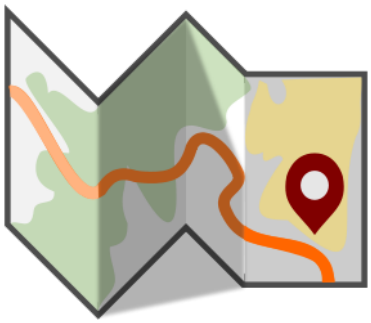
Getting Started with Python Programming

QGIS Tutorials and Tips



Author

Ujaval Gandhi

http://google.com/+UjavalGandhi

Translations by

Pino Nicolosi a.k.a Rattus

This work is licensed under a Creative Commons Attribution 4.0 International License.

Primi passi con la programmazione Python

QGIS fornisce una potente interfaccia di programmazione che vi permette di estendere le funzionalità di base del software e di scrivere codice per automatizzare i vostri lavori. QGIS supporta il celebre linguaggio di scripting Python. Anche se siete dei principianti, imparare un po' di Python e di programmazione delle interfacce QGIS, vi permetterà di essere molto più produttivi nel vostro lavoro. Questa esercitazione non presume competenze pregresse nella programmazione e si propone di offrire un'introduzione allo scripting python in QGIS (PyQGIS).

Descrizione del compito

Caricheremo un vettore puntuale che contiene tutti i principali areoporti e useremo lo scripting in python per creare un file di testo con il nome dell'areoporto, il codice dell'aeroporto, la latitudine e la longitudine di ciascuno degli aeroporti presenti nel layer.

Ottenere i dati necessari

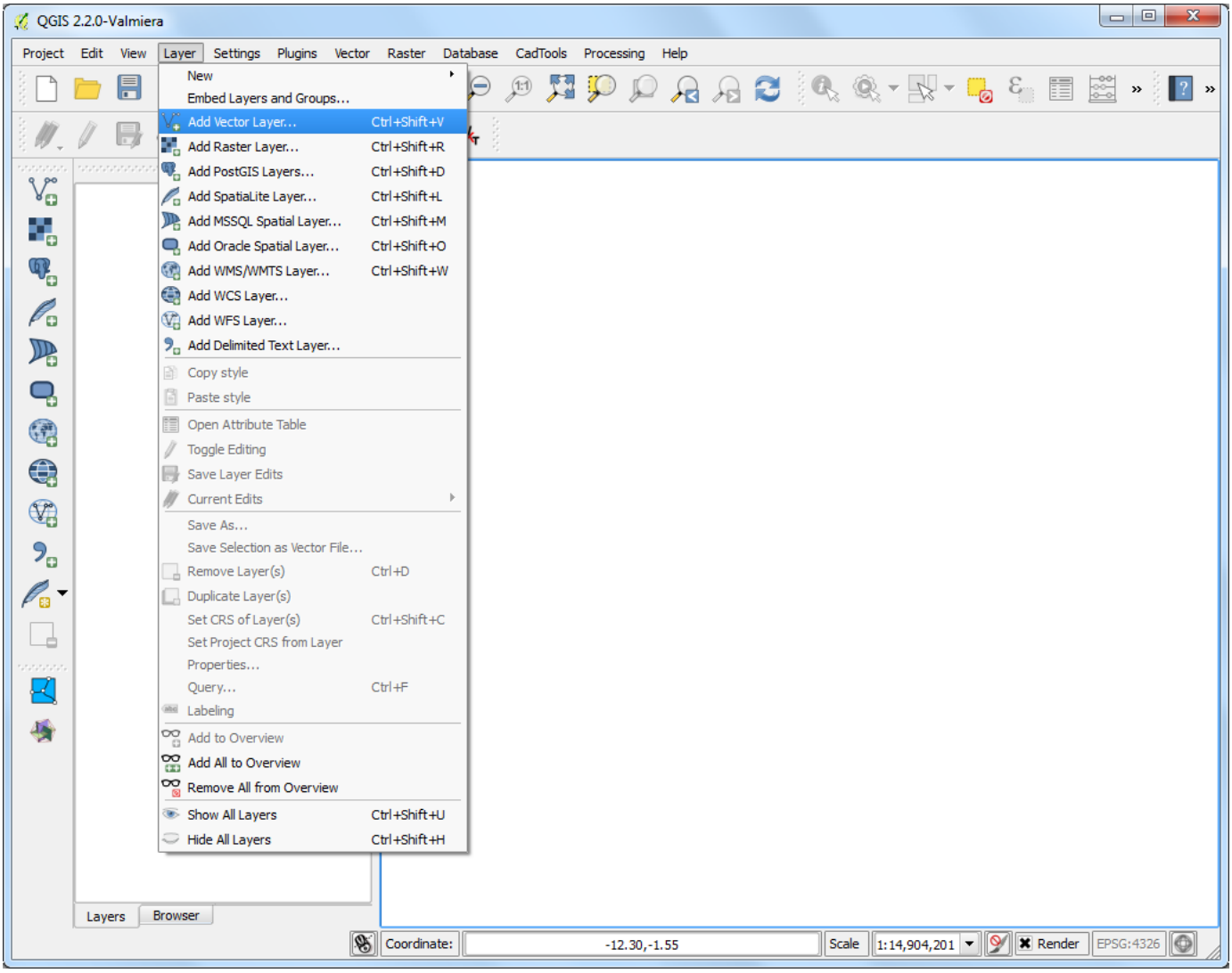
Useremo il dataset Airports tratto da Natural Earth.

Scaricate lo Airports shapefile.

