Exercise 7
Manipulating spatial data

Work with search cursors

Cursors are used to iterate over the rows in a table. Cursor functions create cursor objects, which can be used to access the row objects. Several methods exist to manipulate these row objects.

Search cursors are used to search records and to carry out SQL expressions in Python.

1 **Start ArcMap and open a new blank map document. Open the Catalog window. Navigate to the Exercise07 folder and drag the shapefile airports.shp into the map document.**

2 **In the ArcMap table of contents, right-click the airports layer and click Open Attribute Table. Review the fields of the airports.** Notice that there is a field called NAME and that the table contains 221 records.

3 **Close ArcMap. There is no need to save your map document.**

4 **Start PythonWin. Create a new Python script and save as** printvalues.py **to your C:\EsriPress\Python\Data\Exercise07\Results folder.**

5 **Enter the following code:**

```
import arcpy
from arcpy import env
env.workspace = "C:/EsriPress/Python/Data/Exercise07"
fc = "airports.shp"
cursor = arcpy.da.SearchCursor(fc, ["NAME"])
for row in cursor:
    print "Airport name = {0}".format(row[0])
```

Notice that the script creates a search cursor on the feature class and uses a `for` loop to iterate over all the rows of the attribute table.

6 Save and run the script.

The result is a list of the names of all the airports, as follows:

```
Airport name = Hyder
Airport name = Chignik Lagoon
Airport name = Koyuk
Airport name = Kivalina
Airport name = Ketchikan Harbor
Airport name = Metlakatla
Airport name = Waterfall
 ...
```

Note: *Be careful printing results to the Interactive Window because a feature class or table could contain millions of records. In the preceding example, it was confirmed in advance that the number of records was relatively limited.*

Use search cursors with SQL in Python

Search cursors can be used to carry out SQL expressions in Python.

1 In PythonWin, create a new Python script and save as SQL.py **to the Results folder for exercise 7.**

2 Enter the following code:

```
import arcpy
from arcpy import env
env.workspace = "C:/EsriPress/Python/Data/Exercise07"
fc = "airports.shp"
cursor = arcpy.da.SearchCursor(fc, ["NAME"], '"TOT_ENP" > 100000')
for row in cursor:
    print row[0]
del row
del cursor
```

3 Save and run the script. The result is a list of the names of the airports for which the SQL expression is true. The field TOT_ENP is a measure of the number of passengers. The list is as follows:

```
Ketchikan
Juneau International
Kenai Municipal
Fairbanks International
Bethel
Ted Stevens Anchorage International
```

Take another look at the SQL expression used: `"TOT_ENP" > 100000`. Field delimiters for shapefiles consist of double quotation marks—for example, `"TOT_ENP"`—but there are no quotation marks around the value of `100000` because TOT_ENP is a numeric field. The entire SQL expression needs to be in quotation marks, because the `WHERE` clause in the syntax of the search cursor is a string. This results in the SQL expression, `'"TOT_ENP" > 100000'`.

This syntax can create complications. For example, for text fields in SQL expressions, the values require single quotation marks—for example, `"NAME" = 'Ketchikan'`. The statement in the `WHERE` clause needs to be in quotation marks, but whether you use double quotation marks (" ") or single quotation marks (' '), the statement will produce a syntax error. The solution is to use the escape character (\), which would otherwise cause a syntax error, in front of the quotation marks. In Python, a backslash within a string is interpreted as an escape character, which is a signal that the next character is to be given a special interpretation. So instead of `'"NAME" = 'Ketchikan''`, the expression becomes `'"NAME" = \'Ketchikan\''`.

4 Modify the SQL expression in the script as follows:

```
    cursor = arcpy.da.SearchCursor(fc, ["NAME"], '"FEATURE" = ➜
  ➜ \'Seaplane Base\'')
```

5 Save and run the script.

The result is a list of the names of the airports for which the SQL expression is true, as follows:

```
Hyder
Ketchikan Harbor
Metlakatla
Waterfall
Kasaan
Hollis
Craig
...
```

There are other complications when working with SQL expressions. Specifically, the field delimiters vary with the format of the feature class, as follows:

- Shapefiles and file geodatabase feature classes use double quotation marks (" ")—for example, `"NAME"`.
- Personal geodatabase feature classes use square brackets ([ ])—for example, [NAME].
- ArcSDE geodatabase feature classes do not use any delimiters—for example, NAME.

