author: Paul Zandbergen
# This script creates a random sample of input features based on
# a specified count and saves the results as a new feature class.

# Import modules.
import arcpy
import random

# Set inputs and outputs. Inputfc can be a shapefile or geodatabase
# feature class. Outcount cannot exceed the feature count of inputfc.
inputfc = "C:/PythonProAdv/Ex03/Data.gdb/points"
outputfc = "C:/PythonProAdv/Ex03/Data.gdb/random"
outcount = 5

# Create a list of all the IDs of the input features.
inlist = []
with arcpy.da.SearchCursor(inputfc, "OID@") as cursor:
    for row in cursor:
        id = row[0]
        inlist.append(id)

# Create a random sample of IDs from the list of all IDs.
randomlist = random.sample(inlist, outcount)

# Use the random sample of IDs to create a new feature class.
desc = arcpy.da.Describe(inputfc)
fldname = desc["OIDFieldName"]
sqlfield = arcpy.AddFieldDelimiters(inputfc, fldname)
sqlexp = f"(sqlfield) IN {tuple(randomlist)}"
arcpy.Select_analysis(inputfc, outputfc, sqlexp)
```

The input feature class, output feature class, and the count are all hard coded in the script. The goal is to develop a script tool in which a user can specify these values using a tool dialog box. First, however, you will edit and test the stand-alone script.

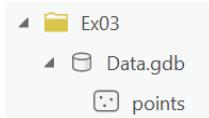
3. Start ArcGIS Pro with a new blank project.

4. Make sure the Catalog pane is visible by clicking Catalog Pane on the View tab.

Dock the Catalog pane to the right side of the ArcGIS Pro interface.

5. Create a new folder connection to the location of the exercise data by right-clicking Folders > Add Folder Connection and navigating to the folder (e.g., C:\PythonProAdv\Ex03).

6. Examine the contents of this folder.



Notice that there is a geodatabase named Data.gdb with a feature class named points.

This is the input feature class for testing purposes.

7. Add the points feature class to a new map and determine the number of records by opening the attribute table.

The record count is 1,314.

8. Return to the random_sample.py script in IDLE.

9. Modify the hard-coded paths to reflect the location of the data for this exercise, as shown. You may need to modify these paths depending on where you extracted the data.

