

Exercise: Use Python to create buffers around forest roads

In this exercise, you will create treatment areas for invasive plant species by creating polygons around the forest roads. These polygons will define the non-chemical treatment areas.

Estimated completion time: 20 minutes

To complete exercises, you need the following:

ArcGIS Desktop 10.0 or ArcGIS Desktop 10.1 or ArcGIS Desktop 10.2 or ArcGIS Desktop 10.3 or ArcGIS Desktop 10.4 (Basic, Standard, or Advanced)

PythonWin 2.6 Or PythonWin 2.7

Note: PythonWin 2.6 and 2.7 are available as a free download. If you do not have PythonWin installed, the first course exercise provides instructions for downloading the appropriate version.

- Step 1: Download the data

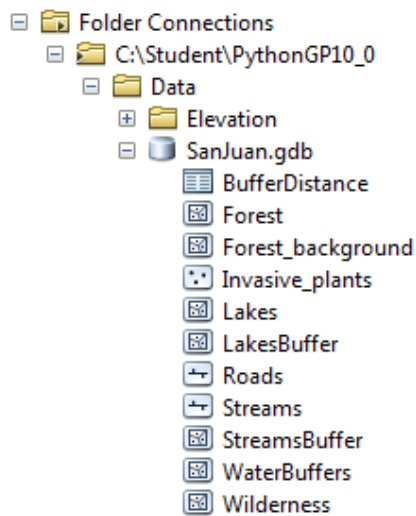
To complete the exercise, you must download the data. If you have already downloaded and installed the data, continue to the next step.

- Step 2: Examine BufferDistance table

In this step, you will examine the data that will be used by your script.

Start ArcMap and open the `..\PythonGP10_0\SanJuan.mxd`, if necessary.

Open the Catalog window and expand the `..\PythonGP10_0\Data\SanJuan` geodatabase.



Drag the BufferDistance table into your map.

In the table of contents, right-click the BufferDistance table and choose Open.

	OBJECTID *	ROUTE_TYPE	DISTANCE
	1	Primary	2000
	2	Secondary	1000
	3	Light_duty	500

This is the completed table of buffer distances that will be used to create buffer polygons around your road features.

Close the BufferDistance table.


Save your map document and close ArcMap.

- Step 3: Begin your script and set the workspace environment

You are now ready to begin your script, which will use your BufferDistance table.

Open PythonWin.

Can't find PythonWin?

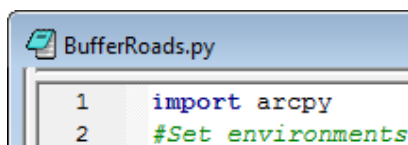
Click the New button  and save your script as **..\Scripts\BufferRoads.py**.

Tile your windows so you can see both your script and the Interactive Window.

In the Interactive Window, type **import arcpy** and press Enter.

In your script window, begin your new script by importing arcpy.

Next, add a comment indicating that you will set geoprocessing environments.



Step 3a: Begin your script and set the workspace environment.

Set the workspace environment to your SanJuan geodatabase.

```
arcpy.env.workspace = "C:/Student/PythonGP10_0/Data/SanJuan.gdb"
```

Note: If you saved the course data to a location other than the default, set your workspace to match the location of your data.

Next, allow your script to overwrite the output if it already exists.

```
arcpy.env.overwriteOutput = True
```

Step 4: Join BufferDistance table to the Roads

The next section of your script will join your BufferDistance table to the Roads feature class. Prior to executing the join, you will create four variables that will store the parameters used by the `arcpy.Join` function.

Add a comment indicating that you are setting the parameters for the Join function.

