

Exercise 11

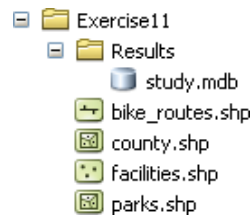
Debugging and error handling

Examine syntax errors and exceptions

To start the exercise, first take a quick look at the data.

- 1 **Start ArcMap. Open the Catalog window. Navigate to the C:\EsriPress\Python\Data\Exercise11 folder. →**

Notice that there are four shapefiles in this folder, plus one empty personal geodatabase in the Results folder.

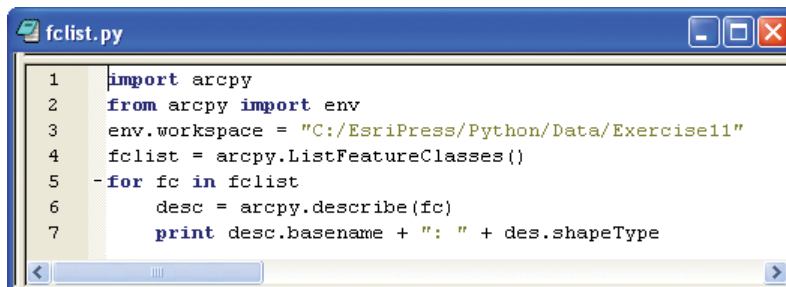


There are also two scripts in the Exercise11 folder, which are not visible in ArcCatalog by default. These scripts contain errors, which you will fix in this exercise.

- 2 **Close ArcMap. There is no need to save the map document.**

Syntax errors prevent code from being executed. In the following examples, you will identify some common syntax errors.

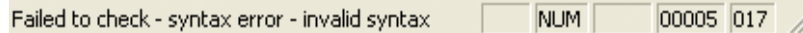
- 3 **Start PythonWin. On the menu bar, click File > Open, navigate to the Exercise11 folder, and open the script fclist.py.**

A screenshot of the PythonWin window with the file 'fclist.py' open. The script contains the following Python code:

```
1 import arcpy
2 from arcpy import env
3 env.workspace = "C:/EsriPress/Python/Data/Exercise11"
4 fclist = arcpy.ListFeatureClasses()
5 -for fc in fclist
6     desc = arcpy.describe(fc)
7     print desc.basename + ": " + des.shapeType
```

This script has a number of syntax errors. You may be able to identify them directly, but even so, it is useful to see how they can be found.

- 4 With the cursor placed inside the script, click the Check button.** This displays an error message on the PythonWin status bar. The cursor is placed at the end of the line where the syntax error occurred: line 5, character position 17.



Failed to check - syntax error - invalid syntax NUM 00005 017

The error message does not report what the error is, but only where it occurs—you still have to identify the exact nature of the syntax error. In this case, the error is a missing colon (:) at the end of the line of code.

- 5 Correct the code on line 5 of the script as follows:**

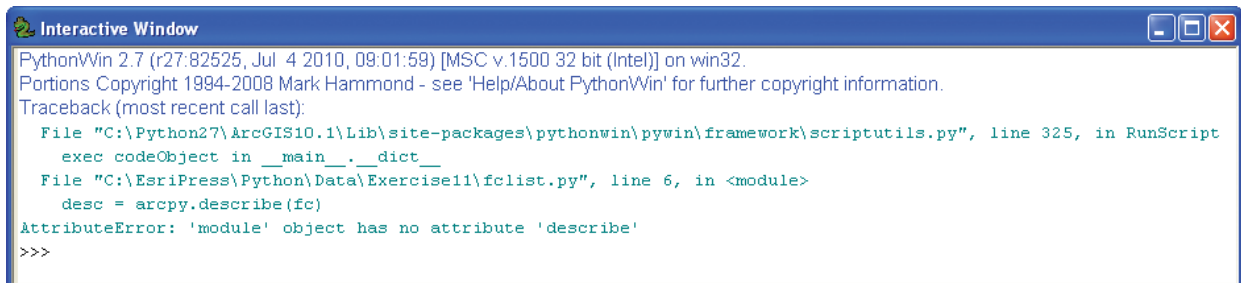
```
for fc in fclist:
```

- 6 Save and run the script.** The script runs but is interrupted and an error message appears on the PythonWin status bar, as shown in the figure.