```
inputfc = "C:/PythonProAdv/Ex03/Data.gdb/points"
```

```
outputfc = "C:/PythonProAdv/Ex03/Data.gdb/random"
```

The script is now ready to run.

10. Save and run the script.

11. Return to ArcGIS Pro and confirm that the random feature class has been created.

Note: When you leave ArcGIS Pro open, the results from running a stand-alone script are not immediately visible. You typically need to refresh the workspace. Right-click on the geodatabase (or folder) and click Refresh.

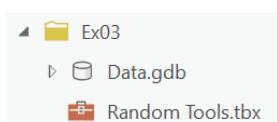
12. Add the random feature class to the current map and confirm that it contains five features.

If you ran the script again, you would get a different random selection. Now that you have confirmed that the stand-alone script works as intended, you can start developing the script tool.

Create a script tool

Script tools are located inside a toolbox. First, you will create a new custom toolbox. Then within that toolbox, you will create a new script tool that makes use of the preceding script.

1. Return to ArcGIS Pro.
2. In the Catalog pane, right-click the C:\PythonProAdv\Ex03 folder and click New > Toolbox.
3. Name the new toolbox **Random Tools.tbx**.



A toolbox can contain one or more script tools.

4. Right-click the new toolbox and click New > Script.

This brings up the New Script dialog box.

5. In the General panel, enter the following information for the name and label:

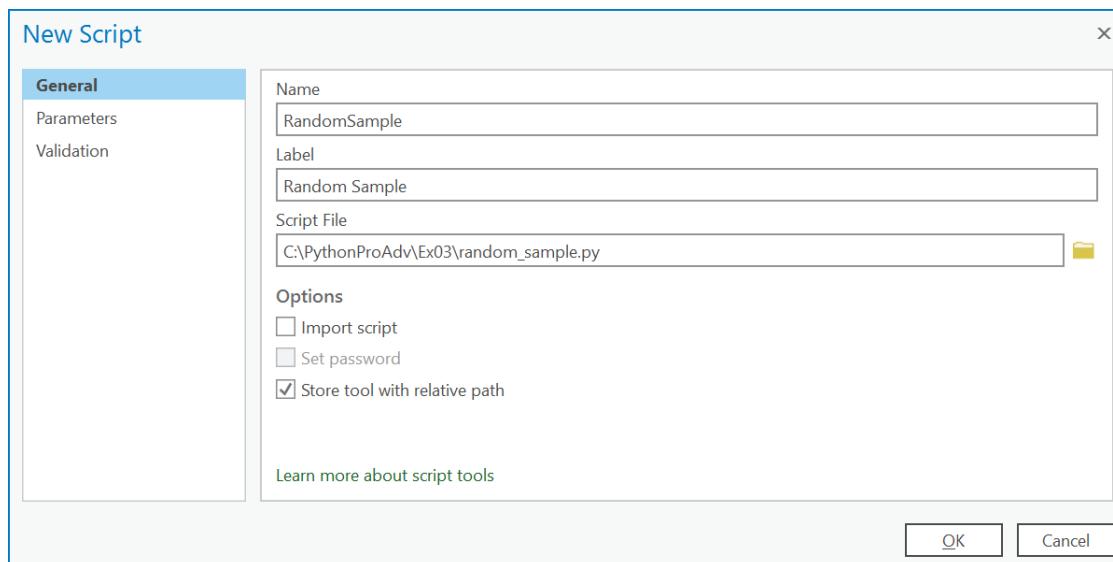
- **Name:** RandomSample
- **Label:** Random Sample

6. For the Script File entry, browse to the script used earlier, located at

C:\PythonProAdv\Ex03\random_sample.py.

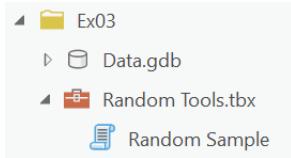
7. Select the check box "Store tool with relative path."

The New Script dialog box now looks like the figure.



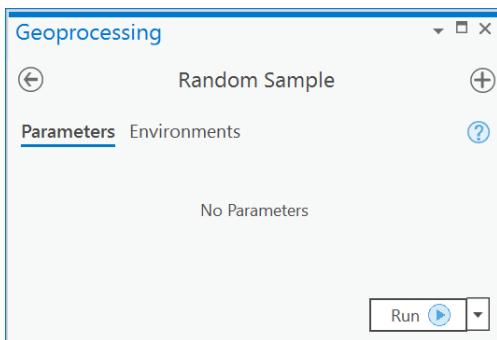
8. Click OK to save the settings and close the dialog box.

The Random Sample script tool has been added to the Random Tools toolbox.



9. Double-click the Random Sample tool to open the tool dialog box.

The tool dialog box opens but does not contain any parameters.



If you clicked Run, the script would execute, but this is not a meaningful way to run a script. You need to add tool parameters.

10. Close the Random Sample tool dialog box.

Configure script tool parameters

The new tool will have three parameters: (1) an input feature class, (2) an output feature class containing randomly selected features, and (3) the number of features to be selected.

1. Right-click on the Random Sample tool in the toolbox and click Properties.

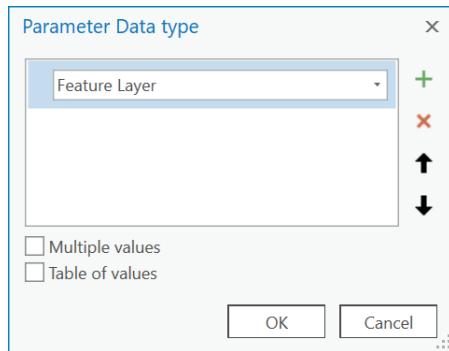
2. Click Parameters.

