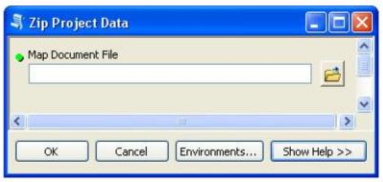
This video will discuss how to prepare a script for importation into ArcGIS.

ArcScripts

- Scripts that have been imported into ArcToolbox are run like the built-in tools.
- Run through dialog window interface…
  - automatically generated by ArcMap.
  - can be thoroughly documented.
  - no programming ability required to run script.
- Excellent for scripts that will be shared with other GIS users.



2

A python script can be imported into ArcToolbox to create an “ArcScript”. The ArcScript is run the same way as any of the built-in tools in ArcToolbox.

ArcScripts are run through the **dialog window** interface. This interface is automatically created when a script is imported into ArcToolbox. The ArcScript can be thoroughly documented and is run without the user having any direct interaction with the script.

An ArcScript is easy to share with other GIS users and can be run without any knowledge of python.

Catching errors

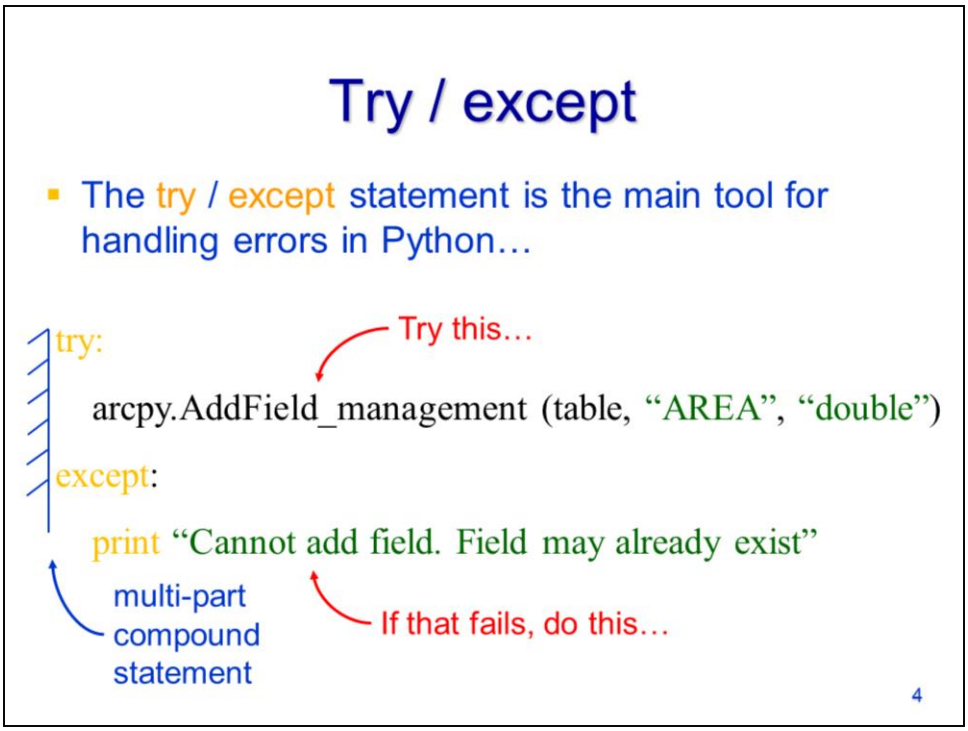
- Script should be robust before importing into ArcToolbox…
  - troubleshooting is difficult in ArcToolbox
- Python has tools that "catch" errors:
  - allow script to deal with them and continue
  - or exit the script with explanation and location of the failure

Image at: http://www.nyworkerscompensationalliance.org/uploads/image/sbSafetyNet.jpg

3

Before converting a python script into an ArcScript, the code must be robust and thoroughly tested. Troubleshooting a script is much more difficult after it has been imported into ArcGIS.

