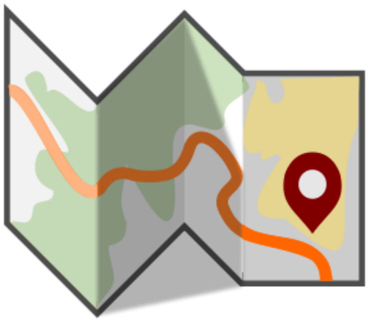
Getting Started with Python Programming

QGIS Tutorials and Tips



Author

Ujaval Gandhi

http://google.com/+UjavalGandhi

Translations by

SongHyun Choi

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

파이썬 프로그래밍 시작

QGIS는 강력한 프로그래밍 인터페이스를 가지고 있는데 이것으로 소프트웨어의 핵심 기능을 확장할 수 있고 작업을 자동화하는 스크립트를 만들어 낼 수 있습니다 . QGIS는 요즘 인기있는 파이썬 스크립팅 언어를 지원합니다 . 초보자라할 지라도 조금만 파이썬과 QGIS 프로그래밍 인터페이스를 배우면 작업이 매우 생산적이 될 것입니다 . 이 예제에서는 사전에 프로그램에 대한 지식이 없다고 가정하고 QGIS (PyQGIS)엣 파이썬을 소개하고자 합니다 .

과업 개요

예제에서는 모든 주요 공항을 표현하는 벡터 점 레이어를 불러와서 파이썬 스크립팅으로 레이어에서 공항의 이름 , 공항코드, 각 공항의 위도와 경도를 텍스트파일로 만들 것입니다 .

데이터 획득

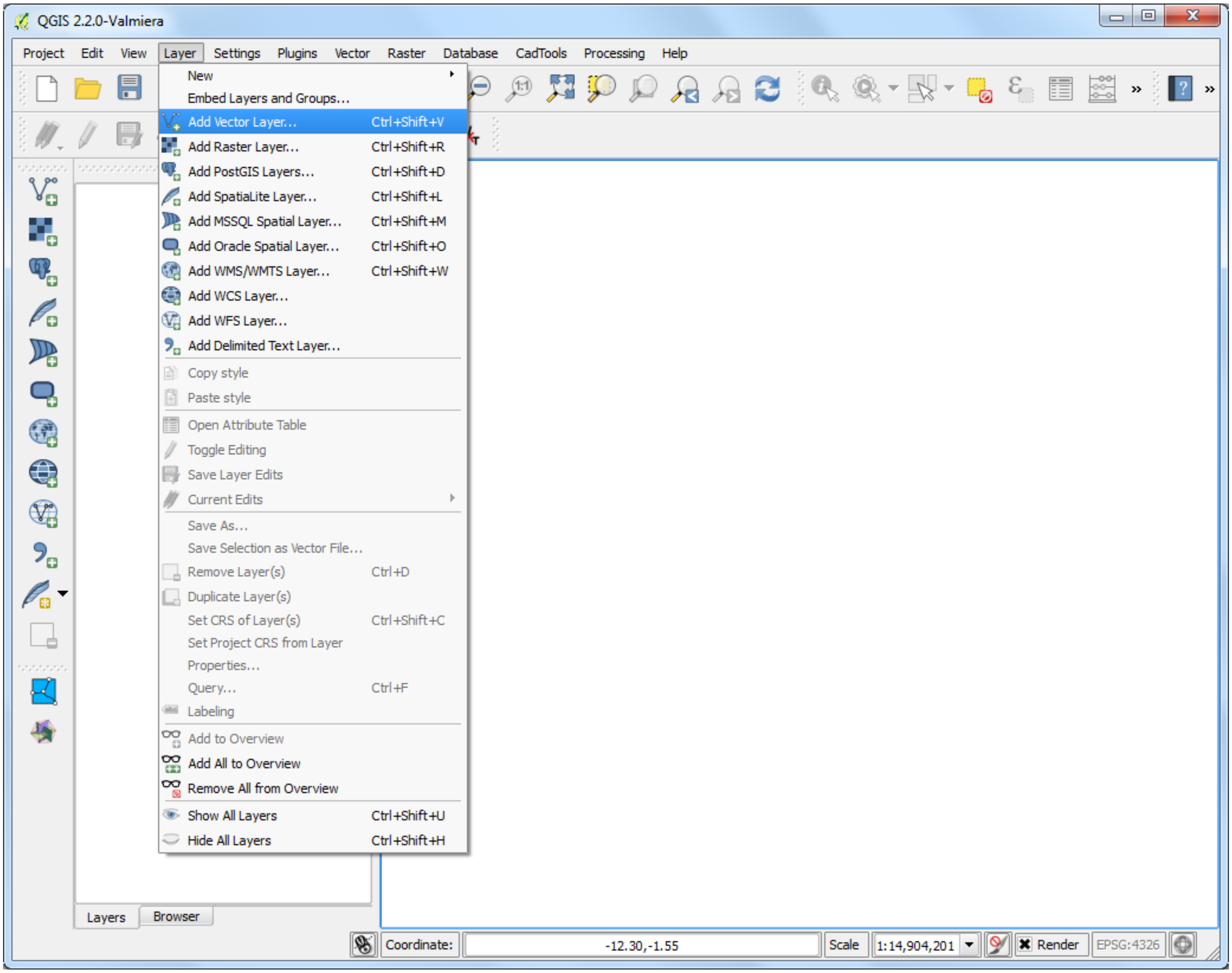
Natural Earth의 Airports 데이터셋을 사용합니다 .

