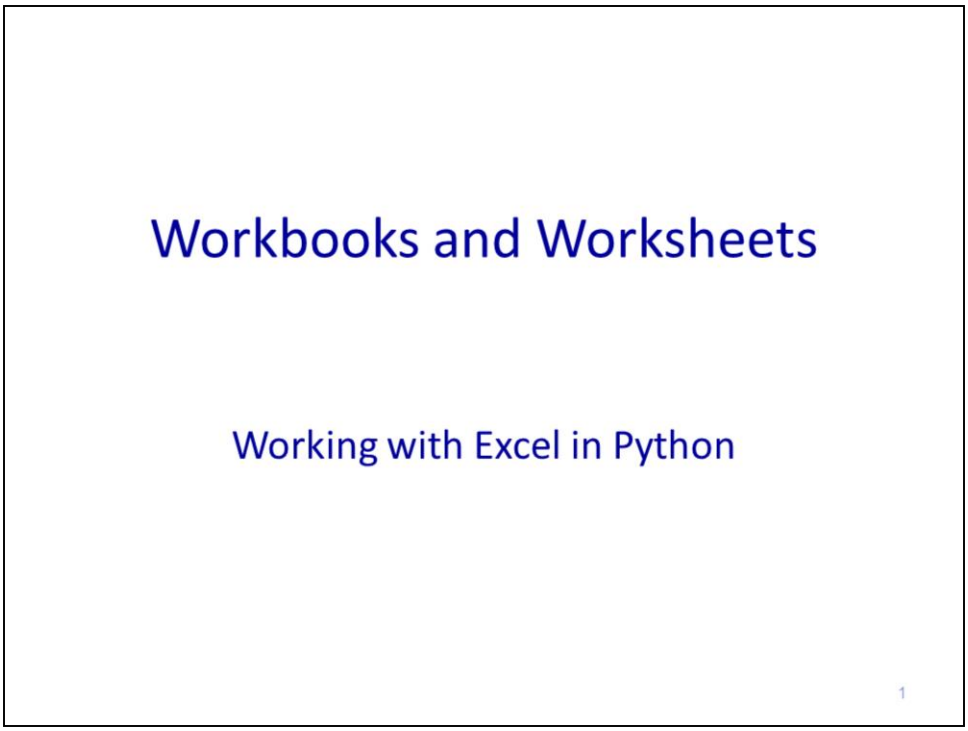
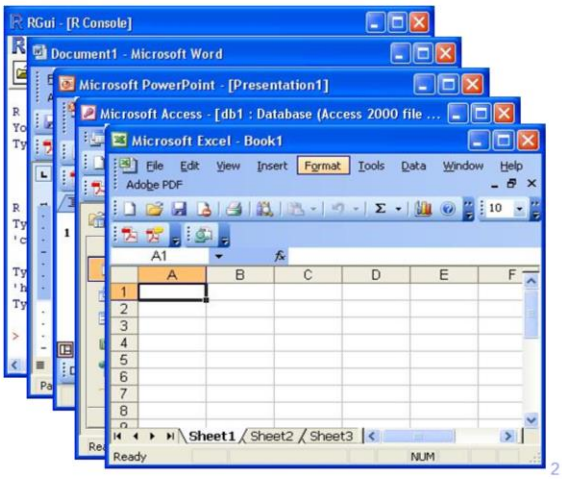



This video will discuss how to use the win32com module to work with Excel workbooks and worksheets.

