# Introduction to ArcPy

ArcGIS python package

# Slides content and code examples credits

The slides with python content and code snippets are taken from

- https://www.esri.com/training/catalog/57630436851d31e02a43f13c/python-for-everyone/
- http://desktop.arcgis.com/en/arcmap/10.4/analyze/arcpy/a-quick-tour-of-arcpy.htm

# Python and scripting environment

- Python is a scripting language commonly used to automate tasks.
- Advantage: it can be written and run in different environments, including ArcGIS.
- To edit and debug scripts, you can use an integrated development environment (IDE).
- There's no correct environment to use when writing and running scripts. It depends on the script task and personal preference.
- In the esri tutorial, the IDE used in ArcGIS 10.4 and 10.5 with Python 2.7 is PythonWin. This is a simple editor.
- In this training, we also took a look to another IDEs (Spyder and PyCharm) during the Intro to Python session.

# What is a Python script?

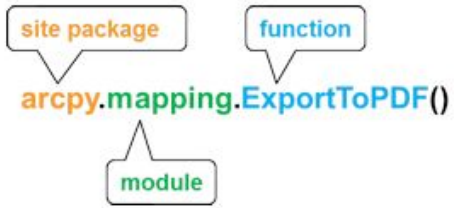
- It is a sequence of instructions.
- Structure and arrangement of instructions are referred to as syntax.
- Python uses a variety of language elements to create the instructions (data types, statements, and functions).
- Python script = recipe where
  - Data types = ingredients
  - Statements and functions = instructions

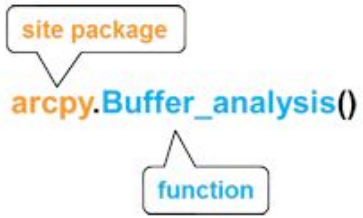
# Python in ArcGIS

- You can use a Python script, function, or statement inside ArcGIS to change labels, automate analyses, and more.
- In ArcGIS we can use the Python windows to
  - Run a python command
  - Load a python script and run it
- In PPG, the python scripts are normally run from the windows command prompt.

# ArcPy site package

- Python functionality in ArcGIS can range from a simple label expression to complex geoprocessing script.
- Geoprocessing functions, along with other GIS-related functions, are available through the ArcPy site package.
- The ArcPy site package is preloaded into ArcMap. Outside ArcMap, the site package acts a module which needs to be imported to make its functionality available.
- Examples:



# ArcPy modules

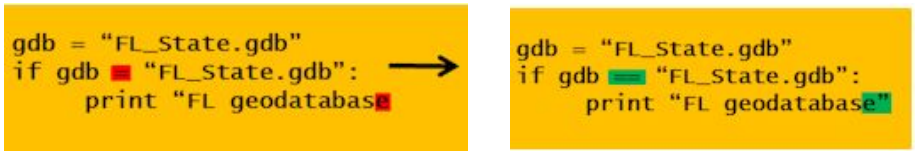
- ArcPy includes modules covering different areas of ArcGIS
  - arcpy.da - data access module
  - arcpy.mapping - mapping module
  - arcpy.sa - ArcGIS Spatial Analyst extension module
  - arcpy.na - ArcGIS Network Analyst extension module
- The tools from the arcpy.sa module use tools in the Spatial Analyst toolbox but are configure to support Map Algebra.
  - arcpy.sa.Slope is the same Slope tool from the Spatial Analyst toolbox.
- In PPG the most common modules used are arcpy.mapping and arcpy.sa

# Handling errors

- When developing and running script, errors may occur
- If Python encounters an error in a script, the first step is to determine which type of error has occurred to handle the errors.
- Two type of errors may occur when running python scripts
  - Syntax errors
  - Exceptions