When a tool like Select By Attributes or other dialog-driven queries is used, this syntax is automatically applied, but in scripting, this can be handled using the `AddFieldDelimiters` function.

6 Modify the script as follows:

```
import arcpy
from arcpy import env
env.workspace = "C:/EsriPress/Python/Data/Exercise07"
fc = "airports.shp"
delimitedField = arcpy.AddFieldDelimiters(fc, "COUNTY")
cursor = arcpy.da.SearchCursor(fc, ["NAME"], delimitedField + " = ➔
➔ 'Anchorage Borough'")
for row in cursor:
    print row[0]
del row
del cursor
```

7 Save and run the script.

The result is a list of the names of the airports for which the SQL expression is true, as follows:

```
Girdwood
Merrill Field
Lake Hood
Elmendorf Air Force Base
Ted Stevens Anchorage International
```

SQL is also used in a number of geoprocessing tools, and a similar approach can be used to create valid SQL expressions, in general.

8 Save the existing SQL.py script as Select.py **to the Exercise07 folder.**

9 Modify the script as follows:

```
    import arcpy
    from arcpy import env
    env.workspace = "C:/EsriPress/Python/Data/Exercise07"
    infc = "airports.shp"
    outfc = "Results/airports_anchorage.shp"
    delimitedfield = arcpy.AddFieldDelimiters(infc, "COUNTY")
    arcpy.Select_analysis(infc, outfc, delimitedfield + " = ➜
  ➜ 'Anchorage Borough'")
```

10 Save and run the script.

11 Start ArcMap and open the Catalog window.

12 In the Catalog window, navigate to the Results folder for exercise 7 and confirm that the new shapefile was created with five point features.

Work with update cursors

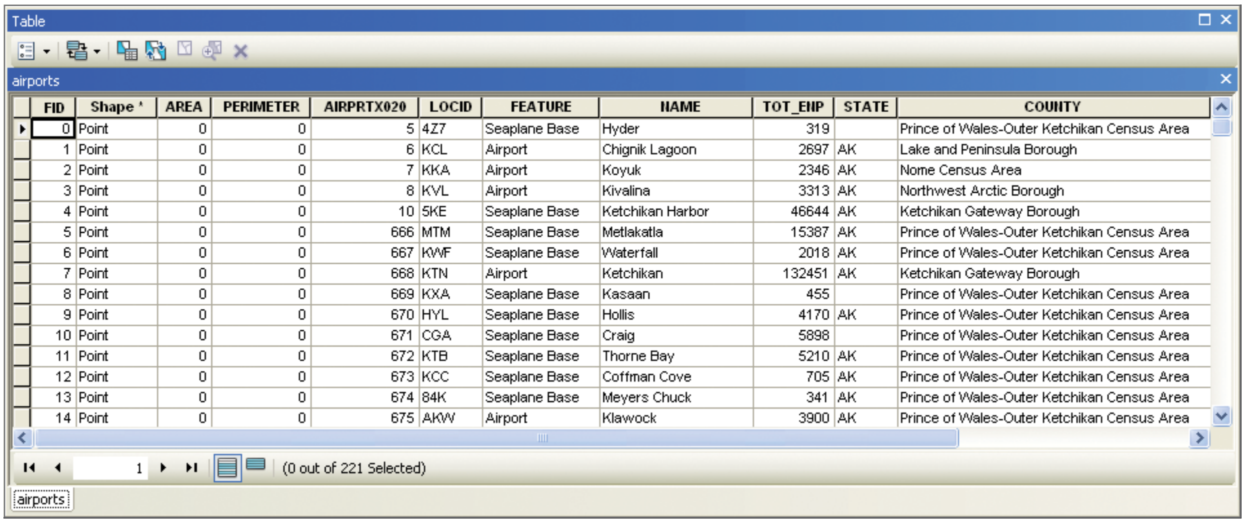
Two other cursor types can be used to work with row objects. Update cursors are used to make changes to existing records, and insert cursors are used to add new records. First, you will use an update cursor to update attribute values and delete records. Because this will permanently modify the data, it is a good idea to copy the data first.