`Airports shapefile <http://www.naturalearthdata.com/http//www.naturalearthdata.com/download/10m/cultural/ne_10m_airports.zip>`_를 다운로드 하십시오 .

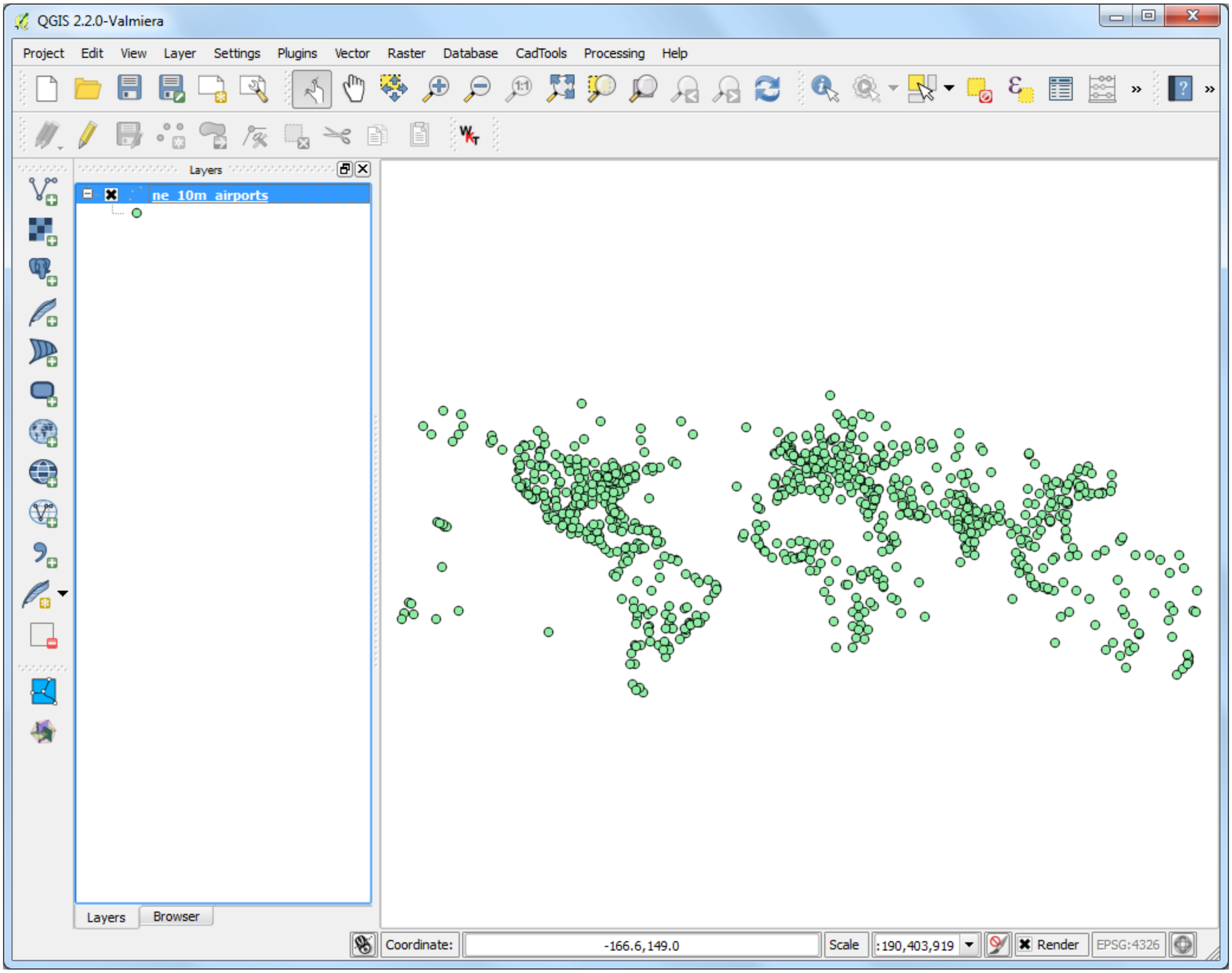
데이터 출처 : [NATURALEARTH]

과정

1. QGIS에서 메뉴 레이어 --> 레이어 추가 --> 벡터 레이어 추가 :menuselection:`Layers --> Add Vector Layer`로 가십시오. 다운로드한 ``ne_10m_airports.zip``파일을 찾고 열기 :guilabel:`Open`를 누릅니다. ``ne_10m_airports.shp``레이어를 선택하고 확인 :guilabel:`OK`을 누릅니다.


