

# Getting Started with Python Programming

## QGIS Tutorials and Tips



Author

Ujaval Gandhi

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

Translations by

SongHyun Choi

□ □ □ □ □ □ □ □

QGIS는 오픈 소스 GIS 소프트웨어로, GIS를 사용하여 데이터를 시각화하고 분석하는 데 사용됩니다. QGIS는 다양한 데이터 소스에서 데이터를 가져올 수 있으며, 데이터를 처리하고 분석할 수 있습니다. QGIS는 GIS를 사용하여 데이터를 시각화하고 분석하는 데 사용됩니다. QGIS (PyQGIS)는 오픈 소스 GIS 소프트웨어입니다.

□ □ □ □

QGIS는 오픈 소스 GIS 소프트웨어로, GIS를 사용하여 데이터를 시각화하고 분석하는 데 사용됩니다. QGIS는 다양한 데이터 소스에서 데이터를 가져올 수 있으며, 데이터를 처리하고 분석할 수 있습니다. QGIS는 GIS를 사용하여 데이터를 시각화하고 분석하는 데 사용됩니다. QGIS, QGIS, QGIS는 오픈 소스 GIS 소프트웨어입니다.

□ □ □ □

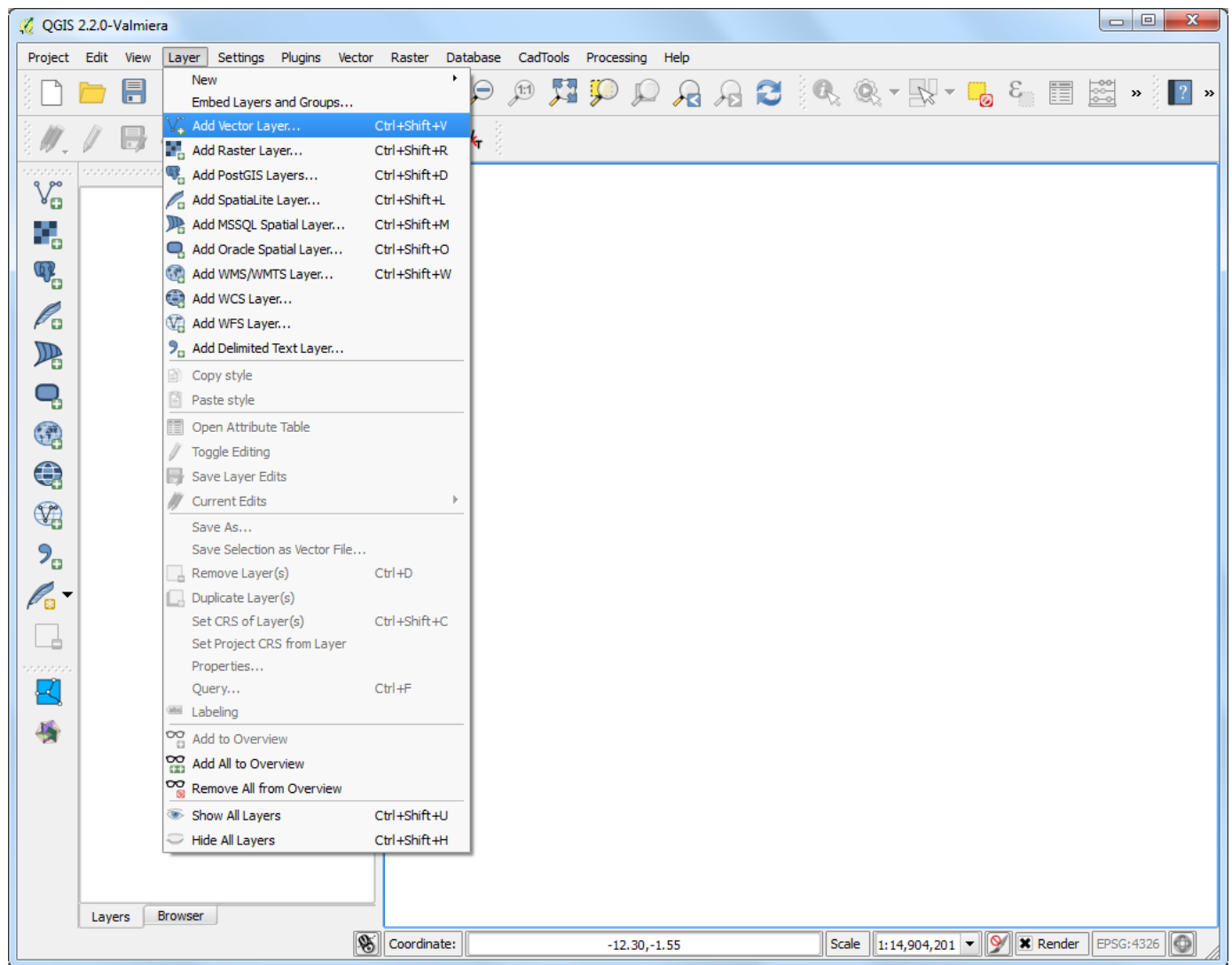
Natural Earth의 Airports 데이터를 다운로드합니다.

`Airports shapefile <[http://www.naturalearthdata.com/http://www.naturalearthdata.com/download/10m/cultural/ne\\_10m\\_airports.zip](http://www.naturalearthdata.com/http://www.naturalearthdata.com/download/10m/cultural/ne_10m_airports.zip)>`\_를 다운로드합니다.

□ □ □ □: [NATURALEARTH]

□ □

1. QGIS에서 데이터 파일을 --> 데이터 파일 --> 데이터 파일 :menuselection: `Layers --> Add Vector Layer`로 불러옵니다. 데이터 파일 `ne\_10m\_airports.zip`을 불러옵니다. :guilabel: `Open`을 클릭합니다. `ne\_10m\_airports.shp`을 불러옵니다. :guilabel: `OK`를 클릭합니다.



2. QGIS `ne\_10m\_airports` 。



3. Identify the following fields in the 'ne\_10m\_airports' layer. The fields are: 'name', 'iata\_code', and 'icao\_code'. The 'name' field is the name of the airport, the 'iata\_code' is the International Air Transport Association (IATA) code, and the 'icao\_code' is the International Civil Aviation Organization (ICAO) code.







6. `dir()` returns a list of attributes and methods of the active layer. `layer` is the name of the active layer.

```
dir(layer)
```



7. `getFeatures()` returns a list of features. Each feature is a dictionary with keys for the feature's attributes and values for the attribute data. The first key is always 'id', which is the feature's unique identifier. The other keys are the names of the attributes. The values are the corresponding attribute values. For example, if a feature has attributes 'name' and 'population', the dictionary would look like: `{ 'id': 1, 'name': 'New York', 'population': 20000000 }`.

```

for f in layer.getFeatures():
    print f

```





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

```

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

```



```

    geom = f.geometry()
    print geom.asPoint()

```

```

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

```



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

```

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

```



11. 00 000 000 000 00 00 00 000 0 0 00 00 000 000 000000. 0 00 000  
 00, 0000, 00 000 000 000000 000 000 000000. ``%s`` 0 ``%f`` 000 000  
 000000 00000 000000.

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





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
```