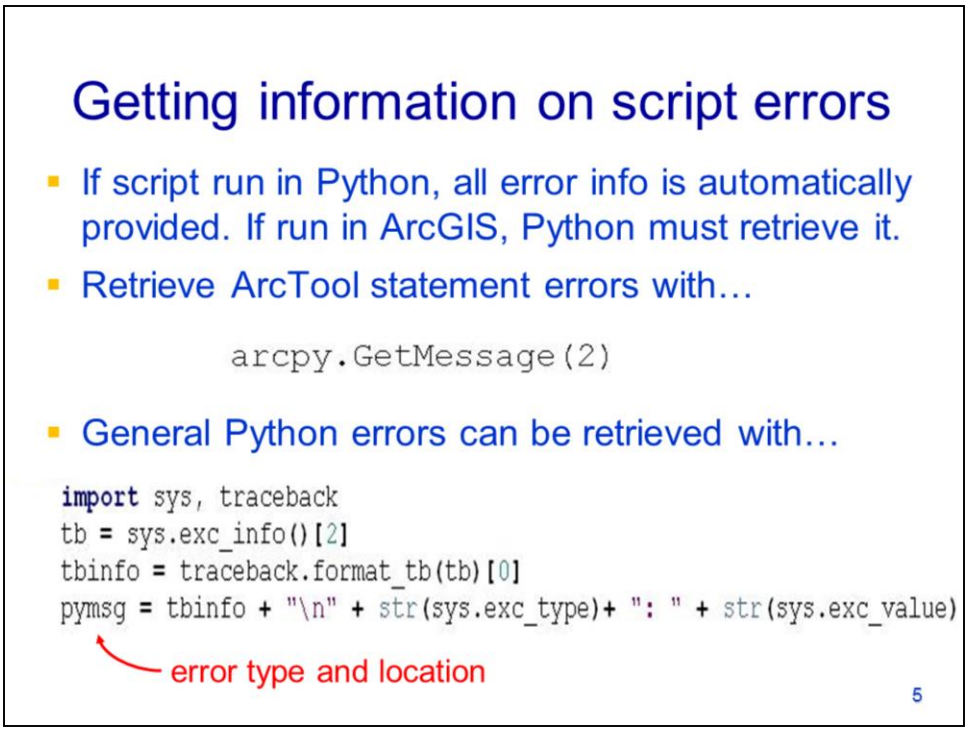
The script should be able to "catch" and deal with errors that may occur. If the script cannot continue because of the error, then the script should exit gracefully and provide any information that is known about the error including the associated line number.



The **try/except** statement can be used to catch and deal with unanticipated errors in a script.

The statement consists of two parts – note that the header lines for both parts have the same level of indentation.

