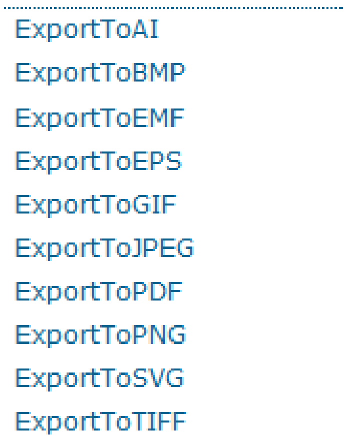
Chapter 10: Printing and Exporting Maps

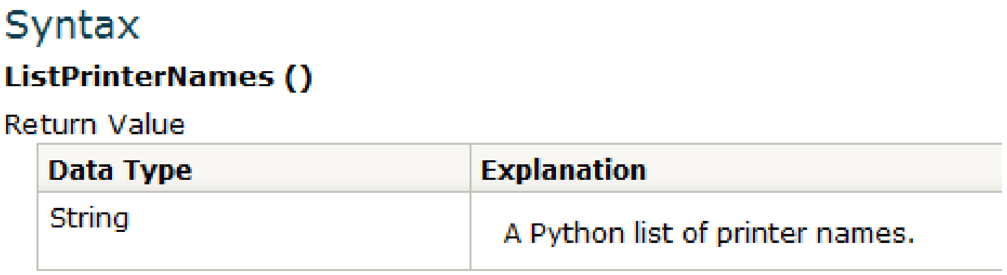
- Arcpy mapping provides a PrintMap() function for print page layouts or data frames within ArcMap
- Other functions include for exporting to various image formats



ExportToAI
ExportToBMP
ExportToEMF
ExportToEPS
ExportToGIF
ExportToJPEG
ExportToPDF
ExportToPNG
ExportToSVG
ExportToTIFF

10.1 Getting a list of Printers

- Before printing a map, it is common to get a list of available printers
- ListPrinterNames() returns a Python list of Printers on the local computer
- The returned value of ListPrinterNames() can then be used as input into the PrintMap() function



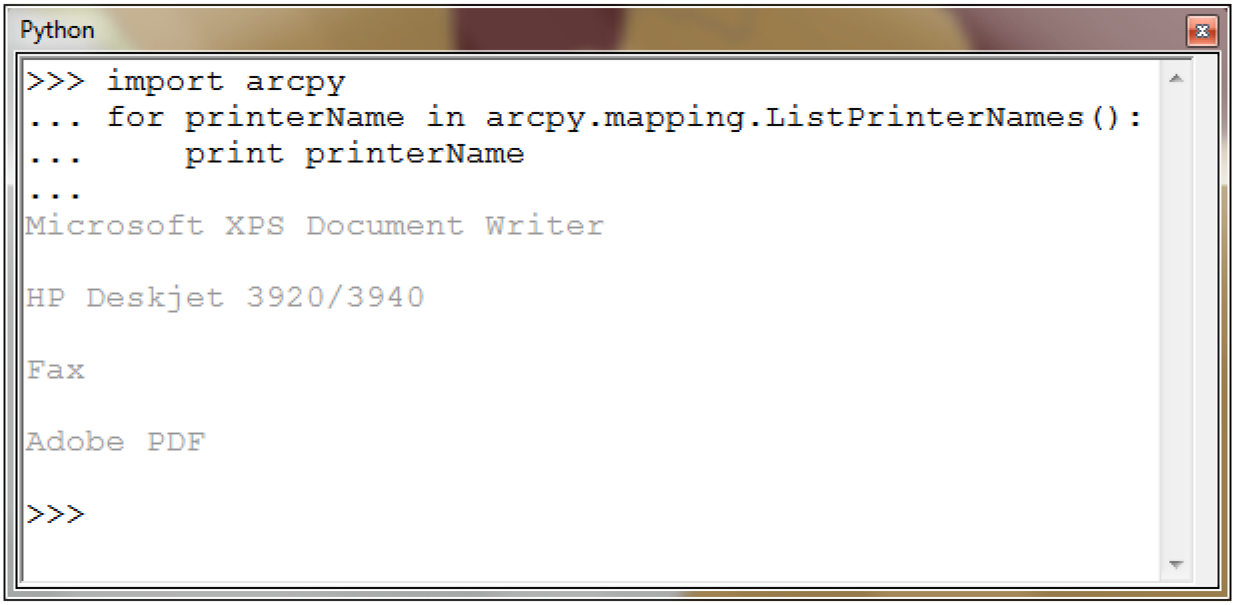
Syntax

ListPrinterNames ()

Return Value

Data Type	Explanation
String	A Python list of printer names.

