PYTHON

PYTHON PROGRAMMING IN GIS

WHY USE PYTHON PROGRAMMING IN GIS?

- Workflow automation
- Downloading data
- Geoprocessing
- Automation of repetitive tasks
- Functionality that would otherwise be unavailable

WHY PYTHON?

- Interpreted programming language
- Built from C
- Platform Independent
- No need to compile
- Many different versions
- ArcGIS and QGIS already installed it!
- Execute tools inside ArcMap and QGIS
- Execute tools/code using CMD/Terminal/Console
- Many Interactive Development Environments

BENEFITS

- Programs are clean, easy to understand and maintainable
- Programs are portable across Windows, Mac, Linux, etc.
  - ArcGIS Desktop only on Windows, but as we will see, lots of GIS tasks can be achieved directly using Python modules
- Many open-source libraries and modules available
  - Including with spatial programming support
- Plays nicely with programs written in other languages
  - ArcGIS is written in C++, but many Toolbox functions are
  - written in Python

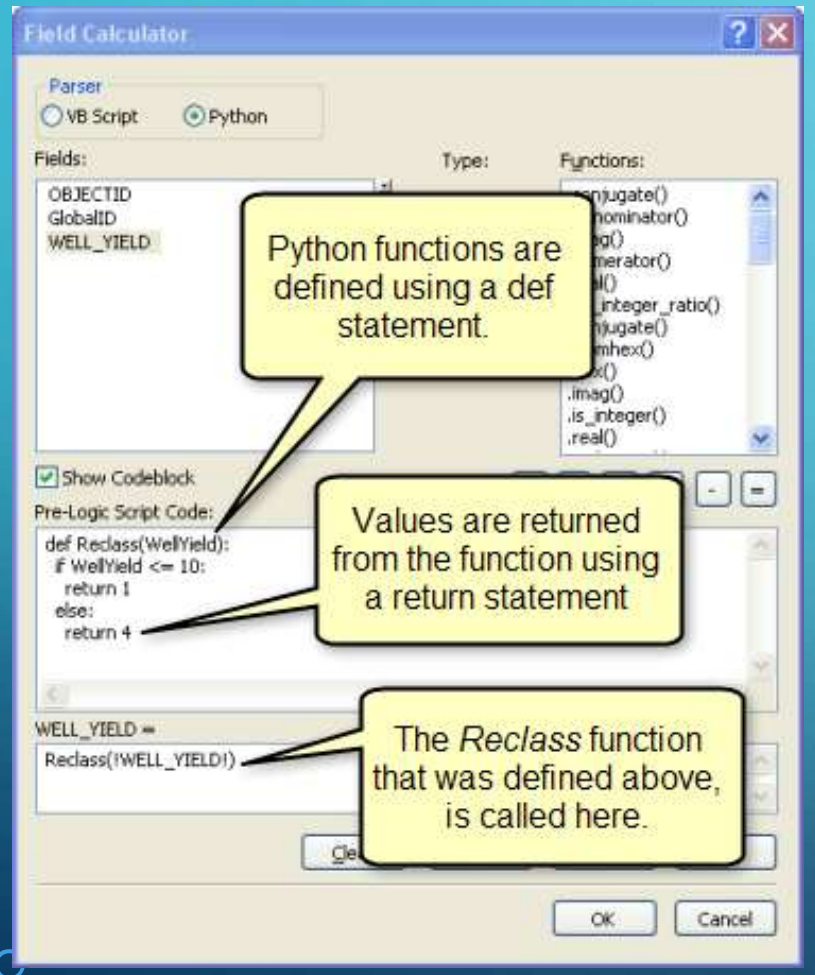
CONS/DRAWBACKS

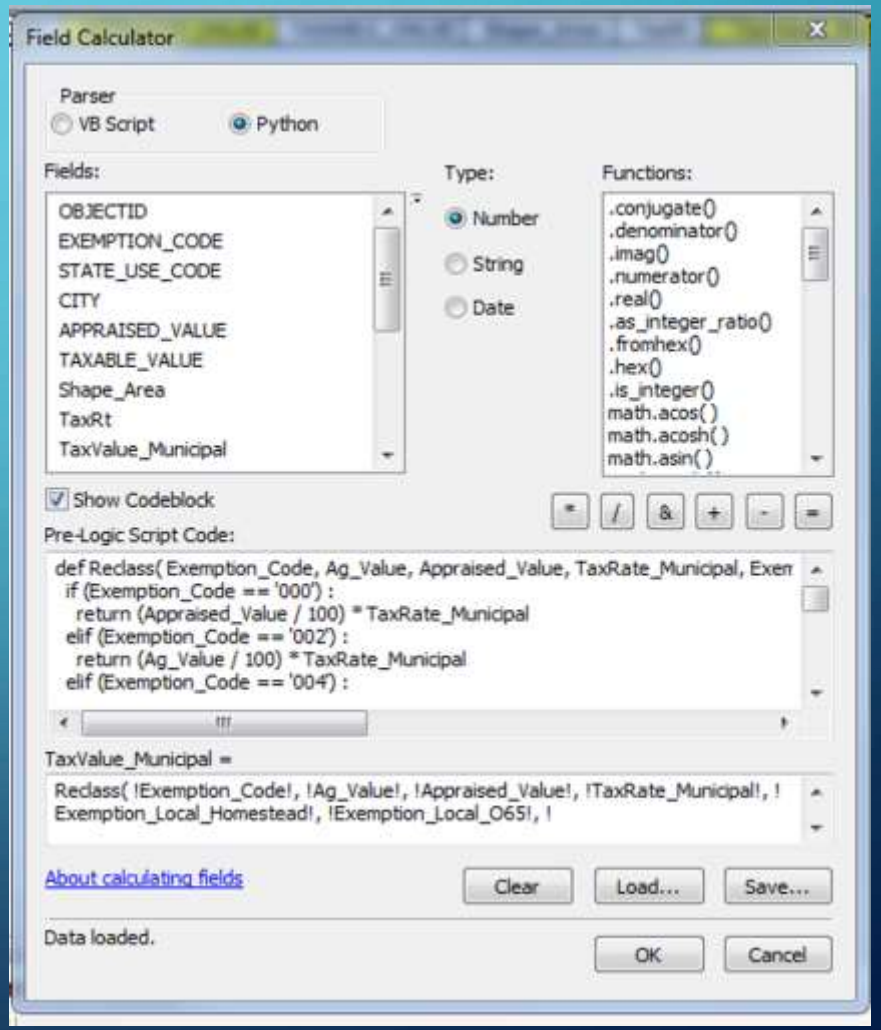
- Python is not the fastest kid on the block
  - Especially for heavy-duty number crunching
  - May want to write such modules in C/C++ and invoke them from Python
    - Another option to write such modules in Python, compile it and then link the compiled extensions into your code
- Still, Python is sufficient for many programming tasks
  - And hardware is getting more and more capable

- **Variable** is a way of storing values into the memory of the computer by using specific names that you define.
- **Data types**
- Integer (int) = Whole number
- Float (float) = Decimal number
- String (str) = Text
- Boolean (bool) = True / False
- List (list) = A “container” that can store any kind of values. You can create a list with square brackets e.g. [1, 2, 3, 'a', 'b', 'c'].
- Tuple (tuple) = A similar “container” as list with a difference that you cannot update the values in a tuple. You can create a tuple with parentheses (1, 2, 3, 'a', 'b', 'c').
- **Index** number is the location of specific value stored in Python lists or tuples. The first index value of list is always **0**.
- **Script** is a dedicated document for writing Python code that you can execute. Python script files should always have the `` .py `` file extension.

