**📚 GISC 2335 – 2025-04-22 Lecture Notes**

**Chapter 11: Map Scripting**

**🧠 Big Picture**

* **Map scripting** automates map-making tasks in ArcGIS Pro using **Python** and the **arcpy.mp** module.
* Instead of manually clicking in ArcGIS Pro, you write scripts to:
  + Open a project 📂
  + Modify maps 🗺️
  + Change symbology 🎨
  + Export layouts 📄
* Useful for **repetitive tasks** (e.g., changing icons, updating data sources) across many maps.

**🛠️ Tools Introduced**

* **arcpy.mp** — the Python module for manipulating ArcGIS Pro projects (.aprx files).
* **Key Objects:**
  + ArcGISProject (manages the project file)
  + Map (manages a map)
  + Layer (manages a layer)
  + Layout (manages a layout)
  + MapFrame (container for maps on a layout)
  + Camera (controls map zoom and viewpoint)

**📄 Project Management Basics**

* Open a project:
* aprx = arcpy.mp.ArcGISProject("CURRENT")
* Save a copy to avoid overwriting:
* aprx.saveACopy("Austin\_Copy.aprx")
* Delete the object from memory (to unlock the file):
* del aprx

💬 *Note:* Use CURRENT only **inside** ArcGIS Pro (Python window). For standalone scripts, use full paths!

**🗺️ Working with Maps**

* List maps:
* maps = aprx.listMaps()
* Print map names:
* for m in maps:
* print(m.name)
* Change map names, add basemaps, remove or insert layers programmatically.

**📚 Working with Layers**

* List layers in a map:
* layers = m.listLayers()
* Identify types:
* if lyr.isFeatureLayer:
* print(lyr.name + " is a feature layer")
* Add a basemap:
* m.addBasemap("Light Gray Canvas")
* Add new data:
* m.addDataFromPath("C:/path/to/your/data.gdb/featureclass")

**🎨 Working with Symbology**

* Access and modify symbology:
* sym = lyr.symbology
* if lyr.isFeatureLayer and hasattr(sym, "renderer"):
* sym.renderer.symbol.color = {"RGB": [255, 0, 0, 100]} # Red
* lyr.symbology = sym
* Use **Graduated Colors**, **Unique Values**, or **SimpleRenderer** for feature layers.
* Raster layers use **Stretch**, **Classify**, or **Unique Value** colorizers.

**📄 Working with Layouts**

* List layouts:
* layouts = aprx.listLayouts()
* Find and modify layout elements (like map frames, legends, text).
* Example to update a picture element:
* photo = lyt.listElements("PICTURE\_ELEMENT")[0]
* photo.sourceImage = "C:/path/to/new/logo.png"

**🖼️ Exporting Maps**

* Export layout to PNG:
* lyt.exportToPNG("C:/path/to/output.png")
* Export layout to PDF:
* lyt.exportToPDF("C:/path/to/output.pdf")

📝 *Tip:* You can automate **batch exporting** for multiple layouts!

**🧩 Extra Power with CIM (Cartographic Information Model)**

* **CIM access** lets you modify *advanced* cartographic settings (e.g., special labels, styles).
* More complex, recommended for advanced users.

**⚠️ Limitations to Remember**

* **You cannot:**
  + Create a brand-new .aprx from scratch (must start with a template)
  + Create new layout elements (you can only modify existing ones)
  + Easily apply multi-variable symbology (like combining poverty + health access)
* **Workarounds:** Preprocess fields in advance or stack multiple layers with transparency.

**📌 Deadlines Reminder**

* ✅ Chapter 11 Lab due: **May 6, 2025**
* ✅ All textbook-related assignments (Ch. 1–11): **due May 7, 2025**
* ❗ Late work NOT accepted after May 7 unless discussed with Professor Bushland!

**🔥 Highlights You Should Practice**

* Understand difference between save() and saveACopy().
* Practice listing and modifying **maps, layers, and layouts**.
* Change **symbology** using color dictionaries or style galleries.
* Export maps automatically using **Python scripts**.