

# Getting Started with Python Programming

## QGIS Tutorials and Tips



Author

Ujaval Gandhi

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

Translations by

Dick Groskamp

# Beginnen met programmeren in Python

QGIS heeft een krachtige interface voor programmeren die u in staat stelt de bron-functionaliteit van de software uit te breiden en ook om scripts te schrijven om uw taken te automatiseren. QGIS ondersteunt de populaire programmeertaal voor scripten Python. Zelfs als u een beginner bent, leer een klein beetje Python en de interface voor programmeren van QGIS zal u in staat stellen veel productiever te werk te gaan. Deze handleiding gaat er van uit dat u geen eerdere kennis van programmeren heeft en is bedoeld als introductie voor het scripten met Python in QGIS (PyQGIS).

## Overzicht van de taak

We zullen een vector puntlaag laden die alle belangrijke vliegvelden weergeeft en scripten in Python gebruiken om een tekstbestand te maken met de naam van het vliegveld, de code van het vliegveld en latitude en longitude voor elk vliegveld op de laag.

## De gegevens ophalen

We zullen de gegevensset [Airports](#) van Natural Earth gebruiken.

Download het [shapefile Airports](#).

Gegevensbron [NATURALEARTH]

## Procedure

1. Ga in QGIS naar Kaartlagen › Laag toevoegen › Vectorlaag toevoegen. Blader naar het gedownloade bestand ***ne\_10m\_airports.zip*** en klik op Openen. Selecteer de laag ***ne\_10m\_airports.shp*** en klik op OK.



2. U zult de laag *ne\_10m\_airports* zien geladen in QGIS.



3. Selecteer het gereedschap Objecten identificeren en klik op één van de punten om daarvan de beschikbare attributen te bekijken. U zult zien dat de naam van het vliegveld en de 3-cijferige code ervan zijn opgenomen in de respectievelijke attributen *name* en *iata\_code*.



4. QGIS verschaft een ingebouwde console waar u opdrachten voor Python kunt typen en de resultaten verkrijgt. Deze console is een fantastische manier om te leren scripten en ook om snel gegevens te verwerken. Open de Python Console door te gaan naar Plug-ins › Python Console.



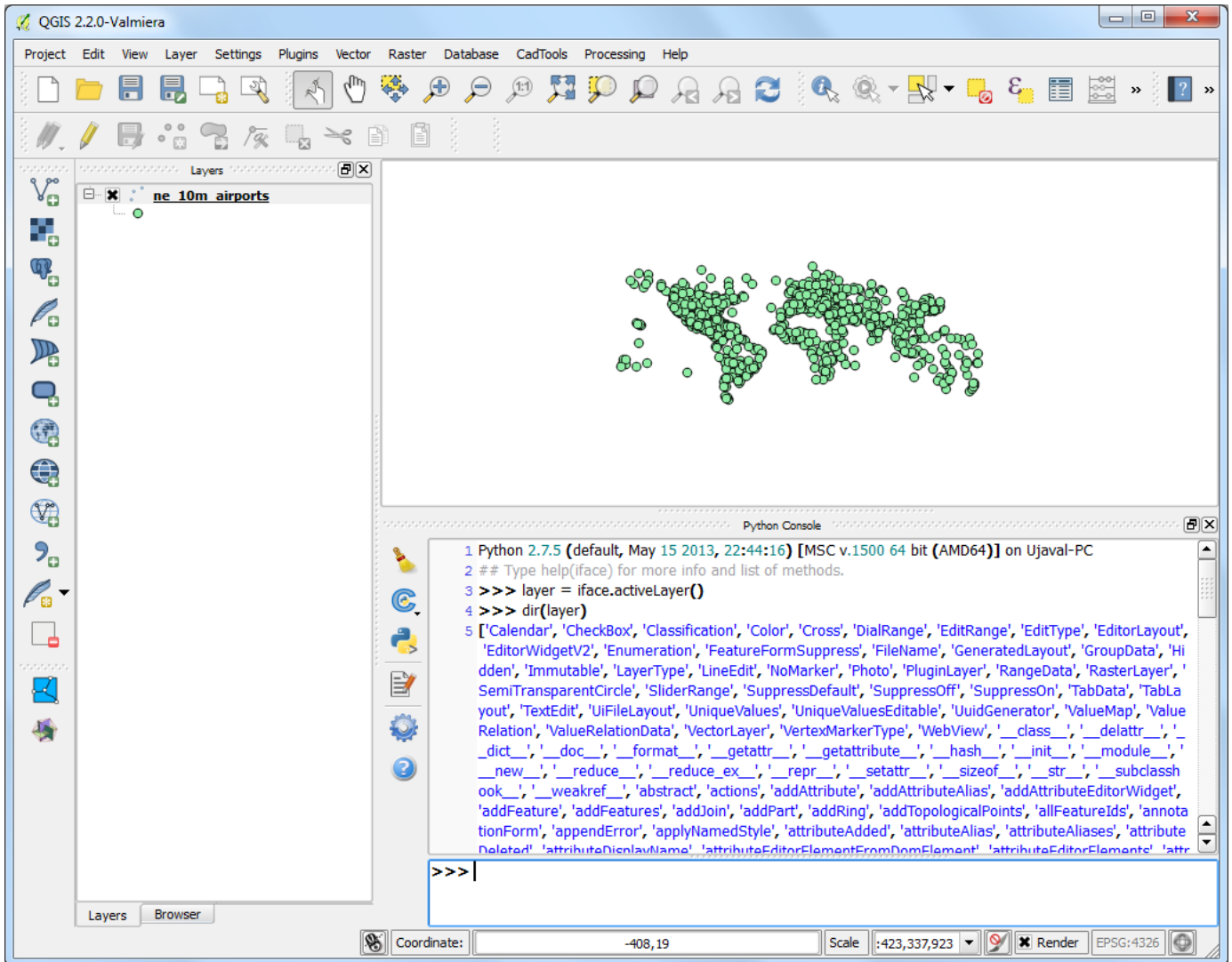
5. U zult zien dat een nieuw paneel wordt geopend aan de onderzijde van het kaartvenster van QGIS. U zult een prompt als `>>>` zien aan de onderzijde waar u opdrachten kunt typen. Voor de interactie met de omgeving van QGIS, moeten we de variabele `iface` gebruiken. U kunt het volgende typen en op Enter drukken om toegang te krijgen tot de huidige actieve laag in QGIS. Deze opdracht haalt de verwijzing naar de huidige geladen laag op en slaat die op in de variabele `layer`.