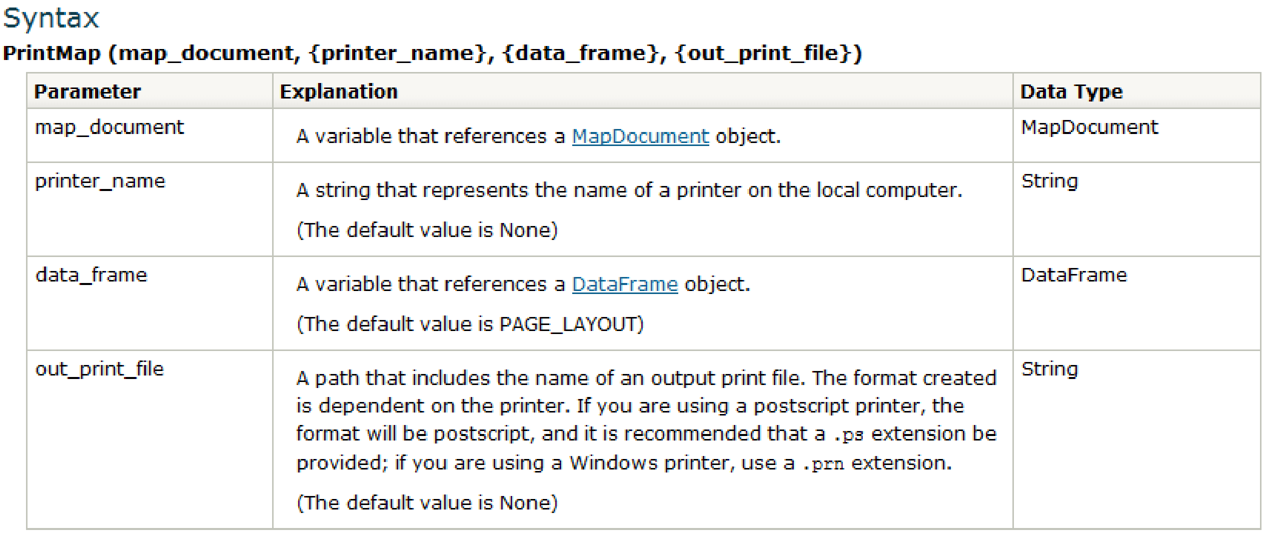
Example:



```
>>> import arcpy
... for printerName in arcpy.mapping.ListPrinterNames():
...     print printerName
...
Microsoft XPS Document Writer

HP Deskjet 3920/3940

Fax

Adobe PDF

>>>
```

10.2 Print Maps

- PrintMap provides the ability to print a specific data frame or a map document layout to a system printer or a print file.
- If a printer name is not provided, PrintMap will use the printer that is saved with the map document or will use the default system printer if the map document does not have a printer saved.
- PrintMap() is usually used together with ListPrinterNames()



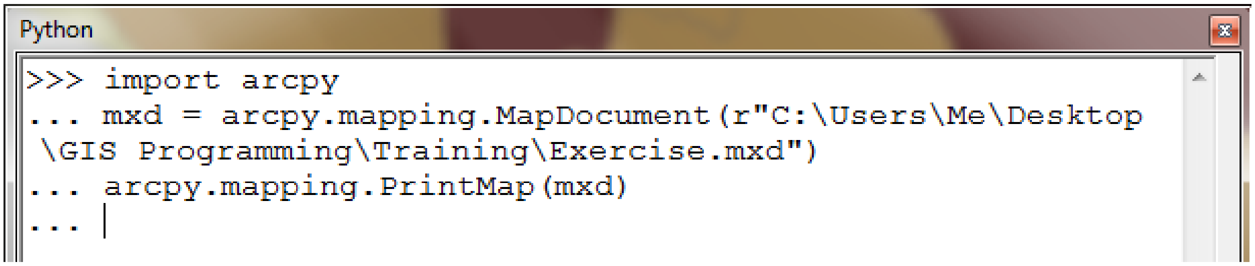
Syntax

PrintMap (map_document, {printer_name}, {data_frame}, {out_print_file})

Parameter	Explanation	Data Type
map_document	A variable that references a MapDocument object.	MapDocument
printer_name	A string that represents the name of a printer on the local computer. (The default value is None)	String
data_frame	A variable that references a DataFrame object. (The default value is PAGE_LAYOUT)	DataFrame
out_print_file	A path that includes the name of an output print file. The format created is dependent on the printer. If you are using a postscript printer, the format will be postscript, and it is recommended that a `.ps` extension be provided; if you are using a Windows printer, use a `.prn` extension. (The default value is None)	String