1 In the Catalog window, navigate to the Exercise07 folder.

2 Drag the shapefile airports.shp into the map document.

3 In the ArcMap table of contents, right-click the airports layer and click Open Attribute Table.

4 Scroll over to the field STATE. Notice that some of the values in this field are blank.



Table

airports

FID	Shape *	AREA	PERIMETER	AIRPRTX020	LOCID	FEATURE	NAME	TOT_ENP	STATE	COUNTY
0	Point	0	0	5	4Z7	Seaplane Base	Hyder	319		Prince of Wales-Outer Ketchikan Census Area
1	Point	0	0	6	KCL	Airport	Chignik Lagoon	2697	AK	Lake and Peninsula Borough
2	Point	0	0	7	KKA	Airport	Koyuk	2346	AK	Nome Census Area
3	Point	0	0	8	KVL	Airport	Kivalina	3313	AK	Northwest Arctic Borough
4	Point	0	0	10	5KE	Seaplane Base	Ketchikan Harbor	46644	AK	Ketchikan Gateway Borough
5	Point	0	0	666	MTM	Seaplane Base	Metlakatla	15387	AK	Prince of Wales-Outer Ketchikan Census Area
6	Point	0	0	667	KWF	Seaplane Base	Waterfall	2018	AK	Prince of Wales-Outer Ketchikan Census Area
7	Point	0	0	668	KTN	Airport	Ketchikan	132451	AK	Ketchikan Gateway Borough
8	Point	0	0	669	KXA	Seaplane Base	Kasaan	455		Prince of Wales-Outer Ketchikan Census Area
9	Point	0	0	670	HYL	Seaplane Base	Hollis	4170	AK	Prince of Wales-Outer Ketchikan Census Area
10	Point	0	0	671	CGA	Seaplane Base	Craig	5898		Prince of Wales-Outer Ketchikan Census Area
11	Point	0	0	672	KTB	Seaplane Base	Thorne Bay	5210	AK	Prince of Wales-Outer Ketchikan Census Area
12	Point	0	0	673	KCC	Seaplane Base	Coffman Cove	705	AK	Prince of Wales-Outer Ketchikan Census Area
13	Point	0	0	674	84K	Seaplane Base	Meyers Chuck	341	AK	Prince of Wales-Outer Ketchikan Census Area
14	Point	0	0	675	AKW	Airport	Klawock	3900	AK	Prince of Wales-Outer Ketchikan Census Area

1 (0 out of 221 Selected)

airports

5 Close the attribute table.

6 In the Catalog window, navigate to the Exercise07 folder and copy the airports.shp feature class to the Results folder so you can make edits without having to worry about keeping the original intact.

7 Close ArcMap. There is no need to save your map document.

Note: *Working with insert and update cursors is just like editing, and having the feature class you want to work with in a script be open at the same time in ArcMap may result in errors because of a shared lock ArcMap places on the feature class.*

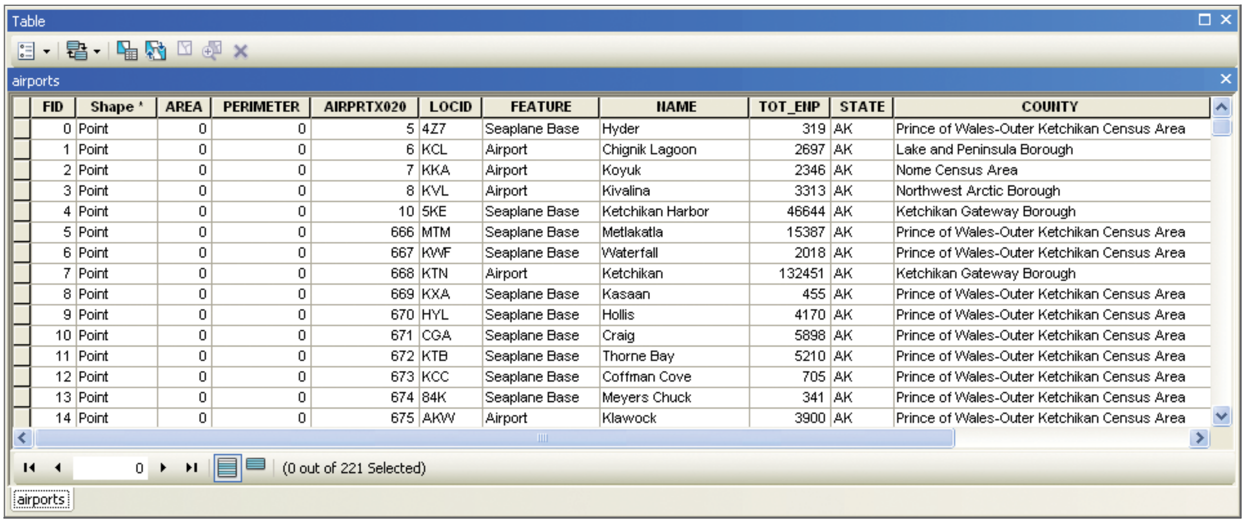
8 In PythonWin, create a new Python script and save as Update.py **to the Results folder for exercise 7.**