#Set parameters used to join the BufferDistance table to the Roads feature class

In the Interactive Window, enter the following:

arcpy.Usage("JoinField_management")



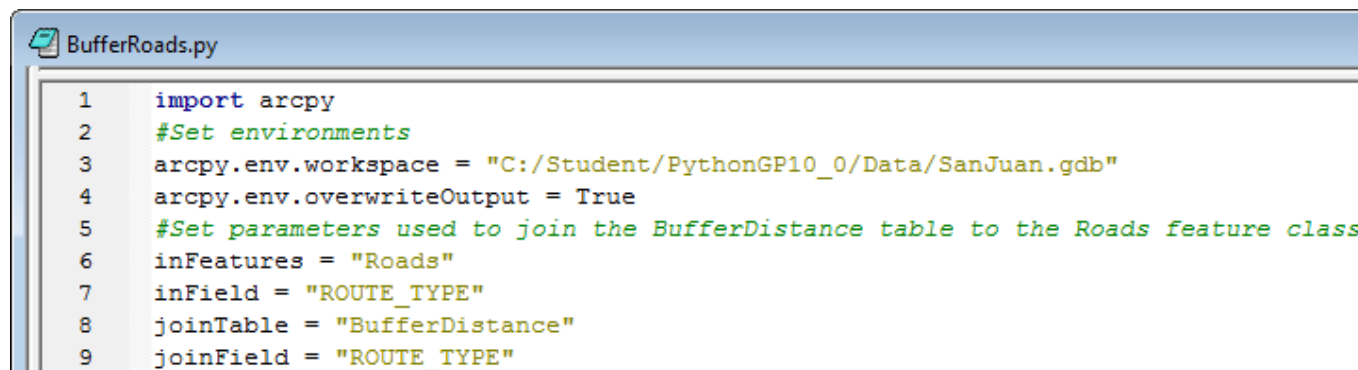
What are the four required parameters for the JoinField function?

You will store each of these parameters as a variable.

Create four variables with their values as shown in the following table.

Note: Make sure to specify each value as a Python string data type.

Variable name	Value
inFeatures	Roads
inField	ROUTE_TYPE
joinTable	BufferDistance
joinField	ROUTE_TYPE



```
1 import arcpy
2 #Set environments
3 arcpy.env.workspace = "C:/Student/PythonGP10_0/Data/SanJuan.gdb"
4 arcpy.env.overwriteOutput = True
5 #Set parameters used to join the BufferDistance table to the Roads feature class
6 inFeatures = "Roads"
7 inField = "ROUTE_TYPE"
8 joinTable = "BufferDistance"
9 joinField = "ROUTE_TYPE"
```

Step 4a: Join BufferDistance table to the Roads.

Next, you will write the code to complete the join.

Add a comment indicating that the script will create the join.

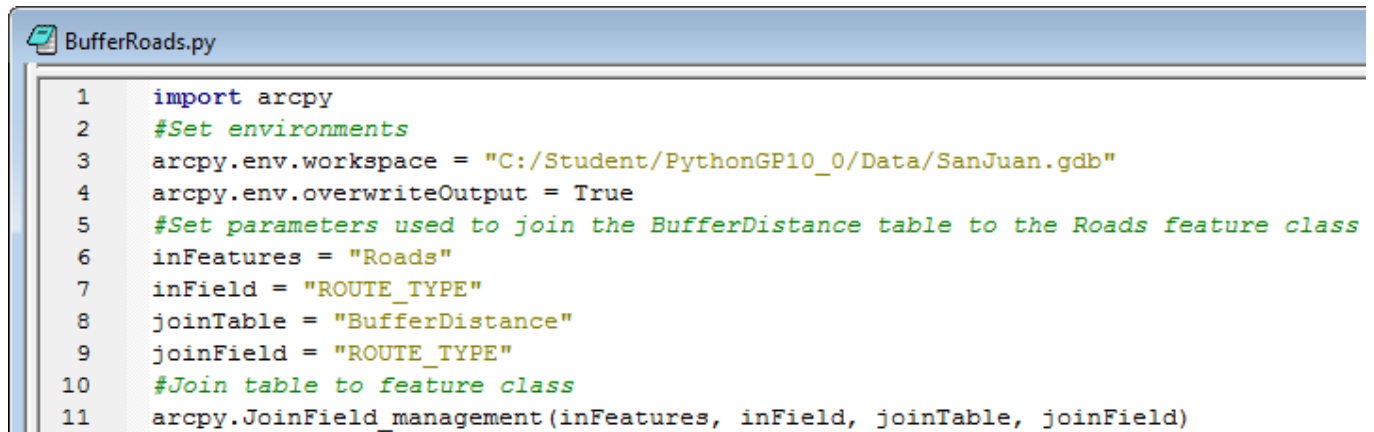
#Join table to feature class

Type **arcpy.Join** and add the `JoinField` function from the list.