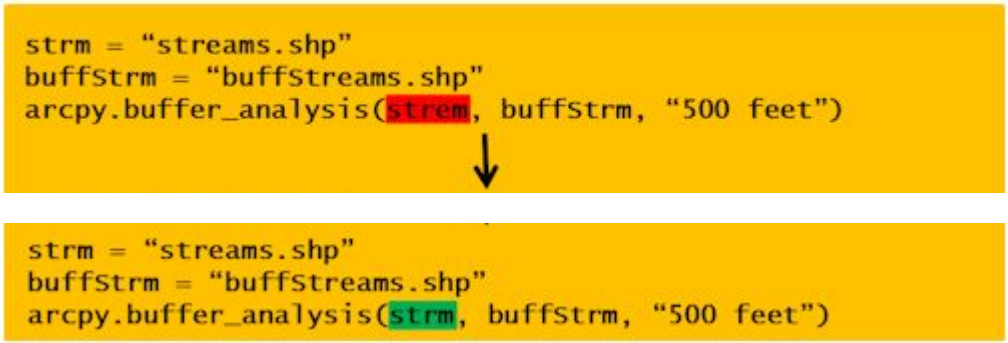
# Syntax errors

- A syntax error applies to a structural error in which a rule was broken or not enforced
- A helpful acronym for common syntax errors is ICCE
  - Indentation - spacing
  - Capitalization - case sensitivity
  - Closures - close of a syntax structure
  - Evaluations - conditional statements



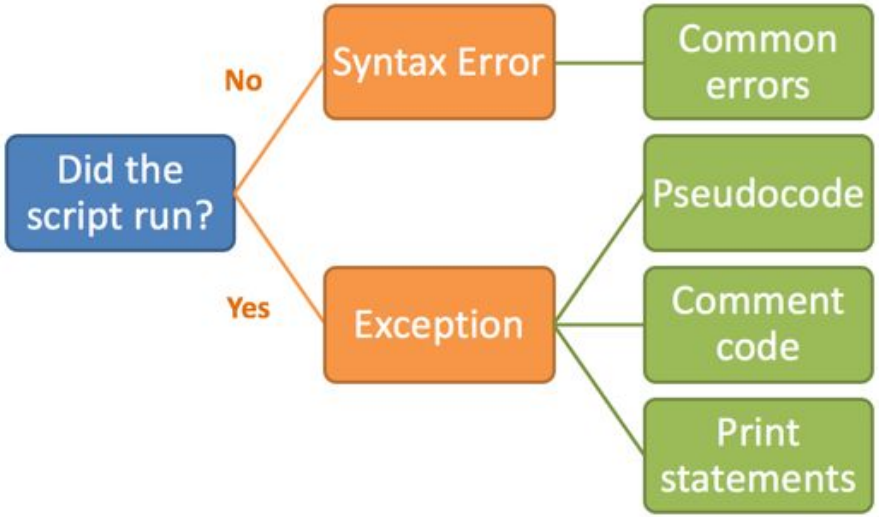
# Exceptions

- After correcting all syntax errors, the script can still fail and generate an error. These errors are referred as exceptions and are usually caused by errors during the execution of the script.
- Exceptions are typically raised because of an error in the logic or decision-making steps of your script. An exception can also occur if your script tries to work with data that it is not expecting.



# Basic debugging tips

After you determine the type of error encountered in the script, the next step is to debug. Using a debugging decision tree can help you debug your script.



# Basic debugging tips

- Syntax Error: common errors: use the ICCE concept

| | |
|---|---|
| Indentation | Confirm that any code within a looping or conditional statement is indented and that the indents line up. |
| Capitalization | Python is a case-sensitive language. Confirm that statements, such as print or import, are lowercase. |
| Closures | Confirm that lists, strings, and other data types are enclosed using the appropriate character (e.g., brackets for lists and quotation marks for strings). |
| Evaluations | Confirm that conditional statements (e.g., if statements) use two equal signs to evaluate a value. |

- Exception: pseudocode: review the code to determine if any steps are missing

```
#Assign variables
strm = r"C:/Data/Strms.shp"
buff = r"C:/Data/StrmsBuff.shp"

#Create 1000ft buffer
arcpy.Buffer_analysis(strm, buff, "1000 feet")
```