Exception raised while running script fclist.py NUM 00005 018

An error message is also printed to the Interactive Window, as shown in the figure.



An `AttributeError` exception was raised at line 6 in the script. The module `arcpy` does not have an attribute called `describe`. The correct spelling of the function is `Describe`. Remember that Python is case sensitive, for the most part.

- 7 Correct the code on line 6 of the script as follows:**

```
desc = arcpy.Describe(fc)
```

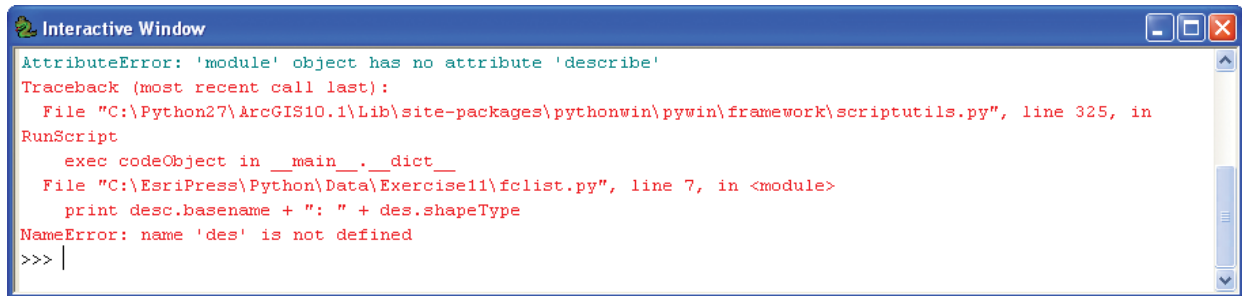
>>> TIP

Line numbering can be made visible in PythonWin by clicking View > Options from the menu bar. On the PythonWin Options dialog box, click the Editor tab and under Margin Widths, increase the size for Line Numbers—for example, to 30.

PythonWin also shows the position of the cursor within a script window by a designation on the far right of the status bar. The first number is the line number, and the second number is the character position.

- 8 Save and run the script.** Again, an error message appears on the PythonWin status bar, and the details are printed to the Interactive Window.

Note: For longer error messages, it is best to start reading them from the bottom up. For example, the last line shows the information about the exact error. Above that, it shows the line that's in error. And above that, it reports the line number as line 7.



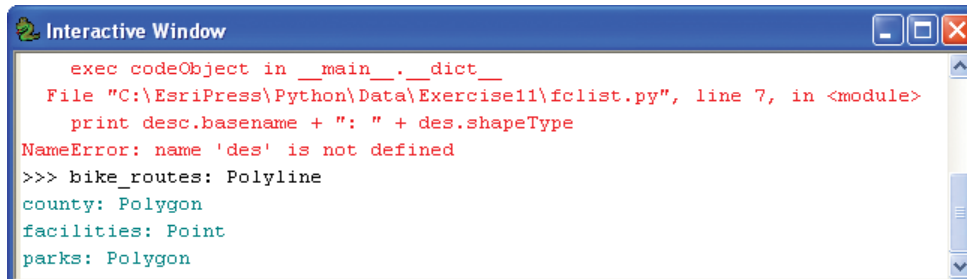
```
Interactive Window
AttributeError: 'module' object has no attribute 'describe'
Traceback (most recent call last):
  File "C:\Python27\ArcGIS10.1\Lib\site-packages\pythonwin\pywin\framework\scriptutils.py", line 325, in
RunScript
    exec codeObject in __main__.__dict__
  File "C:\EsriPress\Python\Data\Exercise11\fclist.py", line 7, in <module>
    print desc.basename + ": " + des.shapeType
NameError: name 'des' is not defined
>>> |
```

A NameError exception was raised on line 7 in the script. The name `des` is not defined. The correct spelling of the variable is `desc`.