Example:

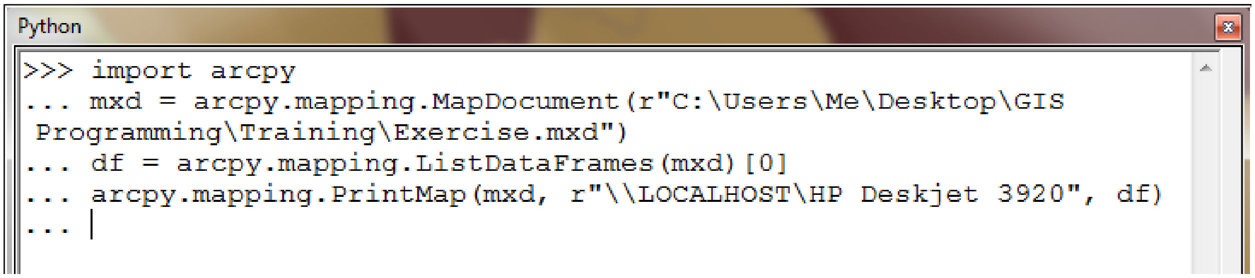
The following script prints a map using the default printer options.



```
>>> import arcpy
... mxd = arcpy.mapping.MapDocument(r"C:\Users\Me\Desktop
 \GIS Programming\Training\Exercise.mxd")
... arcpy.mapping.PrintMap(mxd)
... |
```

Example:

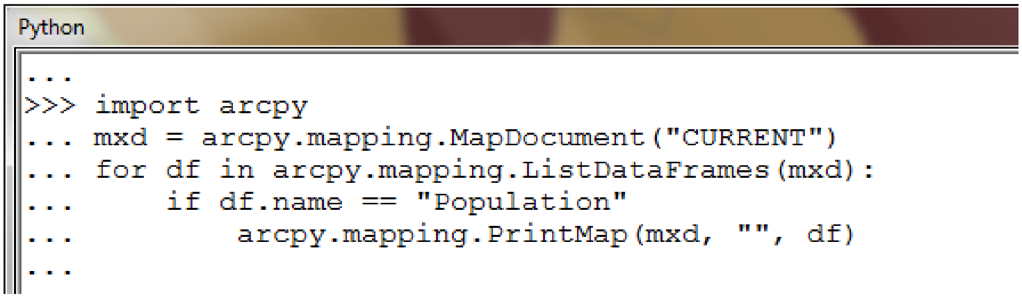
The following script prints the first data frame within a map document using a specified printer name.



```
>>> import arcpy
... mxd = arcpy.mapping.MapDocument(r"C:\Users\Me\Desktop\GIS
 Programming\Training\Exercise.mxd")
... df = arcpy.mapping.ListDataFrames(mxd)[0]
... arcpy.mapping.PrintMap(mxd, r"\\LOCALHOST\HP Deskjet 3920", df)
... |
```

Example:

Printing a specific data frame



```
...
>>> import arcpy
... mxd = arcpy.mapping.MapDocument("CURRENT")
... for df in arcpy.mapping.ListDataFrames(mxd):
...     if df.name == "Population"
...         arcpy.mapping.PrintMap(mxd, "", df)
...
```

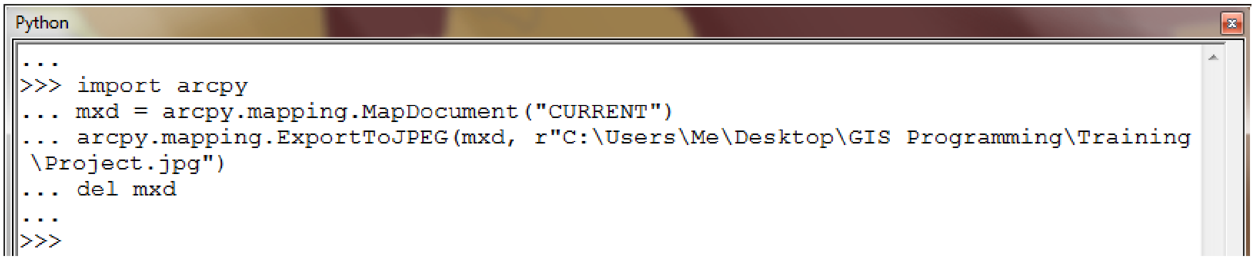
10.3 Exporting Maps to Images

- You can use one of the many export functions in Arcpy to export maps from mxds or specific data frames to image files.
- Parameters provided to each function will vary

- Check out some of the varying parameters at http://help.arcgis.com/en/arcgisdesktop/10.0/help/index.html#/ExportToJPEG/00s300000038000000/

