Python térinformatikai programozás

Siki Zoltán

A következő anyagok felhasználásával:

https://github.com/elpaso/python-gis-workshop/blob/master/python_gis_part1.rst
https://github.com/elpaso/python-gis-workshop/blob/master/python_gis_part2.rst
https://pcjericks.github.io/py-gdalogr-cookbook
http://www.gdal.org/gdal_tutorial.html
http://www.gis.usu.edu/~chrisg/python/2008/os5_slides.pdf
Lawehead, J: Learning Geospatial Analysis wuth Python, PacktPub 2013
Westra, E: Python Geospatial Development, PacktPub 2013

Python térinformatikai programozás

Siki Zoltán

A következő anyagok felhasználásával:

https://github.com/elpaso/python-gis-workshop/blob/master/python_gis_part1.rst
https://github.com/elpaso/python-gis-workshop/blob/master/python_gis_part2.rst
https://pcjericks.github.io/py-gdalogr-cookbook
http://www.gdal.org/gdal_tutorial.html
http://www.gis.usu.edu/~chrisg/python/2008/os5_slides.pdf
Lawehead, J: Learning Geospatial Analysis wuth Python, PacktPub 2013
Westra, E: Python Geospatial Development, PacktPub 2013

Python térinformatikai programozás

Siki Zoltán

Állományok letöltése: http://www.geod.bme.hu/gis/workshop4/eloadasok/python.zip

A következő anyagok felhasználásával:

https://github.com/elpaso/python-gis-workshop/blob/master/python_gis_part1.rst
https://github.com/elpaso/python-gis-workshop/blob/master/python_gis_part2.rst
https://pcjericks.github.io/py-gdalogr-cookbook
http://www.gdal.org/gdal_tutorial.html
http://www.gis.usu.edu/~chrisg/python/2008/os5_slides.pdf
Lawehead, J: Learning Geospatial Analysis wuth Python, PacktPub 2013
Westra, E: Python Geospatial Development, PacktPub 2013

Minimális alapok

Windows: OSGeo4w Shell indítása
Linux: burok indítása
python

```
>>> print 'hello world'
'hello world'
>>> a = 1.342
>>> 2 * a + 4 * a * a
9.887856000000001
>>> import math
>>> math.sin(math.pi / 4.0)
0.7071067811865475
>>> l = [ 'hello', 102, 1.453]
>>> l[0]
'hello'
>>> l[1:]
[102, 1.453]
```

```
>>> l.append(10)
>>> l
['hello', 102, 1.453, 10]
>>> s = { 'nev' : 'Python',
.... 'verzio': 2.7}
>>> s['nev']
>>> for i in range(3):
...     print i
...
0
1
2
```

GDAL, raszteres adatok

```
>>> from osgeo import gdal
>>> ds = gdal.Open('resz.tif', gdal.GA_ReadOnly)
>>> ds.GetProjection()
'PROJCS["HD72 / EOV",GEOGCS["HD72",DATUM["Hungarian_Datum...
>>> ds.RasterXSize        # ds.RasterYSize is van!
2000
>>> ds.RasterCount
1
>>> ds.GetGeoTransform()
(641696.9727068803, 25.06431456120198, 0.0, 214703.19343912468,
0.0, -25.06431456120198)
```

Pixelméret

Bal felső sarok

Nincs forgatás

```
>>> ds.GetRasterBand(1).GetStatistics(True, True)
[0.0, 2140.0676269531, 165.98723347207, 191.03936116502]
```

min. maximum átlag szórás

```
>>> data = band.ReadAsArray(0,0, ds.RasterXSize, ds.RasterYSize)
>>> data[200,300]
```

Semmit sem tudsz jól csinálni?

GDAL folytatás

raster_area.py

```
from osgeo import gdal
data = gdal.Open("resz.tif", gdal.GA_ReadOnly) # open dtm image
geotr = data.GetGeoTransform()
pixel_area = abs(geotr[1] * geotr[5])
band = data.GetRasterBand(1)     # get the only band
area = 0.0                       # variable for area sum
for y in range(band.YSize):      # soronkénti feldolgozás
    values = band.ReadAsArray(0, y, band.XSize, 1)
    values = values[0,:]       # 2D -> 1D tömb
    area += sum([value for value in values if value < 300])
total_area = band.XSize * band.YSize * pixel_area
print area, total_area, round(area / total_area * 100)
```