- QGIS – pygis
- ArcGIS – arcpy

PYTHON IN THE FIELD CALCULATOR

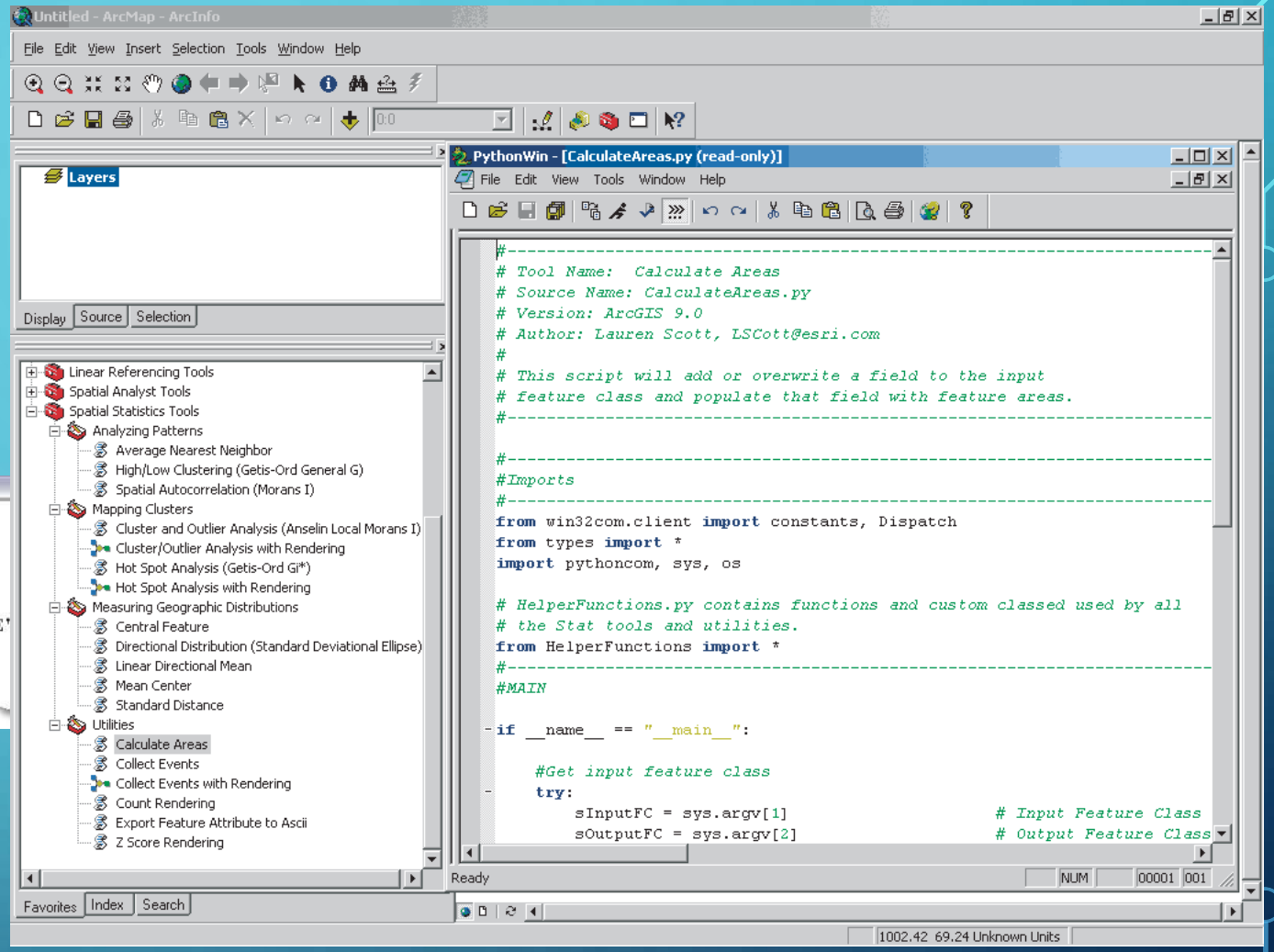




PYTHON CONSOLE

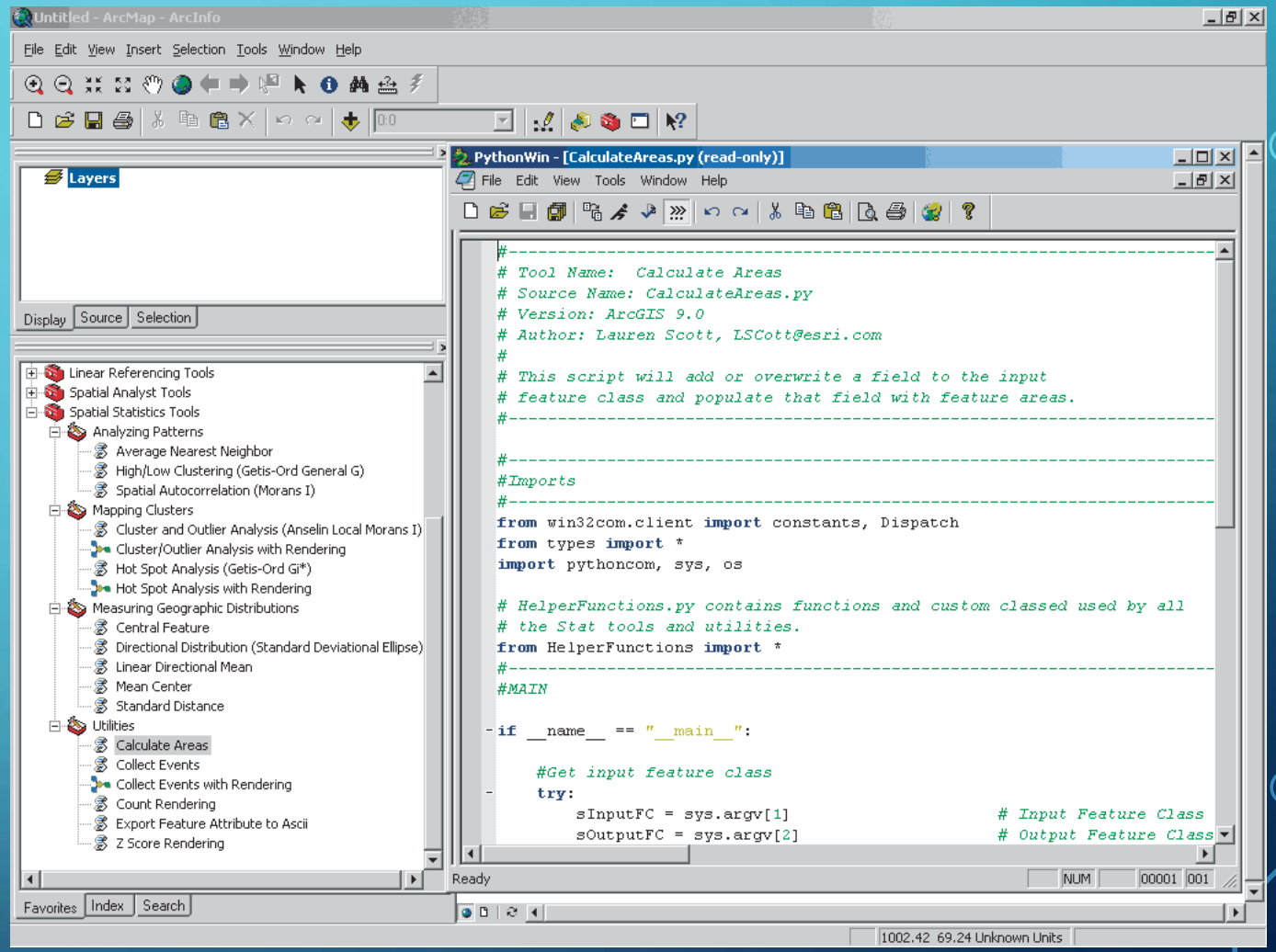
```
Python
>>> import arcpy, glob
>>> mxd = arcpy.mapping.MapDocument('current')
>>> df = arcpy.mapping.ListDataFrames(mxd)[0]
>>> for lyrfile in glob.glob(r"c:\data\*.lyr"):
... print "Adding", lyrfile, "to map"
... arcpy.mapping.AddLayer(df, arcpy.mapping.Layer(lyrfile), "AUTO_ARRANGE"
...
```