# Basic debugging tips

- Exception: Comment code to further narrow down the location of the error

```
#Assign variables
strm = r"C:/Data/Strms.shp"
buff = r"C:/Data/StrmsBuff.shp"

#Create 1000m buffer
#arcpy.Buffer_analysis(strm, buff, #1000 + "meters")
```

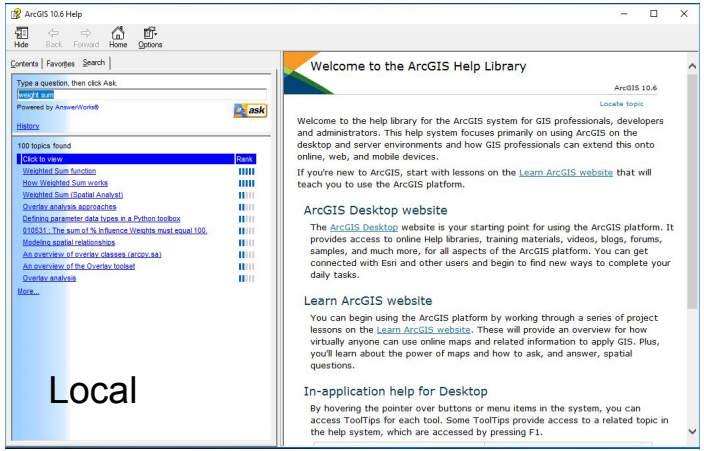
- Exception: Print statements to indicate when steps in the scripts are processing or variables has been assigned.

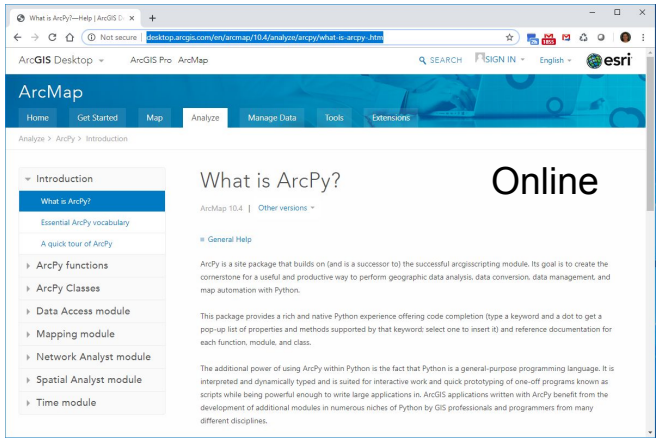
```
#Assign variables
strm = r"C:/Data/Strms.shp"
path = r"C:/Data/StrmsBuff"
buffList = ["100", 500, "700"]

#Create 100, 500, 700 ft buffers
for buff in buffList:
     print (buff + " processing")
     arcpy.Buffer_analysis(strm, path + buff + ".shp", buff + "feet")
```

# ArcGIS Desktop Help

- Provides access to the complete documentation which includes information about arcpy, geoprocessing tool, etc.
- There are 2 ways to get access: local or online
- Local: Start > ArcGIS > ArcGIS 10.x Help
- Online: http://desktop.arcgis.com/en/arcmap/10.4/analyze/arcpy/what-is-arcpy-.htm



# Exercise

- Python for Everyone exercises
- Take Exam to get esri tutorial certificate (recommended)

# References

- Python for Everyone - esri training - requires an esri account - https://www.esri.com/training/catalog/57630436851d31e02a43f13c/python-for-everyone/
- A quick tour of ArcPy - ArcGIS Desktop - http://desktop.arcgis.com/en/arcmap/10.4/analyze/arcpy/a-quick-tour-of-arcpy.htm
- What is ArcPy http://desktop.arcgis.com/en/arcmap/10.3/analyze/arcpy/what-is-arcpy-.htm