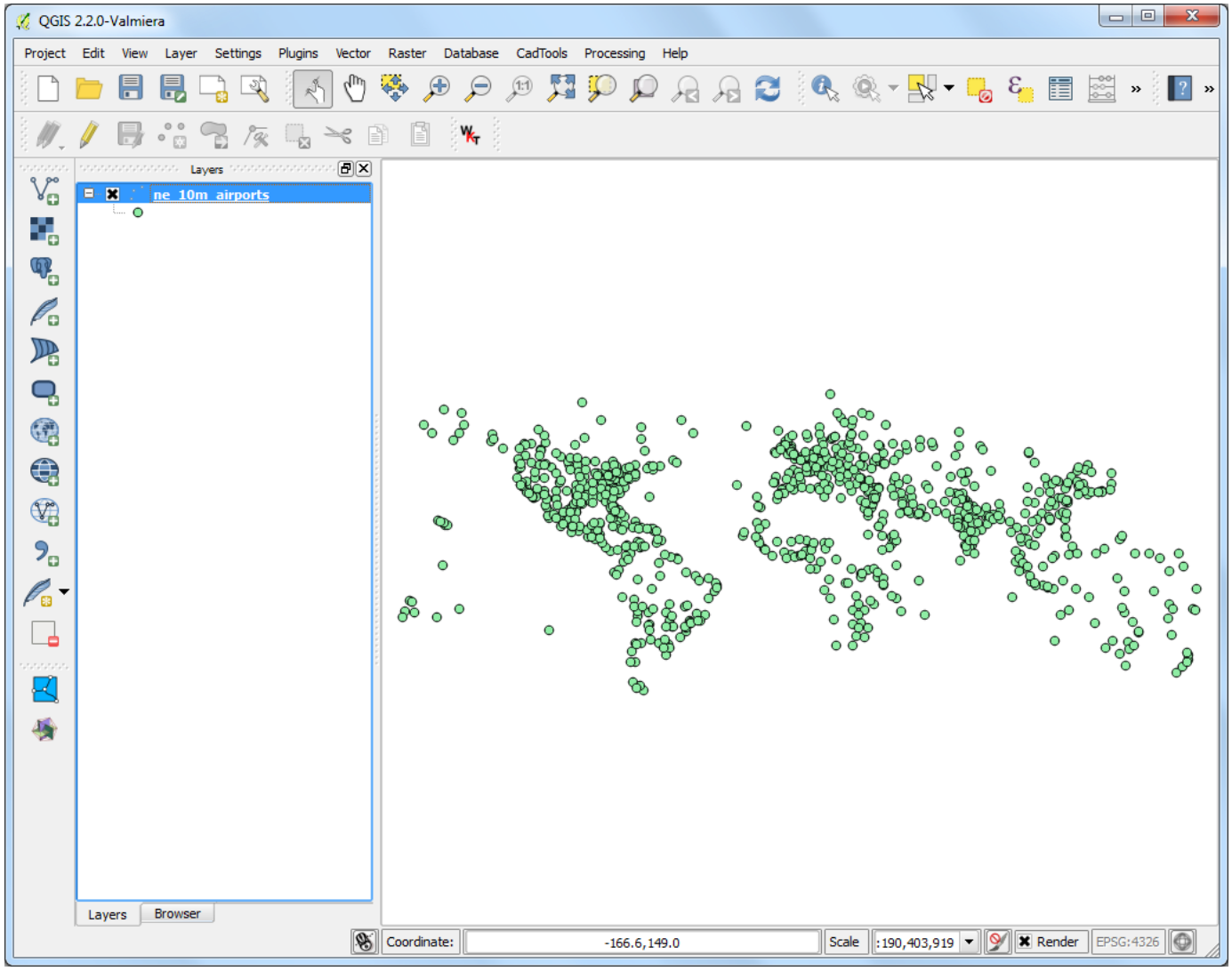
Fonte Dati [NATURALEARTH]

Procedimento

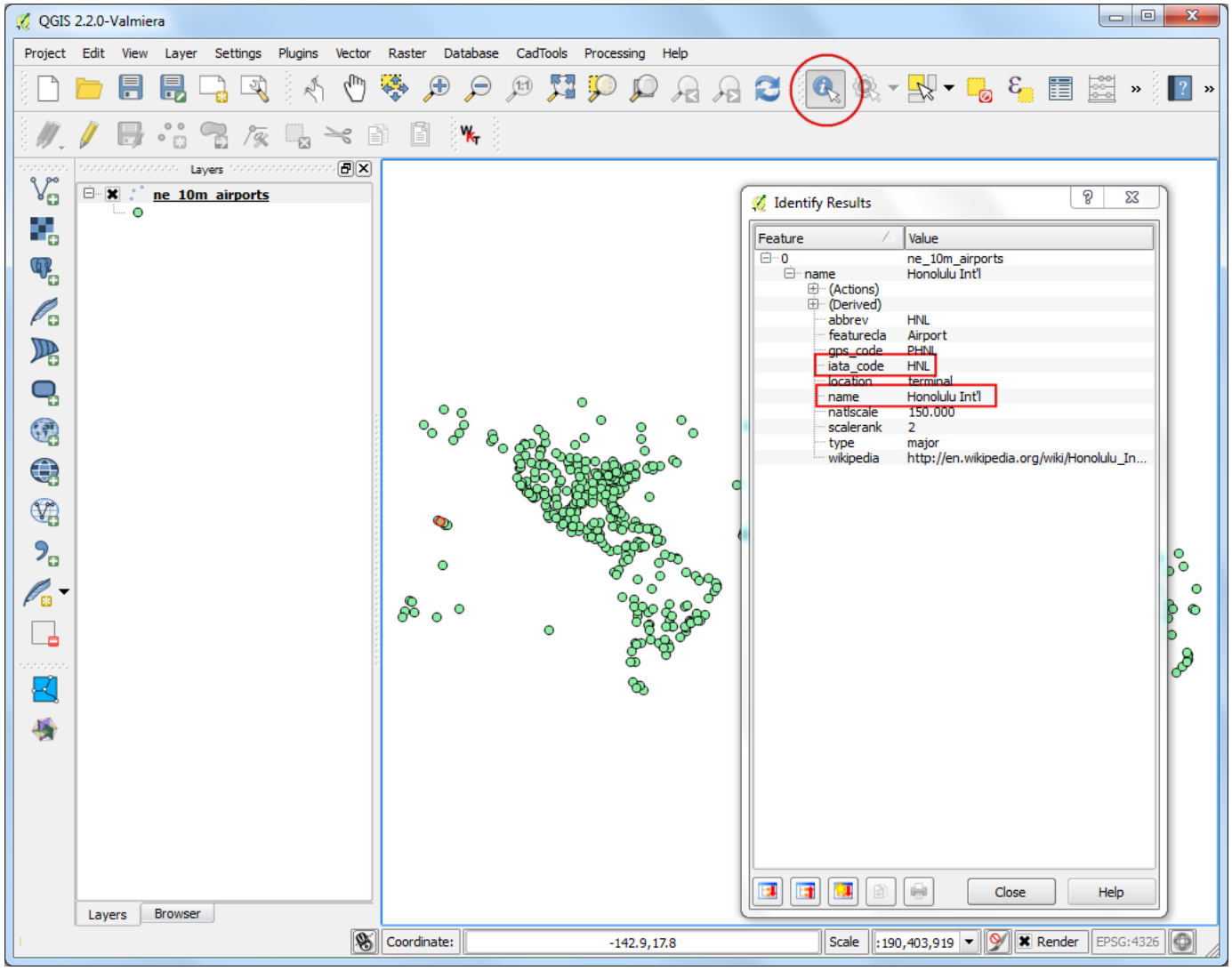
1. Aprite QGIS andate su Layers ‣ Aggiungi vettore. Individuate il file scaricato ***ne_10m_airports.zip*** e fate click su Apri. Selezionate il layer ***ne_10m_airports.shp*** e fate click su OK.



