

Getting Started with Python Programming

QGIS Tutorials and Tips



Author

Ujaval Gandhi

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

Translations by

Maruli Tua Manullang

Bakhtiar Arif

Memulai Pemrograman Phyton

QGIS memiliki sebuah antarmuka pemrograman yang ampuh sehingga kita bisa memperlebar fungsi inti dari software dan juga menulis skrip untuk otomatisasi pelaksanaan tugas. QGIS mendukung Bahasa skript Phyton yang terkenal. Walaupun anda seorang pemula, mempelajari sedikit phyton dan antarmuka Pemrogramman QGIS akan membuat anda jauh lebih produktif dalam pekerjaan anda. Tutorial ini mengasumsikan anda tanpa pengalaman pemrograman dan ditujukan untuk memberikan pengenalan pada scripting Python di QGIS (PyQGIS).

Tinjauan Tugas

Kita akan membuka sebuah layer poin yang merepresentasikan semua airport besan dan menggunakan skript python untuk menghasilkan sebuah file teks dengan nama airport, kode airport, garis lintang dan garis bujur untuk setiap airport pada layer.

Mendapatkan data

kita akan menggunakan dataset dari Natural Earth [Airports](#)

Unduh [Airports shapefile](#).

Sumber data [NATURALEARTH]

Prosedur

1. Dalam QGIS, akses Layers › Add Vector Layer . Jelajah ke file *ne_10m_airports.zip* yang sudah terunduh dan klik *ne_10m_airports.zip* . Pilih layer *ne_10m_airports.shp* dan klik OK.



2. Anda akan melihat layer *ne_10m_airports* terbuka di QGIS



3. Pilih tool Identify dan klik pada poin mana saja untuk memeriksa attribut yang tersedia. Anda akan melihat bahwa nama airport dan 3 digit kode didalamnya adalah attribut *name* dan *iata_code* secara berurutan



4. QGIS menyediakan konsol built-in dimana anda bisa mengetik command phyton dan mendapatkan hasilnya. Konsol ini merupakan cara yang sangat baik untuk belajar scripting dan juga melakukan proses data cepat. Buka Python Console dengan mengakses Plugins > Python Console.



5. Anda akan melihat sebuah panel baru di bawah kanvas QGIS. Anda akan melihat sebuah tanda seperti >>> di bawah dimana anda dapat mengetik command. Untuk interaksi dengan environment QGIS, anda harus menggunakan variabel *iface* . Untuk mengakses layer yang sedang aktif di QGIS, anda dapat mengetik sebagai berikut dan tekan *iface* . Command ini menarik referensi ke layer yang sedang dibuka dan menyimpannya sebagai variabel *layer*.

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



6. Ada function berguna yang disebut `dir()` di phyton yang menunjukkan anda semua metode yang tersedia untuk objek mana saja. Ini berguna ketika anda tidak yakin function mana yang tersedia untuk obyek tertentu. Jalankan command berikut dan lihat operasi apa yang bisa kita lakukan pada variabel `layer` .

```
dir(layer)
```



7. Anda akan melihat sebuah daftar panjang dari function yang tersedia. Untuk sekarang, kita akan menggunakan sebuah fungsi bernama `getFeatures()` dimana akan memberikan anda referensi untuk semua fitur pada sebuah layer. Pada kasus kita, setiap fitur akan menjadi sebuah poin yang merepresentasikan airport. Anda dapat mengetik command untuk mengiterasi melalui setiap fitur pada layer yang sedang aktif. Pastikan untuk menambah 2 spasi sebelum mengetik garis kedua.

```

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




seperti yang anda lihat di output, setiap garis mengandung sebuah referensi ke sebuah fitur di layer. Referensi ke fitur disimpan dalam variabel `f`. Kita dapat menggunakan variabel `f` untuk mengakses atribut pada setiap fitur. Ketik sebagai berikut untuk mencetak kode IATA untuk setiap fitur airport.

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



9. Jadi sekarang anda tahu bagaimana secara programatik mengakses atribut dari setiap fitur di layer. Sekarang, mari kita lihat bagaimana mengakses koordinat fitur. Koordinat dari sebuah fitur vektor dapat diakses dengan memanggil function **geometry()**. Function ini memberikan sebuah obyek geometri yang dapat kita simpan pada variable **geom**. Anda dapat menjalankan function **asPoint()** pada obyek geometri untuk mendapatkan koordinat x dan y poin tersebut. Jika fitur anda adalah sebuah garis atau sebuah poligon, anda dapat menggunakan fungsi **asPolyline()** atau **asPolygon()**. Ketik Kode berikut pada prompt dan tekan Enter untuk melihat koordinat x dan y dari setiap fitur.

```

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

```



10. Bagaimana jika kita hanya ingin mendapatkan koordinat x dari fitur ? Anda dapat memanggil fungsi $x()$ pada obyek poin dan mendapatkan koordinat x nya.

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



11. Sekarang kita mempunyai semua kepingan yang akan kita satukan untuk menghasilkan output yang kita inginkan. Ketik kode berikut untuk mencetak name, iata_code, latitude dan longitude dari setiap fitur airport. Notasi %s dan %f adalah cara untuk memformat sebuah variabel string dan numerik

```

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. Anda dapat melihat output tercetak pada konsol. Sebuah cara yang lebih berguna untuk menyimpan output adalah dengan sebuah file. Anda dapat mengetik kode berikut untuk membuat sebuah file dan menulis outputnya di sana, Ganti path atau alamat file dengan sebuah path dalam sistem anda sendiri. Catat bahwa kita menambahkan `\n` di akhir baris. Ini untuk menambah sebuah garis baru setelah kita menambahkan data untuk setiap fitur. Anda seharusnya juga memperhatikan garis `unicode_line = line.encode('utf-8')`. Karena layer kita mengandung sejumlah fitur dengan karakter unicode, kita tidak bisa begitu saja menulisnya ke dalam file teks. Kita encode teks menggunakan UTF-8 encoding dan kemudian menulis file teks tersebut.

```

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. Anda dapat mengakses lokasi file output yang sudah ditentukan dan membuka file teks. Anda akan melihat data dari shapefile airport yang kita ekstraksi menggunakan skripting 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