This brings up the Parameters panel in which you can specify the parameters for the script tool.

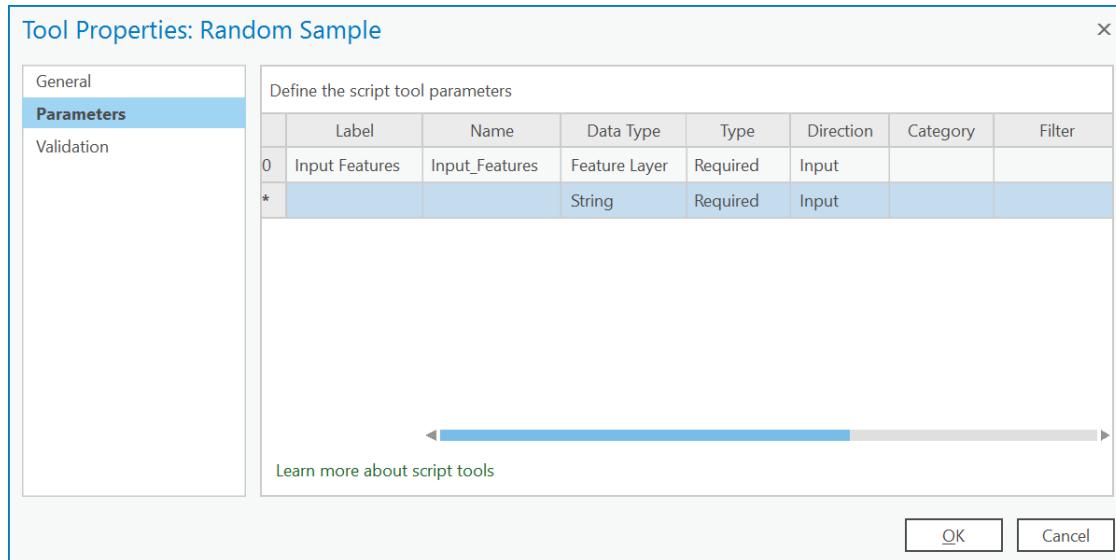
3. For the first parameter, enter the following information:

- **Label:** **Input Features**
- **Name:** **Input_Features**
- **Data Type:** Feature Layer (see details below)
- **Type:** Required
- **Direction:** Input

4. To change the data type, when you hover over the cell, an icon with three dots appears. Click on this icon to bring up the Parameter Data type dialog box. You can select the desired data type from the list of drop-down options.



The Tool Properties dialog box now looks like the following figure.



The second parameter consists of the output features to be created.

5. For the second parameter, enter the following information:

- **Label:** Output Features
- **Name:** Output_Features
- **Data Type:** Feature Class
- **Type:** Required
- **Direction:** Output

The third parameter consists of the number of features to be selected at random.

6. For the third parameter, enter the following information:

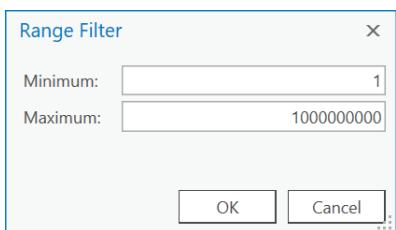
- **Label:** Number of Features
- **Name:** Number_of_Features

- **Data Type: Long**
- **Type: Required**
- **Direction: Input**

The third parameter requires one additional setting to ensure that only positive values greater than zero can be entered.

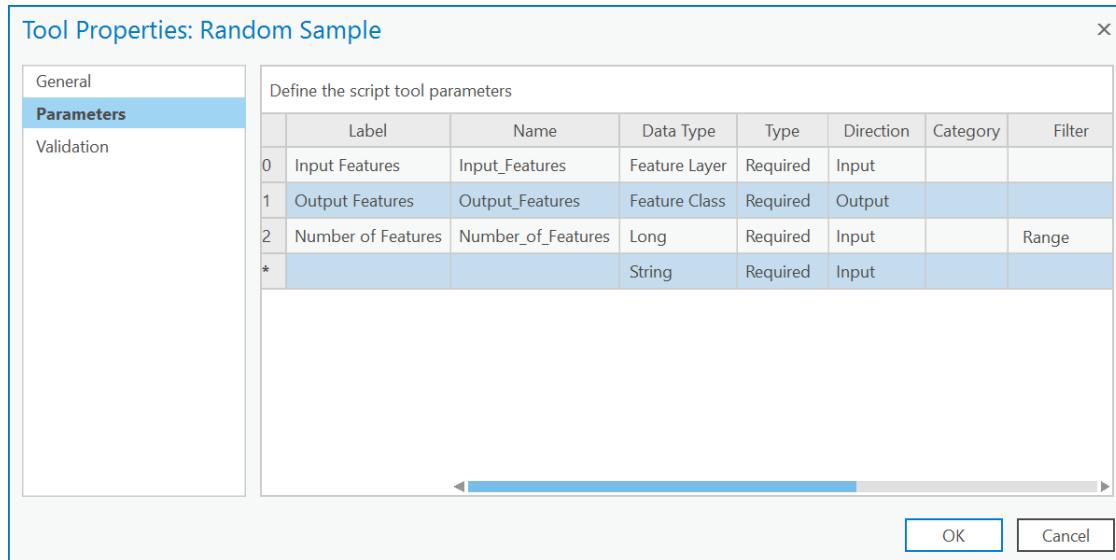
7. For the third parameter, click inside the cell in the Filter column and click Range.

8. In the Range Filter dialog box, enter a minimum value of 1 and a maximum value of 1000000000.



9. Click OK to apply the filter.

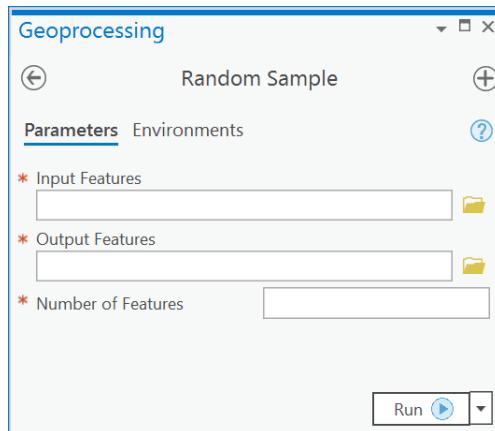
The Tool Properties dialog box now looks like the following figure.



10. Click OK to close the Tool Properties dialog box.

11. Double-click the Random Sample tool inside the Random Tools toolbox.

The tool dialog box now includes the three newly created parameters.



Even though a user can enter parameters and run the tool, the parameter values are not correctly received by the script because the script contains hard-coded values. Next, you will modify the script to receive the parameters.

Modify the script to receive parameters

You will edit the script by replacing the hard-coded values with the parameter values passed by the script when the tool is run.

1. Right-click on the Random Sample tool in the toolbox and click Edit.

This opens the random_sample.py script in the default Python editor configured in ArcGIS Pro. The default editor is IDLE. Next, you will make changes to the code, save the script, and run the tool to see if it works correctly.

2. Modify the lines of code with the hard-coded parameters as follows:

```
inputfc = arcpy.GetParameterAsText(0)

outputfc = arcpy.GetParameterAsText(1)

outcount = arcpy.GetParameter(2)
```

This code effectively replaces the hard-coded values in the script with the parameters passed by the tool. The first two parameters are received as a string, but the third parameter is received as a number.

3. Save the script and close IDLE.

Now you can test the Random Sample tool.

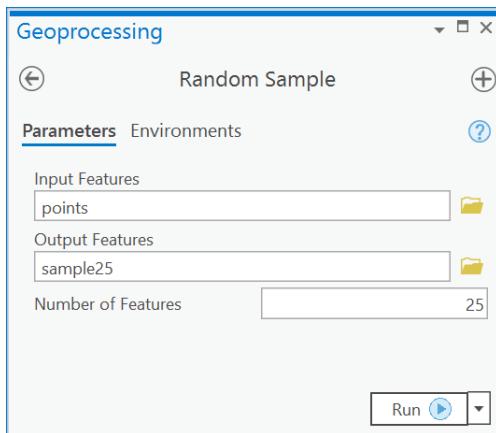
4. In the tool dialog box of the Random Sample tool, enter the following values:

- **Input Features:** points
- **Output Features:** sample25

- **Number of Features: 25**

For the output features, you can save the feature class to the Data geodatabase provided with this exercise. The full path for this parameter is C:\PythonProAdv\Ex03\Data.gdb\sample25.

The tool dialog box now looks like the figure.



5. Click Run.

When the tool has finished, the new feature class sample25 is added to the current map with 25 records.

[End of exercise 3.]