9 Enter the following code:

```
import arcpy
from arcpy import env
env.workspace = "C:/EsriPress/Python/Data/Exercise07"
fc = "Results/airports.shp"
delimfield = arcpy.AddFieldDelimiters(fc, "STATE")
cursor = arcpy.da.UpdateCursor(fc, ["STATE"], delimfield + " <> 'AK'")
for row in cursor:
    row[0] = "AK"
    cursor.updateRow(row)
del row
del cursor
```

10 Save and run the script.

11 Start ArcMap, open the Catalog window, navigate to the Exercise07 folder, and drag the airports.shp file into the data frame. Confirm that the values in the STATE field have been updated.



FID	Shape *	AREA	PERIMETER	AIRPRTX020	LOCID	FEATURE	NAME	TOT_ENP	STATE	COUNTY
0	Point	0	0	5	4Z7	Seaplane Base	Hyder	319	AK	Prince of Wales-Outer Ketchikan Census Area
1	Point	0	0	6	KCL	Airport	Chignik Lagoon	2697	AK	Lake and Peninsula Borough
2	Point	0	0	7	KKA	Airport	Koyuk	2346	AK	Nome Census Area
3	Point	0	0	8	KVL	Airport	Kivalina	3313	AK	Northwest Arctic Borough
4	Point	0	0	10	5KE	Seaplane Base	Ketchikan Harbor	46644	AK	Ketchikan Gateway Borough
5	Point	0	0	666	MTM	Seaplane Base	Metlakatla	15387	AK	Prince of Wales-Outer Ketchikan Census Area
6	Point	0	0	667	KWF	Seaplane Base	Waterfall	2018	AK	Prince of Wales-Outer Ketchikan Census Area
7	Point	0	0	668	KTN	Airport	Ketchikan	132451	AK	Ketchikan Gateway Borough
8	Point	0	0	669	KXA	Seaplane Base	Kasaan	455	AK	Prince of Wales-Outer Ketchikan Census Area
9	Point	0	0	670	HYL	Seaplane Base	Hollis	4170	AK	Prince of Wales-Outer Ketchikan Census Area
10	Point	0	0	671	CGA	Seaplane Base	Craig	5898	AK	Prince of Wales-Outer Ketchikan Census Area
11	Point	0	0	672	KTB	Seaplane Base	Thorne Bay	5210	AK	Prince of Wales-Outer Ketchikan Census Area
12	Point	0	0	673	KCC	Seaplane Base	Coffman Cove	705	AK	Prince of Wales-Outer Ketchikan Census Area
13	Point	0	0	674	84K	Seaplane Base	Meyers Chuck	341	AK	Prince of Wales-Outer Ketchikan Census Area
14	Point	0	0	675	AKW	Airport	Klawock	3900	AK	Prince of Wales-Outer Ketchikan Census Area

0 (0 out of 221 Selected)

airports

12 Close ArcMap. There is no need to save your map document.

In addition to updating attributes using the `updateRow` method, search cursors can also be used to delete records, which you'll do next.