QGIS 2.2.0-Valmiera
Project Edit View Layer Settings Plugins Vector Raster Database CadTools Processing Help
New
Embed Layers and Groups...
Add Vector Layer... Ctrl+Shift+V
Add Raster Layer... Ctrl+Shift+R
Add PostGIS Layers... Ctrl+Shift+D
Add SpatiaLite Layer... Ctrl+Shift+L
Add MSSQL Spatial Layer... Ctrl+Shift+M
Add Oracle Spatial Layer... Ctrl+Shift+O
Add WMS/WMTS Layer... Ctrl+Shift+W
Add WCS Layer...
Add WFS Layer...
Add Delimited Text Layer...
Copy style
Paste style
Open Attribute Table
Toggle Editing
Save Layer Edits
Current Edits
Save As...
Save Selection as Vector File...
Remove Layer(s) Ctrl+D
Duplicate Layer(s)
Set CRS of Layer(s) Ctrl+Shift+C
Set Project CRS from Layer
Properties...
Query... Ctrl+F
Labeling
Add to Overview
Add All to Overview
Remove All from Overview
Show All Layers Ctrl+Shift+U
Hide All Layers Ctrl+Shift+H
Layers
Browser
Coordinate:
-12.30,-1.55
Scale
1:14,904,201
Render
EPSG:4326

2. QGIS에 ``ne_10m_airports`` 레이어가 불러들여질 것입니다 .


QGIS 2.2.0-Valmiera
Project
Edit
View
Layer
Settings
Plugins
Vector
Raster
Database
CadTools
Processing
Help
Layers
Layers
Browser
Render

