

Noțiuni de bază despre programarea în Python

QGIS Tutorials and Tips



Author

Ujaval Gandhi

<http://google.com/+UjavalGandhi>

Translations by

Sorin Călinică

Noțiuni de bază despre programarea în Python

QGIS are o interfață de programare puternică, care vă îngăduie să extindeți funcționalitatea de bază a aplicației, precum și să scrieți script-uri pentru a automatiza sarcinile. QGIS suportă limbajul popular de scripting Python. Chiar dacă sunteți un începător, învățarea unor noțiuni despre Python și despre interfața de programare a QGIS va permite să fiți mult mai productivi în munca dvs. Acest tutorial nu necesită cunoștințe de programare prealabile, având scopul de a oferi o introducere în scriptarea Python în cadrul QGIS (PyQGIS).

Privire de ansamblu asupra activității

Vom încărca un strat vectorial de tip punct, reprezentând toate aeroporturile majore, folosind scriptarea Python pentru a crea un fișier text cu numele, codul, latitudinea și longitudinea pentru fiecare dintre aeroporturile stratului.

Obținerea datelor

Vom folosi setul de date [Airports](#) de la Natural Earth.

Decărcați [Fișierul shape de Aeroporturi](#).

Sursa de date [NATURALEARTH]

Procedura

1. În QGIS, mergeți la Layers ▸ Add Vector Layer. Navigați la fișierul descărcat ***ne_10m_airports.zip*** și faceți clic pe Open. Selectați stratul ***ne_10m_airports.shp*** și faceți clic pe OK.



2. Veți vedea stratul *ne_10m_airports* încărcat în QGIS.



3. Selectați instrumentul Identify și faceți clic pe oricare dintre puncte pentru a examina atributele disponibile. Veți vedea că numele aeroportului și codul de 3 cifre al acestuia sunt conținute în atributele **name** și respectiv **iata_code**.