Example of more advanced Pyton code in the Python window



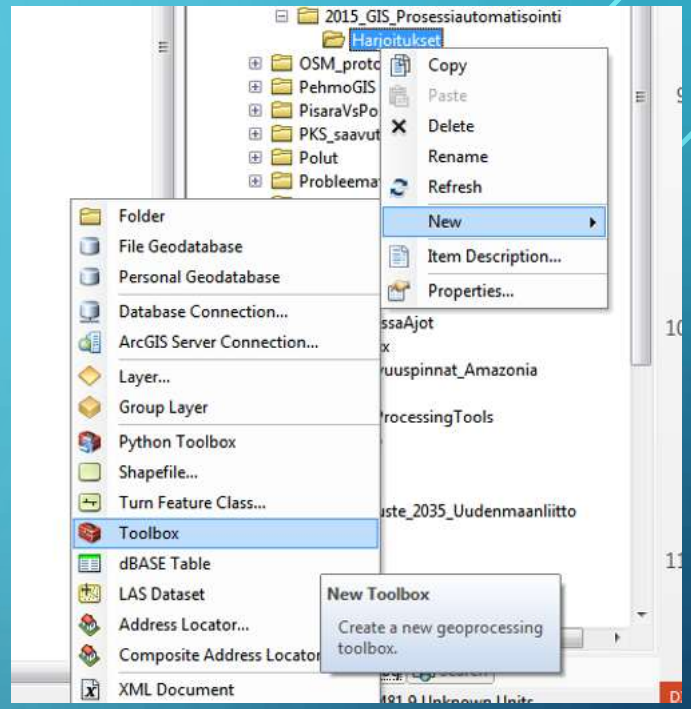
EXAMPLE OF SCRIPT LIBRARY "MODULES"

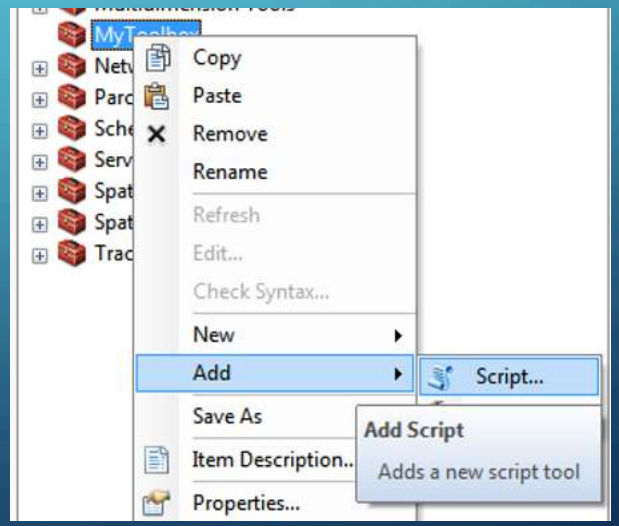
- NumPy, Shapely, GDAL Python bindings, PySAL, pyshp and GeoDjango
- SciPy is a Python-based ecosystem of open-source software for mathematics, science, and engineering.