GDAL folytatás

raster_area1.py

```
from osgeo import gdal
import struct
data = gdal.Open("resz.tif", gdal.GA_ReadOnly) # open dtm image
geotr = data.GetGeoTransform()
pixel_area = abs(geotr[1] * geotr[5])
band = data.GetRasterBand(1)     # get the only band
fmt = "<" + ("f" * band.XSize)   # float32 data
area = 0.0                       # variable for area sum
for y in range(band.YSize):
    scanline = band.ReadRaster(0, y, band.XSize, 1, band.XSize,
          1, band.DataType)
    values = struct.unpack(fmt, scanline)
    area += sum([value for value in values if value < 300])
total_area = band.XSize * band.YSize * pixel_area
print area, total_area, round(area / total_area * 100)
```

Hatékonyabb megoldás, de kevésbé érthető

Bár a praktikusság veri a tisztaságot.

OGR, vektoros adatok

```
>>> from osgeo import ogr
>>> driver = ogr.GetDriverByName('ESRI Shapefile')
>>> datasource = driver.Open('megye.shp', 0)
>>> print datasource.GetLayerCount()
1
>>> layer = datasource.GetLayer()
>>> print layer.GetFeatureCount()
20
>>> print layer.GetExtent()
(426738.12004999997, 937422.49975, 43841.009849999995, 360722.17...
>>> layerDefn = layer.GetLayerDefn()
>>> layerDefn.GetFieldCount()
5
>>> layerDefn.GetGeomType()
3
>>> fieldDefn = layerDefn.GetFieldDefn(0)
>>> fieldDefn.GetName()
'stsum'
```

Néha úgy érzem, hogy az enyém a legrosszabb munka a világon!

Igen ... Igaz!

OGR folyt.

```
>>> feature = layer.GetFeature(0)
>>> feature.GetFID()
0
>>> feature.GetField('Nev')
'Budapest'
>>> geometry = feature.GetGeometryRef()
>>> geometry.GetEnvelope()
(313352.32445650722, 517043.7912779671, 4879624.4439933635, ...
>>> geometry.GetGeometryName()
'POLYGON'
>>> geometry.IsValid()
True
>>> geometry.GetDimension()
2
>>> geometry.GetArea()
565957218.4118297
```

OGR folyt.

list.py

```
from osgeo import ogr
shapefile = ogr.Open("megye.shp")
layer = shapefile.GetLayer(0)

for i in range(layer.GetFeatureCount()):
    feature = layer.GetFeature(i)
    name = feature.GetField("Nev")
    geometry = feature.GetGeometryRef()
    print i, name, geometry.GetGeometryName()
```

OGR folyt.

box.py

```
#!/usr/bin/python    shebang for Unix
""" calculate bounding box for counties
"""

import ogr
shapefile = ogr.Open("megye.shp")
layer = shapefile.GetLayer(0)
counties = [] # List of (name,minLat,maxLat,minLong,maxLong) tupl
for i in range(layer.GetFeatureCount()):
    feature = layer.GetFeature(i)
    name = feature.GetField("NEV")
    geometry = feature.GetGeometryRef()
    minLong,maxLong,minLat,maxLat = geometry.GetEnvelope()
    counties.append((name, minLat, maxLat, minLong, maxLong))
counties.sort()
for name,minLat,maxLat,minLong,maxLong in counties:
    print "%s Y=%0.4f..%0.4f, X=%0.4f..%0.4f" \
        % (name, minLong, maxLong, minLat, maxLat)
```

Ébren van még valaki?

OSR, vetületi transzformáció

wgs2eov.py

```
import sys
from osgeo import osr
from osgeo import ogr
to = osr.SpatialReference()
to.ImportFromEPSG(23700)    # EOV
fr = osr.SpatialReference()
fr.ImportFromEPSG(4326)     # WGS84
trans = osr.CoordinateTransformation(fr, to)
point = ogr.Geometry(ogr.wkbPoint)
if len(sys.argv) > 2:                # parancssori paraméterek
    lat = float(sys.argv[1])
    lon = float(sys.argv[2])
    point.AddPoint(lon, lat)
    point.Transform(trans)
    print point.GetX()
    print point.GetY()
```