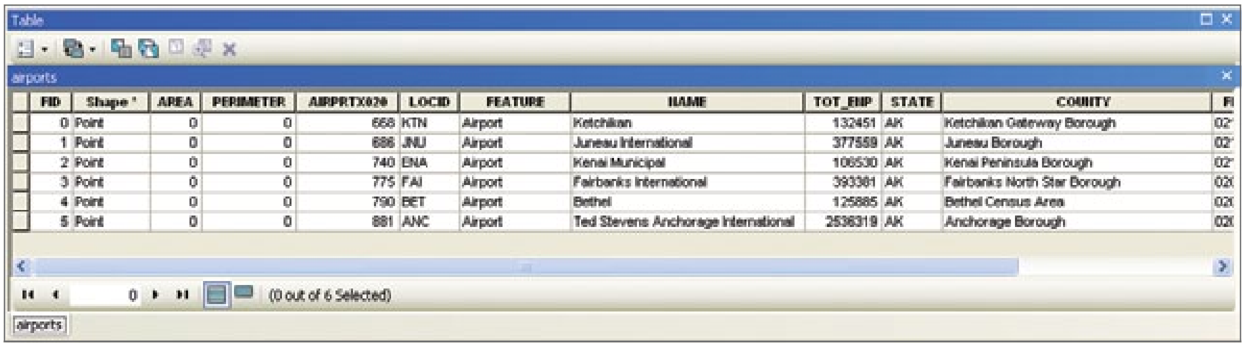
13 In PythonWin, create a new Python script and save as Delete.py to the Results folder for exercise 7.

14 Enter the following code:

```
import arcpy
from arcpy import env
env.workspace = "C:/EsriPress/Python/Data/Exercise07"
fc = "Results/airports.shp"
cursor = arcpy.da.UpdateCursor(fc, ["TOT_ENP"])
for row in cursor:
    if row[0] < 100000:
        cursor.deleteRow()
del row
del cursor
```

15 Save and run the script.

16 Start ArcMap, open the Catalog window, navigate to the Results folder for exercise 7, and drag the airports.shp file into the data frame. Confirm that all the records with fewer than 100,000 passengers have been deleted.



Table

airports

FID	Shape *	AREA	PERIMETER	AIRPRTX020	LOCID	FEATURE	NAME	TOT_ENP	STATE	COUNTY	FI
0	Point	0	0	668	KTN	Airport	Ketchikan	132451	AK	Ketchikan Gateway Borough	02
1	Point	0	0	686	JNU	Airport	Juneau International	377559	AK	Juneau Borough	02
2	Point	0	0	740	ENA	Airport	Kenai Municipal	106530	AK	Kenai Peninsula Borough	02
3	Point	0	0	775	FAI	Airport	Fairbanks International	393381	AK	Fairbanks North Star Borough	02
4	Point	0	0	790	BET	Airport	Bethel	125885	AK	Bethel Census Area	02
5	Point	0	0	881	ANC	Airport	Ted Stevens Anchorage International	2536319	AK	Anchorage Borough	02

0 (0 out of 6 Selected)

airports

17 Close ArcMap. There is no need to save your map document.

Work with insert cursors

Insert cursors are used to create new records.

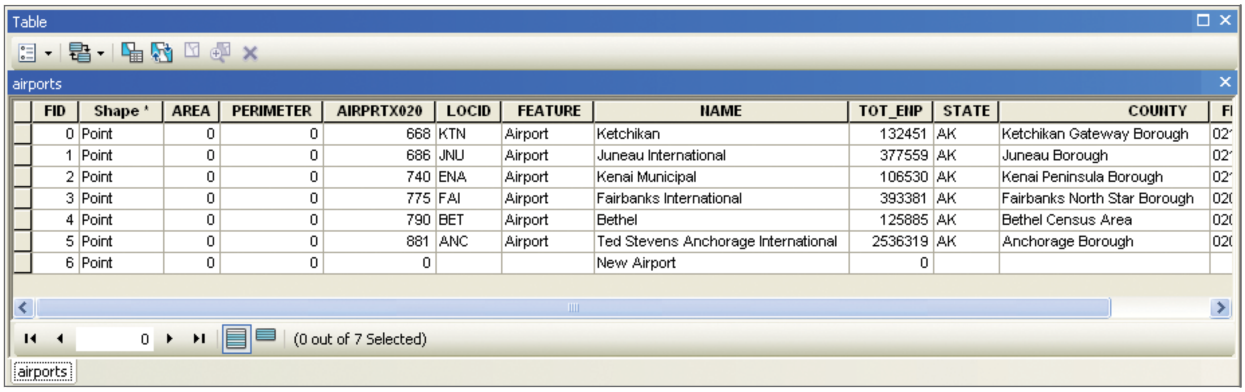
1 In PythonWin, create a new Python script and save as insert.py **to the Results folder for exercise 7.**