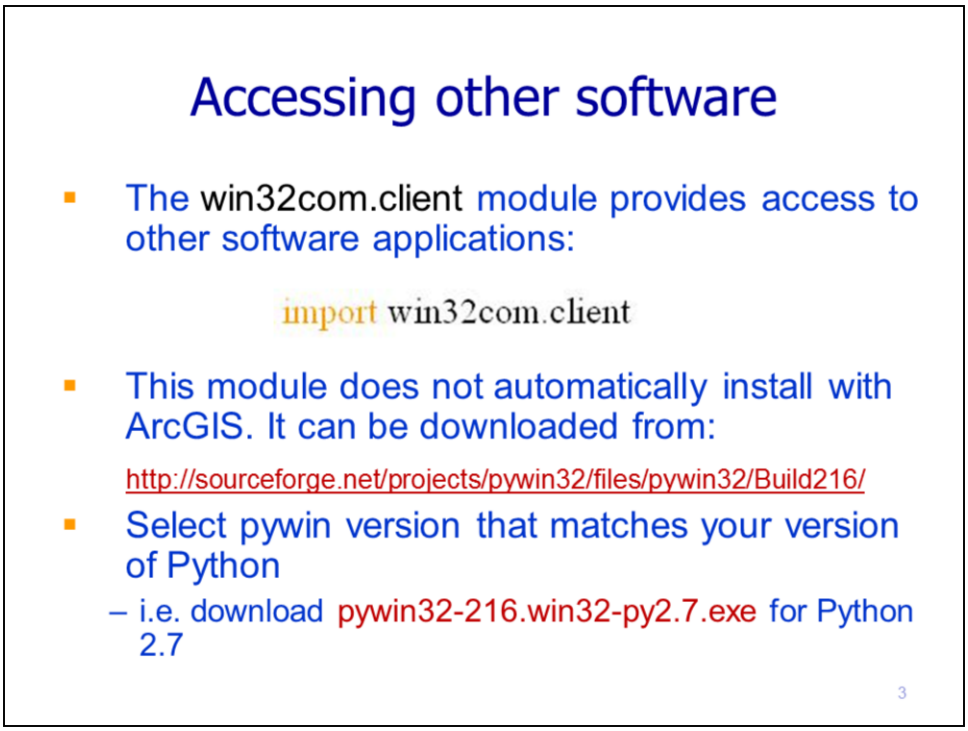
# Scripting is not just for ArcGIS

- Python can be used with other programs including:
  - R statistical software
  - Word
  - PowerPoint
  - Access
  - Excel




Python can access many other software applications besides ArcGIS.

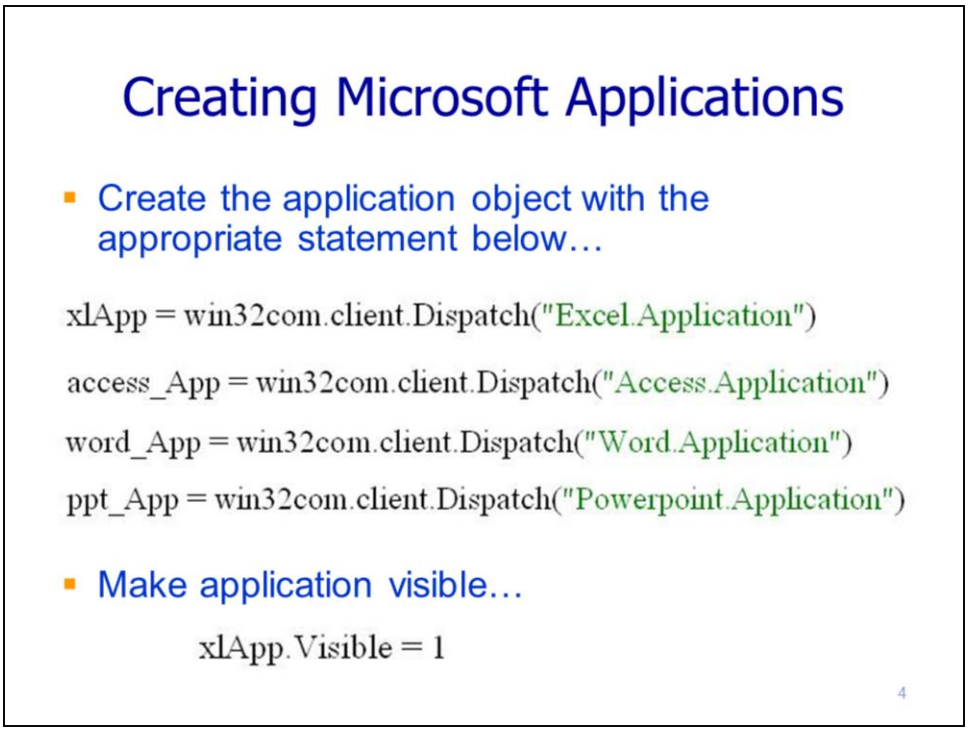
These applications include R and Microsoft Office products.




Modules, like arcpy, give Python to access other software applications. We'll use the **win32com.client** module for working with Microsoft Excel.

