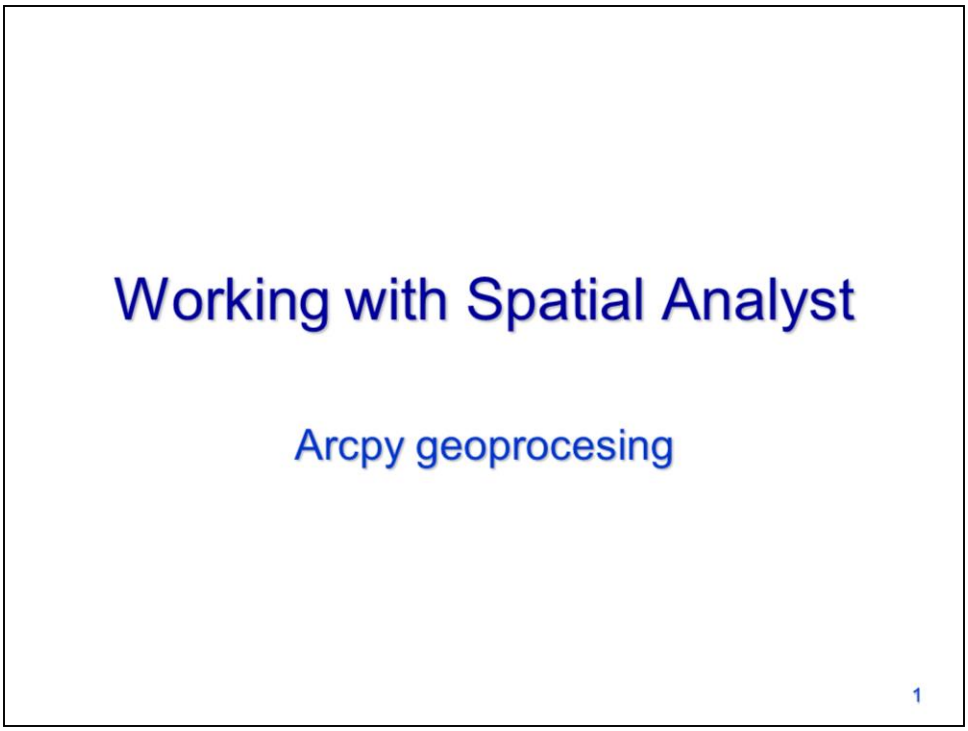
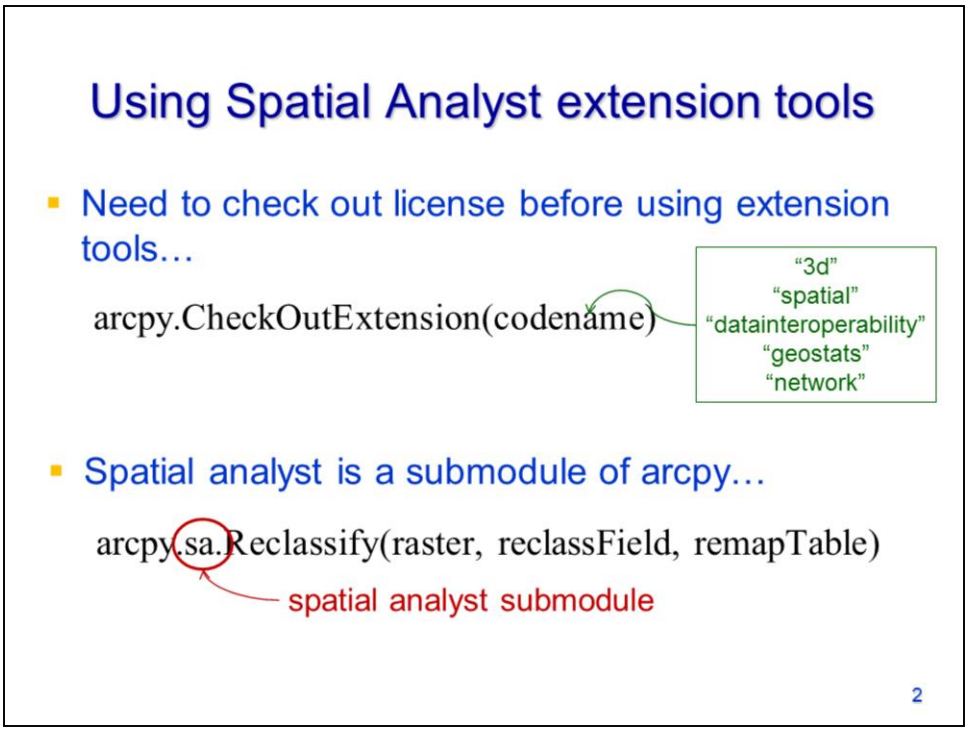


