Krótkie wprowadzenie do GeoPythona

Małgorzata Papież

Geo czyli co ?

GIS (ang. Geographic Information System)

Geoinformacja

Informacja przestrzenna

Informacja geoprzestrzenna

Dane geoprzestrzenne

Geodane

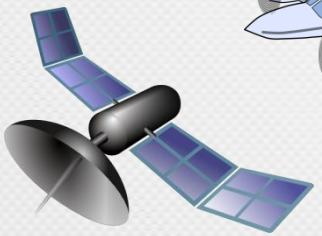
Lokalizacja obiektów geograficznych na powierzchni ziemi poprzez współrzędne (x,y,z) oraz dane opisowe.

PRZECHOWYWANIE

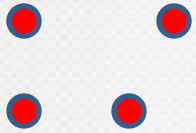
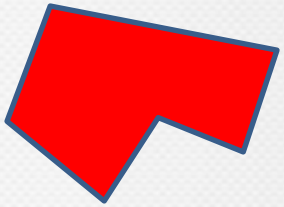
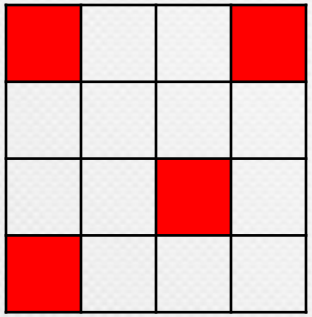
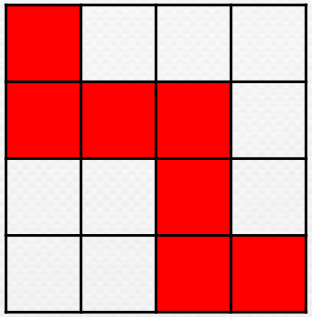
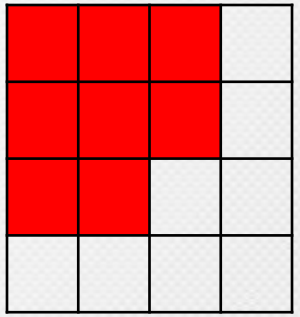
ANALIZOWANIE