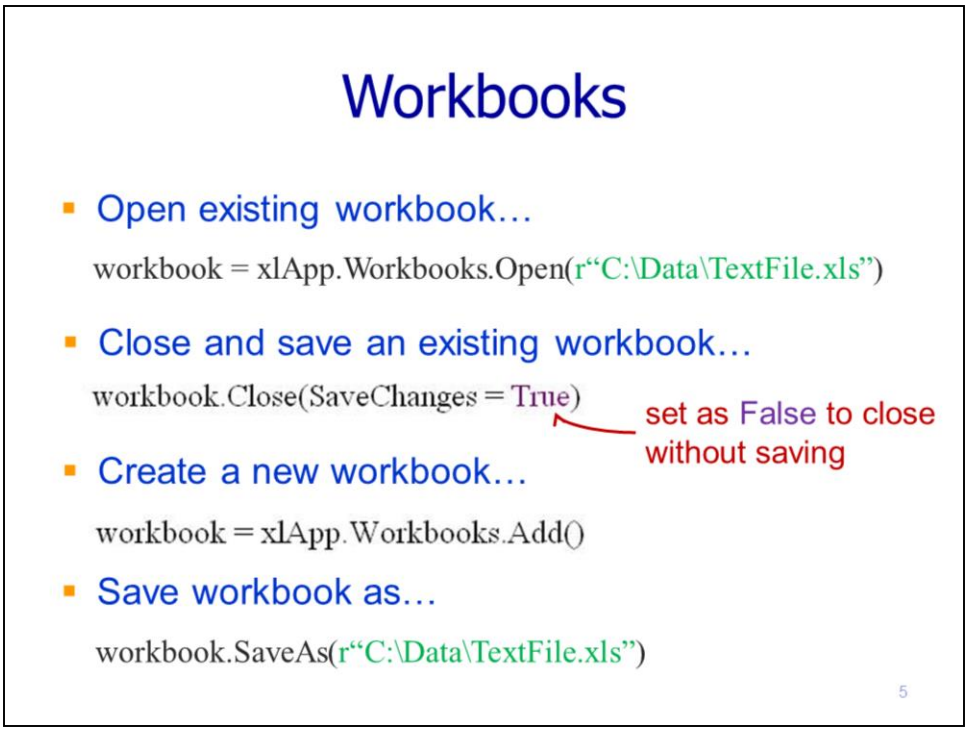
win32com is not an out-of-the-box module and must be downloaded and installed. The link provided in the notes is one location from which it can be downloaded.

The version must match your version of Python (i.e. python 2.7 for ArcGIS 10.3). If your Python installation is 32-bit, then you will need the win32 version, even if your computer is 64-bit.




Applications can be accessed using win32com's **Dispatch** method and specifying the appropriate name of the application. These example statements show how to create application objects for several Microsoft applications including Excel, Access, Word, and Powerpoint. This lecture will focus on the Excel application.

Setting the application's **visible** property to 1 (i.e. True) will make it visible while the script works with it. Setting the visible property to zero will make the application run in the background.

