

```
#####
# Description: This script is being created to find suitable areas to grow grapes in the
#             Annapolis Royal area. The datasets that this script takes a look at are Slope
#             (created by a DEM), Aspect (created by a DEM), Canada Land Inventory (CLI),
#             Properties, and Buildings. Then the following tools were used to create the
#             final output. Clip(x2), Aspect, Slope, Area Solar Radiation, Reclassify, Join
#             Field, Make Feature Layer(x7), Copy Features(x3), Select Analysis(x3)), Int(x2),
#             Raster to Polygon(x4), Merge, Select Layer By Location(x2), Polygon to Raster(x2),
#             Raster Calculator(x3), and Search Cursor. The final output shows which properties are
#             most suitable to start growing grapes. This final output is based on south/southwest
#             facing properties, based on the slope, if the property has a building on it or not,
#             how much sun reaches the properties and a number of other things.
#
# Name: Grape_Growing_Katie.py
# Created by: Katie Chute
# Created: December 2015
#
#Geoprocessing and Problem Solving 1 - Week 14
#####

import arcpy

arcpy.CheckOutExtension("Spatial")

from arcpy import *
from arcpy.sa import *

env.workspace =arcpy.GetParameter(0)

arcpy.AddMessage("Environment Settings set")

#Environment Overwrite settings are set to True.
arcpy.env.overwriteOutput = True

arcpy.AddMessage ("Overwrite Ouput set to True")

#Setting Variables
##Workspace= r"D:/Python/Data"
Clip_Feature= arcpy.GetParameter(1)
##Clip_Feature= "D:/Python/Data/AnnapolisRoyal_Clip2015.shp"
DEM = arcpy.GetParameter(2)
##DEM = "dem020hy1.img"
CLI_beforeclip= arcpy.GetParameter(3)
##CLI_beforeClip= "a021a.shp"
Join_Table= arcpy.GetParameter(4)
##Join_Table="a021a_oldpat.dbf"

try:
    # Replace a layer/table view name with a path to a dataset (which can be a layer file) or
    create the layer/table view within the script
    # The following inputs are layers or table views: "dem020hy1.img", "AnnapolisRoyal_Clip2015"
    arcpy.Clip_management(DEM,"298220.193980415 4954814.49748954 313220.193980415 4964814.49748954",
    "Anna_DEM.img", Clip_Feature,"-32768", "NONE", "NO_MAINTAIN_EXTENT")
    # Replace a layer/table view name with a path to a dataset (which can be a layer file) or
    create the layer/table view within the script
    # The following inputs are layers or table views: "Anna_DEM"
    arcpy.gp.Aspect_sa("D:/Python/Data/Anna_DEM.img", "D:/Python/Data/Anna_Aspect.img")
    # Replace a layer/table view name with a path to a dataset (which can be a layer file) or
    create the layer/table view within the script
    # The following inputs are layers or table views: "Anna_DEM"
    arcpy.gp.Slope_sa("Anna_DEM.img", "D:/Python/Data/Anna_Slope.img", "PERCENT_RISE")

except:
```

```

#Print statement saying that the DEM has been clipped to our study area
arcpy.AddMessage("Clipping DEM down to study area completed")
#Print statement saying that the Aspect has been created successfully
arcpy.AddMessage("Creating Aspect completed")
#Print statement saying that the Slope has been created successfully
arcpy.AddMessage("Creating Slope completed")
#Prints any errors that come up
arcpy.GetMessage(2)

try:
    # Replace a layer/table view name with a path to a dataset (which can be a layer file) or
    create the layer/table view within the script
    # The following inputs are layers or table views: "Anna_DEM"
    arcpy.gp.AreaSolarRadiation_sa("Anna_DEM.img", "Anna_Solar.img", "44.7652387459911", "200",
"MultiDays    2015    121    303", "14", "0.5", "NOINTERVAL", "1", "FROM_DEM", "32", "8", "8",
"STANDARD_OVERCAST_SKY", "0.3", "0.5", "", "", "")
    # Replace a layer/table view name with a path to a dataset (which can be a layer file) or
    create the layer/table view within the script
    # The following inputs are layers or table views: "anna_solar"
    arcpy.gp.Reclassify_sa("Anna_Solar.img", "Value", "676465.6875 776580.67377450969
1;776580.67377450969 799187.28357843123 2;799187.28357843123 816142.24093137239
3;816142.24093137239 837134.09289215668 4;837134.09289215668 882347.3125 5",
"D:/Python/Data/Solar_Reclass", "NODATA")
    # Replace a layer/table view name with a path to a dataset (which can be a layer file) or
    create the layer/table view within the script
    # The following inputs are layers or table views: "a021a", "AnnapolisRoyal_Clip2015"
    arcpy.Clip_analysis(CLI_beforeclip, Clip_Feature, "D:/Python/Data/AnnaCLI_Clip.shp", "")

except:
    #Print statement saying that the Area Solar Radiation has been completed
    arcpy.AddMessage("Area Solar Radiation creation completed")
    #Print statement saying that the reclassifying of the Area Solar Radiation output has been
    completed
    arcpy.AddMessage("Reclassifying the solar output completed")
    #Print Statement saying that the CLI has been clipped to our study area has been completed
    arcpy.AddMessage("Clipping CLI to study area completed")
    #Prints any errors that come up
    arcpy.GetMessage(2)

try:
    # Replace a layer/table view name with a path to a dataset (which can be a layer file) or
    create the layer/table view within the script
    # The following inputs are layers or table views: "AnnaForest_Clip", "a021a_oldpat"
    arcpy.JoinField_management("AnnaCLI_Clip.shp", "FID", Join_Table, "OID", "")
    #Creating a layer file from the joined CLI data above
    arcpy.MakeFeatureLayer_management("AnnaCLI_Clip.shp", "AnnaCLI.lyr")
    #Creating a new shapefile from the above layer file
    arcpy.CopyFeatures_management("AnnaCLI.lyr", "Anna_CLI.shp")

except:
    #Print statements saying that the Join has been completed
    arcpy.AddMessage("Joining the CLI to an external table completed")
    #Print statement saying that the creation of the layer file has been completed.