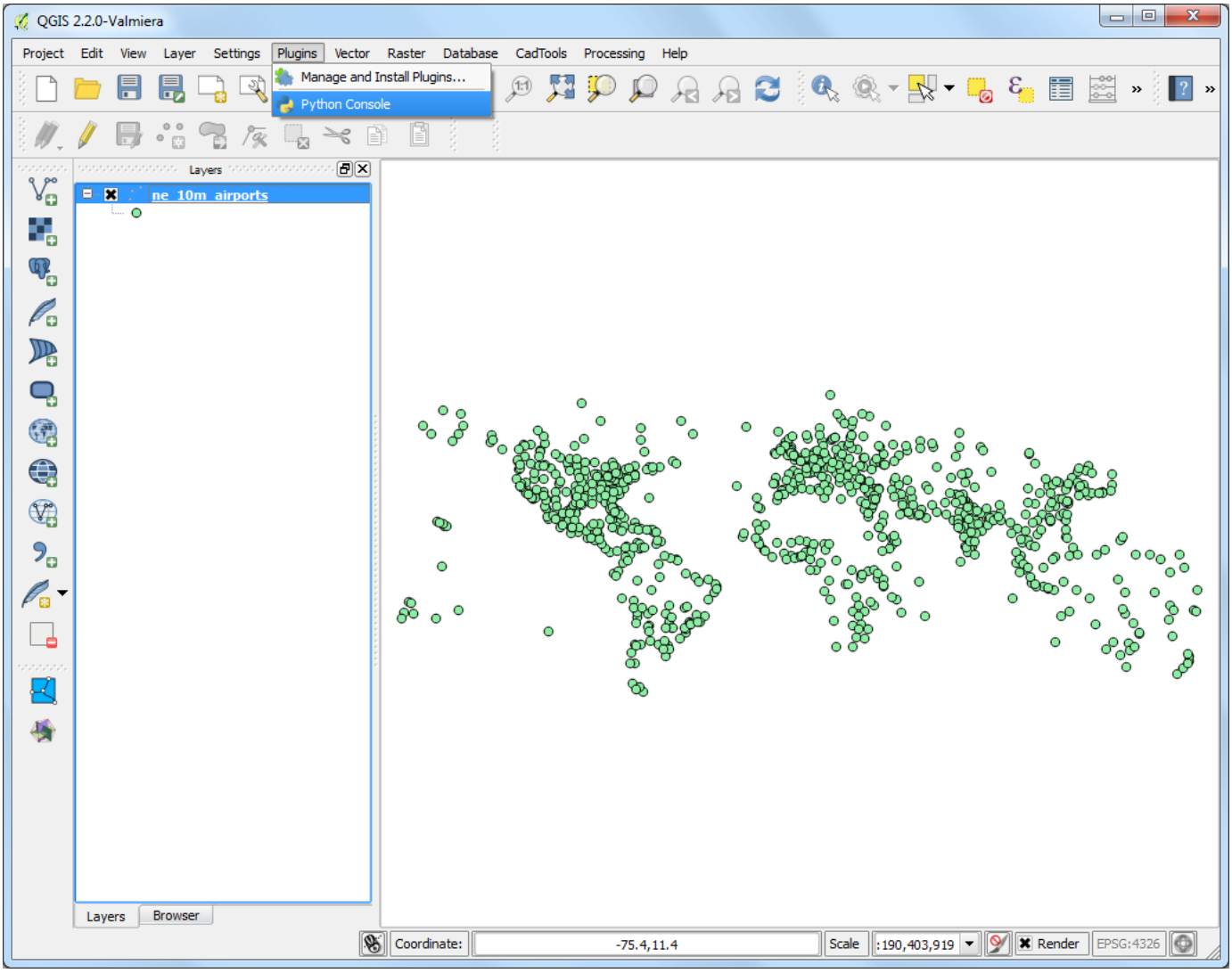
2. Ora vedrete il layer ***ne_10m_airports*** caricato in QGIS.



3. Selezionate lo strumento Identifica elemento e fate click su uno dei punti per esaminare gli attributi che abbiamo a disposizione. Vedrete che il nome dell'aeroporto e il suo codice di 3 lettere sono contenuti, rispettivamente, negli attributi ***`name`*** e ***`iata_code`***.

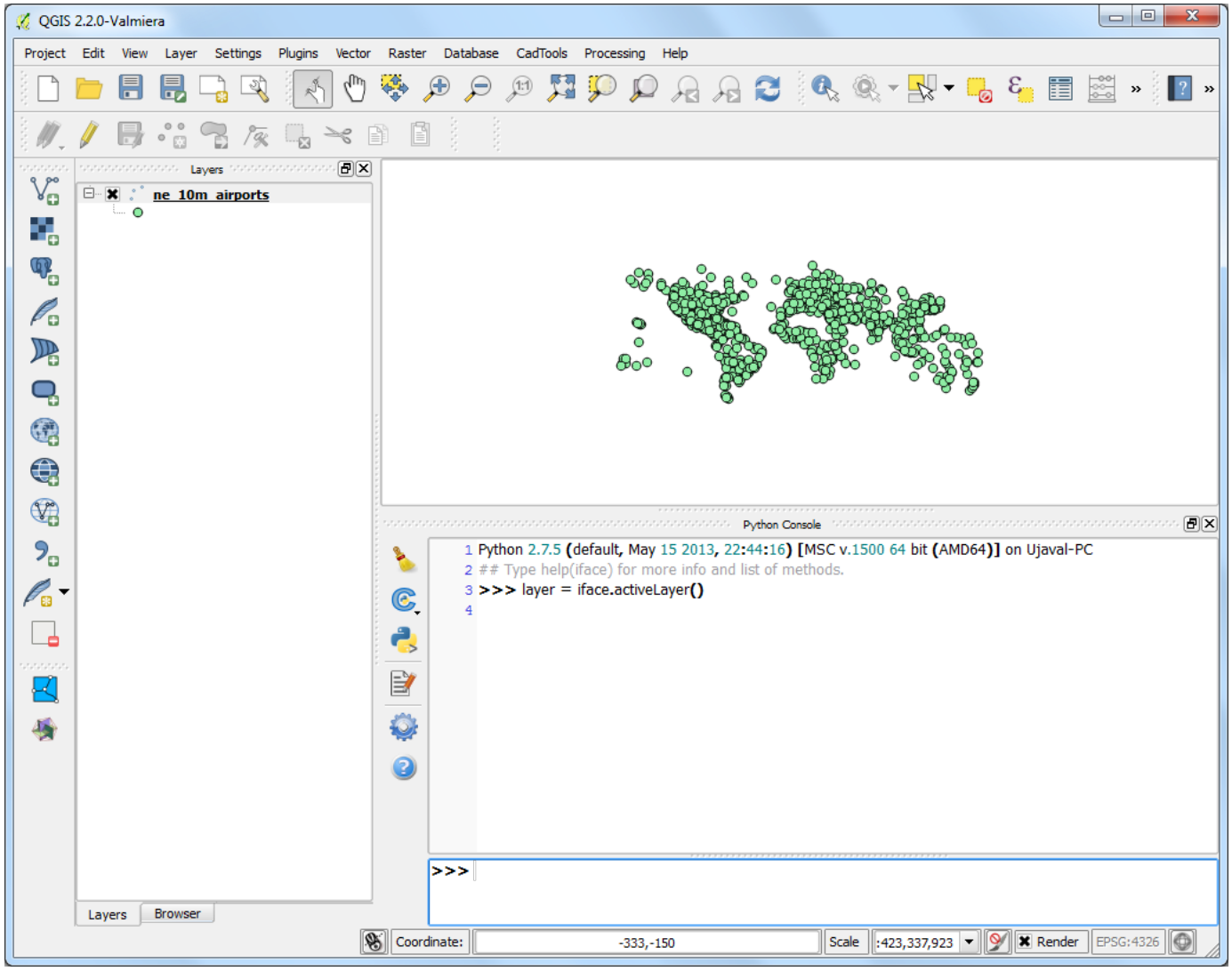


4. QGIS fornisce una console residente dove potete scrivere comandi in Python e verificare il risultato. Questa console è una strada privilegiata per apprendere a scrivere gli script e per effettuare delle rapide elaborazioni di dati. Apriamo la Python Console andando su Plugins ‣ Console Python.