WYŚWIETLANIE

POZYSKIWANIE



Typy danych

	PUNKTY	LINIE	POWIERZCHNIE
WEKTOR			
RASTER			
			
	http://www.garnek.pl/	https://pl.wikipedia.org	https://pl.wikipedia.org

Formaty danych

Formaty danych rastrowych

GeoTiff
JPEG, JPEG2000
BMP
GIF
KEA

http://www.gdal.org/formats_list.html

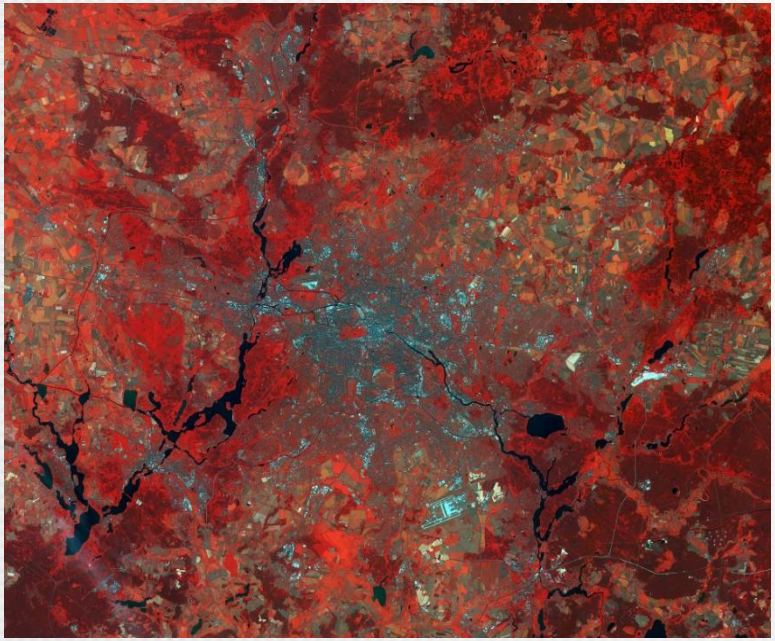
Formaty danych wektorowych

ESRI Shapefile
KML
GML
GPX
XSL,CSV
GeoJSON

http://www.gdal.org/ogr_formats.html

Przykładowe dane

Przykład danych rastrowych



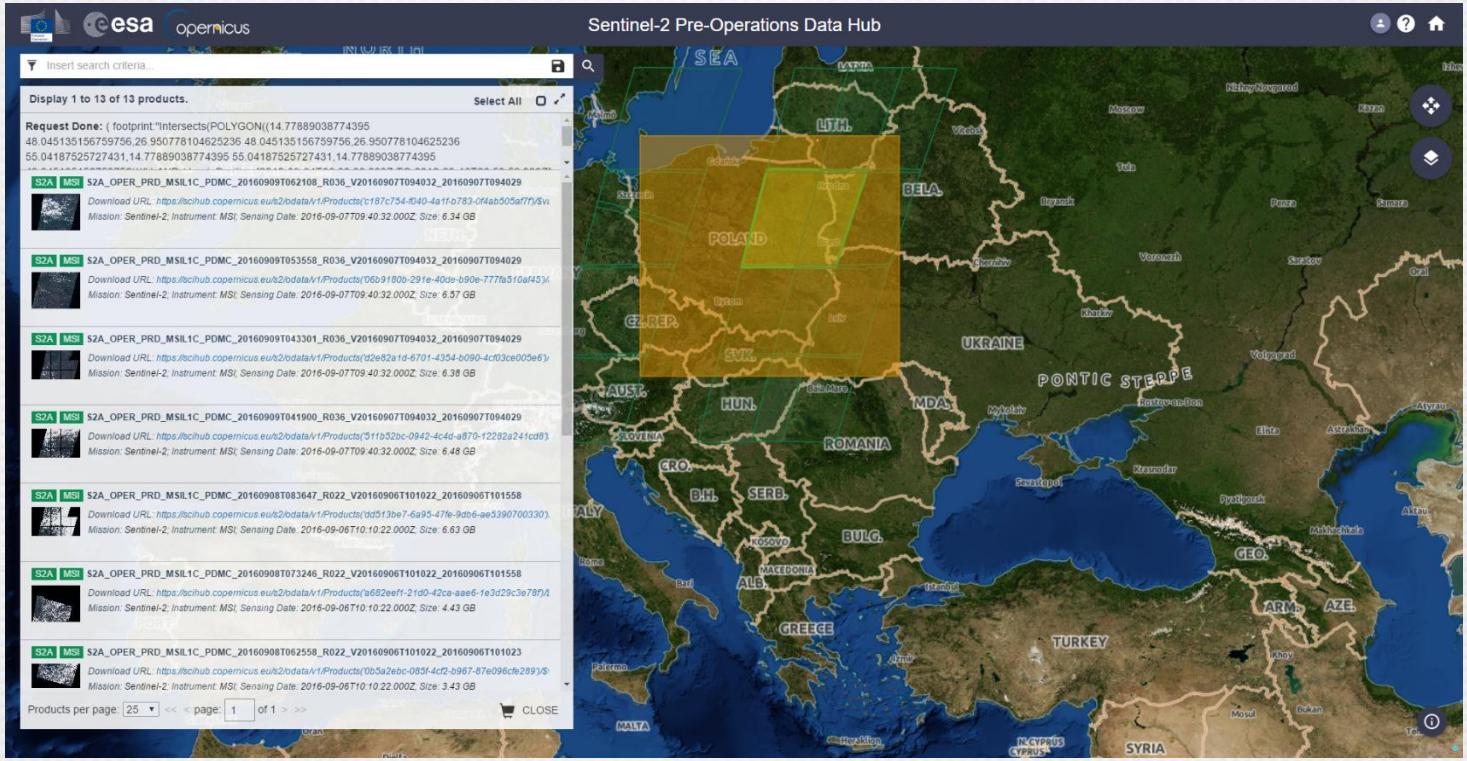
Źródło: Sentinel data (2015)/ESA
Berlin

Przykład danych wektorowych

Źródło: CODIK
Mapa wektorowa Polski

Darmowe dane rastrowe

https://scihub.copernicus.eu/



Współrzędne geograficzne

- Współrzędne wyznaczają położenie punktu na powierzchni Ziemi
- Szerokość i długość geograficzna
- Dotyczą one powierzchni kuli albo elipsoidy
- Nie nadają się do dokładnego wyznaczania punktów na mapie