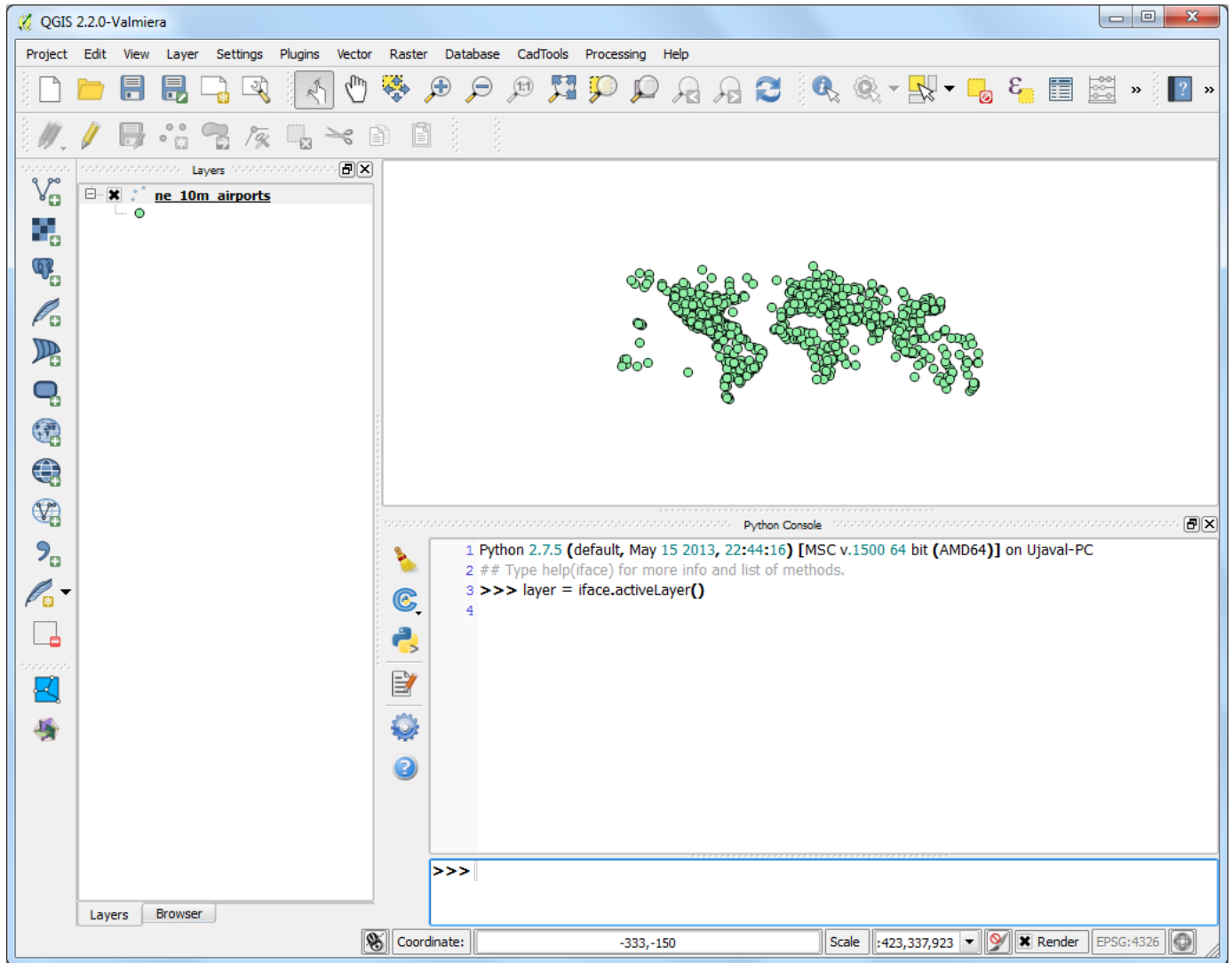


4. QGIS oferă o consolă încorporată în care puteți introduce comenzi Python și să obțineți rezultatul. Această consolă este o modalitate foarte bună de a învăța scriptarea și, de asemenea, de a procesa datele rapid. Deschideți Python Console din Plugins › Python Console.



5. Veți vedea un nou panou, deschis în partea de jos a suportului de hărți al QGIS. Veți vedea un prompt cum ar fi `>>>`, unde aveți posibilitatea să tastați comenzi. Interacționa cu mediul QGIS, se face folosind variabila `iface`. Pentru a accesa stratul din QGIS, activ în mod curent, aveți posibilitatea să tastați următoarele, apoi să apăsați Enter. Această comandă obține referința către stratul încărcat și o stochează în variabila `layer`.

```
layer = iface.activeLayer()
```



6. Există în Python o funcție utilă, numită *dir()*, care vă arată toate metodele disponibile pentru orice obiect. Acest lucru este folositor atunci când nu știți precis care funcții sunt disponibile pentru un anumit obiect. Executați următoarea comandă pentru a vedea ce operații putem executa cu variabila *layer*.

```
dir(layer)
```



7. Veți vedea o listă lungă de funcții disponibile. Pentru moment, vom folosi o funcție numită **getFeatures()**, care va obține referința către toate entitățile unui strat. În cazul nostru, fiecare entitate va fi un punct, care reprezintă un aeroport. Puteți tasta următoarea comandă, pentru a parcurge fiecare dintre entitățile din stratul curent. Asigurați-vă că ați adăugat 2 spații, înainte de a introduce a doua linie.

```

for f in layer.getFeatures():
    print f
  
```




8. După cum veți vedea în ieșire, fiecare linie conține o referire la o entitate din cadrul stratului. Referința către entitate este stocată în variabila `f`. Putem folosi variabila `f` pentru a accesa atributele fiecărei entități. Introduceți următoarele pentru a imprima `name` și `iata_code` pentru fiecare entitate de tip aeroport.

```

for f in layer.getFeatures():
    print f['name'], f['iata_code']

```



9. So now you know how to programatically access the attribute of each feature in a layer. Now, let's see how we can access the coordinates of the feature. The coordinates of a vector feature can be accessed by calling the **geometry()** function. This function returns a geometry object that we can store in the variable **geom**. You can run **asPoint()** function on the geometry object to get the x and y coordinates of the point. If your feature is a line or a polygon, you can use **asPolyline()** or **asPolygon()** functions. Type the folloing code at the prompt and press Enter to see the x and y coordinates of each feature.

```

for f in layer.getFeatures():
    geom = f.geometry()
    print geom.asPoint()

```



10. Ce se întâmplă dacă vrem să obținem numai coordonata x a entității? Puteți apela funcția `x()` pe obiectul punct, pentru a-i obține coordonata x .

```

for f in layer.getFeatures():
    geom = f.geometry()
    print geom.asPoint().x()