Proj.4/pyproj
vetületi számítások

```
>>> import pyproj
>>> help(pyproj)
>>> lon1, lat1 = (19.054419, 47.481921)
>>> lon2, lat2 = (19.053724, 47.479310)
>>> geod = pyproj.Geod(ellps='WGS84')
>>> ans =geod.inv(lon1, lat1, lon2, lat2) # oda, vissza azimut, táv.
>>> ans
(-169.7708089360682, 10.228678815066615, 294.97992656182424)
>>> geod.fwd(lon1, lat1, ans[0], ans[2])
(19.053724, 47.47931000000001, 10.228678815066615)
>>> p = pyproj.Proj('+init=EPSG:23700')
>>> p(lon1, lat1)
(650440.6801851124, 237522.99221407762)
```

QGIS Python konzol

Indítsuk el a QGIST és nyissuk meg a megye.shp-t

QGIS által kiadott parancsok:
from qgis.core import *
import qgis.utils

A konzolban Python parancsokat adhatunk ki

```
>>> layer = qgis.utils.iface.activeLayer()
>>> layer.featureCount()
20L
>>> layer.geometryType()
2
>>> layer.extent().xMinimum()
426738.12004999997
```

QGIS szkript futtató modul

```
from glob import glob
from os import path

def run_script(iface):
    ldr = Loader(iface)    # az osztály egy példányának létrehozása
    ldr.load_shapefiles('/home/siki/mo_uj')    # az objektum metódusának
futtatása

# az osztály, mely a betöltést végzi
class Loader:
    def __init__(self, iface):
        self.iface = iface
    def load_shapefiles(self, shp_path):
        """Load all shapefiles found in shp_path"""
        print "Loading shapes from %s" % path.join(shp_path, "*.shp")
        shps = glob(path.join(shp_path, "*.shp"))   # minden shape-re
        for shp in shps:
            (shpdir, shpfile) = path.split(shp)    # könyvtár és név szétvál.
            self.iface.addVectorLayer(shp, shpfile, 'ogr' )    # betöltés
```

Szkipt futtató ezt indítja el

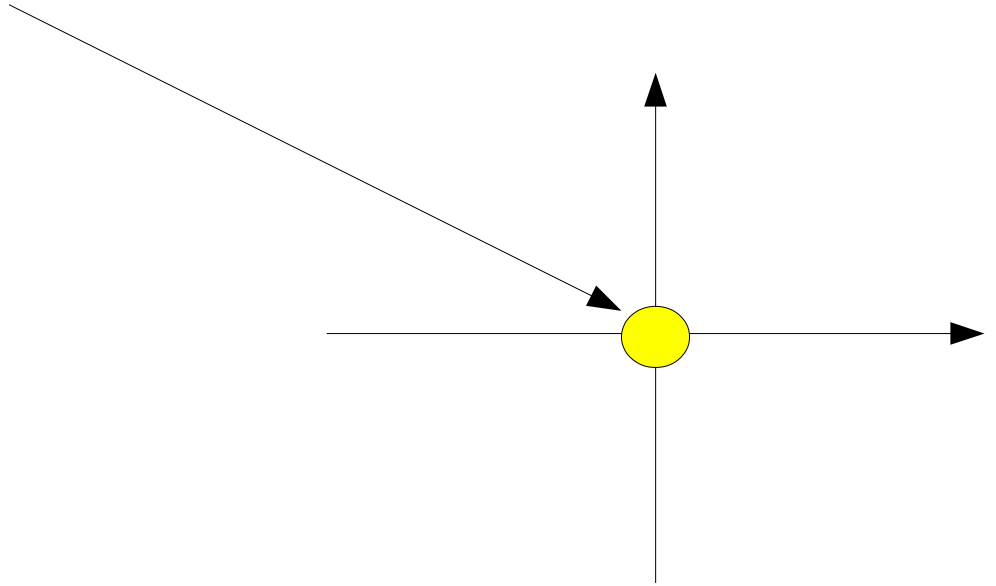
Ezt le kell cserélni!

További magyar nyelvű anyagok

Python mogyoróhéjban
http://www.geod.bme.hu/gis/python/python_oktato.pdf
Python GDAL/OGR programozás
http://www.geod.bme.hu/gis/gdal/ogr_python.pdf
DXF fájl konvertálása Shape fájlba
http://www.geod.bme.hu/gis/gdal/dxf2shp_py.pdf
Python kód használata QGIS-ben
http://www.geod.bme.hu/gis/qgis/qgis_and_python.pdf
Python konzol (QGIS)
http://www.geod.bme.hu/gis/qgis/python_konzol.pdf
QGIS Python modul készítés
http://www.geod.bme.hu/gis/qgis/plugins_tutorial.pdf
QGIS szkript futtató modul
http://www.geod.bme.hu/gis/qgis/script_runner.pdf