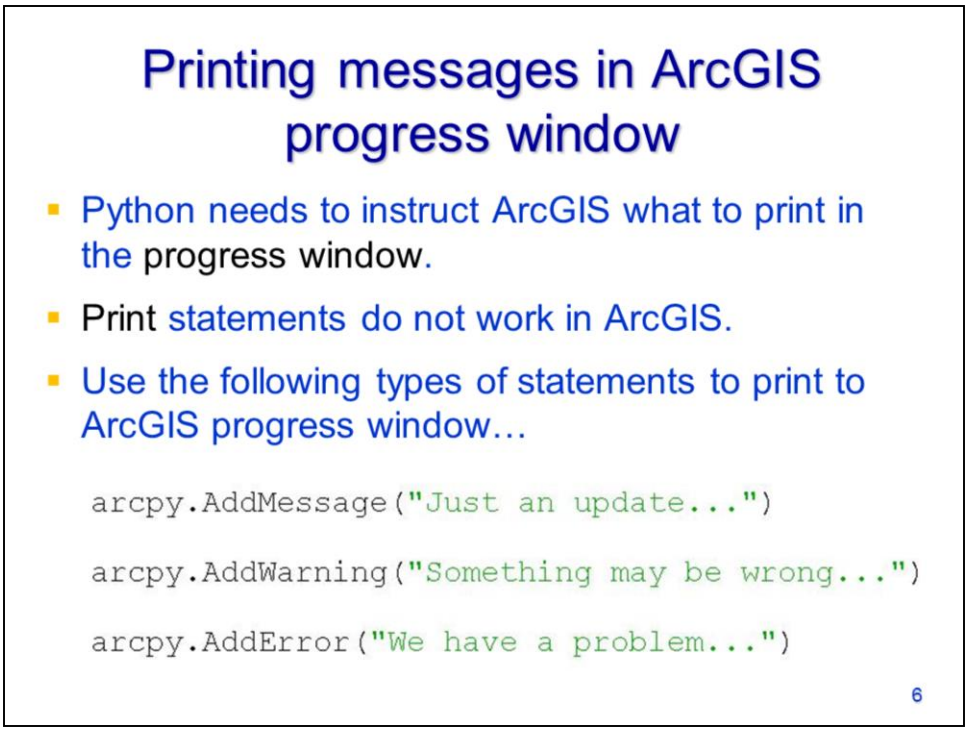
The script will try running the code in the **try** part of the statement. If an error occurs, then the script will immediately skip to the **except** part of the statement and run the code associated with it. If the code in the try part completes successfully, then the except part will be skipped.



When a script is run in Python, error messages are automatically reported in the Python Shell. However, when running a script in ArcGIS, the error message does not automatically print to ArcGIS's progress window.

The script needs to retrieve error messages and explicitly print them in ArcGIS. Errors and messages associated with ArcTool statements can be retrieved using arcpy's **GetMessage** method. Specifying a parameter value of 2 will return only the error messages; a value of 1 will get the warning messages; and a value of zero will get all messages.

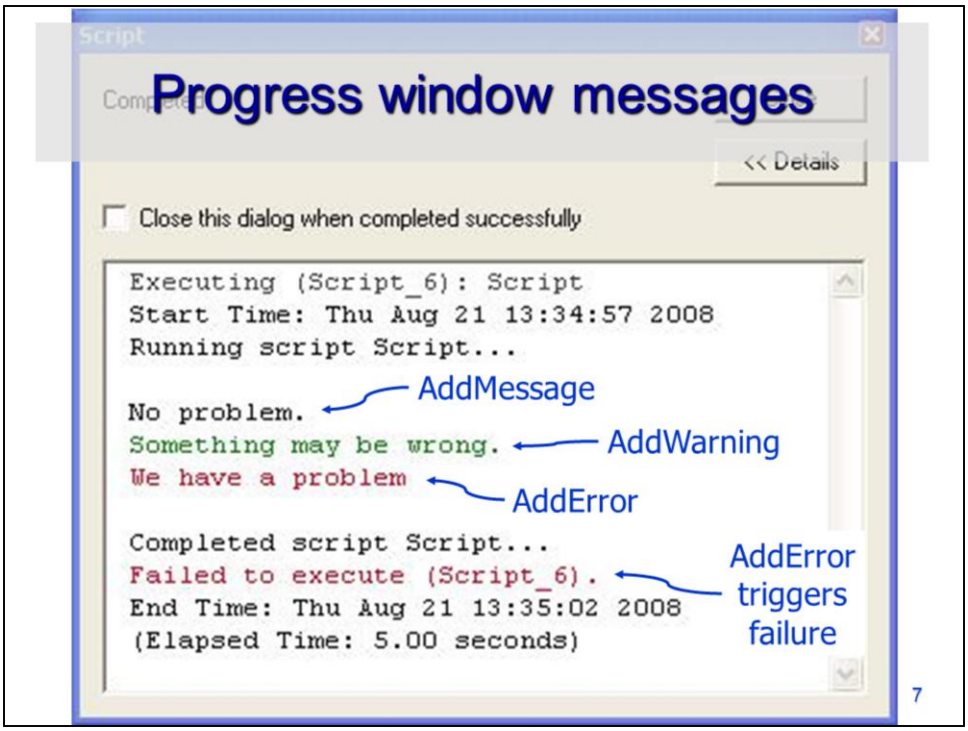
The Python error report can be obtained using the **traceback** module. The code provided here will get the type and location of the error in the script – this information is included in the string assigned to **pymsg.**



Messages only appear in ArcGIS's progress window when the script prints to it.