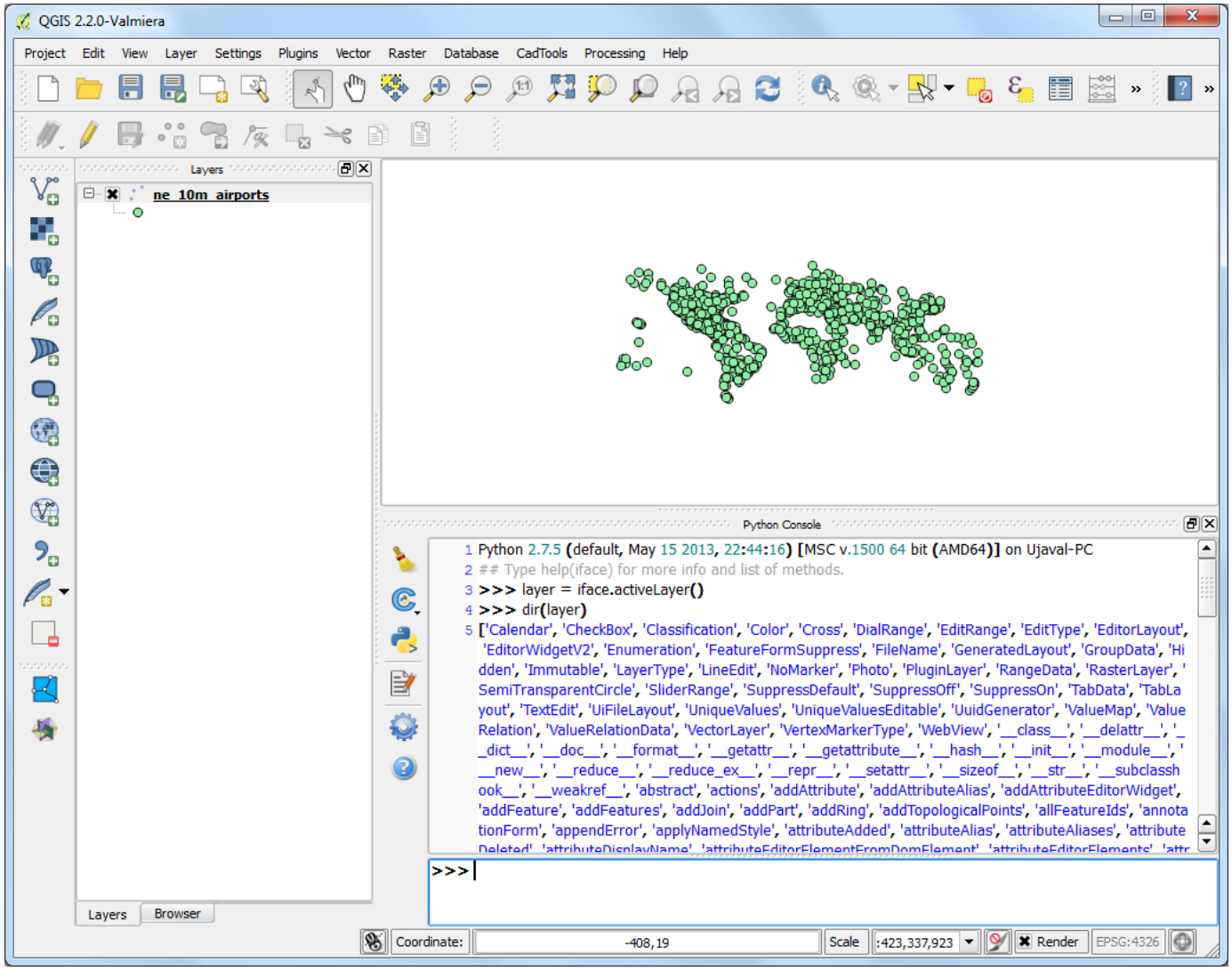
5. Vedrete comparire un nuovo pannello che si aprirà alla base della finestra principale di QGIS. Noterete un prompt fatto così **>>>** alla base del pannello, dove si digitano i comandi. Per interagire con l'ambiente QGIS dobbiamo usare la variabile ***iface***. Per avere accesso al layer attualmente attivo in QGIS, dovete digitare la riga di codice che segue e quindi premere il tasto Enter. Questo comando porta il controllo sul layer corrente e lo memorizza nella varabile ``layer`.

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



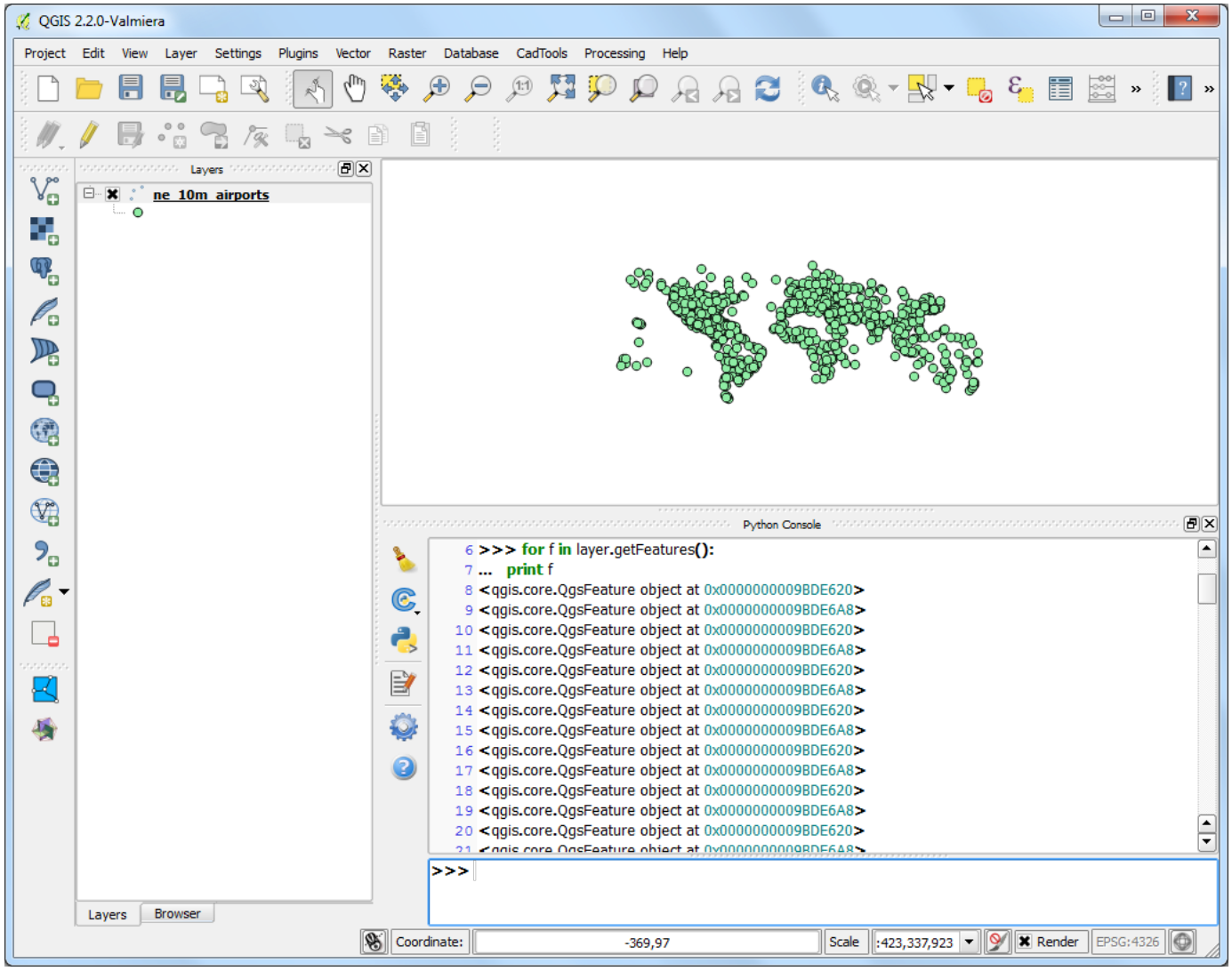
6. Esiste in Python una comoda funzione chiamata ***dir()*** che vi mostra tutti i metodi a disposizione per ciascun oggetto. Questo è utile quando non avete chiaro quali funzioni sono a disposizione per un dato oggetto. Lanciate il comando che segue per vedere quali operazioni possiamo fare sulla variabile ***layer*** .

```
dir(layer)
```