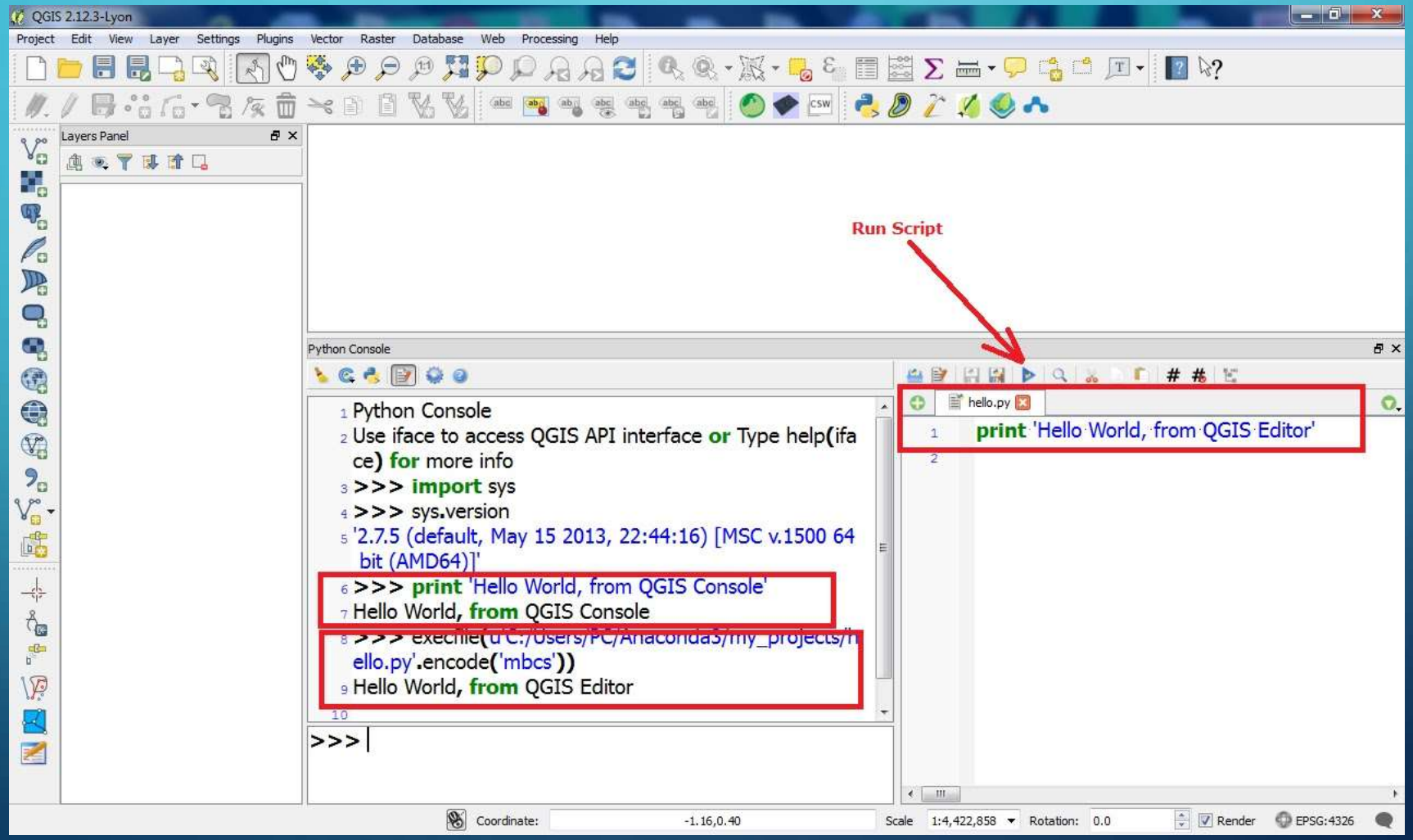
BUILD A TOOL WITH SCRIPT IN ARCGIS

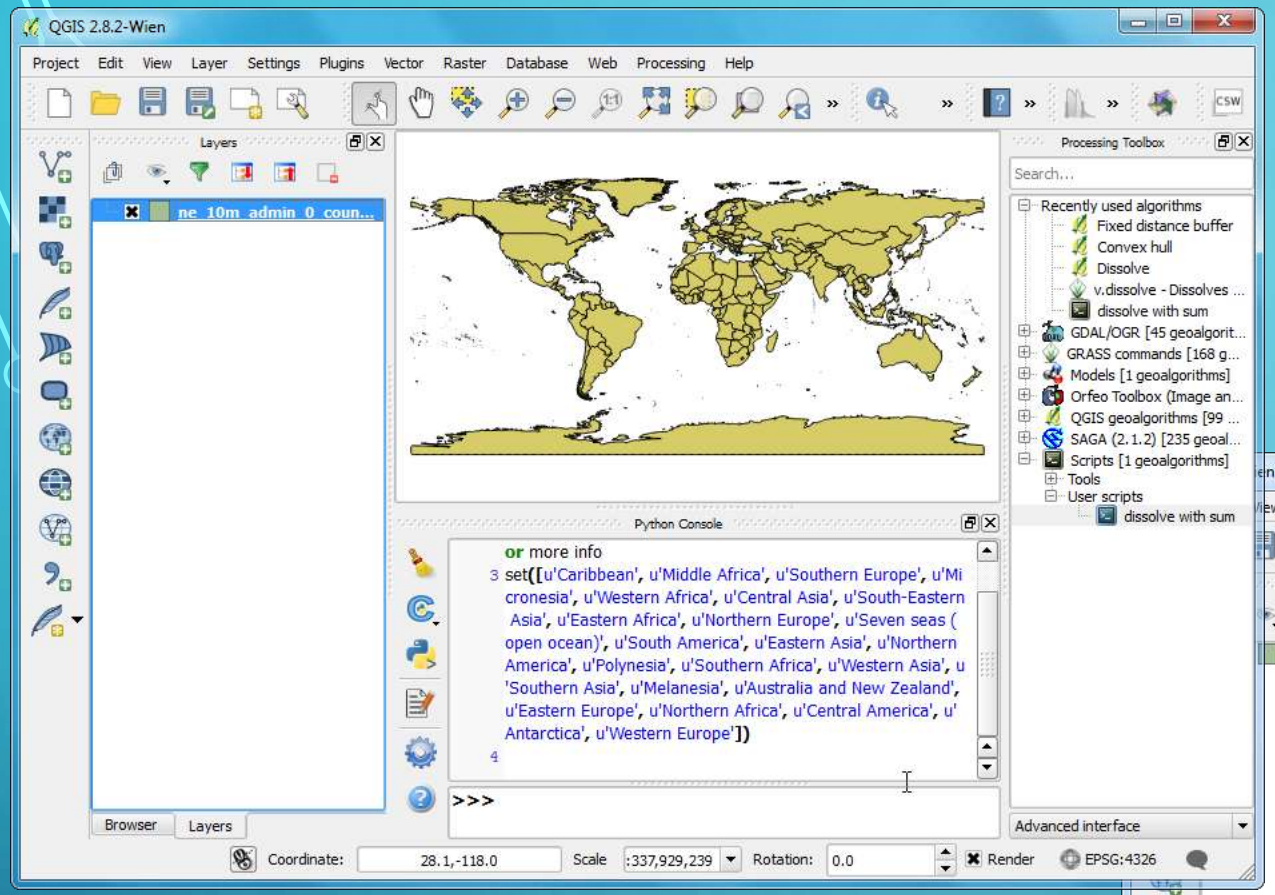
- Creating a script tool allows you to turn your own Python scripts and functionality into your own geoprocessing tools—tools that look and act like system geoprocessing tools.