- 9 Correct the code on line 7 of the script as follows:**

```
print desc.basename + ": " + desc.shapeType
```

- 10 Save and run the script.** This time the script runs correctly, and the result is printed to the Interactive Window.



```
Interactive Window
exec codeObject in __main__.__dict__
File "C:\EsriPress\Python\Data\Exercise11\fclist.py", line 7, in <module>
    print desc.basename + ": " + desc.shapeType
NameError: name 'des' is not defined
>>> bike_routes: Polyline
county: Polygon
facilities: Point
parks: Polygon
```

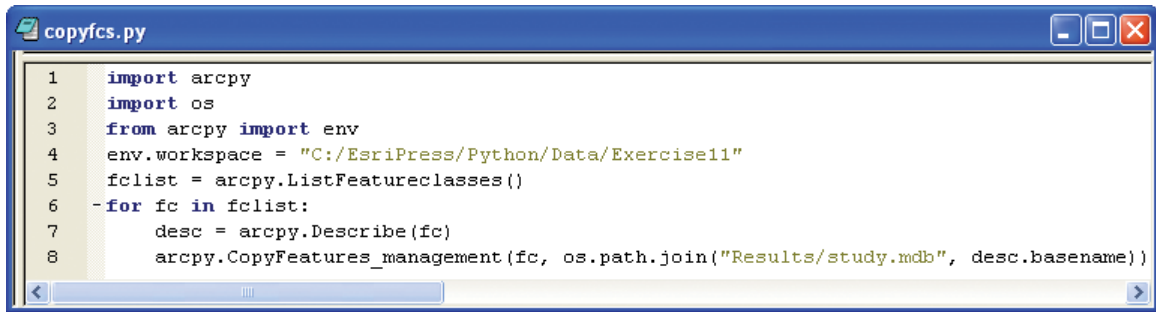
This example illustrates some of the most common errors in Python scripts: punctuation, capitalization, and spelling.

- 11 Close the fclist.py script.**

Implement debugging procedures

Many errors can be identified simply by trying to run a script and investigating the error messages. An alternative is to step through your code line by line using the Python debugger.

- 1 **On the PythonWin Standard toolbar, click the Open button, navigate to the Exercise11 folder, and open the script copyfcs.py.**



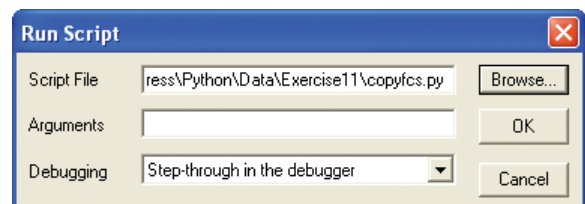
```
1 import arcpy
2 import os
3 from arcpy import env
4 env.workspace = "C:/EsriPress/Python/Data/Exercise11"
5 fclist = arcpy.ListFeatureclasses()
6 -for fc in fclist:
7     desc = arcpy.Describe(fc)
8     arcpy.CopyFeatures_management(fc, os.path.join("Results/study.mdb", desc.basename))
```

Take a moment to examine the code. The script creates a list of all the feature classes in a workspace and copies them to a personal geodatabase.