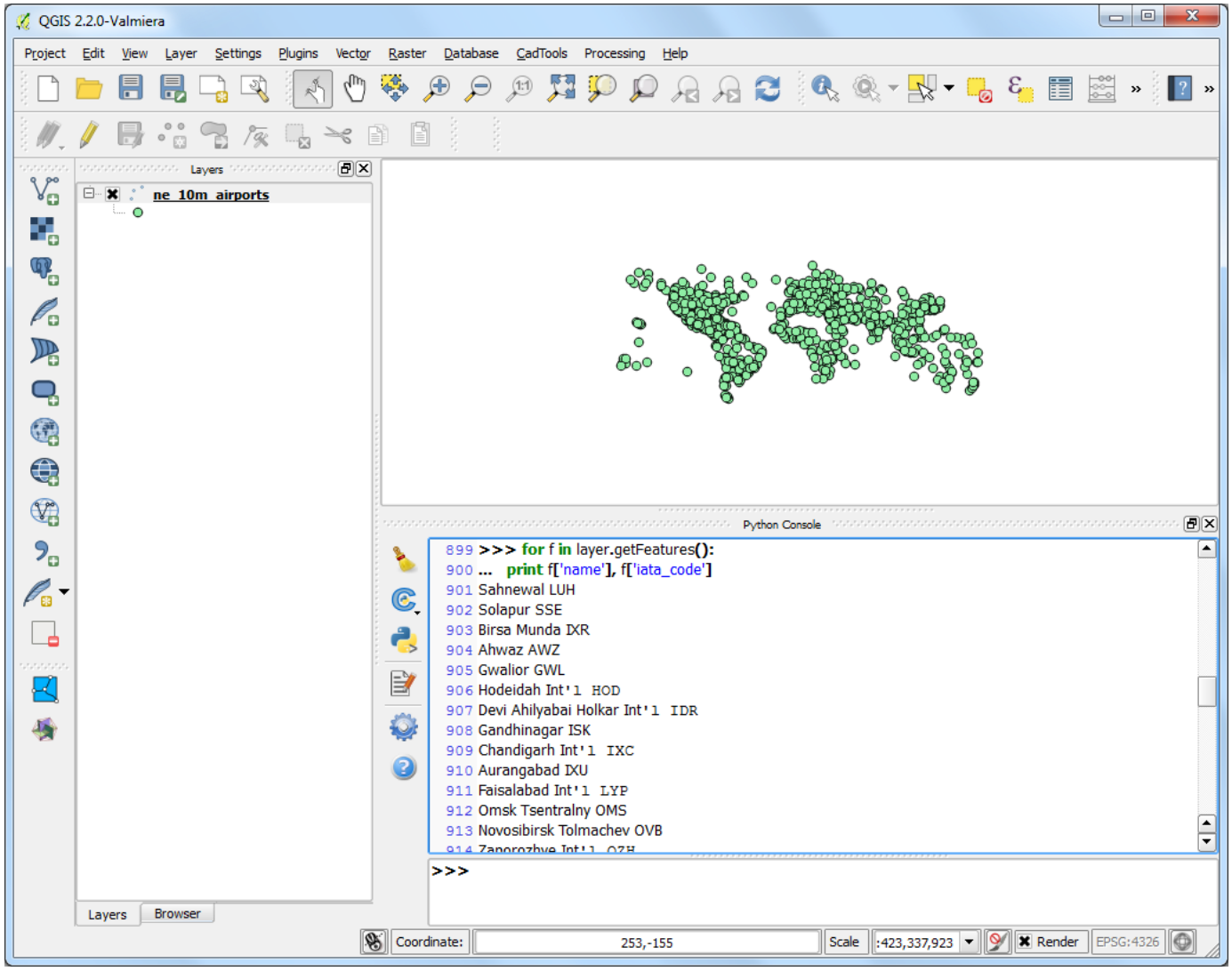
7. Vedrete una lunga lista di funzioni disponibili. Per ora, useremo una funzione chiamata ***getFeatures()*** che vi darà il controllo su tutte le geometrie del layer. Nel nostro caso, ciscun elemento sarà un punto che rappresenta un areoporto. Potete digitare il comando che segue per realizzare un'iterazione di ciascuna delle geometrie presenti nel layer corrente. Assicuratevi di aver battuto 2 spazi prima di digitare la seconda linea di codice.