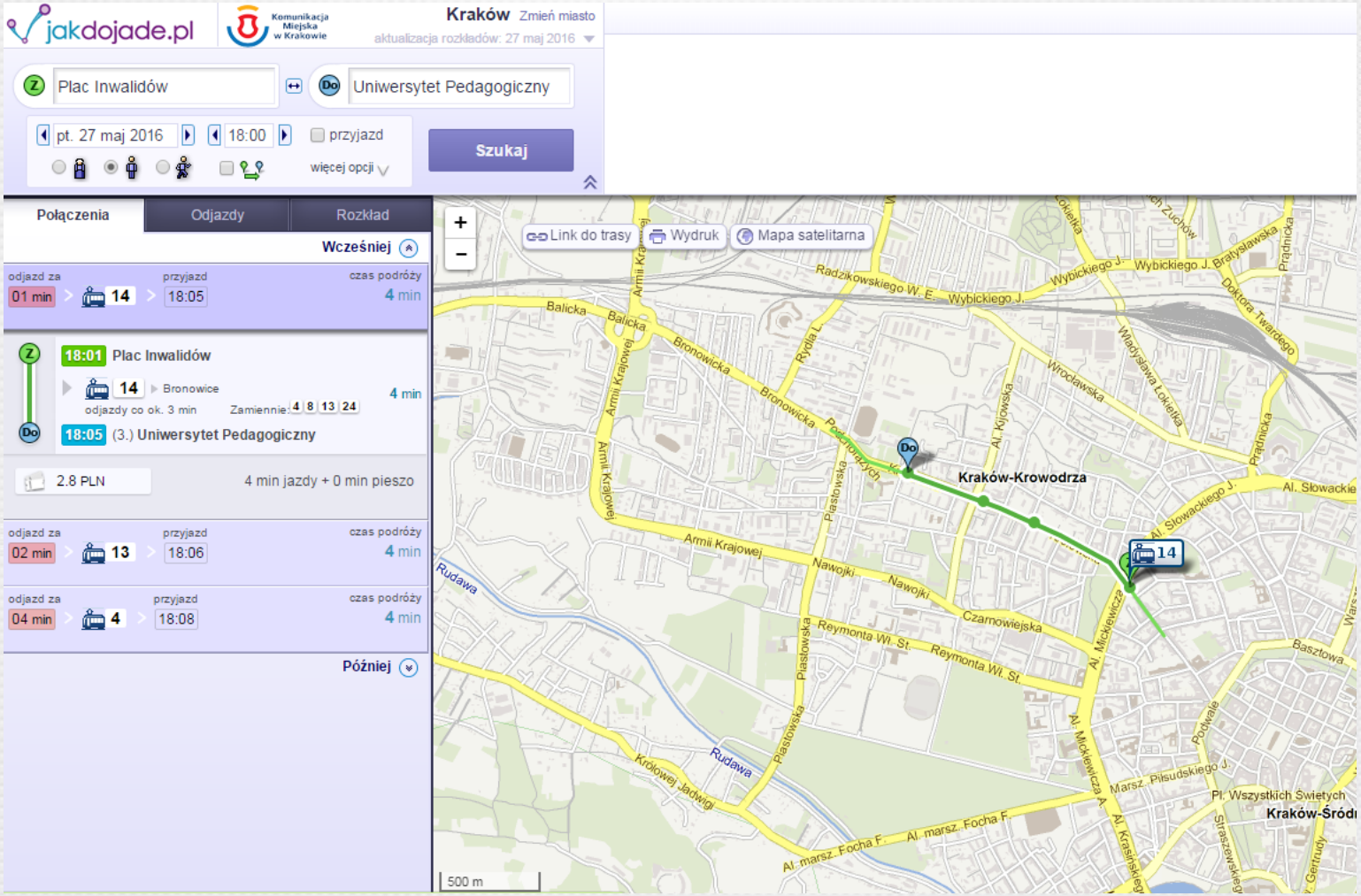
Współrzędne prostokątne płaskie

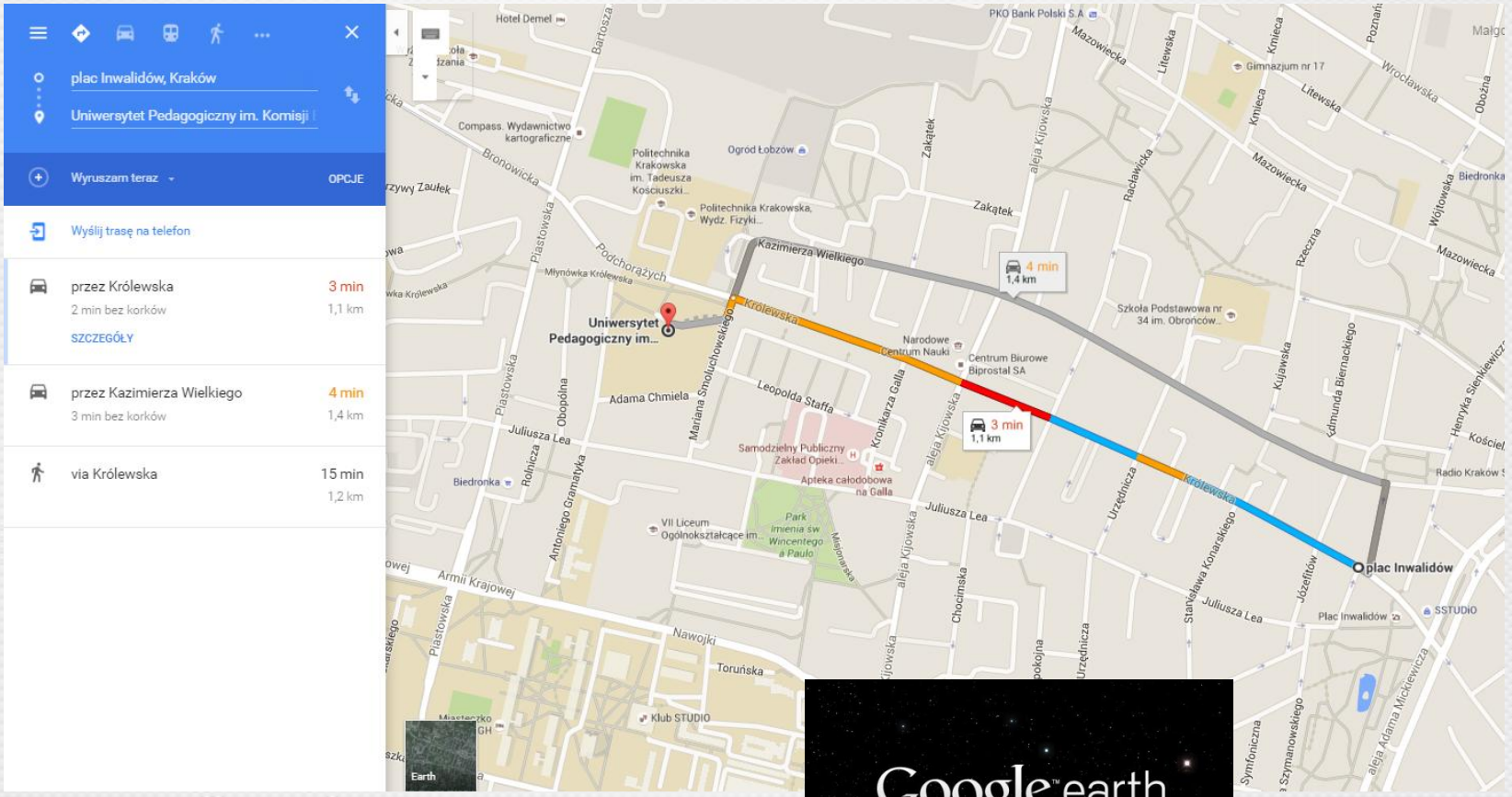
- Określone jako współrzędne x,y
- Aby uzyskać współrzędne płaskie stosuje się odwzorowania płaskie
- Z racji że Ziemia ma kształt geoidy nie łatwo jest odwzorować ją na płaskiej powierzchni – powstają zniekształcenia