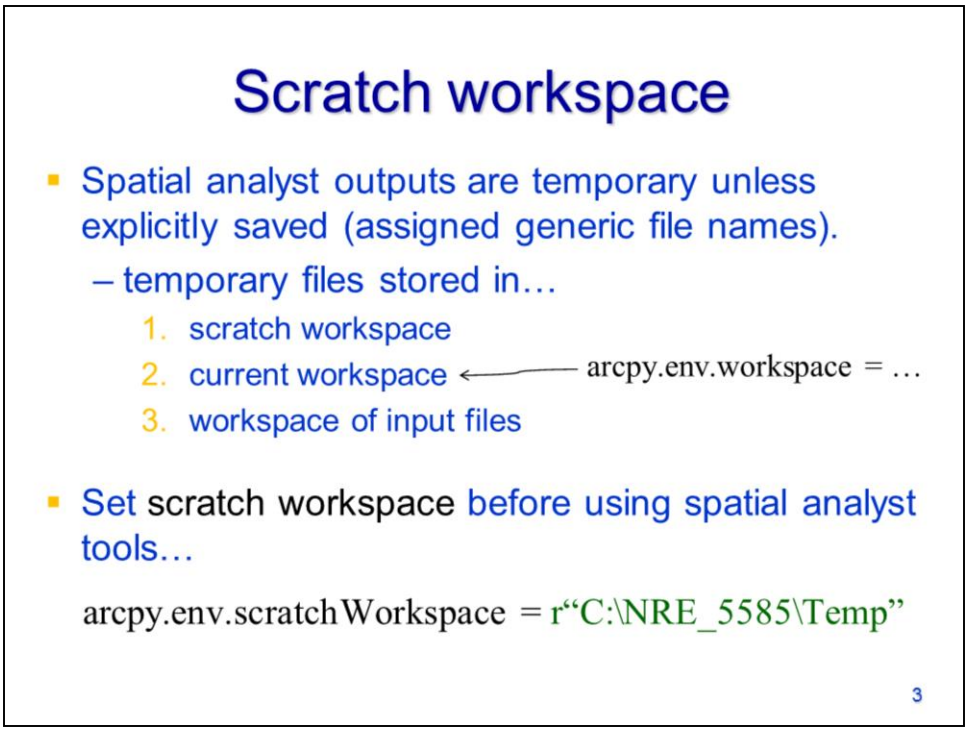
This video will discuss how to use ArcGIS's Spatial Analyst extension in a Python script.



The Spatial Analyst extension contains most of ArcGIS's tools for working with raster data.