2 Enter the following code:

```
import arcpy
from arcpy import env
env.workspace = "C:/EsriPress/Python/Data/Exercise07"
fc = "Results/airports.shp"
cursor = arcpy.da.InsertCursor(fc, "NAME")
cursor.insertRow(["New Airport"])
del cursor
```

3 Save and run the script.

4 Start ArcMap, open the Catalog window, navigate to the Results folder for exercise 7, and drag the airports.shp file into the data frame. Confirm that a new record has been added with the name New Airport**—the other fields are blank.**



Table

airports

FID	Shape *	AREA	PERIMETER	AIRPRTX020	LOCID	FEATURE	NAME	TOT_ENP	STATE	COUNTY	FI
0	Point	0	0	668	KTN	Airport	Ketchikan	132451	AK	Ketchikan Gateway Borough	02
1	Point	0	0	686	JNU	Airport	Juneau International	377559	AK	Juneau Borough	02
2	Point	0	0	740	ENA	Airport	Kenai Municipal	106530	AK	Kenai Peninsula Borough	02
3	Point	0	0	775	FAI	Airport	Fairbanks International	393381	AK	Fairbanks North Star Borough	02
4	Point	0	0	790	BET	Airport	Bethel	125885	AK	Bethel Census Area	02
5	Point	0	0	881	ANC	Airport	Ted Stevens Anchorage International	2536319	AK	Anchorage Borough	02
6	Point	0	0	0			New Airport	0			

0 (0 out of 7 Selected)

airports

***Note:** Although a new record has been added and the attributes can be given values using the insert cursor, the new record does not have a geometry yet. This is covered in chapter 8.*

5 Close ArcMap. There is no need to save your map document.

Validate table and field names

ArcPy contains functions to validate the names of tables and fields. This prevents attempts to create invalid names, such as those with spaces or invalid characters.

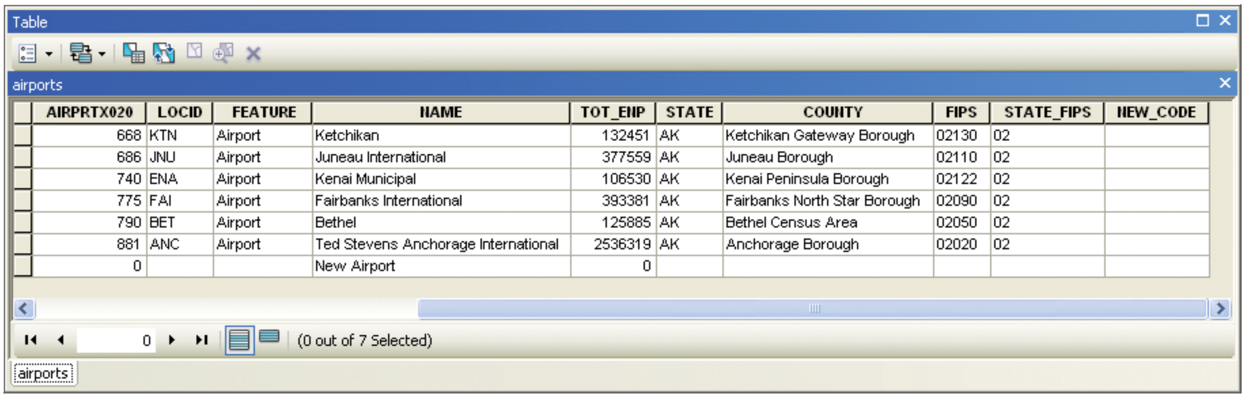
1 In PythonWin, create a new Python script and save as validatefield.py **to the Results folder for exercise 7.**

2 Enter the following code:

```
import arcpy
from arcpy import env
env.workspace = "C:/EsriPress/Python/Data/Exercise07"
fc = "Results/airports.shp"
newfield = "NEW CODE"
fieldtype = "TEXT"
fieldname = arcpy.ValidateFieldName(newfield)
arcpy.AddField_management(fc, fieldname, fieldtype, "", "", 12)
```

3 Save and run the script.

4 Start ArcMap, open the Catalog window, navigate to the Results folder for exercise 7, and drag the airports.shp file into the data frame. Confirm that a new field has been added with the name NEW_CODE**.** The space has been replaced by an underscore (_).