However, the Python print statement does not work in ArcGIS.

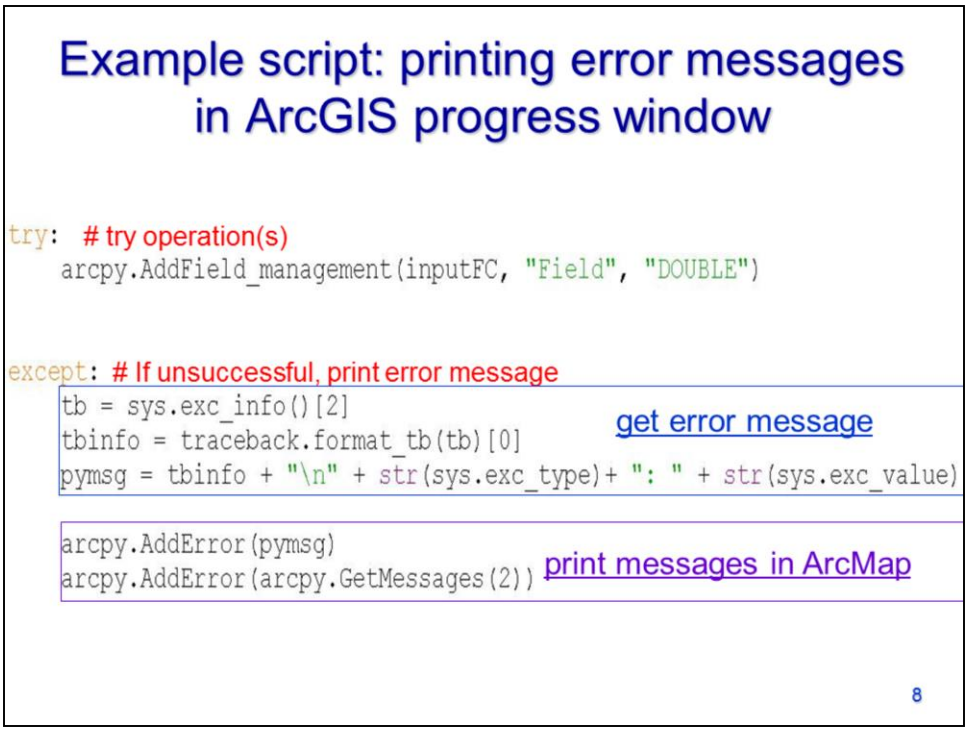
To print messages, warnings, and errors to the ArcGIS progress window, use arcpy's **AddMessage, AddWarning,** and **AddError** methods, respectively.



In the ArcGIS progress window…

- Messages appear in black text
- Warnings appear in green text
- Errors appear in red text.

The **AddError** method triggers the script to fail immediately.



This example script demonstrates how to use the try/except statement to catch errors and print the error report to the ArcGIS progress window.