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



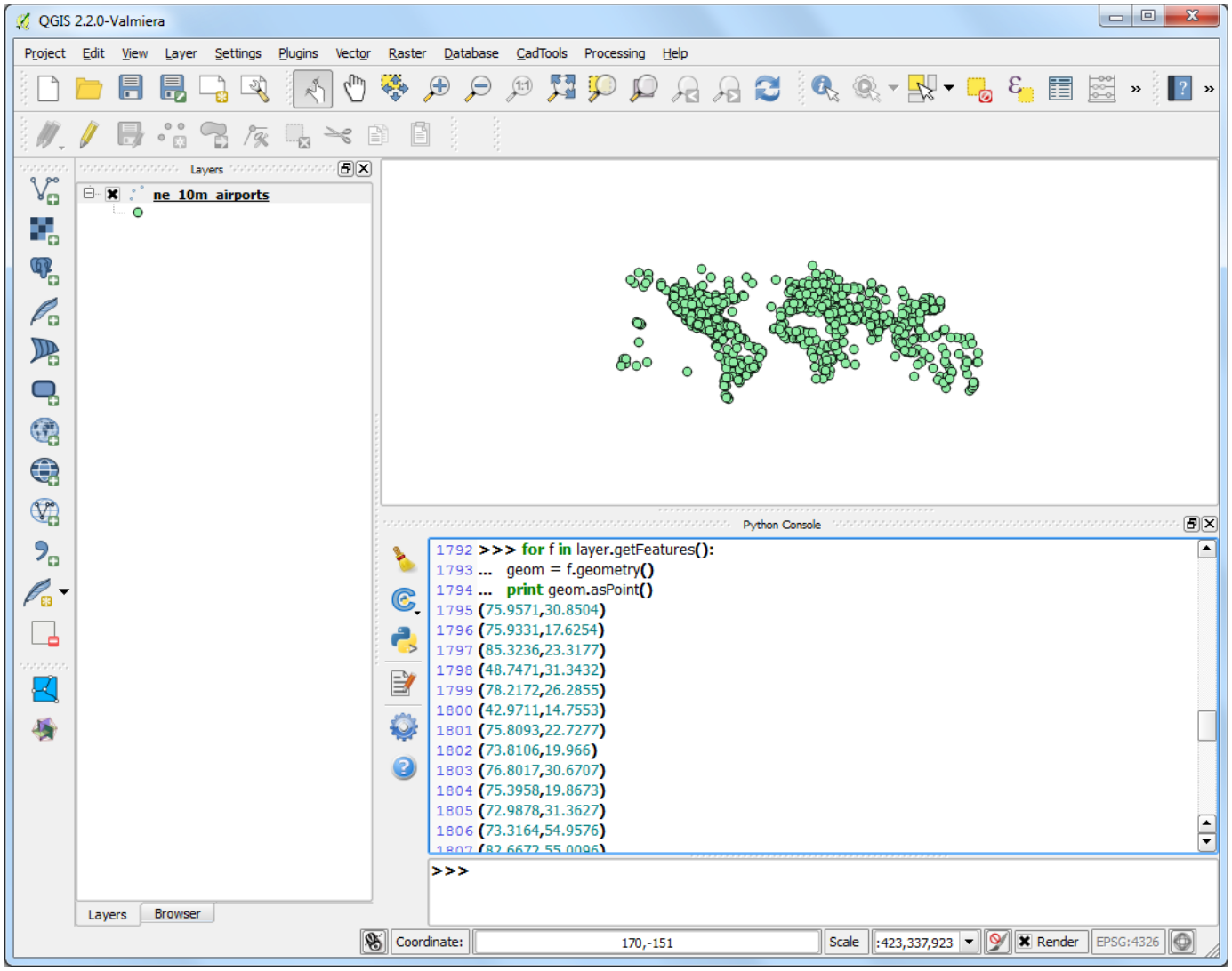
8. Come potete vedere nell'output, ciascuna linea contiene un riferimento ad una geometria presente all'interno del layer. Il riferimento all'elemento è contenuto nella variabile ***f***. Possiamo quindi usare la variabile ***f*** per accedere agli attributi di ciascun elemento. Digitate il comando seguente per stampare il ***name*** e lo ***iata_code*** per ciascuna geometria rappresentata da un areoporto.

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



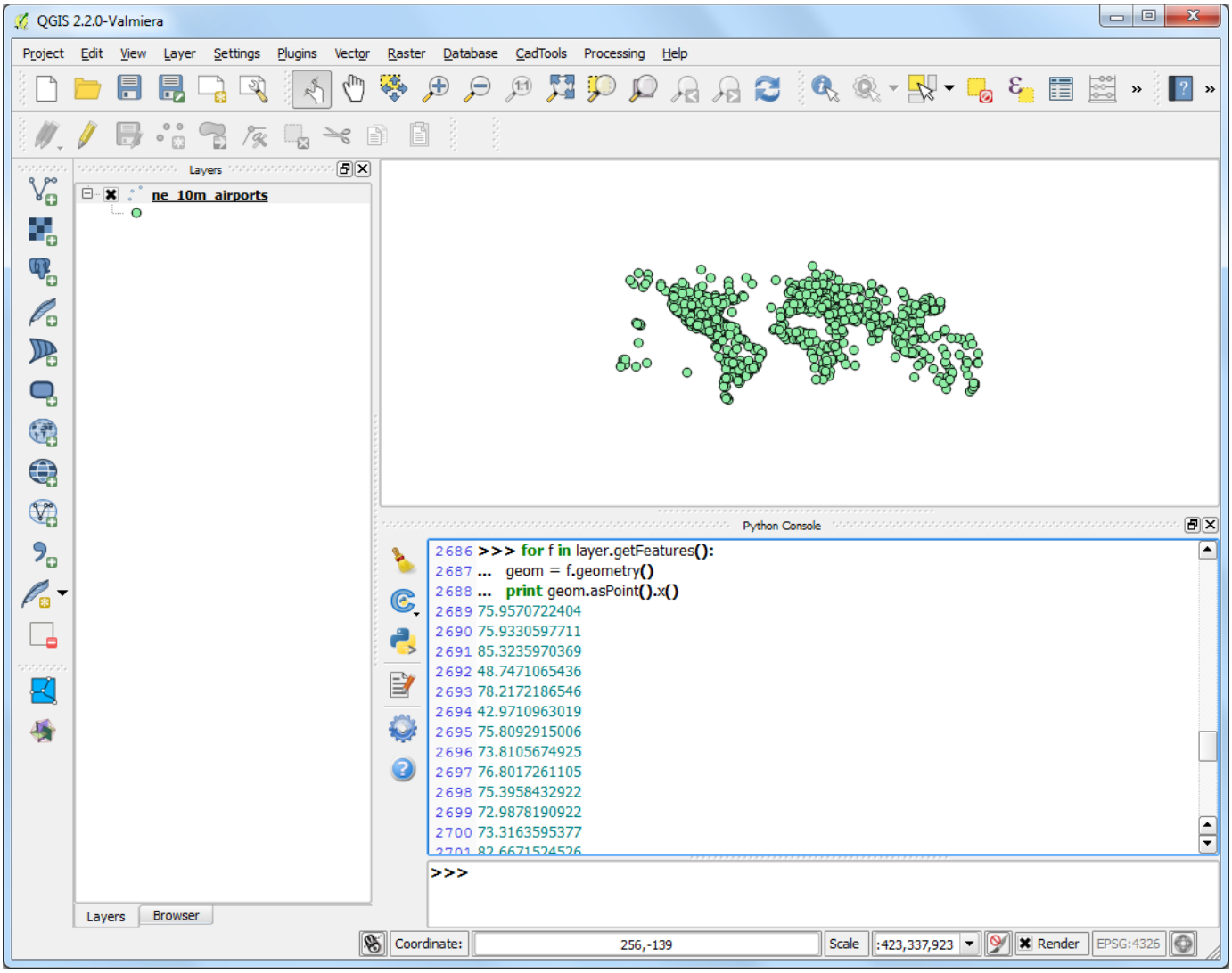
9. Così, adesso sapete come accedere agli attributi di ciascuna geometria di un layer. Adesso vediamo come possiamo accedere alle coordinate della geometria. Le coordinate di un elemento vettoriale possono essere ottenute con una chiamata alla funzione ***geometry()*** . Questa funzione restituisce un oggetto geometrico che noi possiamo depositare nella variabile ***geom***. Potete lanciare la funzione ***asPoint()*** per quegli oggetti che restituiscono le coordinate X e Y di un punto. Nel caso in cui la geometria fosse una linea o un poligono, dovreste usare, rispettivamente, le funzioni ***asPolyline()*** e ***asPolygon()***. Scrivete il codice seguente al prompt e premete Enter per vedere le coordinate X e Y di ciascun elemento.