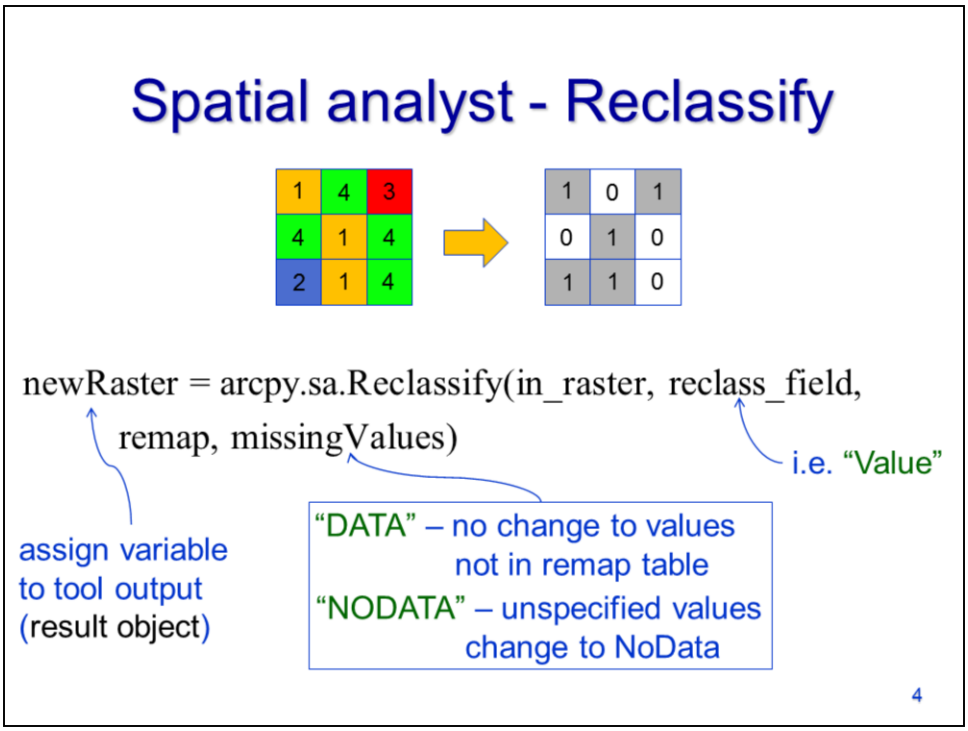
In order to use any ArcGIS extension in a script, you must first check out the appropriate extension license.

The tools in the Spatial Analyst extension use a slightly different syntax than other Arcpy tools which we'll see later in this video. The Spatial Analyst tools are contained within the **sa** submodule of arcpy.



The outputs of spatial analyst tools are temporary unless explicitly saved. These temporary files are stored in the following locations, in order of priority: 1) arcpy's scratch workspace, 2) arcpy's workspace, or 3) the workspace of the input files.

The **scratch workspace** should be set when working with spatial analyst to avoid cluttering the input workspace with temporary files. The scratch workspace is set through arcpy's **env** properties.