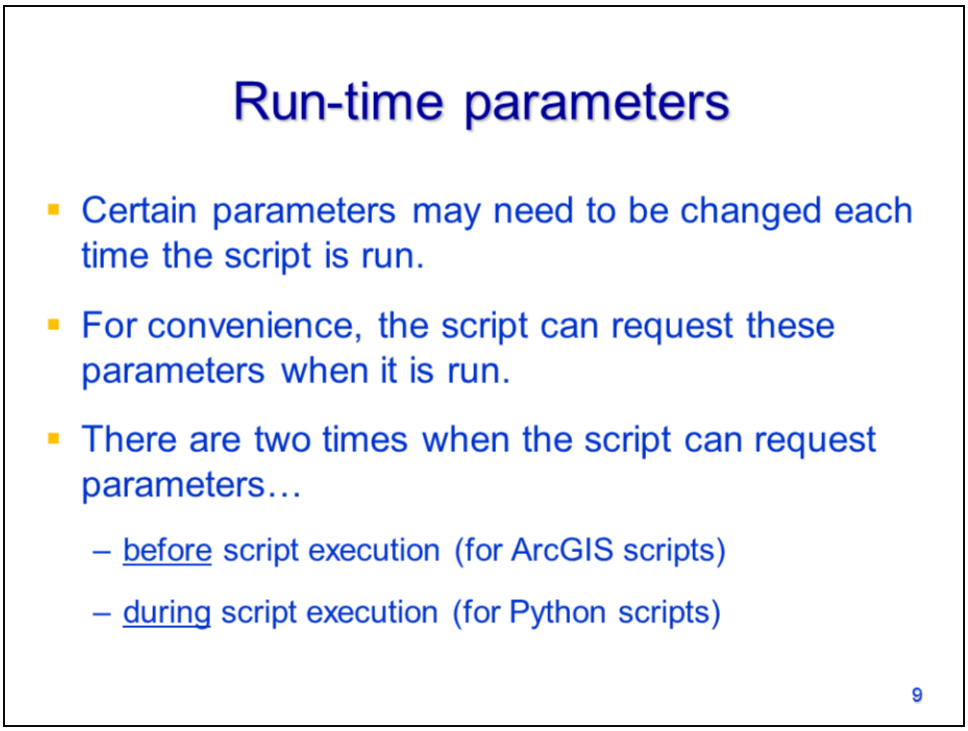
The script tries to run the code in the try part.

If an error is encountered, the except part executes.

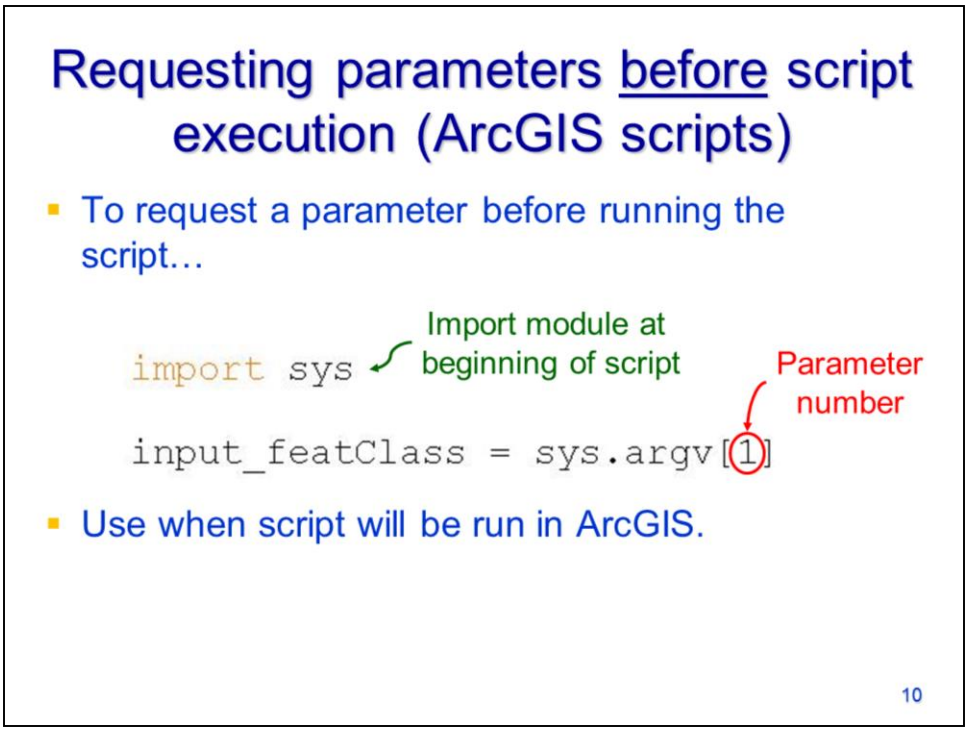
The error message information is retrieved using the traceback module – note that in this example, the traceback module was imported prior to the try/except statement.