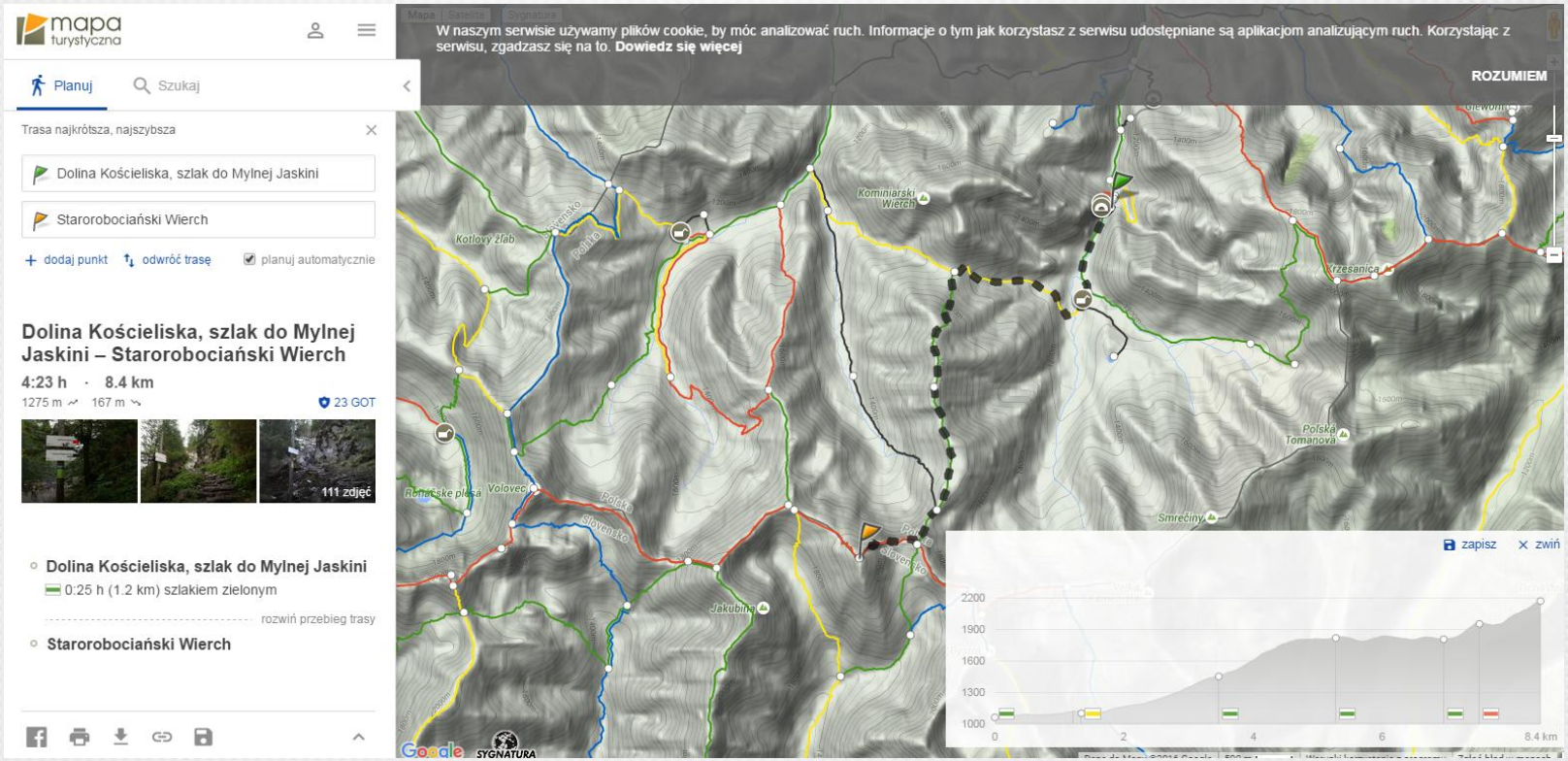
Układy współrzędnych

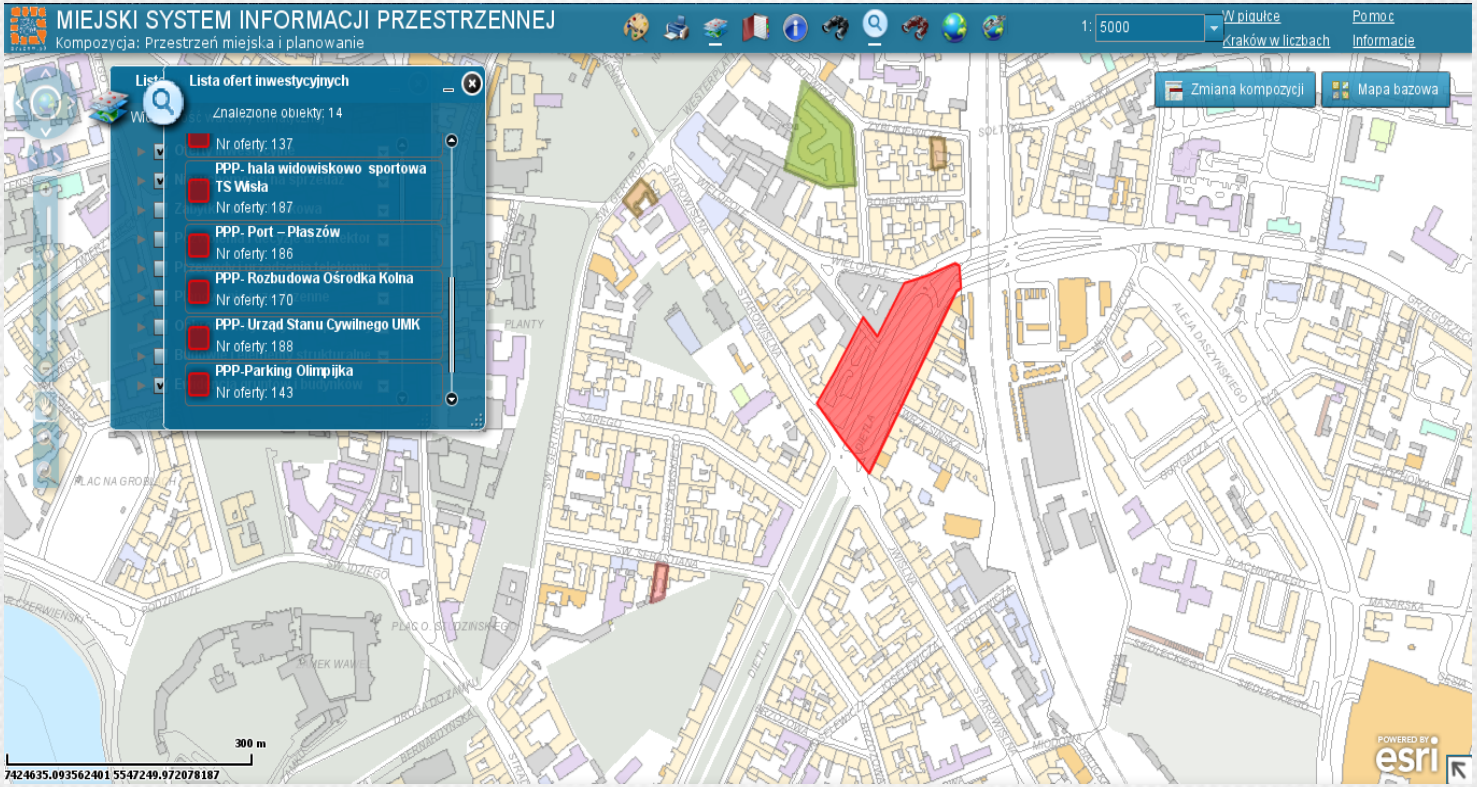
- Matematyczny model przeniesienia powierzchni kuli na powierzchnię płaską
- Powstały aby minimalizować błędy przekształceń współrzędnych
- WGS84 – jednolity dla całego świata
- Kody EPSG - kody dla układów współrzędnych