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



6. Er bestaat een handige functie, genaamd `dir()`, in Python die u alle beschikbare methoden voor een object laat zien. Dat is handig als u niet precies weet welke functies beschikbaar zijn voor het object. Voer de volgende opdracht uit om te zien welke bewerkingen we kunnen doen met de variabele `layer`.

```
dir(layer)
```



7. U zult ene lange lijst met beschikbare functies zien. Voor nu zullen we de functie, genaamd `getFeatures()`, gebruiken wat u een verwijzing zal geven naar alle objecten van een laag. In ons geval zal elk object een punt zijn dat een vliegveld weergeeft. U kunt de volgende opdracht typen om door elk van de objecten op de huidige laag te gaan. Zorg er voor dat u 2 spaties toevoegt vóórdat u de tweede regel typt.

```

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





8. Zoals u in de uitvoer zult zien, bevat elke regel een verwijzing naar een object op de laag. De verwijzing naar het object wordt opgeslagen in de variabele `f`. We kunnen de variabele `f` gebruiken om toegang te krijgen tot de attributen van elk object. Typ het volgende om de `name` en `iata_code` voor elk object vliegveld af te drukken.

```

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

```



9. U weet dus nu hoe u programmatisch toegang kunt krijgen tot de attributen van elk object op een laag. laten we dan nu eens zien hoe we toegang krijgen tot de coördinaten van het object. Toegang tot de coördinaten van het vectorobject kan worden verkregen door de functie **geometry()** aan te roepen. Deze functie geeft een object geometrie terug dat we kunnen opslaan in de variabele **geom**. U kunt de functie **asPoint()** op het object geometrie uitvoeren om de X- en Y-coördinaten van het punt te verkrijgen. Als uw object een lijn of polygoon is, kunt u de functies **asPolyline()** of **asPolygon()** gebruiken. Typ de volgende code bij de prompt en druk op Enter om de X- en Y-coördinaten van elk object te bekijken.

```

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



10. Wat als we alleen het  $x$ -coördinaat van het object wilden? U kunt de functie `x()` aanroepen voor het object punt en daarvan het  $X$ -coördinaat verkrijgen.

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



11. Nu hebben we alle stukjes die we aan elkaar kunnen naaien om de door ons gewenste uitvoer te verkrijgen. Typ de volgende code om de naam, iata\_code, latitude en longitude van alle objecten vliegveld af te drukken. De notaties `%s` en `%f` zijn manieren om de string en number variabelen op te maken.

```

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. U kunt de uitvoer zien worden afgedrukt op de console. Een meer handige manier om de uitvoer op te slaan zou een bestand zijn. U kunt de volgende code typen om een bestand te maken en de uitvoer daar naartoe weg te schrijven. vervang het bestandspad door een bestandspad op uw eigen systeem. Onthoud dat we `\n` toevoegen aan het einde van de opmaak van onze regel. Dat is om ene nieuwe regel toe te voegen na de gegevens voor elke object. U zou ook de regel `unicode_line = line.encode('utf-8')` moeten onthouden. Omdat onze laag enkele objecten bevat met tekens in Unicode, kunnen we het niet eenvoudigweg wegschrijven naar een tekstbestand. We coderen de tekst met behulp van de codering UTF-8 en schrijven dan weg naar het tekstbestand.

```

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. U kunt naar de locatie van het uitvoerbestand gaan ie u heeft gespecificeerd en het tekstbestand openen. U zult de gegevens uit het shapefile Airports zien die we hebben uitgenomen door middle van scripten met 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