Note: The name of the function is JoinField. The name of the toolbox in which the function is stored will appear after the function name in your script.

Complete the JoinField function by entering the variable parameters you specified earlier.

Tip: Use the function usage in the Interactive Window to help you enter your variables in the correct order.



```

1  import arcpy
2  #Set environments
3  arcpy.env.workspace = "C:/Student/PythonGP10_0/Data/SanJuan.gdb"
4  arcpy.env.overwriteOutput = True
5  #Set parameters used to join the BufferDistance table to the Roads feature class
6  inFeatures = "Roads"
7  inField = "ROUTE_TYPE"
8  joinTable = "BufferDistance"
9  joinField = "ROUTE_TYPE"
10 #Join table to feature class
11 arcpy.JoinField_management(inFeatures, inField, joinTable, joinField)

```

Step 4b: Join BufferDistance table to the Roads.

- Step 5: Buffer the Roads

Now you are ready to buffer the roads using the joined DISTANCE attribute. Before you execute the Buffer function, you will store the parameters as variables.

Create a comment for this section of your script

#Set parameters used to buffer Roads feature class

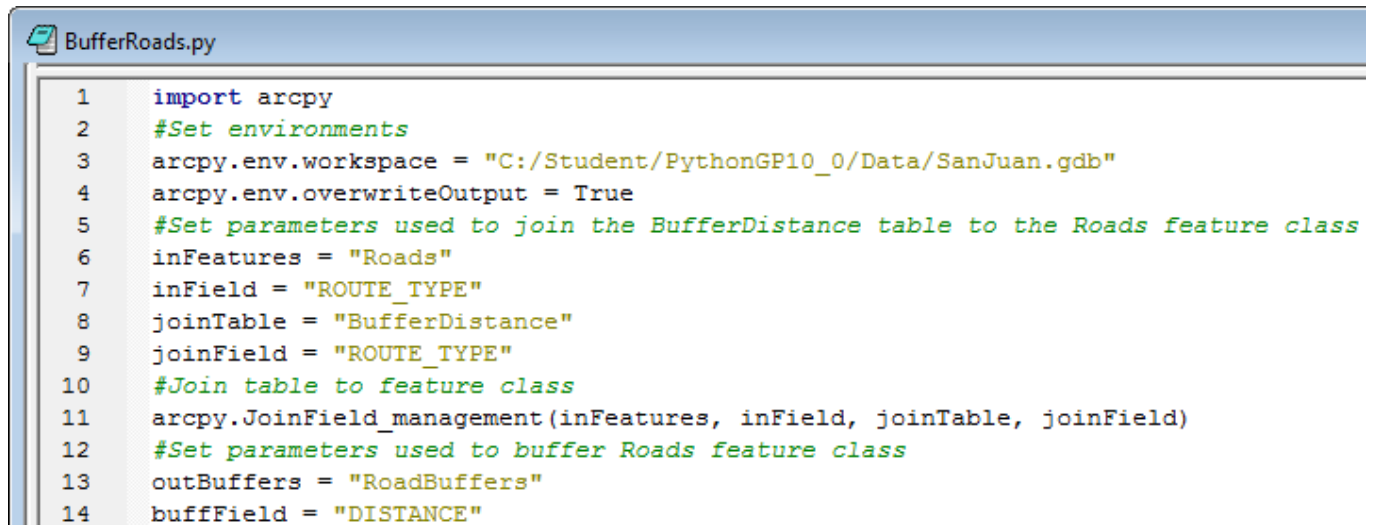
Use the arcpy.Usage function in the Interactive Window to find the parameters for the Buffer function, or refer to the Buffer tool (ArcGIS 10.2 users | ArcGIS 10.1 users | ArcGIS 10.0 users).

? What are the required parameters you will need to enter?

? Do you need to specify a variable for the in_features?

Use the following table to add the two needed variables for the Buffer function. (Remember to add each value as a string.)