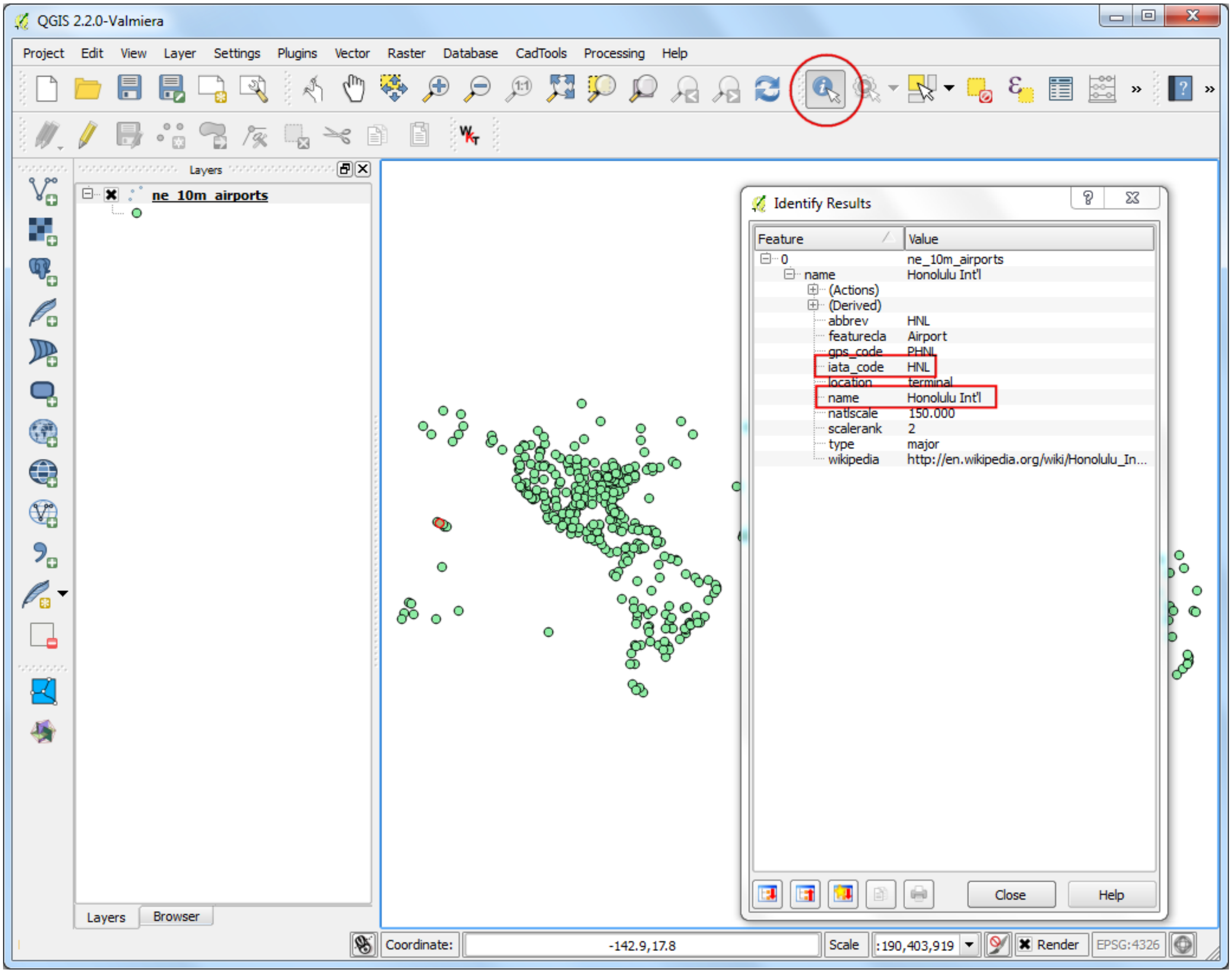
ne_10m_airports

Coordinate: -166.6,149.0

Scale :190,403,919

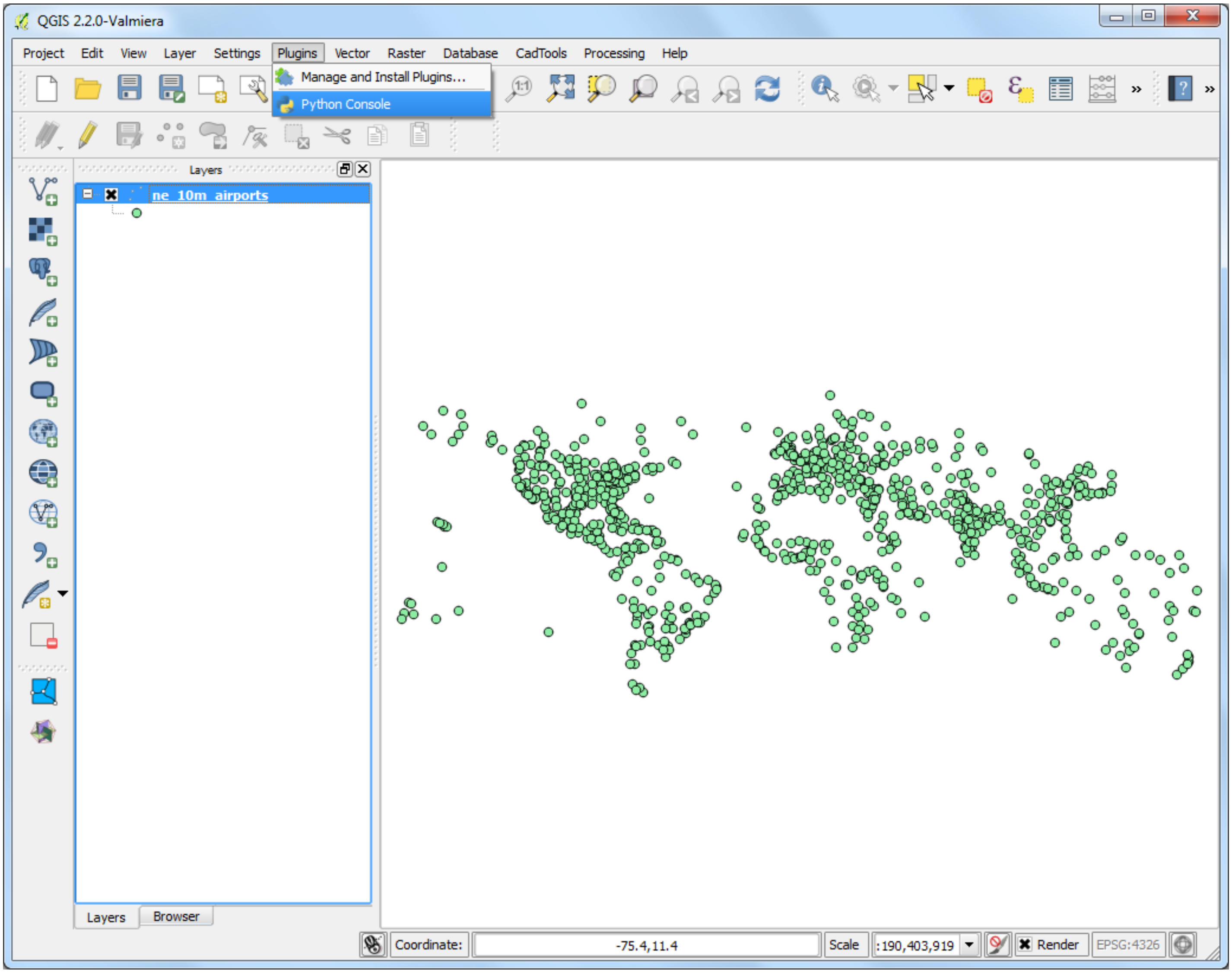
EPSG:4326

3. 객체 확인 Identify 툴을 선택하고 사용가능한 속성을 확인하기 위해 아무 점이나 누릅니다. 공항이름과 공항의 3자리수 코드가 속성의 ``name``과 ``iata_code``로 각각 포함되어 있는 것을 보게됩니다.


QGIS 2.2.0-Valmiera
Project
Edit
View
Layer
Settings
Plugins
Vector
Raster
Database
CadTools
Processing
Help
Layers
ne_10m_airports