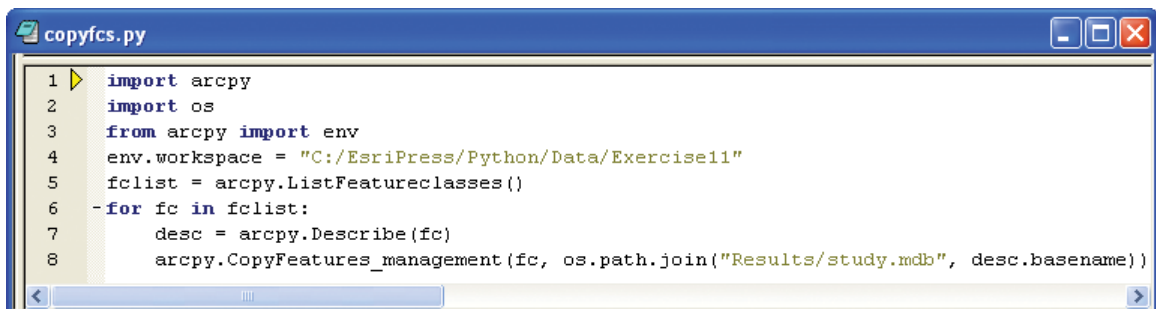
- 2 **With your pointer placed inside the script, click the Check button.**
There are no syntax errors.

- 3 **Click the Run button.**

- 4 **On the Run Script dialog box, under the Debugging options, click "Step-through in the debugger".** ➔



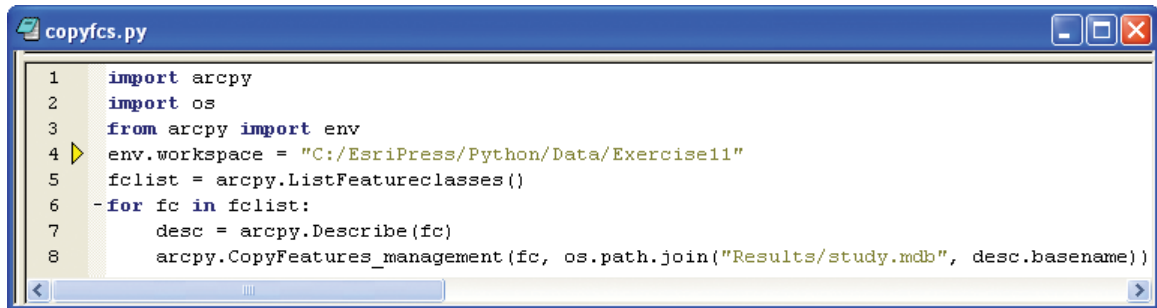
- 5 **Click OK.** This brings up the Debugger toolbar, and in the script, a yellow arrow points to the first line of executable code.



```
1 import arcpy
2 import os
3 from arcpy import env
4 env.workspace = "C:/EsriPress/Python/Data/Exercise11"
5 fclist = arcpy.ListFeatureclasses()
6 -for fc in fclist:
7     desc = arcpy.Describe(fc)
8     arcpy.CopyFeatures_management(fc, os.path.join("Results/study.mdb", desc.basename))
```

Using the Debugger tools, you can step through the code line by line. If any errors occur while running the code line by line, the error messages will be printed to the Interactive Window.

- 6 On the Debugger toolbar, click the Step button.** This runs the first line of code, and the yellow arrow points to the second line of code.
- 7 Click the Step button again twice.** The yellow arrow now points to the fourth line of code.



- 8 Click the Step button again.** This opens the `_base.py` module, with the yellow arrow pointing to line 512 of the script, although the line number may change between different versions of Python. This is one of the built-in ArcPy scripts. When the workspace is set as part of the environment settings, this module is called. You probably don't want to step through this module but let the code run and return to your script, so that you can then continue to step through your own script.
- 9 On the Debugger toolbar, click the Step Out button twice.**
- 10 Close the `_base.py` script.** This brings you back to your own script, with the yellow arrow pointing to line 5.

To avoid opening other modules, you will use the Step Over button next, instead of Step. The Step Over button runs the current line of code, and if it includes any Python modules or functions, they will be run as well, and the cursor will return to the script itself.