Przykład użycia

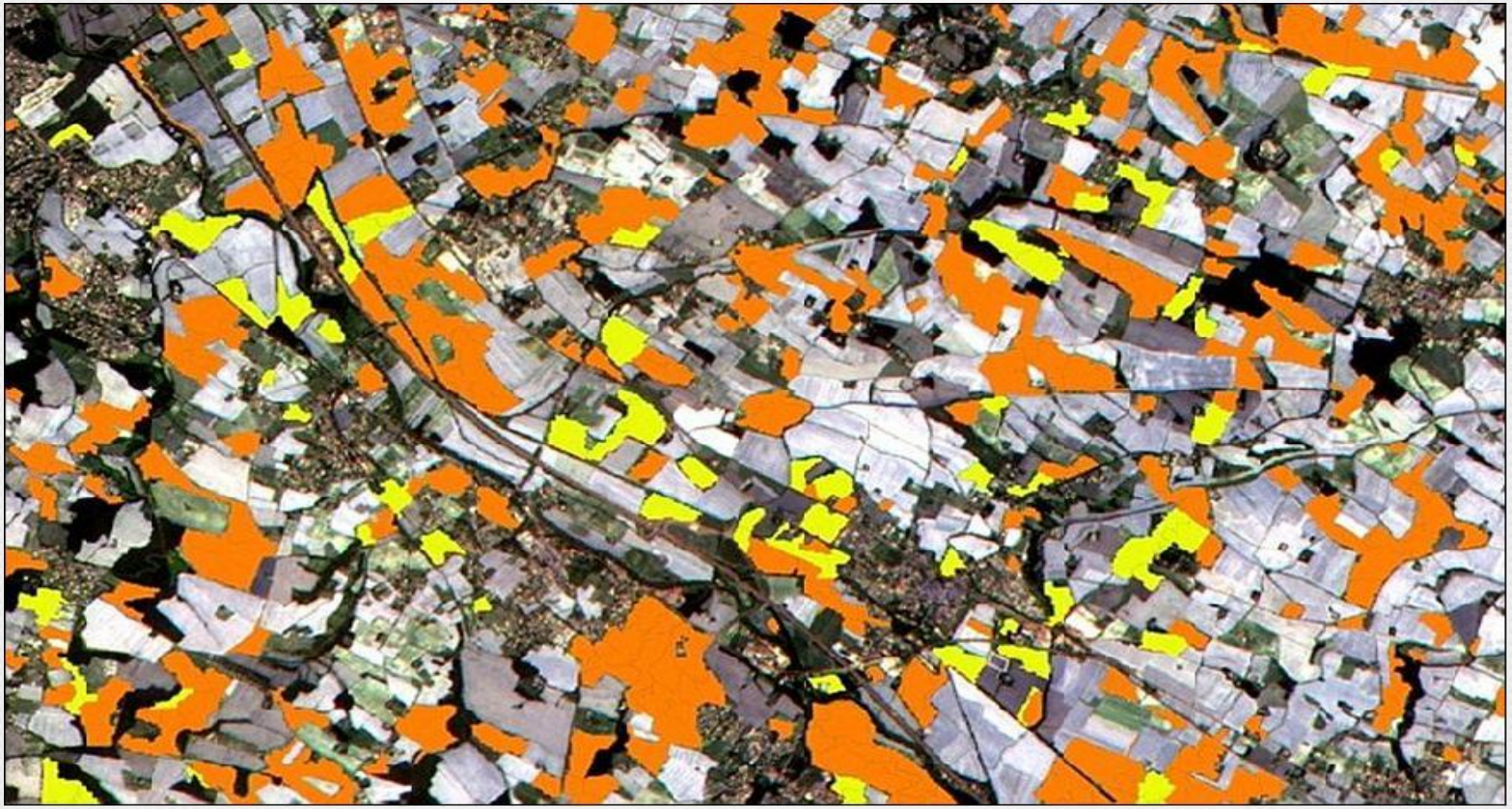



plac Inwalidów, Kraków
Uniwersytet Pedagogiczny im. Komisji
Wyruszam teraz
OPCJE
Wyślij trasę na telefon
przez Królewska
3 min
2 min bez korków
1,1 km
SZCZEGÓŁY
przez Kazimierza Wielkiego
4 min
3 min bez korków
1,4 km
via Królewska
15 min
1,2 km
Uniwersytet Pedagogiczny im...
Kazimierza Wielkiego
Królewska
4 min
1,4 km
3 min
1,1 km
plac Inwalidów
Juliusza Lea
Armii Krajowej
Nawojki
Toruńska
Leopolda Staffa
Adama Chmiela
Mariana Smoluchowskiego
Podchorążych
Piastowska
Bronowicka
Zakątek
aleja Kijowska
Mazowiecka
Litewska
Racławicka
Urzędnicza
Chocimska
Stanisława Konarskiego
Józefitów
Kujawska
Edmunda Biernackiego
Rzeczna
Wrocławska
Obożna
Wójtowska
Kmiecia
Poznańska
Hotel Demel
PKO Bank Polski S.A.
Gimnazjum nr 17
Compass. Wydawnictwo kartograficzne
Politechnika Krakowska im. Tadeusza Kościuszki...
Ogród Łobzów
Politechnika Krakowska, Wydz. Fizyki...
Młynówka Królewska
Narodowe Centrum Nauki
Centrum Biurowe Biprostal SA
Szkoła Podstawowa nr 34 im. Obrońców...
Samodzielny Publiczny Zakład Opieki...
Apteka całodobowa na Galla
Biedronka
VII Liceum Ogólnokształcące im...
Park imienia św Wincentego a Paulo
Klub STUDIO
Plac Inwalidów
SSTUDIO
Radio Kraków
Earth
Google earth


mapa turystyczna
Planuj
Szukaj
Trasa najkrótsza, najszybsza
Dolina Kościeliska, szlak do Mylnej Jaskini
Starorobociański Wierch
dodaj punkt
odwróć trasę
planuj automatycznie
Dolina Kościeliska, szlak do Mylnej Jaskini – Starorobociański Wierch
4:23 h · 8.4 km
1275 m 167 m
23 GOT
111 zdjęć
Dolina Kościeliska, szlak do Mylnej Jaskini
0:25 h (1.2 km) szlakiem zielonym
rozwiń przebieg trasy
Starorobociański Wierch
W naszym serwisie używamy plików cookie, by móc analizować ruch. Informacje o tym jak korzystasz z serwisu udostępniane są aplikacjom analizującym ruch. Korzystając z serwisu, zgadzasz się na to. Dowiedz się więcej
ROZUMIEM
Kominiarski Wierch
Krzesanica
Polska Tomanová
Smrečiny
Jakubina
Kotlový žľab
Volovec
zapisz
zwiń
2200
1900
1600
1300
1000
0
2
4
6
8.4 km
Google
SYGNATURA


MIEJSKI SYSTEM INFORMACJI PRZESTRZENNEJ
Kompozycja: Przestrzeń miejska i planowanie
1: 5000
W pigułce
Kraków w liczbach
Pomoc
Informacje
Zmiana kompozycji
Mapa bazowa
Lista ofert inwestycyjnych
Znalezione obiekty: 14
Nr oferty: 137
PPP- hala widowiskowo sportowa TS Wisła
Nr oferty: 187
PPP- Port – Płaszów
Nr oferty: 186
PPP- Rozbudowa Ośrodka Kolna
Nr oferty: 170
PPP- Urząd Stanu Cywilnego UMK
Nr oferty: 188
PPP-Parking Olimpijka
Nr oferty: 143
300 m
7424635.093562401 5547249.972078187
POWERED BY esri

Sentinel-2



Sentinel-2a, Francja, 2015; ESA
(kolor żółty – słoneczniki, kolor pomarańczowy – kukurydza)

Biblioteki przestrzenne - Python

GDAL/OGR

Pyproj

Shapely

Mapnik

Basemap

Fiona

GeoPanda

GDAL/OGR

GDAL (Geospatial Data Abstraction Library) to darmowa biblioteka do odczytywania danych rastrowych. Zawiera również bibliotekę OGR (OpenGIS Reference) do danych wektorowych.