Layers
Browser
Identify Results
Close
Help
Coordinate:
-142.9,17.8
Scale
:190,403,919
Render
EPSG:4326

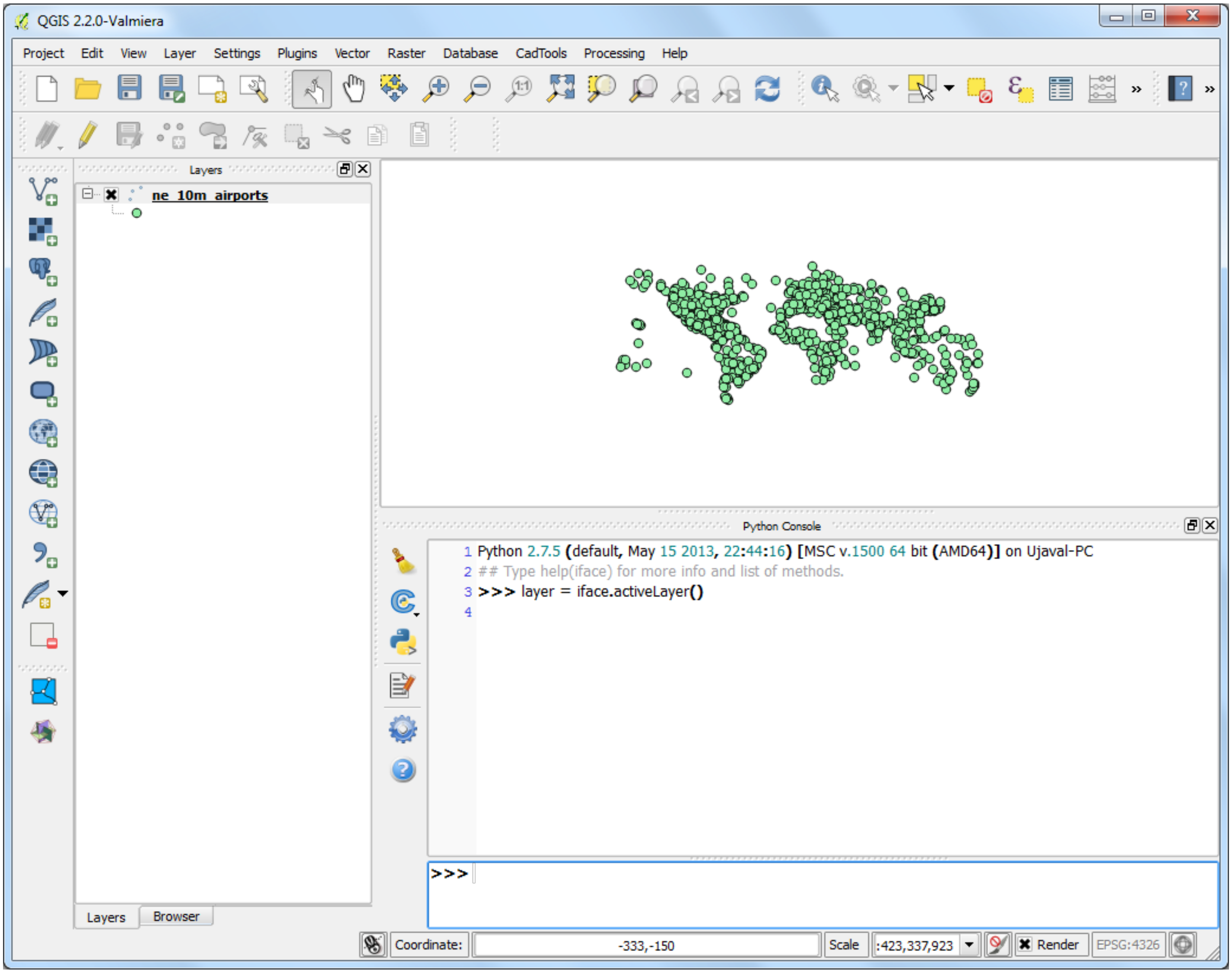
Feature	Value
0	ne_10m_airports
name	Honolulu Int'l
(Actions)	
(Derived)	
abbrev	HNL
featurecla	Airport
gps_code	PHNL
iata_code	HNL
location	terminal
name	Honolulu Int'l
natlscale	150.000
scalerank	2
type	major
wikipedia	http://en.wikipedia.org/wiki/Honolulu_In...