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



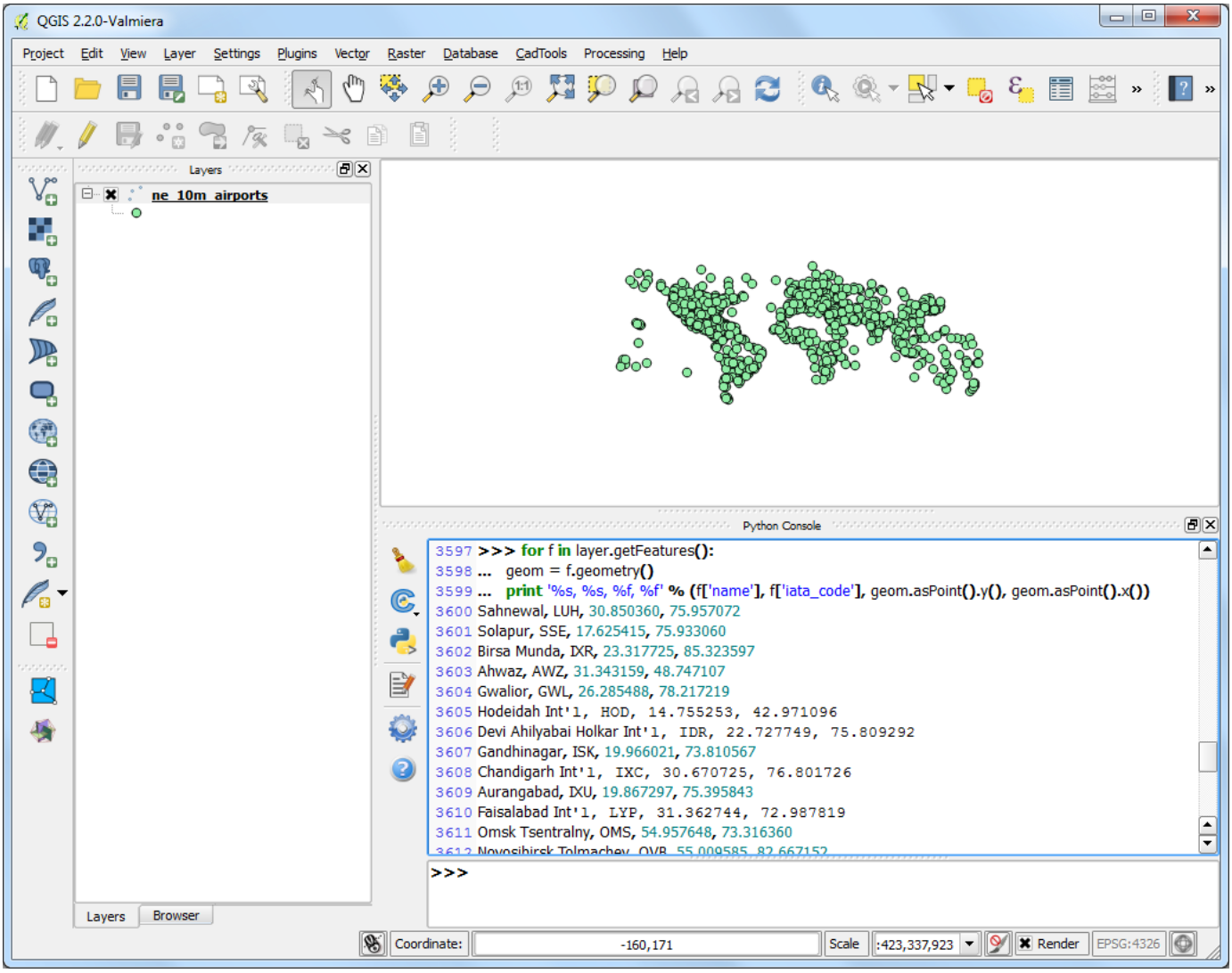
10. Che fare nel caso in cui desiderassimo conoscere la sola coordinata ***x*** del punto? Possiamo chiamare la funzione ***x()*** sull'oggetto punto e ottenere la sua coordinata X.

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



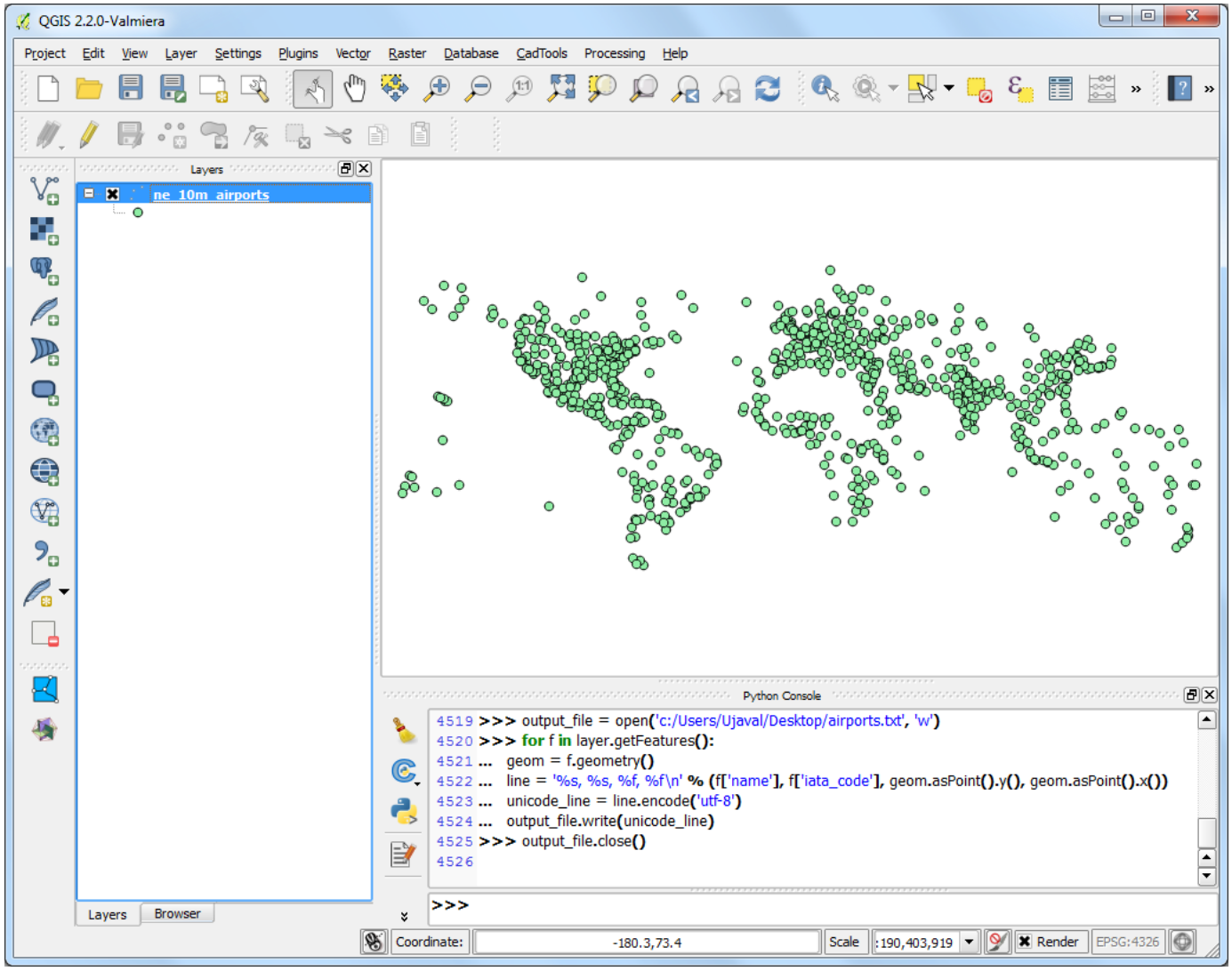
11. Adesso abbiamo tutti gli elementi che occorre cucire insieme per produrre l'output desiderato. Digitate il codice seguente per stampare il nome, lo iata_code la latitudine e la longitudine di ciascuna delle geometrie areoporto presenti in questo layer. I simboli ***%s*** e ***%f*** sono metodi per formattare variabili stringa e variabili numeriche.