GDAL napisany w C++ ale posiada bindingi dla Pythona.

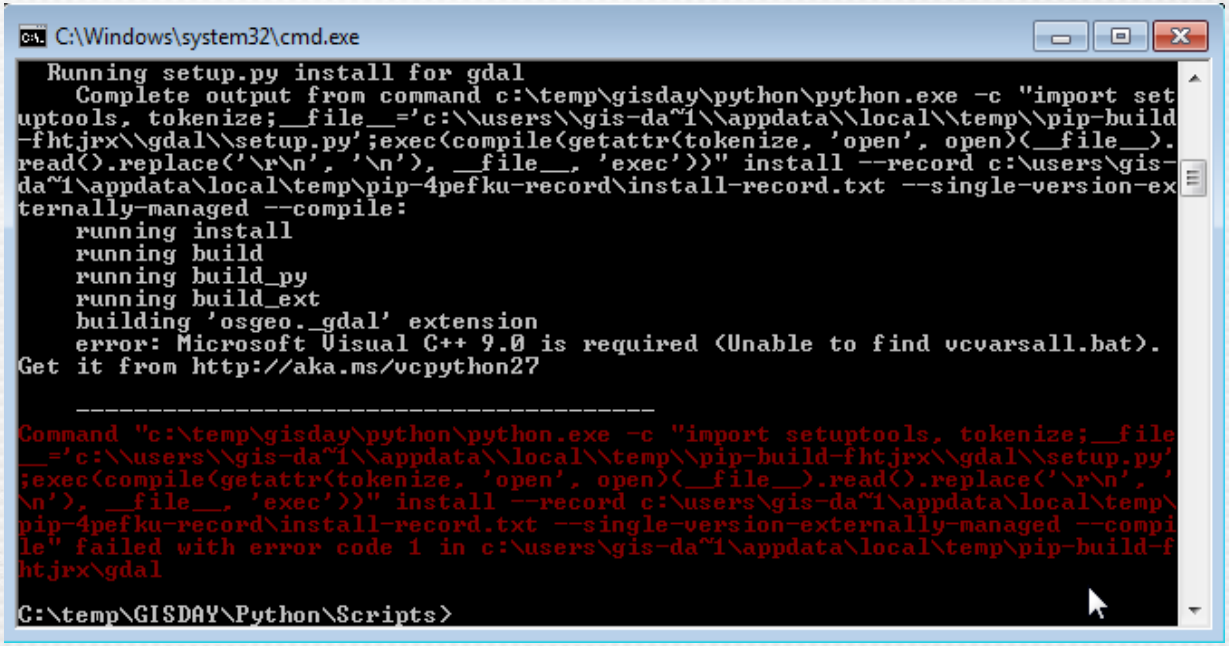
```
from osgeo import gdal
from osgeo import ogr
```

GDAL/OGR – importowanie i instalacja

>> import *gdal*

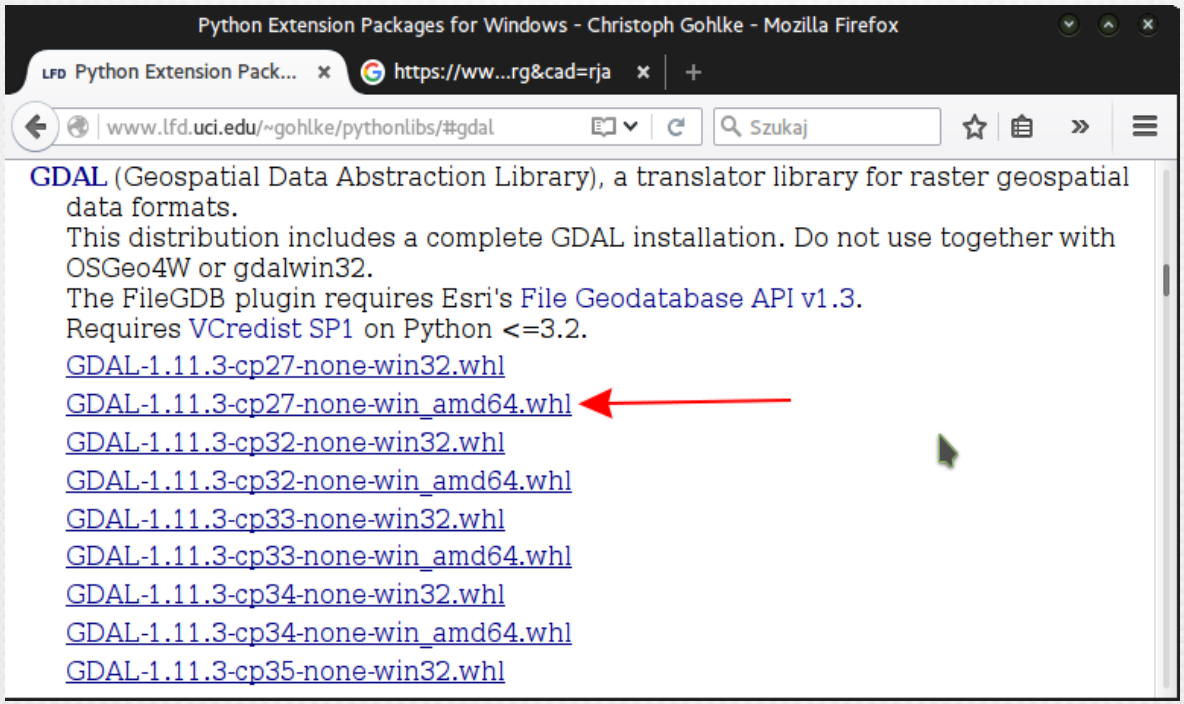
Niestety import nie zadziałała - standardowo GDAL nie jest dołączany do Pythona – należy go doinstalować.