- 11 Click the Step Over button.**

This prints an error message to the Interactive Window, which follows in part:

```
AttributeError: 'module' object has no attribute 'ListFeatureclasses'
```

12 On the Debugger toolbar, click the Close button.

13 Correct line 5 of the script as follows:

```
fcList = arcpy.ListFeatureClasses()
```

14 Save the script.

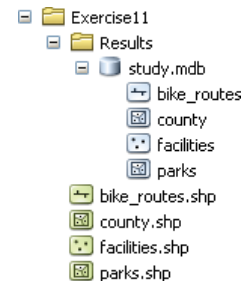
15 Click the Run button.

16 On the Run Script dialog box, under the Debugging options, click "No debugging". This runs the script, and no more errors are encountered. Successful execution is reported on the PythonWin status bar.

Script 'C:\EsriPress\Python\Data\Exercise11\copyfcs.py' returned exit code 0

17 Start ArcMap. Open the Catalog window. Navigate to the Exercise11 folder. Confirm that the geodatabase study.mdb contains four feature classes. ➔

18 Close ArcMap.



Handle some exceptions

Many different types of errors can occur when running a script. Rather than just letting a script cause a runtime error, you can gain more control using certain error-handling procedures. The most widely used error-handling technique uses a `try-except` statement.

Consider the script from the preceding set of steps. What if the geodatabase `study.mdb` did not exist?

1 In PythonWin, make sure the script `copyfcs.py` is still open.

2 On line 8, replace `study.mdb` with `mydata.mdb`.

3 Save and run the script. This produces an error message on the PythonWin status bar, as shown in the figure.

Exception raised while running script `copyfcs.py`

Note: You need to stop the debugger so you can make changes to the script and run it again.

A more detailed error message is also printed to the Interactive Window, as follows:

```
ExecuteError: ERROR 000210: Cannot create output  
Results/mydata.mdb\bike_routes  
Failed to execute (CopyFeatures).
```

Next, you will trap this error using a try-except statement.

4 Modify the code, as follows:

```
import arcpy, os  
from arcpy import env  
try:  
    env.workspace = "C:/EsriPress/Python/Data/Exercise11"  
    fclist = arcpy.ListFeatureClasses()  
    for fc in fclist:  
        desc = arcpy.Describe(fc)  
        arcpy.CopyFeatures_management(fc, os.path.join("Results/ ➔  
➔ mydata.mdb", desc.basename))  
except arcpy.ExecuteError:  
    print arcpy.GetMessages(2)  
except:  
    print "There has been a nontool error."
```

5 Save and run the script. This time the script runs successfully.

Script 'C:\EsriPress\Python\Data\Exercise11\copyfcs.py' returned exit code 0

However, an error message is printed to the Interactive Window, as follows:

```
ExecuteError: ERROR 000210: Cannot create output  
Results/mydata.mdb\bike_routes  
Failed to execute (CopyFeatures).
```

Although the error message is the same as before, there is a very important difference: despite the error, the script ran successfully rather than resulting in a runtime error. This is a very important distinction. Say, for instance, that you called this script as a tool in ArcMap. If the script results in a runtime error, you may not find out what happened because the printed error message does not appear anywhere. With the use of the try-except statement, the script runs successfully, and the error message can be reported back to the tool that called the script.

Challenge exercises

Challenge 1

The following script contains a number of errors. Try to identify all four.

```
import arcpy
from arcpy import env
env.workspace = "C:/EsriPress/Python/Data/Exercise07"
FC = "airports.shp"
rows = arcpy.SearchCursor(fc)
fields = arcpy.ListFields(fc)
for field in fields:
    if fields.name == "NAME"
        for row in rows:
            print "Name = {0}".format(row.getValue(field.name))
```

Challenge 2

The following script contains a number of errors. Try to identify all six.

```
import arcpy
from arcpy import env
env.workspace = "C:/EsriPress/Python/Data\Exercise09"
raster = "landcover.tiff"
desc = arcpy.describe(raster)
x = desc.MeanCellHeight
y = desc.MeanCellWidth
spatialref = desc.spatialReference
units = spatialref.linearUnitName
print "Cells are" + str(x) + " by " + str(y) + " " + units + "."
```