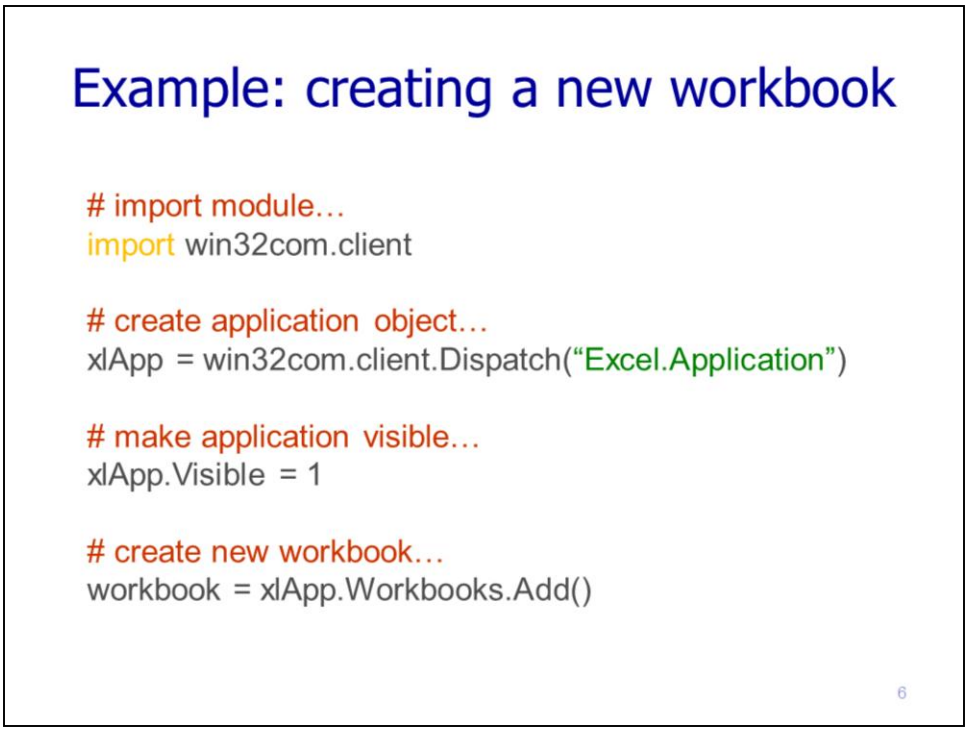



To create a workbook object by opening an existing Excel file, use the **Workbooks.Open** method of the application object.

To close a workbook, use the workbook object's **Close** method. The workbook can be saved by specifying the SaveChanges = True.

To create a new workbook, use the Workbooks.Add method of the application object.

You can save the workbook, with a new file name, by using the workbook's **SaveAs** method.

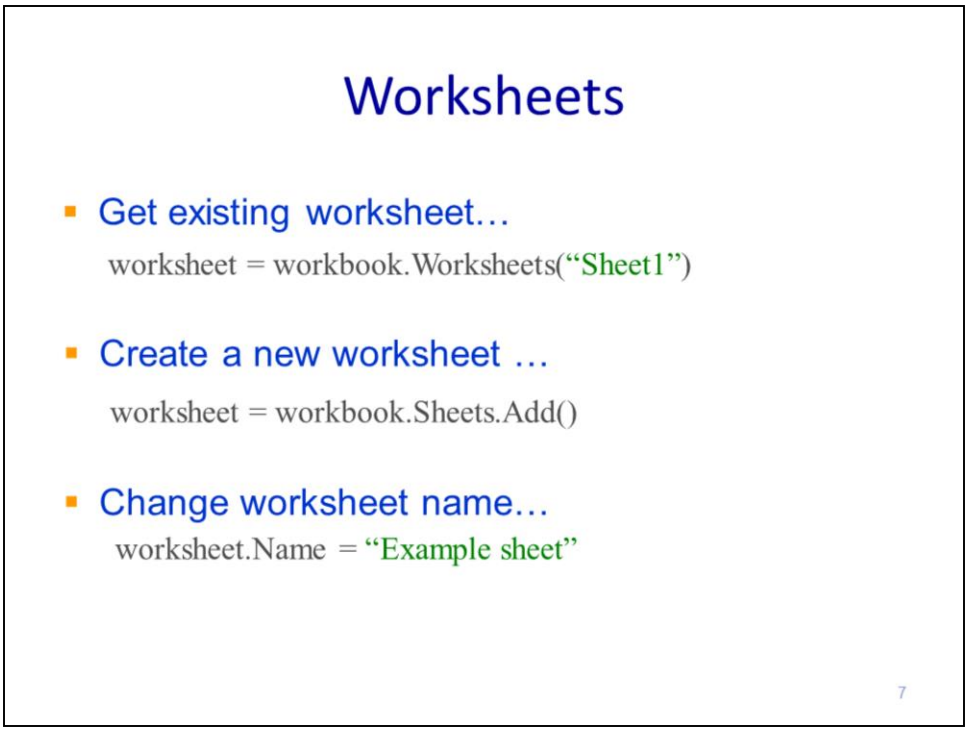



This slide will show an example of the code for creating a new workbook. The first statement imports the win32com.client module.

The **Dispatch** method creates the Excel application object.

The application is made visible.