```



11. Now we have all the pieces that we can stitch together to generate our desired output. Type the following code to print the name, iata_code, latitude and longitude of each of the airport features. The `%s` and `%f` notations are ways to format a string and number variables.

```

for f in layer.getFeatures():
    geom = f.geometry()
    print '%s, %s, %f, %f' % (f['name'], f['iata_code'],
                             geom.asPoint().y(), geom.asPoint().x())

```



12. You can see the output printed on the console. A more useful way to store the output would be in a file. You can type the following code to create a file and write the output there. Replace the file path with a path on your own system. Note that we add `\n` at the end of our line formatting. This is to add a newline after we add the data for each feature. You should also note the `unicode_line = line.encode('utf-8')` line. Since our layer contains some features with unicode characters, we can't simply write it to a text file. We encode the text using the UTF-8 encoding and then write to the text file.

```

output_file = open('c:/Users/Ujaval/Desktop/airports.txt', 'w')
for f in layer.getFeatures():
    geom = f.geometry()
    line = '%s, %s, %f, %f\n' % (f['name'], f['iata_code'],
                                geom.asPoint().y(), geom.asPoint().x())
    unicode_line = line.encode('utf-8')
    output_file.write(unicode_line)
output_file.close()

```



13. Puteți merge la locația fișierului de ieșire specificat și deschideți fișierul text. Veți vedea datele din fișierul shape de aeroporturi, pe care l-am extras folosind scriptarea Python.

airports.txt - Notepad

File Edit Format View Help

```
Sahnewal, LUH, 30.850360, 75.957072
Solapur, SSE, 17.625415, 75.933060
Birsamunda, IXR, 23.317725, 85.323597
Ahwaz, AWZ, 31.343159, 48.747107
Gwalior, GWL, 26.285488, 78.217219
Hodeidah Int'l, HOD, 14.755253, 42.971096
Devi Ahilyabai Holkar Int'l, IDR, 22.727749, 75.809292
Gandhinagar, ISK, 19.966021, 73.810567
Chandigarh Int'l, IXC, 30.670725, 76.801726
Aurangabad, IXU, 19.867297, 75.395843
Faisalabad Int'l, LYP, 31.362744, 72.987819
Omsk Tsentralny, OMS, 54.957648, 73.316360
Novosibirsk Tolmachev, OVB, 55.009585, 82.667152
Zaporozhye Int'l, OZH, 47.873264, 35.301873
Simpang Tiga, PKU, 0.464601, 101.446569
Rota Int'l, ROP, 14.171771, 145.243980
Surgut, SGC, 61.340167, 73.408496
Tiruchirappalli, TRZ, 10.760357, 78.708958
Turbat Int'l, TUK, 25.988795, 63.027933
Quetta Int'l, UET, 30.249043, 66.948731
Zahedan Int'l, ZAH, 29.475294, 60.900709
Abdul Rachman Saleh, MLG, -7.929980, 112.711419
Barnaul, BAX, 53.363385, 83.550453
Adampur, NULL, 31.432942, 75.758483
Bareilly, NULL, 28.421809, 79.452003
Dhamial, NULL, 33.561415, 73.032050
Cheongju Int'l, CJJ, 36.722023, 127.495916
Gwangju, KWJ, 35.140005, 126.810839
Daegu Int'l, TAE, 35.899928, 128.637538
Ulsan, USN, 35.592896, 129.355731
Radin Inten II, TKG, -5.242567, 105.176060
Allahabad, IXD, 25.443522, 81.731727
Chelyabinsk, CEK, 55.297792, 61.512259
Tainan, TNN, 22.950668, 120.209733
Taichung, RMQ, 24.266656, 120.630704
Rotterdam The Hague, RTM, 51.949130, 4.433844
Voronezh-Chertovitskoye, VOZ, 51.812617, 39.225450
Liverpool John Lennon, LPL, 53.336375, -2.858621
Vishakapatnam, VTZ, 17.727958, 83.223522
Sultan Hasanuddin Int'l, UPG, -5.058937, 119.545691
Vava'u Int'l, VAV, -18.586006, -173.968094
Newcastle Int'l, NCL, 55.037085, -1.710346
Goloson Int'l, LCE, 15.745160, -86.851469
```