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

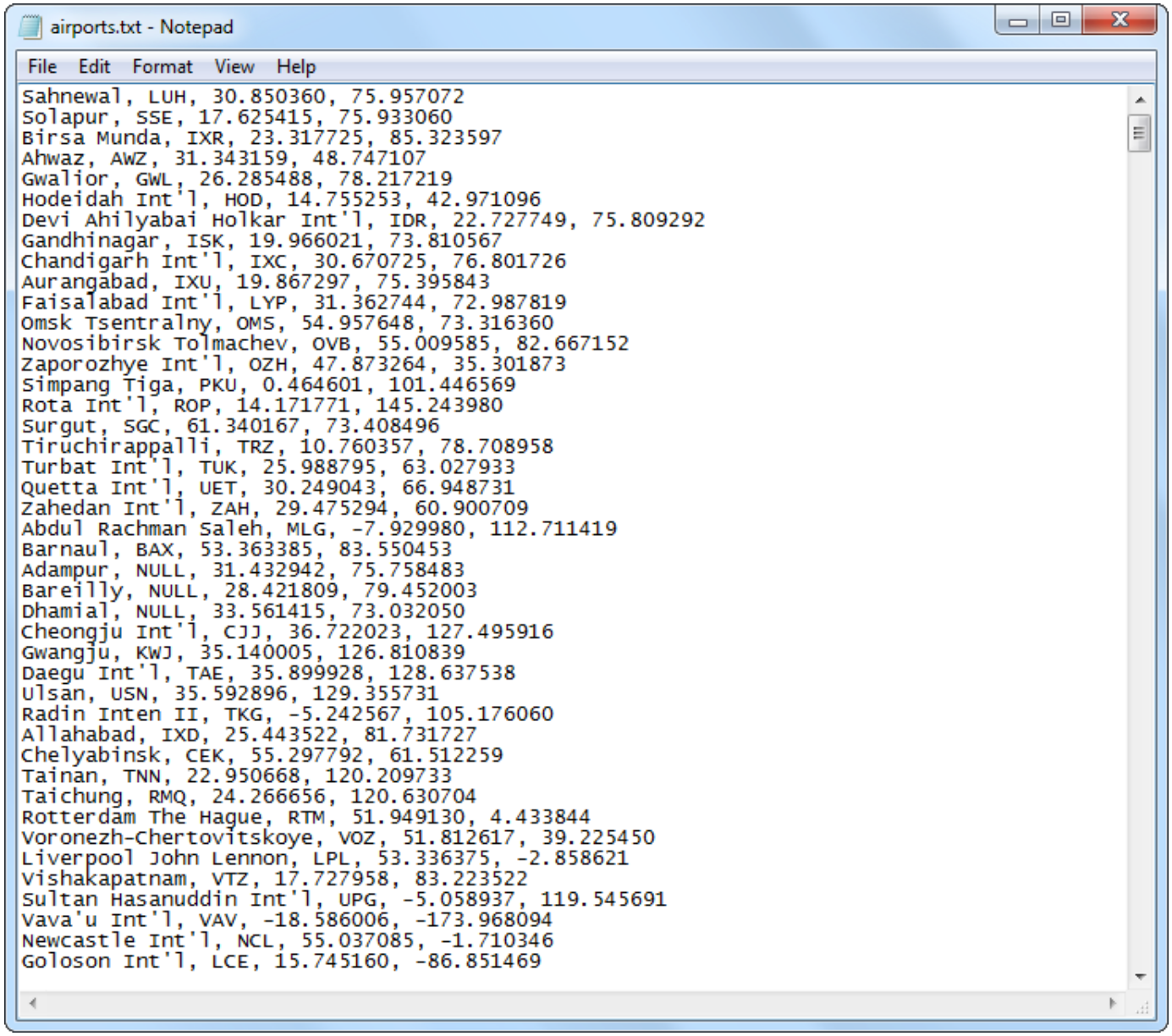


12. Adesso potete osservare l'output stampato sulla console. Un metodo più comodo di immagazzinare questi dati potrebbe essere quello di scriverli in un file. Eseguite il codice che segue per creare un file e scrivere su di esso il nostro output. Ovviamente, dovrete sostituire il percorso per raggiungere il file con un path presente sui vostri sistemi di memoria di massa -- disco fisso, pendrive etc. Come vedete aggiungiamo ***\n*** alla fine di ogni linea che formattiamo. Questo si fa per andare a capo e aggiungere una linea dopo che sono stati inseriti i dati relativi a ciascun elemento. Dovreste notare anche la linea ***unicode_line = line.encode('utf-8')***. Siccome alcune geometrie del nostro layer contengono caratteri unicode, non possiamo semplicemente scriverli in un file di testo. Dobbiamo codificare il testo usando la codifica UTF-8 e solo dopo scrivere sul file di testo.

```
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. Ora potete andare sul file di output nella cartella che avete specificato e aprire finalmente il file di testo. Vedrete i dati provenienti dallo shapefile degli aeroporti, cioè quelli che abbiamo potuto estrarre grazie all'uso dello scrpting python.


airports.txt - Notepad
File Edit Format View Help
Sahnewal, LUH, 30.850360, 75.957072
Solapur, SSE, 17.625415, 75.933060
Birsa Munda, 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