4. QGIS는 파이썬 명령어를 입력할 수 있고 결과를 얻을 수 있는 콘솔을 기본적으로 제공합니다 . 이 콘솔은 스크립팅을 배울 수 있는 가장 좋은 방법이고 또한 데이터를 빠르게 처리할 수 있습니다 . 메뉴 플러그인 --> Python 콘솔 :menuselection:`Plugins --> Python Console`로 가서 Python 콘솔 :guilabel:`Python Console`을 엽니다.


QGIS 2.2.0-Valmiera
Project
Edit
View
Layer
Settings
Plugins
Vector
Raster
Database
CadTools
Processing
Help
Manage and Install Plugins...
Python Console
Layers
ne_10m_airports
Layers
Browser
Coordinate:
-75.4,11.4
Scale
:190,403,919
Render
EPSG:4326

5. QGIS 캔버스 아랫쪽에 새로운 패널이 열리는 것이 보일 것입니다. 명령어를 입력할 수 있는 아랫쪽에 **>>>``■ ■■ ■■■■■ ■■■ ■■■■.** ***QGIS■■■ ■■■■■■■ ``iface*** 변수를 사용해야만 합니다. QGIS에서 현재 활성화된 레이어에 접근하기 위해서 다음과 같이 타이핑하고 엔터 Enter`를 누릅니다. 이 명령어는 레퍼런스를 현재 활성화된 레이어로 가져오고 ``layer` 변수에 저장합니다.