    arcpy.AddMessage("Creating a layer file completed")
    #Print statement saying that the new shapefile has been created successfully
    arcpy.AddMessage("Creating a new shapefile from the layer file completed")
    #Prints any errors that come up
    arcpy.GetMessage(2)

try:
    # Replace a layer/table view name with a path to a dataset (which can be a layer file) or
    create the layer/table view within the script
    # The following inputs are layers or table views: "AnnaForest_Clip"

```

```

    # arcpy.Select_analysis(in_features="AnnaCLI_Clip.shp", out_feature_class="Class2_3.shp",
where_clause="CLASS_A = '2' OR CLASS_A = '3'")
    arcpy.Select_analysis("AnnaCLI_Clip.shp", "Class2_3.shp", "CLASS_A = '2' OR CLASS_A = '3'")
    # Replace a layer/table view name with a path to a dataset (which can be a layer file) or
create the layer/table view within the script
    # The following inputs are layers or table views: "Anna_Aspect
    arcpy.gp.Int_sa("Anna_Aspect.img", "D:/Python/Data/Aspect_Int")
    # Replace a layer/table view name with a path to a dataset (which can be a layer file) or
create the layer/table view within the script
    # The following inputs are layers or table views: "Anna_Slope
    arcpy.gp.Int_sa("Anna_Slope.img", "D:/Python/Data/Slope_Int")

except:
    #Selecting out all of the CLI that are listed under either class 2 or 3
    arcpy.AddMessage("Selecting out all of Class 2 and Class 3 from the CLI data completed")
    arcpy.AddMessage("Creating an Int using the Aspect completed")
    arcpy.AddMessage("Creating an Int using the Slope completed")
    #Prints any errors that come up
    arcpy.GetMessage(2)

try:
    #Taking the Aspect raster and turning it into a Polygon shapefile.
    gp.RasterToPolygon("Aspect_Int", "Aspect", "SIMPLIFY", "VALUE")
    #Taking the Slope raster and turning it into a Polygon shapefile.
    gp.RasterToPolygon("Slope_Int", "Slope", "SIMPLIFY", "VALUE")
    #Selecting all the slope between 5 and 15 percent
    arcpy.Select_analysis("Slope.shp", "Slope5_15.shp", "GRIDCODE >= 5 AND GRIDCODE <= 15")

except:
    #Print statements
    arcpy.AddMessage("Converting the Aspect Raster to a Polygon shapefile completed")
    arcpy.AddMessage("Converting the Slope Raster to a polygon shapefile completed")
    arcpy.AddMessage("Selecting all the slope between 5 and 15 percent completed")
    #Prints any errors that come up
    arcpy.GetMessage(2)

try:
    #Selecting all the Aspect between 112.5 and 247.5
    arcpy.Select_analysis("Aspect.shp", "Aspect_south.shp", "GRIDCODE >= 112.5 AND GRIDCODE <=
247.5")
    #Merging the slope and aspect shapefiles together to make the later processing easier
    arcpy.Merge_management(("Slope5_15.shp", "Aspect_south.shp"), "Slope_Aspect.shp")
    #Creating a new layer file to work with and to be converted into a new shapefile later on in
script
    arcpy.MakeFeatureLayer_management("Slope_Aspect.shp", "Slope_Aspect.lyr")

except:
    #Print statements
    arcpy.AddMessage("Selecting all the Aspect between 112.5 and 247.5 completed")
    arcpy.AddMessage("Merging the slope and aspect shapefiles together completed")
    arcpy.AddMessage("Creating a new layer file completed")
    #Prints any errors that come up
    arcpy.GetMessage(2)

try:
    #Creates a layer file from the CLI Class 2 and 3 shapefile
    arcpy.MakeFeatureLayer_management("Class2_3.shp", "Class2_3.lyr")
    #Select layer by location on the Merged shapefile (that is now a layer file) aby using the
class2_3 layer file
    arcpy.SelectLayerByLocation_management("Slope_Aspect.lyr", "COMPLETELY_CONTAINS", "Class2_3.lyr",
"", "NEW_SELECTION", "NOT_INVERT")
    #Creates a new shapefile that contains the merged slope and aspect that completely contains
class 2 and 3