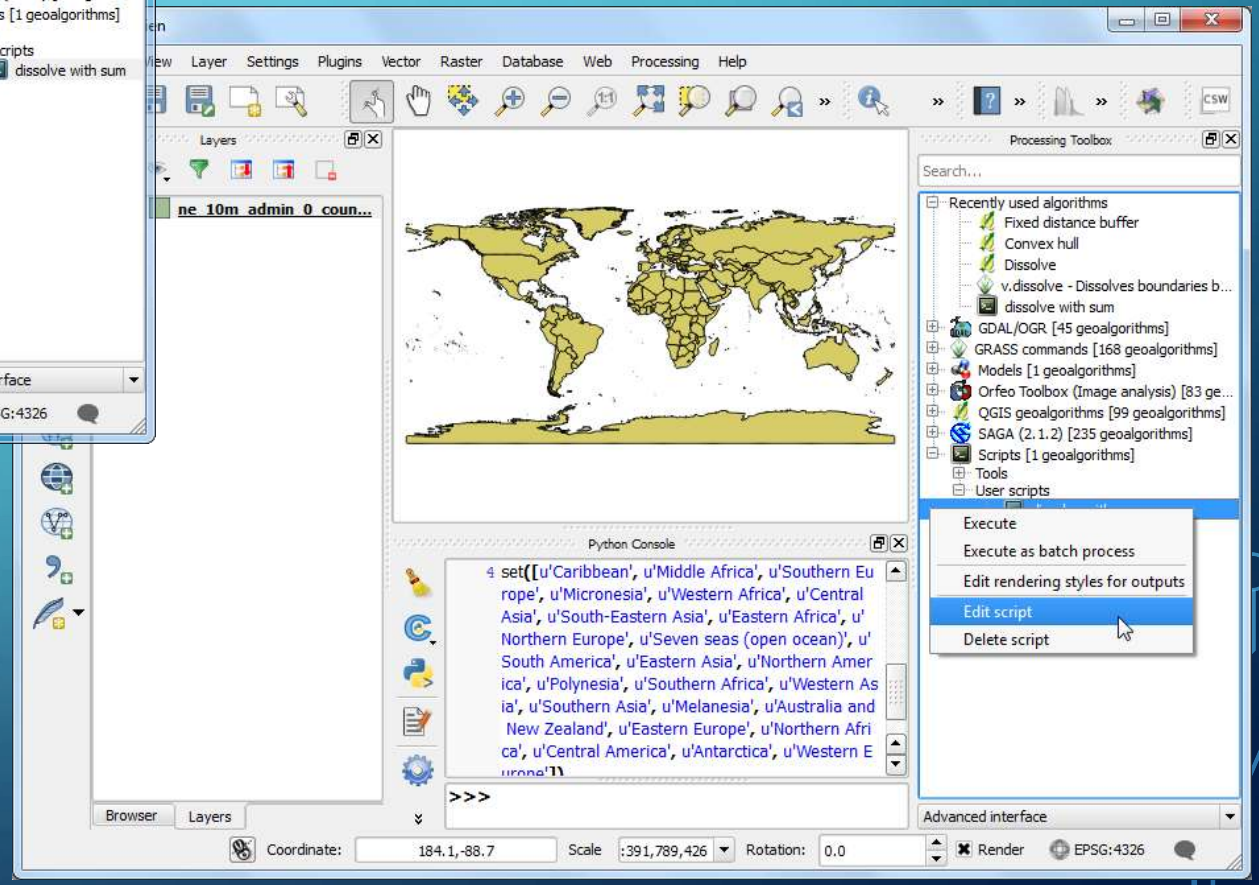




PYTHON CONSOLE IN QGIS




QGIS 2.8.2-Wien
Project Edit View Layer Settings Plugins Vector Raster Database Web Processing Help
Layers
ne_10m_admin_0_coun...
Processing Toolbox
Search...
Recently used algorithms
Fixed distance buffer
Convex hull
Dissolve
v.dissolve - Dissolves ...
dissolve with sum
GDAL/OGR [45 geoalgorit...
GRASS commands [168 g...
Models [1 geoalgorithms]
Orfeo Toolbox (Image an...
QGIS geoalgorithms [99 ...
SAGA (2.1.2) [235 geoal...
Scripts [1 geoalgorithms]
Tools
User scripts
dissolve with sum
Python Console
or more info
3 set([u'Caribbean', u'Middle Africa', u'Southern Europe', u'Micronesia', u'Western Africa', u'Central Asia', u'South-Eastern Asia', u'Eastern Africa', u'Northern Europe', u'Seven seas (open ocean)', u'South America', u'Eastern Asia', u'Northern America', u'Polynesia', u'Southern Africa', u'Western Asia', u'Southern Asia', u'Melanesia', u'Australia and New Zealand', u'Eastern Europe', u'Northern Africa', u'Central America', u'Antarctica', u'Western Europe'])
4
>>>
Browser
Layers
Advanced interface
Coordinate: 28.1,-118.0
Scale :337,929,239
Rotation: 0.0
Render
EPSG:4326


Layers
ne_10m_admin_0_coun...
Processing Toolbox
Search...
Recently used algorithms
Fixed distance buffer
Convex hull
Dissolve
v.dissolve - Dissolves boundaries b...
dissolve with sum
GDAL/OGR [45 geoalgorithms]
GRASS commands [168 geoalgorithms]
Models [1 geoalgorithms]
Orfeo Toolbox (Image analysis) [83 ge...
QGIS geoalgorithms [99 geoalgorithms]
SAGA (2.1.2) [235 geoalgorithms]
Scripts [1 geoalgorithms]
Tools
User scripts
Execute
Execute as batch process
Edit rendering styles for outputs
Edit script
Delete script
Python Console
4 set([u'Caribbean', u'Middle Africa', u'Southern Europe', u'Micronesia', u'Western Africa', u'Central Asia', u'South-Eastern Asia', u'Eastern Africa', u'Northern Europe', u'Seven seas (open ocean)', u'South America', u'Eastern Asia', u'Northern America', u'Polynesia', u'Southern Africa', u'Western Asia', u'Southern Asia', u'Melanesia', u'Australia and New Zealand', u'Eastern Europe', u'Northern Africa', u'Central America', u'Antarctica', u'Western Europe'])
>>>
Browser
Layers
Advanced interface
Coordinate: 184.1,-88.7
Scale :391,789,426
Rotation: 0.0
Render
EPSG:4326

- http://bit.ly/2TT3kwE