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

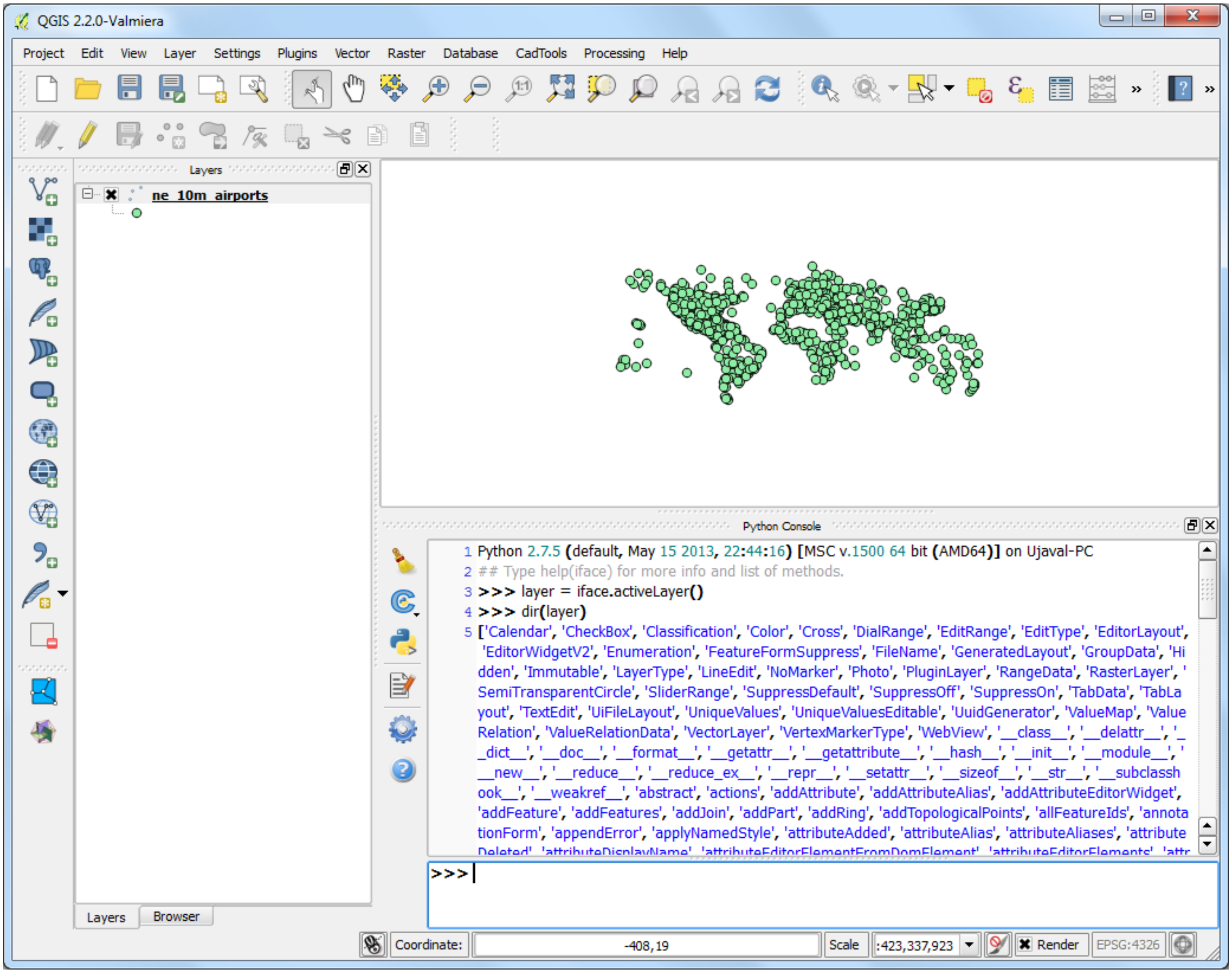

QGIS 2.2.0-Valmiera
Project
Edit
View
Layer
Settings
Plugins
Vector
Raster
Database
CadTools
Processing
Help
Layers
ne_10m_airports
Python Console
Layers
Browser
Coordinate:
-333,-150
Scale
:423,337,923
Render
EPSG:4326

```
Python 2.7.5 (default, May 15 2013, 22:44:16) [MSC v.1500 64 bit (AMD64)] on Ujaval-PC
## Type help(iface) for more info and list of methods.
>>> layer = iface.activeLayer()
```

```
>>>
```

6. 파이썬에는 ``dir()`` 라고 하는 편리한 기능이 있는데 어떤 객체에 대한 모든 가용한 방법을 보여줍니다. 이것은 어떤 함수가 객체에 사용가능한 지 확실하지 않을 때 유용합니다. ``layer`` 변수에 대해 어떤 것을 할 수 있는지 알아보기위해 다음의 명령어를 실행합니다.