Example:

This script uses the current map document and exports the page layout to a JPEG file using default values for all options.



```
...
>>> import arcpy
... mxd = arcpy.mapping.MapDocument("CURRENT")
... arcpy.mapping.ExportToJPEG(mxd, r"C:\Users\Me\Desktop\GIS Programming\Training
\Project.jpg")
... del mxd
...
>>>
```

10.4 Working with the Portable Document Format

Exporting Maps to PDFs

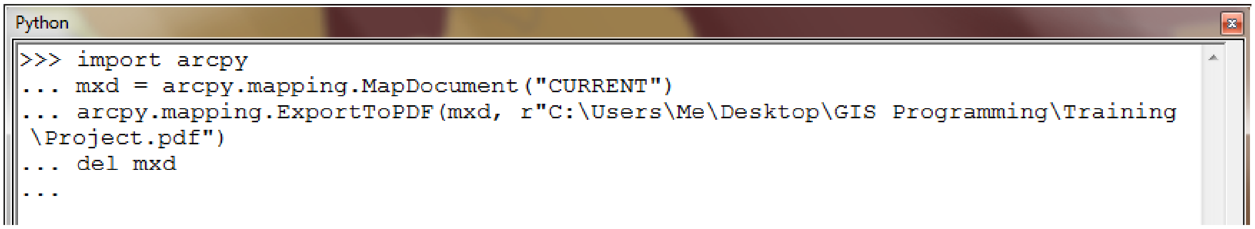
- ExportToPDF exports a page layout or data frame to PDF format
- Syntax

ExportToPDF (map_document, out_pdf, {data_frame}, {df_export_width}, {df_export_height}, {resolution}, {image_quality}, {colorspace}, {compress_vectors}, {image_compression}, {picture_symbol}, {convert_markers}, {embed_fonts}, {layers_attributes}, {georef_info}, {jpeg_compression_quality})

- See the following link for more information on each of the parameters http://help.arcgis.com/en/arcgisdesktop/10.0/help/index.html#/ExportToPDF/00s300000027000000/

Example:

This script opens the current map document and exports the page layout to a PDF file using default values for all options.



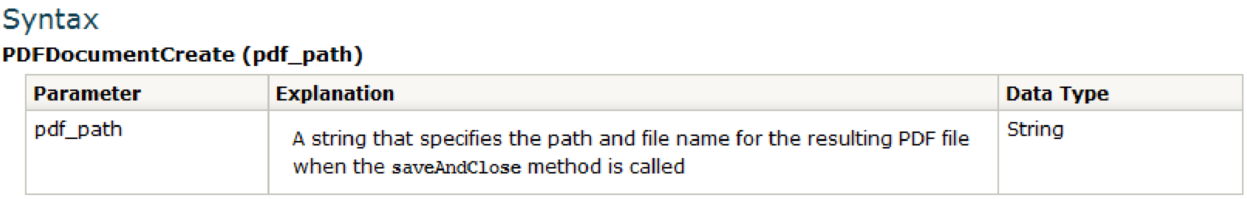
```
>>> import arcpy
... mxd = arcpy.mapping.MapDocument("CURRENT")
... arcpy.mapping.ExportToPDF(mxd, r"C:\Users\Me\Desktop\GIS Programming\Training
\Project.pdf")
... del mxd
...
```

Task:

Write Python code to search a list of data frames, finding a specific data frame using its name and exporting this data frame to a PDF file.

Manipulating PDFs

- The PDFDocument class is used to manipulate PDF documents
- Allows manipulation of PDF documents, including facilities for merging pages, setting document open behavior, adding file attachments, and creating or changing document security settings
- PDFDocumentOpen (opens an existing document) and PDFDocumentCreate (creates a new document) are two functions that provide a reference to a PDFDocument
- Often used in the creation of map books
- More information on Map Books can be found at http://help.arcgis.com/en/arcgisdesktop/10.0/help/index.html#//00s90000002s000000.htm
- To create a pdf use