Variable name	Value
outBuffers	RoadBuffers
buffField	DISTANCE



```

1  import arcpy
2  #Set environments
3  arcpy.env.workspace = "C:/Student/PythonGP10_0/Data/SanJuan.gdb"
4  arcpy.env.overwriteOutput = True
5  #Set parameters used to join the BufferDistance table to the Roads feature class
6  inFeatures = "Roads"
7  inField = "ROUTE_TYPE"
8  joinTable = "BufferDistance"
9  joinField = "ROUTE_TYPE"
10 #Join table to feature class
11 arcpy.JoinField_management(inFeatures, inField, joinTable, joinField)
12 #Set parameters used to buffer Roads feature class
13 outBuffers = "RoadBuffers"
14 buffField = "DISTANCE"

```

Step 5a: Buffer the Roads.

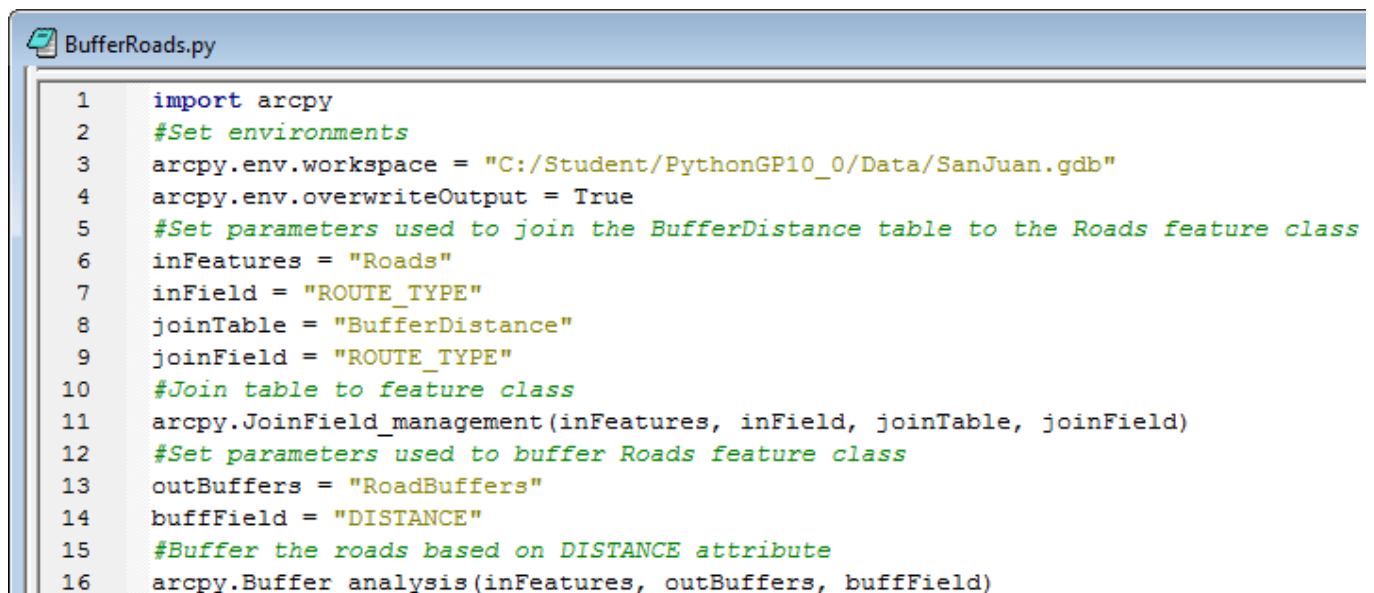
Now you are ready to buffer the roads.

Add a comment.

#Buffer the roads based on DISTANCE attribute

Write the code to buffer the Roads using the DISTANCE attribute.