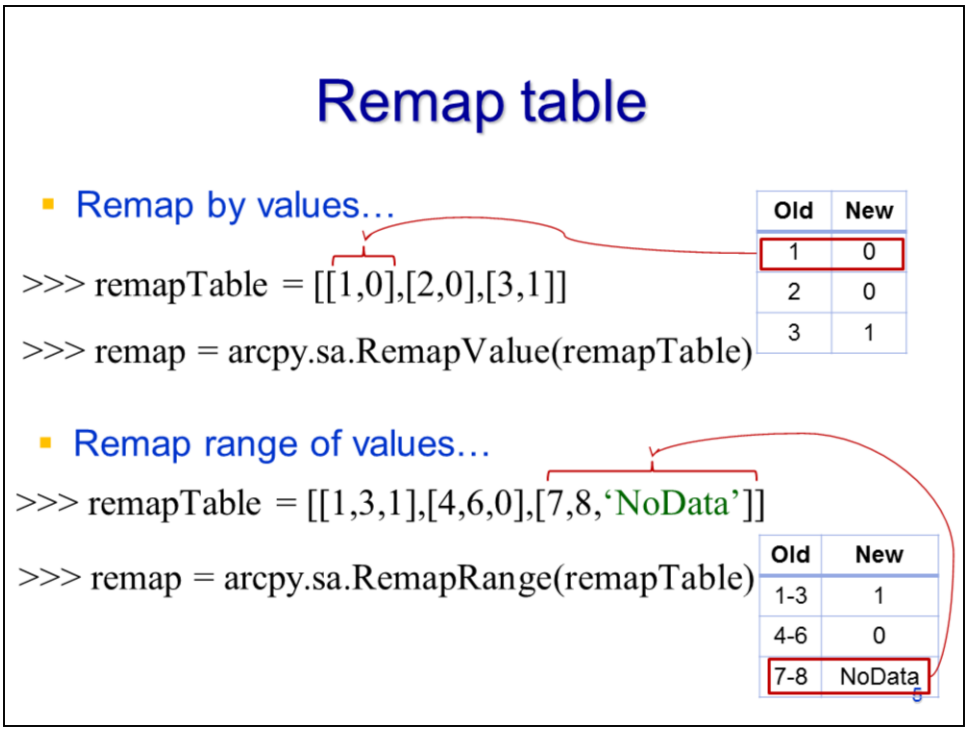


We'll use **Reclassify** tool to show the syntax needed for Spatial Analyst tools. This tool allows the pixel values of a raster to be changed and is commonly used in raster-based analyses.

The spatial analyst tools create a result object as their output – this requires that a variable be assigned to the tool output. Note that variables can be assigned to the outputs of other non-spatial analyst ArcTools for convenience, but it is not required for those tools.

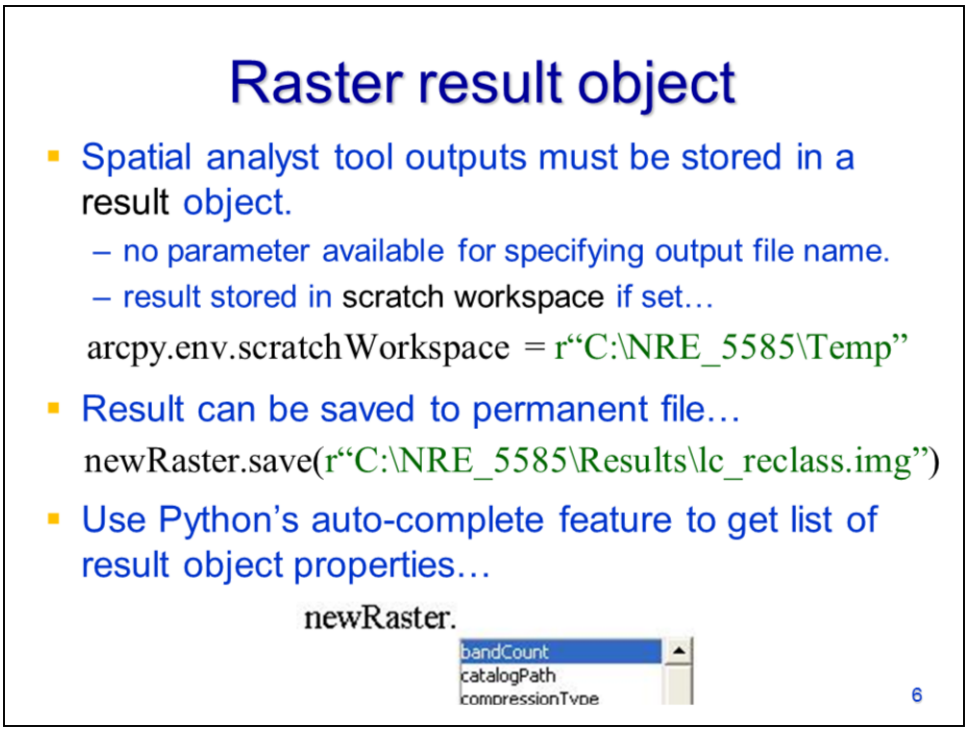
The **reclass_field** is typically the "Value" field which contains the original pixel values. The **remap** parameter requires a **remap object** that indicates the new values that should be assigned to the pixels.

The **missingValues** parameter indicates how to handle pixels whose values were not specified in the remap table – the pixels can either be left alone (the "DATA" option) or they can be changed to NoData (the "NODATA" option).



The **remap** object contains the reclassify table which indicates the new values that should be used to replace the original pixel values. To create the remap object, a 2 level list must be created. The top-level list contains the entire reclass table; the bottom lists contain an old pixel value followed by the new pixel value. The remap object can be created with spatial analyst's **RemapValue** tool using the 2 level list.

The **RemapRange** tool creates a remap object that allows ranges of old values to be assigned a new value. This tool also requires a 2 level list. The bottom level lists indicate the start and end ranges and end ranges for the original values followed by the new values. Note that "NoData' is an acceptable new value.

