# Exercise 1 Introducing Python

## Install the exercise data

## Check ArcGIS settings

## Check the Python version

## Install PythonWin

## Work with different script editors

## Use the ArcGIS Python window

# Exercise 2 Geoprocessing in ArcGIS

## Examine toolboxes and tools

## Run a tool

## Conduct batch processing

## Set the geoprocessing options

## Explore models and ModelBuilder

## Run your model

## Use scripting for geoprocessing

## Use scripts as tools

## Convert a model to a script

## Challenge exercise

### Challenge 1

Create a new model called Soil Analysis that accomplishes the following:

1. Clips the soils layer using the basin layer
2. From the clipped version of the soils layer, selects the features that are “Not prime farmland” (field FARMLNDCL)

Convert the model to a script called soil.py.

# Exercise 3 Using the Python window

## Open the Python window

## Write and run code

## Run a geoprocessing tool

## Get help in the Python window

## Save your work

## Load existing code

# Exercise 4 Learning Python language fundamentals

## Work with numbers

## Work with strings

## Work with variables

All scripting and programming languages work with variables.

A variable is basically a name that represents or refers to a value.

Variables store temporary information that can be manipulated and changed throughout a script.

Many programming languages require that variables be declared before they can be used.

Declaring means that you fist create a variable and specify what type of variable it is—and only then can you actually assign a value to that variable.

In Python, you immediately assign a value to a variable (without declaring it), and from this value, Python then determines the nature of the variable.

This typically saves a lot of code and is one reason why Python scripts are often much shorter than code in other programming languages.

## Work with lists

## Use functions

## Use methods

## Use modules

Hundreds of additional functions are stored in modules.

Before you can use a function, you have to import its module using the import function.

The functions you used in the preceding sections are part of Python’s built-in functions and don’t need to be imported.

One of the most common modules to import is the math module, so you’ll start with that one.

## Save Python code as scripts

## Write conditional statements

## Use loop structures

## Comment scripts

## Check for errors

## Challenge exercises

# Exercise 5 Geoprocessing using Python

# Exercise 6 Exploring spatial data

# Exercise 7 Manipulating spatial data

## Work with search cursors

## Use search cursors with SQL in Python

## Work with update cursors

## Work with insert cursors

## Validate table and field names

## Challenge exercises

### Challenge 1

Write a script that creates a 15,000-meter buffer around features in the airports.shp feature class classified as an airport (based on the FEATURE field) and a 7,500-meter buffer around features classified as a seaplane base.

The results should be two separate feature classes, one for each airport type.

### Challenge 2

Write a script that adds a text field to the roads.shp feature class called FERRY and populates this field with YES and NO values, depending on the value of the FEATURE field.

# Exercise 8 Working with geometries

## Work with geometry objects

Working with full geometry objects can be costly in terms of time.

Geometry tokens provide shortcuts to specific geometry properties of individual features in a feature class.

In the next example, you will see how to work with geometry tokens in scripting.

## Read geometries

In addition to the geometry properties of a feature, a feature’s individual vertices can also be accessed.

You’ll do that next.

## Work with multipart features

## Write geometries

## Challenge exercises

### Challenge 1

Write a script that creates a new polygon feature class containing a single (square) polygon with the following coordinates: (0, 0), (0, 1,000), (1,000, 0), and (1,000, 1,000).

### Challenge 2

Write a script that determines the perimeter (in meters) and area (in square meters) of each of the individual islands of the Hawaii.shp feature class.

Recall that this is a multipart feature class.

### Challenge 3

Write a script that creates an envelope polygon feature class for the Hawaii.shp feature class.

There is actually a tool that accomplishes this called Minimum Bounding Geometry.

You can look at the tool to get some ideas, but your script needs to work directly with the geometry properties.

# Exercise 9 Working with rasters

## List the rasters

The ListRasters function can be used to list all the rasters in a workspace.

## Describe the rasters

The Describe function can be used to describe raster properties.

## Use raster objects in geoprocessing

To use the Spatial Analyst geoprocessing tools, you need an ArcGIS Spatial Analyst license.

When working in the Python window, you can set this from within ArcMap.

## Use map algebra operators

The arcpy.sa module contains a number of map algebra operators, which you’ll use next.

These operators make it easier to write map algebra expressions in Python.

## Work with classes to define raster tool parameters

Many raster tools have parameters that have a varying number of arguments.

The arcpy.sa module has a number of classes to make it easier to work with these parameters.

You will be working with a stand-alone script, and the first task is to make sure a license for the Spatial Analyst extension is available.

## Challenge exercises

### Challenge 1

Create a script that determines what areas meet the following conditions:

Moderate slope—between 5 and 20 degrees

Southerly aspect—between 150 and 270 degrees

Forested—land cover types of 41, 42, or 43

Be sure to use the map algebra expressions of the arcpy.sa module.

### Challenge 2

Write a script that copies all the rasters in a workspace to a new fie geo database.

You can use the rasters in the Exercise09 folder as an example.

# Exercise 10 Map scripting

## Open and save a map document

Map scripting can be used to open a map document and change any of the map document properties.

## Work with data frames

Map scripting can be used to access and modify the properties of one or more data frames in a map document.

In the first example, you will read the names of all the data frames.

## Work with map layers

Map scripting can also be used to work with map layers.

In the next example, you will create a list of all the layers in a map document and modify the properties of a specific layer.

## Work with page layout elements

Map scripting also makes it possible to work with page layout elements.

In the next examples, you will create a list of all the elements of a page layout and modify one of the elements.

## Challenge exercise

### Challenge 1

In ArcGIS Desktop Help, research the AddLayer function of the ArcPy mapping module and use it to write a script that adds the parks layer from the Parks data frame in Austin\_TX.mxd to the other two data frames in the same map document.

# Exercise 11 Debugging and error handling

## Examine syntax errors and exceptions

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

## Implement debugging procedures

## Handle some exceptions

## Challenge exercises

### Challenge 1

### Challenge 2

# Exercise 12Creating Python functions and classes

## Create Python functions

## Call functions from other scripts

## Work with classes

## Challenge exercises

### Challenge 1

### Challenge 2

# Exercise 13 Creating custom tools

## Examine the script

In this exercise, you will create a script tool that creates a random selection of features from an existing feature class and saves the result as a new feature class.

First, examine the script.

## Create a custom tool

## Set the tool parameters

## Select a script editor

## Edit the tool code to read the parameters

## Challenge exercise

### Challenge 1

Make a copy of the random\_sample.py script and call it random\_percent.py.

Modify the script so that the third parameter is a percentage of the number of input records as an integer between 1 and 100.

Modify the script tool settings so that the input for this parameter is validated on the tool dialog box.