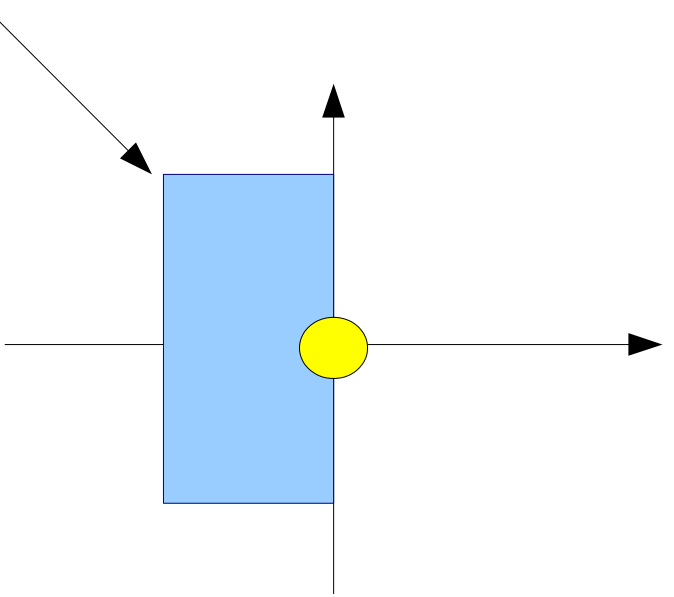
Shapely/GEOS

```
>>> from shapely.geometry import Point
>>> point = Point(0.0, 0.0)
>>> point.area
0.0
>>> point.bounds
(0.0, 0.0, 0.0, 0.0)
>>> point.x, point.y
(0.0, 0.0)
>>> point.area
0.0
>>> point.length
0.0
>>> point.geom_type
'Point'
>>> point.wkt
'POINT (0.0000000000000000 0.0000000000000000)'
```



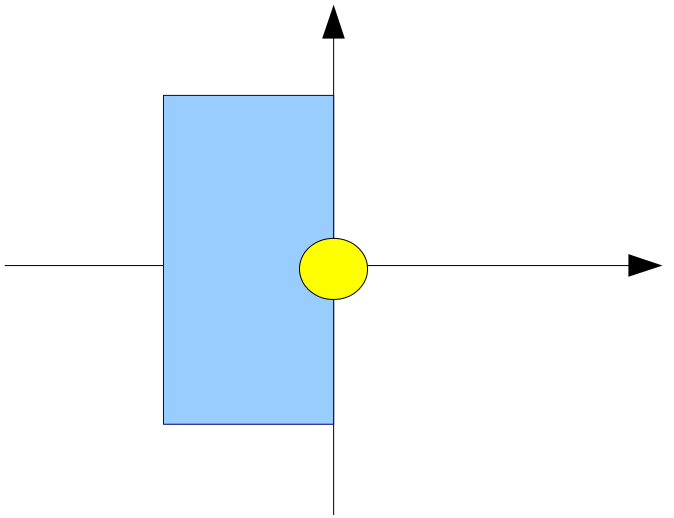
Shapely folyt.

```
>>> from shapely.geometry import Polygon
>>> polygon = Polygon([(-1,-1), (-1,1), (0,1), (0,-1)])
>>> polygon.area
2.0
>>> polygon.length
6.0
>>> polygon.bounds
(-1.0, -1.0, 0.0, 1.0)
>>> polygon.geom_type
'Polygon'
>>> polygon.wkt
'POLYGON ((-1.0000000000000000 -1.0000000000000000, ...
>>> list(polygon.exterior.coords)
[(-1.0, -1.0), (-1.0, 1.0), (0.0, 1.0), (0.0, -1.0), (-1.0, -1.0)]
>>> list(polygon.interiors)
[]
```



Shapely folyt.

```
>>> polygon.has_z
False
>>> polygon.is_empty
False
>>> polygon.is_valid
True
>>> polygon.contains(point)
False
>>> buffer = polygon.buffer(1)
>>> buffer.contains(point)
True
```



Shapely folyt.

```
>>> coords = [(0, 0), (0, 2), (1, 1), (2, 2), (2, 0), (1, 1), (0, 0)]
>>> p = Polygon(coords)
>>> from shapely.validation import explain_validity
>>> explain_validity(p)
'Ring Self-intersection[1 1]'
```