Najprościej:
pip install gdal



GDAL/OGR – błędy instalacji

Strona na której dostępne już skompilowane moduły do Pythona
http://www.lfd.uci.edu/~gohlke/pythonlibs/



GDAL/OGR

Sprawdzenie czy GDAL został poprawnie zainstalowany.

>> import *gdal*

Brak jakiekolwiek informacji oznacza, że import przebiegł bez zakłóceń

```
Python 2.7.10 (default, May 23 2015, 09:44:00) [MSC v.1500 64 bit (AMD64)] on wi
n32
Type "copyright", "credits" or "license()" for more information.
>>> import gdal
>>>
```

GDAL - odczyt rastra

```
dataset = gdal.Open('tempRaster.tif', gdal.GA_ReadOnly)

if dataset is None:
    print 'Could not open file'
    sys.exit(1)
```

```
cols = dataset.RasterXSize
rows = dataset.RasterYSize
bands = dataset.RasterCount

print 'X: ', cols
print 'Y: ', rows
print 'Ilosc kanałów: ', bands
```

```
for band in range(bands):
    band += 1
    srcband = dataset.GetRasterBand(band)
    if srcband:
        stats = srcband.GetStatistics( True, True )
        print 'Minimum = %.3f'%stats[0]
        print 'Maksimum = %.3f'%stats[1]
```

GDAL – położenie w przestrzeni

```
In [5]: raster.GetProjection()

Out[5]: 'PROJCS["WGS 84 / UTM zone 4N",GEOGCS["WGS 84",DATUM["WGS_1984",SPHEROID["W
        GS 84",6378137,298.257223563,AUTHORITY["EPSG","7030"]],AUTHORITY["EPSG","63
        26"]],PRIMEM["Greenwich",0],UNIT["degree",0.0174532925199433],AUTHORITY["EP
        SG","4326"]],PROJECTION["Transverse_Mercator"],PARAMETER["latitude_of_origi
        n",0],PARAMETER["central_meridian",-159],PARAMETER["scale_factor",0.9996],P
        ARAMETER["false_easting",500000],PARAMETER["false_northing",0],UNIT["metre"
        ,1,AUTHORITY["EPSG","9001"]],AUTHORITY["EPSG","32604"]]'

In [6]: raster.GetGeoTransform()

Out[6]: (451581.6, 0.4, 0.0, 2420520.8, 0.0, -0.4)
```

OGR - odczyt wektora

```
import ogr
driver = ogr.GetDriverByName('ESRI Shapefile')
dataSource = driver.Open('polska.shp', 0)
numLayer = dataSource.GetLayerCount()
print 'Ilość warstw: %d'%numLayer

for feat in range(numLayer):
    layer = dataSource.GetLayerByIndex(feat)
    print 'Nazwa warstwy: %s'%layer.GetName()
    numFeatures = layer.GetFeatureCount()
    print 'Ilość obiektów w warstwie: %d'%numFeatures
```

```
Ilość warstw: 1
Nazwa warstwy: polska
Ilość obiektów w warstwie: 16
```

GDAL – układy współrzędnych

Sprawdzanie układu współrzędnych

```
from osgeo import ogr

driver = ogr.GetDriverByName('ESRI Shapefile')
dataset = driver.Open('polska.shp', 0)

layer = dataset.GetLayer()
spatialRefLayer = layer.GetSpatialRef()

feature = layer.GetNextFeature()
geom = feature.GetGeometryRef()
spatialRefObj = geom.GetSpatialReference()
```

Transformacja układów
Moduł OSR

```
from osgeo import ogr
from osgeo import osr

source = osr.SpatialReference()
source.ImportFromEPSG(2927)

target = osr.SpatialReference()
target.ImportFromEPSG(4326)

transform = osr.CoordinateTransformation(source, target)

point = ogr.CreateGeometryFromWkt("POINT (1120351.57 741921.42)")
point.Transform(transform)

print point.ExportToWkt()
```

Fiona

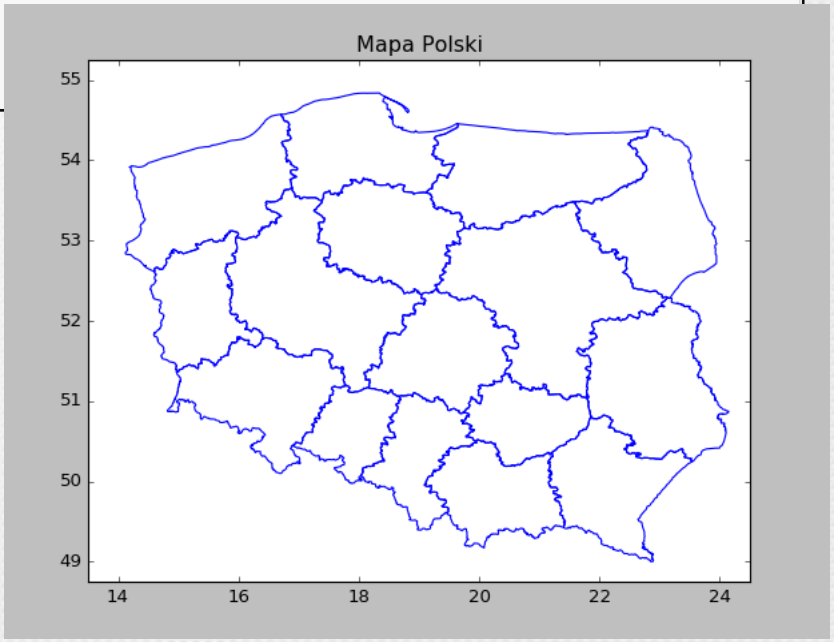
Biblioteka Pythona do zapisu i odczytu danych wektorowych. Do manipulacji danymi używa OGR ale odczyt i zapis danych oparty jest na standardach Pythona – korzysta z plików, słowników, iteratorów.

```
import fiona
f = fiona.open('polska.shp')
print 'Format: ', f.driver
print 'Ilość obiektów w warstwie: ', len(f)

rec = next(f)
print rec.keys()
print(rec['geometry']['type'])
print(rec['properties'])
```

```
Format:  ESRI Shapefile
Ilość obiektów w warstwie:  16
['geometry', 'type', 'id', 'properties']
Polygon
OrderedDict([(u'NAME', u'Zachodniopomorskie')])
```