Tip: Use the `arcpy.Usage` function in the Interactive Window, or the ArcGIS Help to help you enter your variables in the correct order.



```

1  import arcpy
2  #Set environments
3  arcpy.env.workspace = "C:/Student/PythonGP10_0/Data/SanJuan.gdb"
4  arcpy.env.overwriteOutput = True
5  #Set parameters used to join the BufferDistance table to the Roads feature class
6  inFeatures = "Roads"
7  inField = "ROUTE_TYPE"
8  joinTable = "BufferDistance"
9  joinField = "ROUTE_TYPE"
10 #Join table to feature class
11 arcpy.JoinField_management(inFeatures, inField, joinTable, joinField)
12 #Set parameters used to buffer Roads feature class
13 outBuffers = "RoadBuffers"
14 buffField = "DISTANCE"
15 #Buffer the roads based on DISTANCE attribute
16 arcpy.Buffer_analysis(inFeatures, outBuffers, buffField)

```


Step 5b: Buffer the Roads.

- Step 6: Run your script

You are now ready to run your script.

Click the Save button  to save your script.

Make sure that ArcMap is closed before you run your script.

Click the Run button .

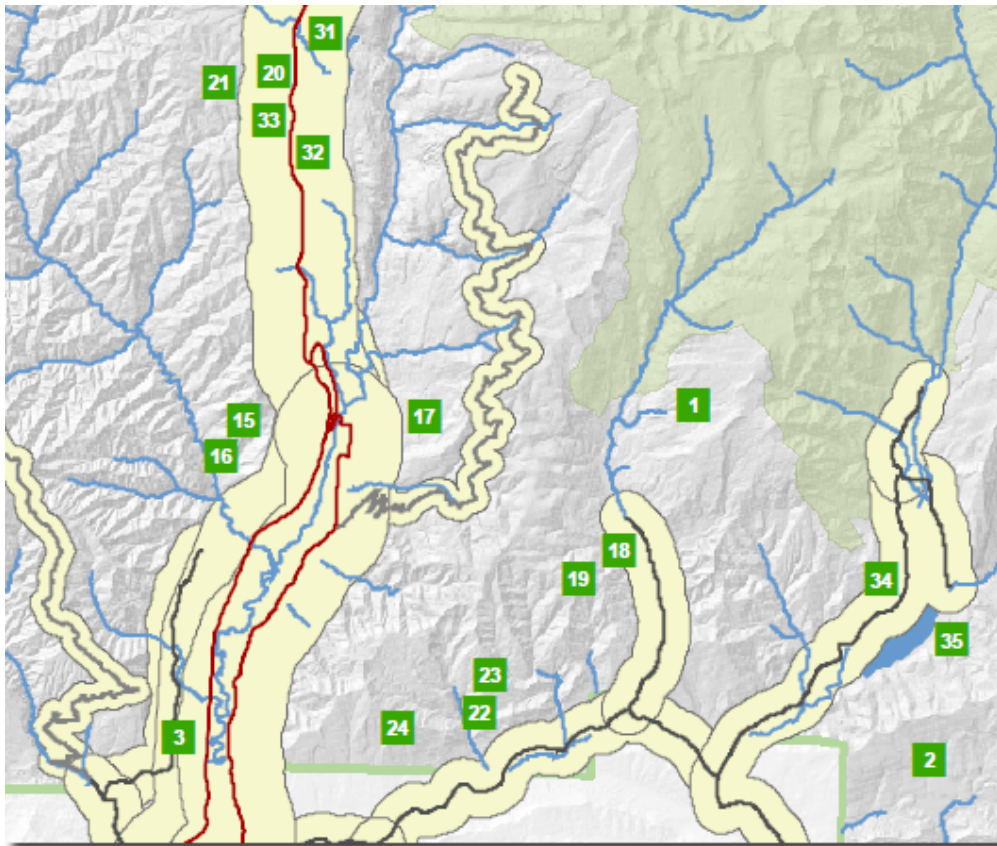
Click OK on the Run Script dialog box.

Your script will take a few moments to run. When PythonWin is finished, you should see the following message at the bottom of the PythonWin window:

Script 'C:\Student\PythonGP10_0\Results\BufferRoads.py' returned exit code 0

 What if I receive an error?

To view the result of your roads buffer, open **SanJuan.mxd** and add the RoadBuffers feature class from the Catalog window.



Step 6a: Run your script.

Notice the varied widths around your roads based on the ROUTE_TYPE attribute.

Close ArcMap.

If you are continuing to the next exercise, leave PythonWin open; otherwise, close PythonWin.