AIRPRTX020	LOCID	FEATURE	NAME	TOT_ENP	STATE	COUNTY	FIPS	STATE_FIPS	NEW_CODE
668	KTN	Airport	Ketchikan	132451	AK	Ketchikan Gateway Borough	02130	02	
686	JNU	Airport	Juneau International	377559	AK	Juneau Borough	02110	02	
740	ENA	Airport	Kenai Municipal	106530	AK	Kenai Peninsula Borough	02122	02	
775	FAI	Airport	Fairbanks International	393381	AK	Fairbanks North Star Borough	02090	02	
790	BET	Airport	Bethel	125885	AK	Bethel Census Area	02050	02	
881	ANC	Airport	Ted Stevens Anchorage International	2536319	AK	Anchorage Borough	02020	02	
0			New Airport	0					

0 (0 out of 7 Selected)

5 Close ArcMap. There is no need to save your map document.

6 In PythonWin, modify the script as follows:

```
newfield = "NEW?*&$"
```

The characters ?, *, &, and $ are all invalid as field names.

7 Save and run the script.

8 Start ArcMap, open the Catalog window, navigate to the Results folder for exercise 7, and drag the airports.shp file into the data frame. Confirm that the name of the new field has been modified to NEW____**.** Each of the invalid characters has been replaced by an underscore.

>>> TIP

Validating table and field names does not determine whether the field name already exists. This requires checking the new name against the names of the existing fields.

9 Close ArcMap. There is no need to save your map document.

10 Modify the script as follows:

```
import arcpy
from arcpy import env
env.workspace = "C:/EsriPress/Python/Data/Exercise07"
fc = "Results/airports.shp"
newfield = "NEW CODE"
fieldtype = "TEXT"
fieldname = arcpy.ValidateFieldName(newfield)
fieldlist = arcpy.ListFields(fc)
fieldnames = []
for field in fieldlist:
    fieldnames.append(field.name)
if fieldname not in fieldnames:
    arcpy.AddField_management(fc, fieldname, fieldtype, "", "", 12)
    print "New field has been added."
else:
    print "Field name already exists."
```

11 Save and run the script.

The preceding script creates a list of field objects using the `ListFields` function. The names of these field objects are placed in a new empty list using the `append` method. The validated name of the new field is compared to this list of field names using an `if ( ) not in` statement.

The `CreateUniqueName` function can also be used to ensure a name for a new feature class or a table is unique, but it is limited to unique names in a workspace. It cannot be used to ensure a field name is unique.

12 In PythonWin, create a new Python script and save as unique_name.py **to the Results folder for exercise 7.**

13 Enter the following code:

```
import arcpy
from arcpy import env
env.workspace = "C:/EsriPress/Python/Data/Exercise07"
fc = "airports.shp"
unique_name = arcpy.CreateUniqueName("Results/buffer.shp")
arcpy.Buffer_analysis(fc, unique_name, "5000 METERS")
```

14 Save and run the script.

15 Start ArcMap, open the Catalog window, navigate to the Results folder for exercise 7, and confirm that a new feature class called buffer.shp **has been created.**

16 Return to PythonWin and run the script again.

17 In ArcMap, in the Catalog window, confirm that a new feature class called buffer0.shp **has been created.**

If you keep running the script again, the next output files will be called **buffer1.shp, buffer2.shp**, and so on. Using the `CreateUniqueName` function can prevent accidentally overwriting files.

>>> TIP

If the feature class is not showing, you may need to refresh the Catalog view by right-clicking the folder in the ArcMap table of contents and clicking Refresh.

Challenge exercises

Challenge 1

Write a script that creates a 15,000-meter buffer around features in the airports.shp feature class classified as an airport (based on the `FEATURE` field) and a 7,500-meter buffer around features classified as a seaplane base. The results should be two separate feature classes, one for each airport type.

Challenge 2

Write a script that adds a text field to the roads.shp feature class called **FERRY** and populates this field with YES and NO values, depending on the value of the FEATURE field.