Syntax

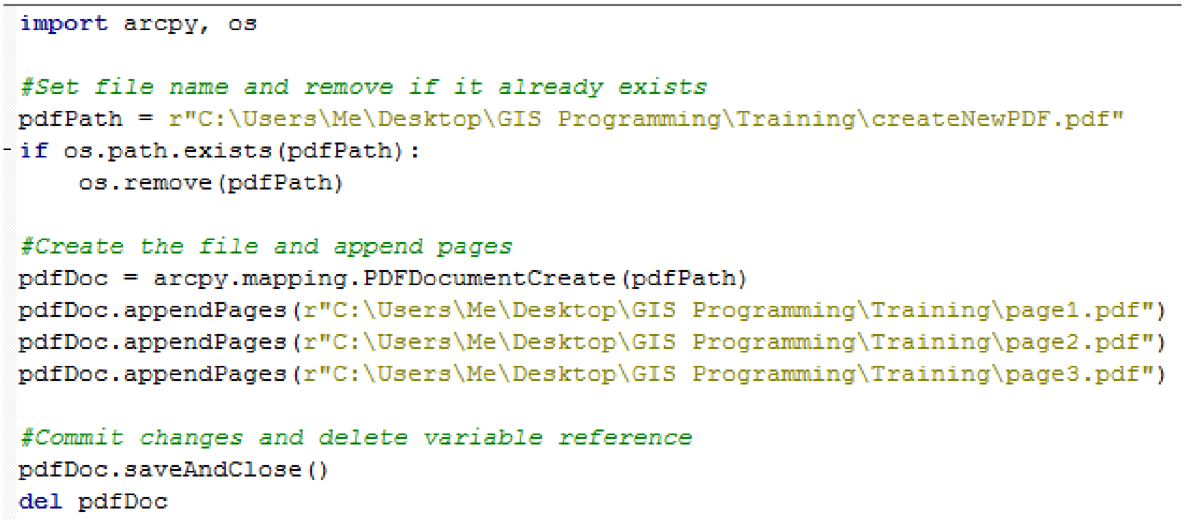
PDFDocumentCreate (pdf_path)

Parameter	Explanation	Data Type
pdf_path	A string that specifies the path and file name for the resulting PDF file when the `saveAndClose` method is called	String

- Accepts a path and file name for the document
- PDF not actually created on disk until
    - You insert or append pages (insertPages() or appendPages())
    - Then calling PDFDocument.saveAndClose()

Example

This script will create a new PDF document, append the contents of three separate PDF documents, and save the resulting PDF file.

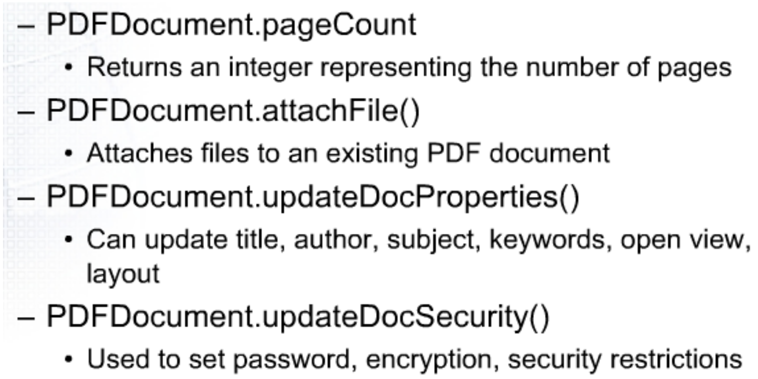


```
import arcpy, os

#Set file name and remove if it already exists
pdfPath = r"C:\Users\Me\Desktop\GIS Programming\Training\createNewPDF.pdf"
if os.path.exists(pdfPath):
    os.remove(pdfPath)

#Create the file and append pages
pdfDoc = arcpy.mapping.PDFDocumentCreate(pdfPath)
pdfDoc.appendPages(r"C:\Users\Me\Desktop\GIS Programming\Training\page1.pdf")
pdfDoc.appendPages(r"C:\Users\Me\Desktop\GIS Programming\Training\page2.pdf")
pdfDoc.appendPages(r"C:\Users\Me\Desktop\GIS Programming\Training\page3.pdf")

#Commit changes and delete variable reference
pdfDoc.saveAndClose()
del pdfDoc
```

Task

Create 3 pdf documents from map files and/or data frames using what you have learned so far in this course and substitute them into the script above, appending them to create a new single pdf document.

- PDFDocumentOpen() accepts a path to a pdf file and returns an instance of PDFDocument
- Once opened, you can make modifications to properties, add or insert files and attach documents,
- Use PDFDocument.saveAndClose after operations
- A number of properties can be saved due to the properties of the PDFDocument object



- PDFDocument.pageCount
  - Returns an integer representing the number of pages
- PDFDocument.attachFile()
  - Attaches files to an existing PDF document
- PDFDocument.updateDocProperties()
  - Can update title, author, subject, keywords, open view, layout
- PDFDocument.updateDocSecurity()
  - Used to set password, encryption, security restrictions

- More on PDFDocument properties at http://help.arcgis.com/en/arcgisdesktop/10.0/help/index.html#/PDFDocument/00s30000002w000000/