The error message obtained from the traceback module is printed in ArcGIS

The error message, associated with any ArcTools, is obtained using the GetMessages method and printed in ArcGIS.



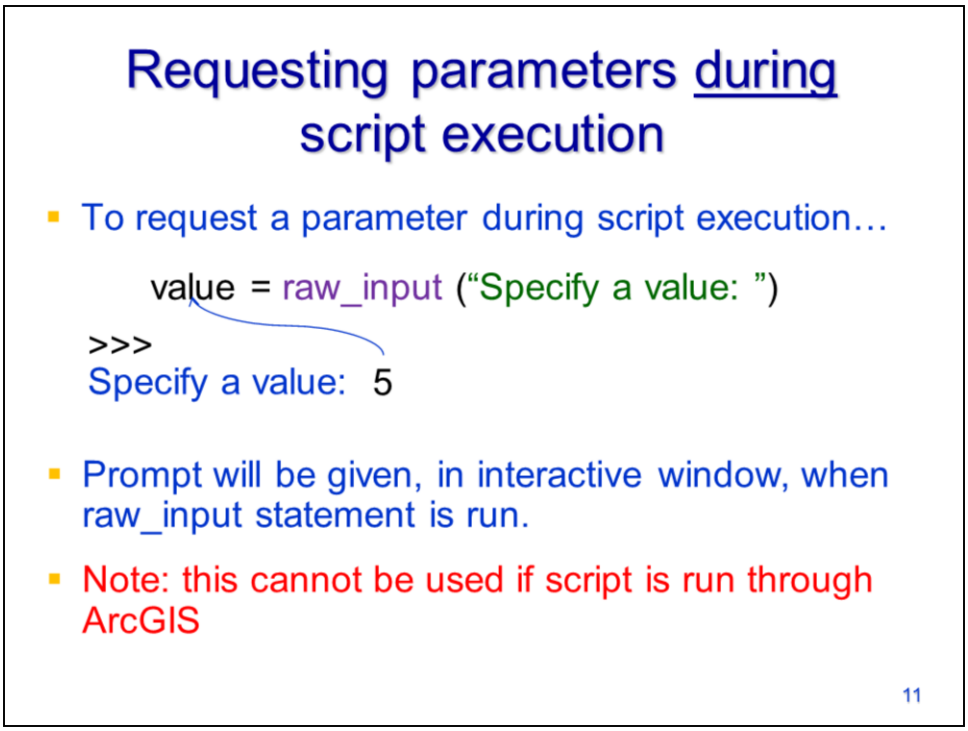
There are usually some parameters in a script that may need to be specified by the user when the script is run. The script can be set to request these user-defined parameters before the script is executed or during the script execution.



Scripts that will be imported into ArcGIS must request parameters before script execution. The sys module

To request parameters before the script is run, use the sys module's **argv** function. The parameter in square brackets corresponds to the order in which the parameter will be requested. The 1st value to be requested will have a parameter number of 1; the 2nd value to be requested will have a parameter number of 2, and so on.

The argv function is suitable to use with ArcGIS.



The **raw_input** command allows a parameter value to be requested when the statement executes in a script.

The prompt, specified in the raw_input statement, will print in the Python Shell and prompt the user to input a value.

The value entered by the user will be assigned to the variable in the raw_input statement.

The raw_input statement only works when the script is run in Python – it will not work for a script that is run in ArcGIS.