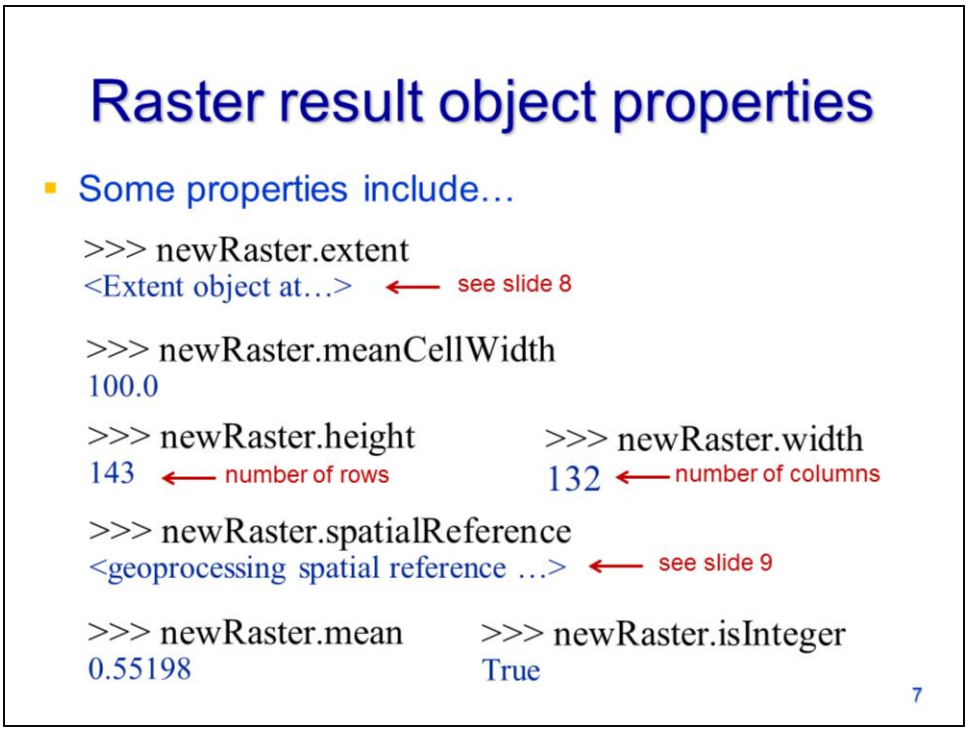


The output of a spatial analyst tool is a **raster result object** which stores information about the temporary raster file. The actual output dataset is stored in the scratch workspace under an automatically generated name.

Make sure to set the **scratch workspace** when using spatial analyst tools.

A dataset created by a spatial analyst tool can be saved using the **save** method of the raster result object. The extension of the output file will determine the format of the saved file. Omitting the extension will create an ESRI grid file for which has a file name limited to 13 characters in length. The .img format is generally a good format choice for raster data because it has no restrictions on the length of file names and the files are easily shared.

Raster result objects provide convenient access to a number of properties of the raster.



The raster result object provides access to a number of raster properties including:

- The **extent object**
- The **meanCellWidth** and **meanCellHeight** – these properties are equivalent for most rasters which have square pixels.
- The **height** of the raster which is number of rows it contains
- The **width** of the raster which is the number of columns it contains
- The **spatialReference** object
- The **mean** of all pixel values
- And whether the pixel values are integer (True) or decimal format (False).