    arcpy.CopyFeatures_management("Class2_3.lyr", "SlopeAspect_CLI.shp")

```

```

except:
    #Print statements
    arcpy.AddMessage("Make Feature Layer completed")
    arcpy.AddMessage("Select Layer By Location completed")
    arcpy.AddMessage("Copy Features completed")
    #Prints any errors that come up
    arcpy.GetMessage(2)

try:
    #Converts polygon to raster
    gp.PolygonToRaster_conversion("SlopeAspect_CLI.shp", "CLASS_A", "SlopeAspCLI_Suit.img", "", "",
    "20")
    #Raster Calculator adding the Solar Radiation reclass to the Slope Aspect CLI shapefile
    outGrapeSuit = Raster("Solar_Reclass.img") + Raster("SlopeAspCLI_Suit.img")
    outGrapeSuit.save ("Grape_Suit.img")
    #Creating layer files for the Properties and Building shapefiles
    arcpy.MakeFeatureLayer_management("Property.shp", "Prop.lyr")
    arcpy.MakeFeatureLayer_management("Buildings.shp", "Build.lyr")
    #Creating shapefiles from the layer files
    arcpy.CopyFeatures_management("Prop.lyr", "Properties.shp")
    arcpy.CopyFeatures_management("Build.lyr", "Builds.shp")

except:
    #Print statements
    arcpy.AddMessage("Converting Polygon to Raster completed")
    arcpy.AddMessage("Make Feature Layer for the Property and Building data completed")
    arcpy.AddMessage("Copy Features Properties and Builds completed")
    #Prints any errors that come up
    arcpy.GetMessage(2)

try:
    # Replace a layer/table view name with a path to a dataset (which can be a layer file) or
    create the layer/table view within the script
    # The following inputs are layers or table views: "Property", "Building"
    arcpy.SelectLayerByLocation_management(in_layer="Prop.lyr", overlap_type="COMPLETELY_CONTAINS",
select_features="Build.lyr", search_distance="", selection_type="NEW_SELECTION",
invert_spatial_relationship="NOT_INVERT")
    #Creating a new layer file from another layer file
    arcpy.MakeFeatureLayer_management("Prop.lyr", "GrapeProp.lyr")
    #Creating a new shapefile from the layer file from the Make Features Layer tool above.
    arcpy.CopyFeatures_management("GrapeProp.lyr", "Grape_Prop.shp")

except:
    #Print Statements
    arcpy.AddMessage("Select Layer By Location completed")
    arcpy.AddMessage("Make Feature Layer completed")
    arcpy.AddMessage("Copy Feature completed")
    #Prints any errors that come up
    arcpy.GetMessage(2)

try:
    #Converting the Grape_Prop shapefile to a raster imagen file
    gp.PolygonToRaster_conversion("Grape_Prop.shp", "PID", "Prop_Grape.img", "", "", "20")
    #Set Prop_Grape.img to IsNULL
    arcpy.gp.RasterCalculator_sa("Con(IsNull("Prop_Grape.img"),0,"Grape_Suit.img")""",
"D:/Python/Data/PropGrape_NULL.img")
    #Subtract property from GrapeSuit
    arcpy.gp.RasterCalculator_sa("Con("Grape_Suit.img" - "PropGrape_NULL.img","Grape_Suit.img")""",
, "D:/Python/Data/Grape_Final.img")

except:
    #Print Statements
    arcpy.AddMessage("Converting Polygon to Raster completed")

```

```
arcpy.AddMessage("Raster Calculator Con IsNnull completed")
arcpy.AddMessage("Raster Calculator Con completed")
#Prints any errors that come up
arcpy.GetMessage(2)

try:
    #Search Cursor
    cursor= da.SearchCursor("Grape_Final.img",["OID","VALUE","COUNT"])
    for row in cursor:
        print ("{0},{1},{2}".format(row[0],row[1],row[2]))

except:
    import traceback                                #
    tb = sys.exc_info()[2]                          # Get the traceback object
    tbinfo = traceback.format_tb(tb)[0]             # this contains the error line number
    #Prints any errors that come up
    arcpy.GetMessages(2)                            # print any error messages to the toolbox window
    #Prints that the Search Cursor has been completed
    arcpy.AddMessage("Search Cursor Completed!!!!")

#Prints that the script has finished
print ("Script Complete")
```