Fiona – odczyt i wyświetlenie wektora

```
import fiona
import numpy as np
import matplotlib.pyplot as plt

f = fiona.open('polska.shp')

for rec in f:
    for poly in rec['geometry']['coordinates']:
        coords = np.array(poly).squeeze()
        plt.plot(coords[:,0], coords[:,1], 'r')


plt.axis([13.5,24.5,48.5,55])
plt.show()
```



Shapely

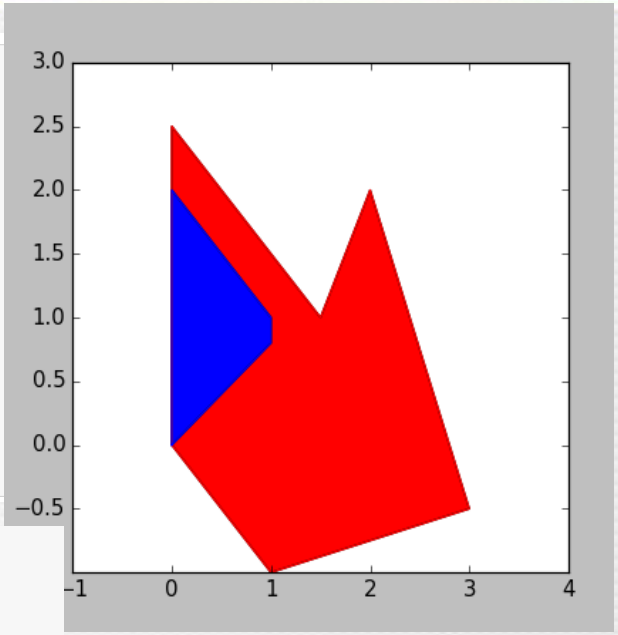
- Biblioteka Pythona do analizy i manipulacji danymi wektorowymi
- Tworzenie geometrii
- Sprawdzanie poprawności geometrii
- Operacje geometryczne

```
from shapely.geometry import LineString
line = LineString([(0, 0), (8, 1)])
print 'długość linii: ', line.length
print 'odelgłość do punktu(1,1): ', line.distance(Point(1,1))
```

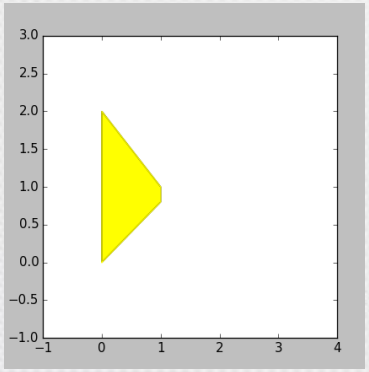
```
długość linii:  8.0622577483
odelgłość do punktu(1,1):  0.868243142124
```

Shapely

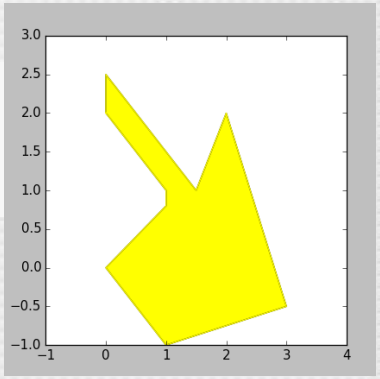
```
from matplotlib import pyplot as plt

poly1 = Polygon([(0, 0), (1, -1), (3, -0.5), (2, 2), (1.5, 1), (0, 2.5), (0, 0)])
poly2 = Polygon([(0, 0), (0, 2), (1, 1), (1, 0.8), (0, 0)])

x1,y1 = poly1.exterior.xy
x2,y2 = poly2.exterior.xy

fig = plt.figure(1, figsize=(5,5), dpi=90)
ax = fig.add_subplot(111)
ax.plot(x1, y1, color='red')
ax.fill(x1, y1, 'red')
ax.plot(x2, y2, color='blue')
ax.fill(x2, y2, 'blue')

plt.axis([-1,4,-1,3])
plt.show()
```




```
interGeom = poly1.intersection(poly2)
x3,y3 = interGeom.exterior.xy
ax.plot(x3, y3, color='yellow')
```



```
difGeom = poly1.difference(poly2)
x3,y3 = difGeom.exterior.xy
ax.plot(x3, y3, color='yellow')
```

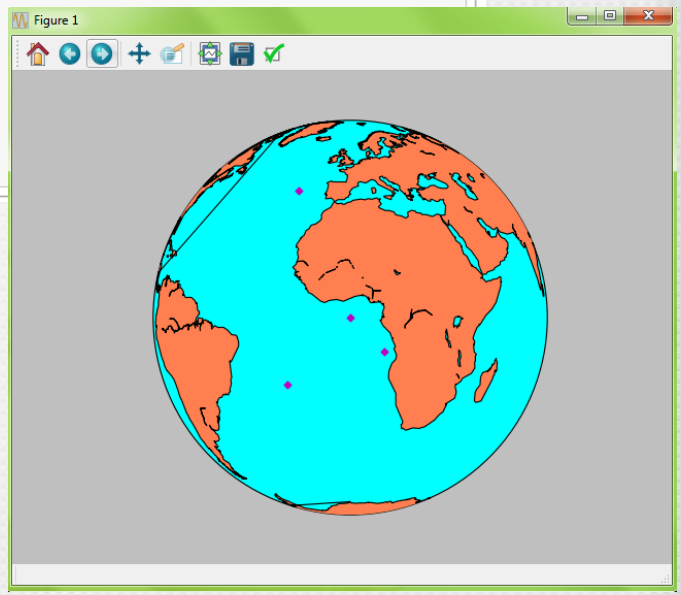


Basemap

- Biblioteka do wizualizacji 2D w Pythonie
- Oparte na toolboxie Matlaba
- Odczyt plików wektorowych w formacie Shapefile

Rysowanie mapy świata

```
from mpl_toolkits.basemap import Basemap
import matplotlib.pyplot as plt

map = Basemap(projection='ortho',
              lat_0=0, lon_0=0)

map.drawmapboundary(fill_color='aqua')
map.fillcontinents(color='coral',lake_color='aqua')
map.drawcoastlines()

lons = [0, 10, -20, -20]
lats = [0, -10, 40, -20]

x, y = map(lons, lats)

map.scatter(x, y, marker='D',color='m')

plt.show()
```



Dziękuje