Example script – reclassifying a raster

```
# check out extension license…
arcpy.CheckOutExtension('spatial')

# create remap object…
remap = arcpy.sa.RemapValue([[5,1],[6,1]])

# reclassify inRaster. Values not in remap change to NoData…
newRaster = arcpy.sa.Reclassify(inRaster, "Value", remap,
   "NODATA")

# save result object…
newRaster.save(r"C:\NRE_5585\Results\outputRas.img")
```

8

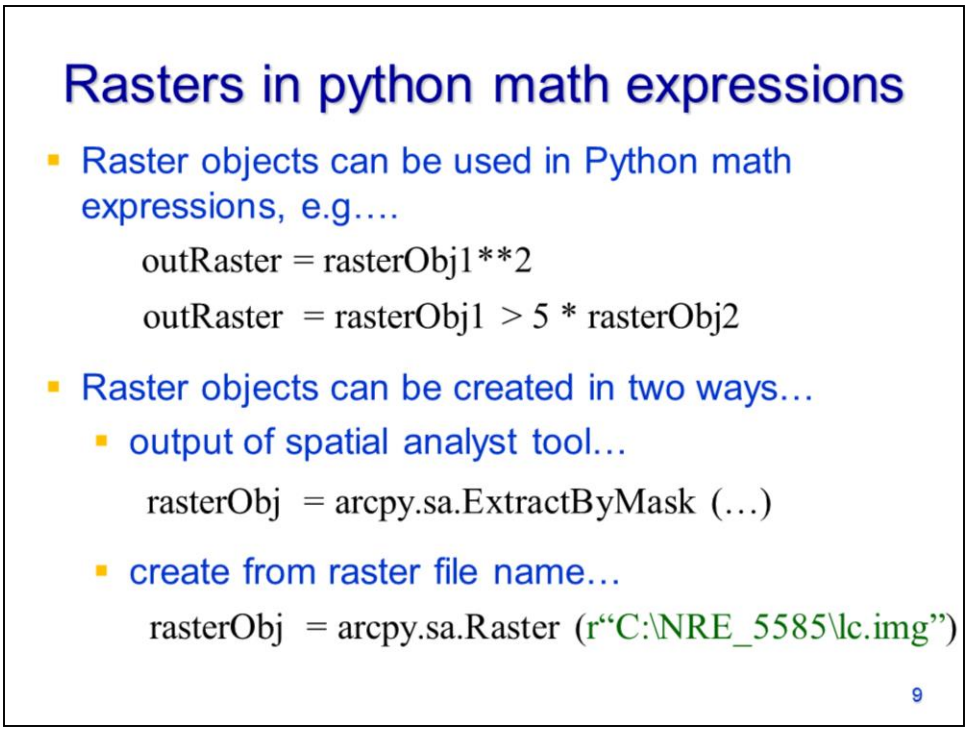
This slide shows an example of code that uses spatial analyst's **reclassify** tool.

The code first checks out the spatial analyst extension license.

Next, a remap object is created in which pixels with original values of 5 will be assigned new values of 1 and pixels with original values of 6 will be assigned new values of 1.

The Reclassify tool is run and any pixels with original values not specified in the remap object will be assigned the NODATA value in the output raster. The Reclassify output is stored in the newRaster result object.

The newRaster is saved as a permanent dataset.



Raster objects can be used with Python operators.

- Operators that can be used on raster objects include math operations such as multiplication, division, addition, subtraction, and exponents…

- as well as tests for equality, inequality, etc.

Raster objects are created as the output for spatial analyst tools.

File names can be used to create raster objects with spatial analyst's Raster Tool.

Example script – raster math

```
# check out extension license…
arcpy.CheckOutExtension('spatial')

# raster file names…
lc_name = r"C:\NRE_5585\Data\land_cover.img"
height_name = r"G:\Lidar\Height\height.img"

# create raster objects…
lc = arcpy.sa.Raster(lc_name )
ht = arcpy.sa.Raster(height_name)

# math expression (clip height to lc pixels with value of 5)…
forestHt = lc == 5 * ht

# save result object…
forestHt.save (r"C:\NRE_5585\Results\forestHt.img")
```

10

This script will show an example of using Python operators with raster objects.

Raster objects are created using the **Raster** tool on the file names.

Pixels with a value of 5 are extracted from the **lc** raster object – this creates a binary raster in which all pixels with original values of 5 become 1 and all other original pixel values become zero. The resulting binary raster is then multiplied by the height raster. This multiplication essentially clips the height raster to the pixels that had an original value of 5 in the lc raster; all other lc values receive a height of zero in the forestHt raster.

The forestHt raster is then saved as a permanent .img file.