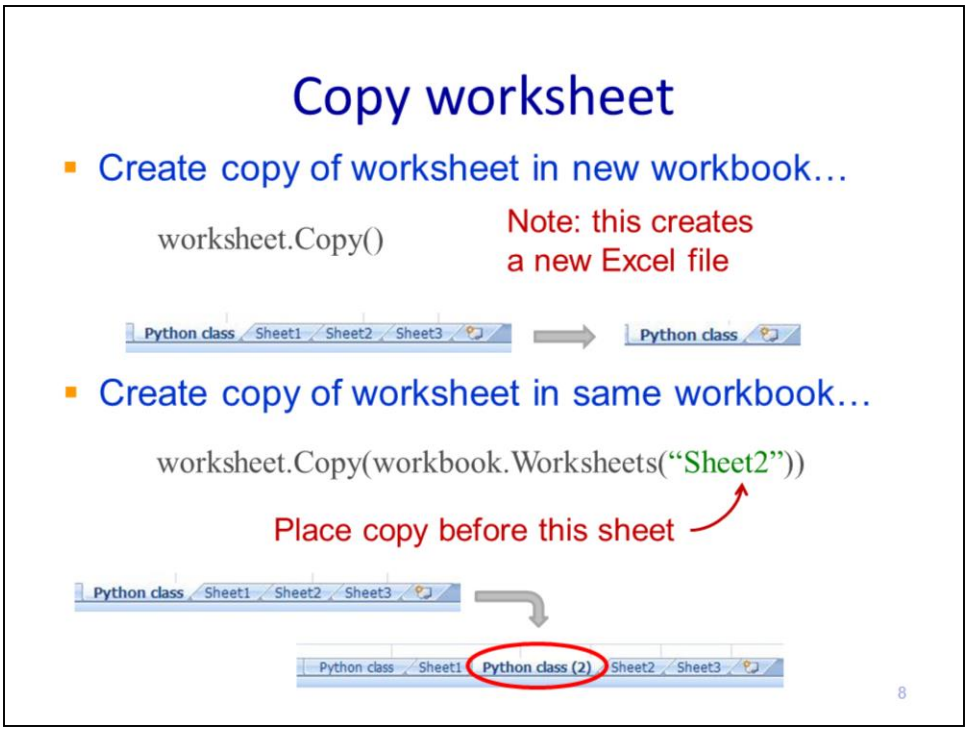
And a new workbook is created using the **Workbooks.Add** method.




This statement shows how to get the “Sheet1” worksheet from the workbook.

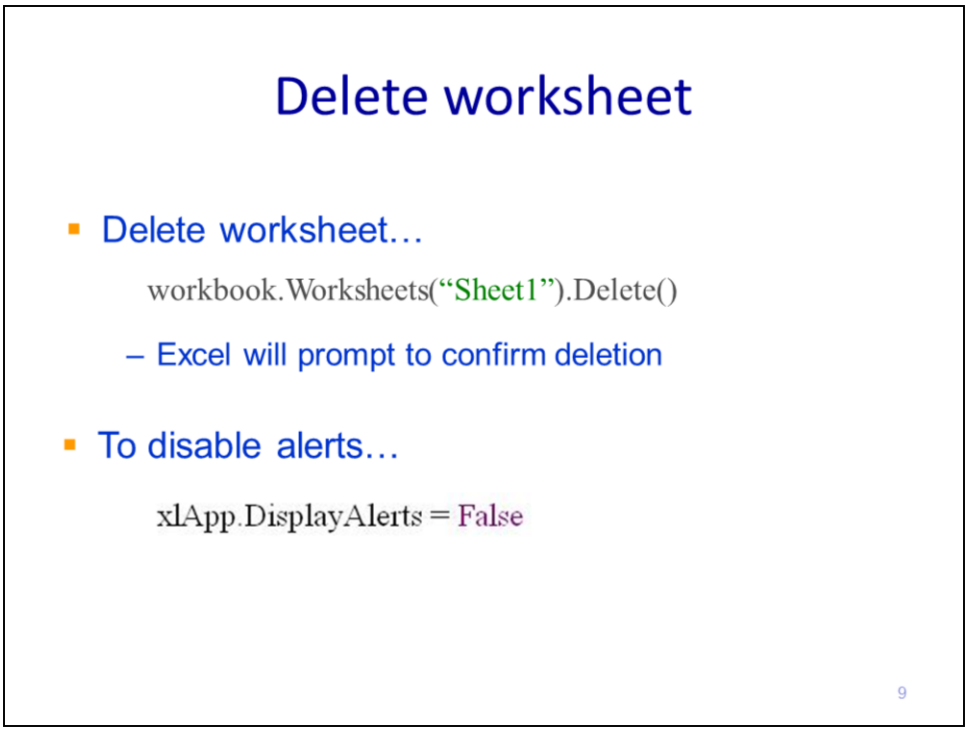
To create a new worksheet, use the workbook’s **Sheets.Add** method.

You can change the name of a worksheet by modifying the **Name** property.




A worksheet can be copied to a new Excel file.

Or a worksheet can be copied within its current workbook. The copy is placed before the specified worksheet.




A worksheet can be deleted from a workbook. By default, Excel will prompt you to confirm the deletion of a worksheet – the script would pause until you respond to the prompt.

The **DisplayAlerts** can be set to False which will disable any prompts. Deletions or saving over existing files will simply be performed without needing confirmation.