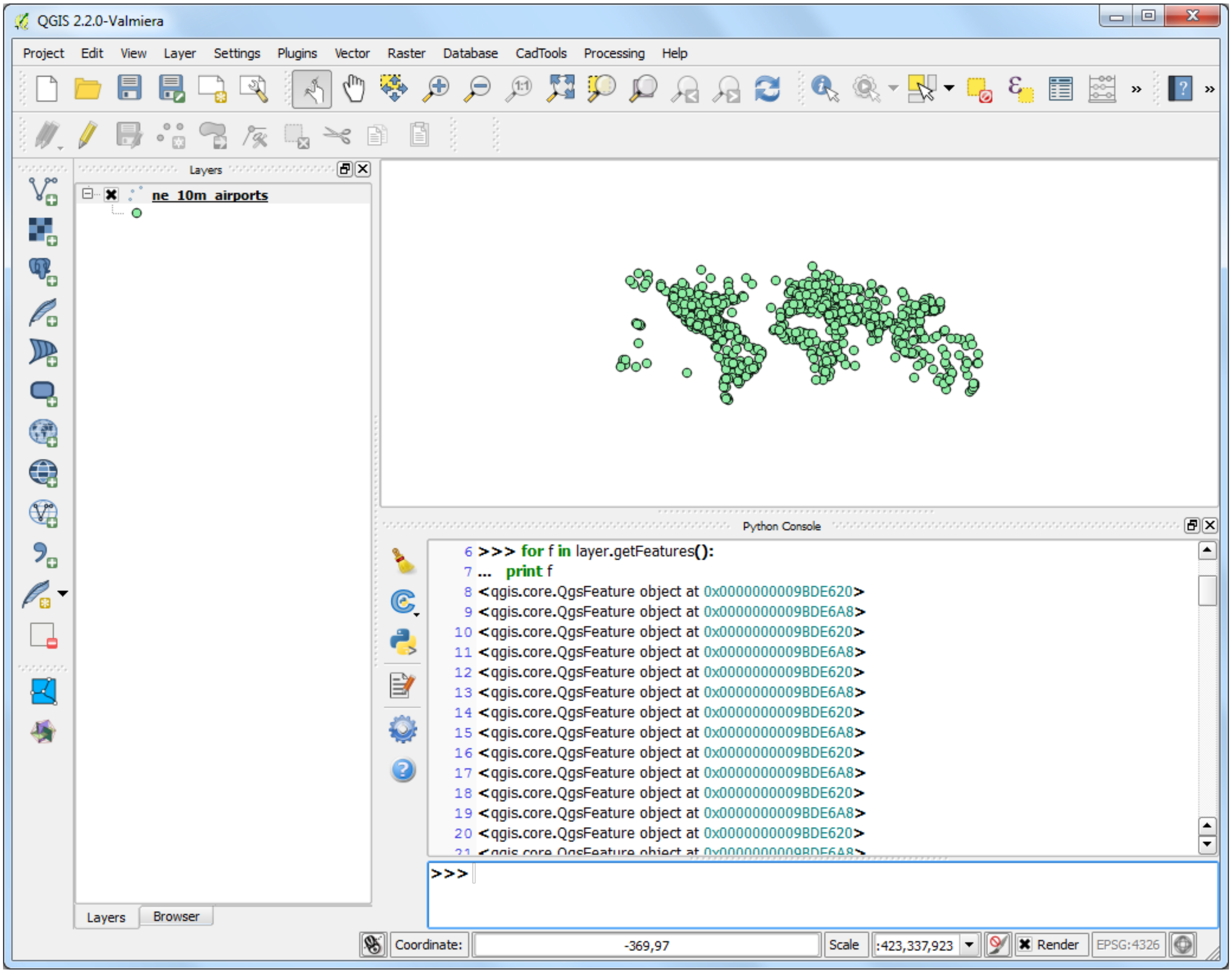
```
dir(layer)
```


QGIS 2.2.0-Valmiera
Project Edit View Layer Settings Plugins Vector Raster Database CadTools Processing Help
Layers
ne_10m_airports
Layers Browser
Python Console
Coordinate: -408,19
Scale :423,337,923
Render
EPSG:4326

```
Python 2.7.5 (default, May 15 2013, 22:44:16) [MSC v.1500 64 bit (AMD64)] on Ujaval-PC
## Type help(iface) for more info and list of methods.
>>> layer = iface.activeLayer()
>>> dir(layer)
['Calendar', 'CheckBox', 'Classification', 'Color', 'Cross', 'DialRange', 'EditRange', 'EditType', 'EditorLayout',
 'EditorWidgetV2', 'Enumeration', 'FeatureFormSuppress', 'FileName', 'GeneratedLayout', 'GroupData', 'Hi
dden', 'Immutable', 'LayerType', 'LineEdit', 'NoMarker', 'Photo', 'PluginLayer', 'RangeData', 'RasterLayer', '
SemiTransparentCircle', 'SliderRange', 'SuppressDefault', 'SuppressOff', 'SuppressOn', 'TabData', 'TabLa
yout', 'TextEdit', 'UiFileLayout', 'UniqueValues', 'UniqueValuesEditable', 'UuidGenerator', 'ValueMap', 'Value
Relation', 'ValueRelationData', 'VectorLayer', 'VertexMarkerType', 'WebView', '__class__', '__delattr__', '_
_dict__', '__doc__', '__format__', '__getattr__', '__getattribute__', '__hash__', '__init__', '__module__', '
__new__', '__reduce__', '__reduce_ex__', '__repr__', '__setattr__', '__sizeof__', '__str__', '__subclassh
ook__', '__weakref__', 'abstract', 'actions', 'addAttribute', 'addAttributeAlias', 'addAttributeEditorWidget',
'addFeature', 'addFeatures', 'addJoin', 'addPart', 'addRing', 'addTopologicalPoints', 'allFeatureIds', 'annota
tionForm', 'appendError', 'applyNamedStyle', 'attributeAdded', 'attributeAlias', 'attributeAliases', 'attribute
Deleted', 'attributeDisplayName', 'attributeEditorElementFromDomElement', 'attributeEditorElements', 'attr
>>>
```

7. 긴 목록의 사용가능한 함수를 보게될 것입니다. 이제 레이어의 모든 객체의 레퍼런스를 보여주는 ``getFeatures()``라는 함수를 사용할 것입니다. 이 경우 각 객체는 공항을 나타내는 점이 될 것입니다. 현재의 레이어에서 각 객체를 통해 반복하는 다음의 명령을 입력할 수 있습니다. 두번째 줄에 입력을 하기 전에 두칸의 스페이스를 추가하는 것을 잊지마십시오.

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


QGIS 2.2.0-Valmiera
Project
Edit
View
Layer
Settings
Plugins
Vector
Raster
Database
CadTools
Processing
Help
Layers
ne_10m_airports
Python Console
Layers
Browser
Render

```
>>> for f in layer.getFeatures():
...     print f
<qgis.core.QgsFeature object at 0x000000009BDE620>
<qgis.core.QgsFeature object at 0x000000009BDE6A8>
<qgis.core.QgsFeature object at 0x000000009BDE620>
<qgis.core.QgsFeature object at 0x000000009BDE6A8>
<qgis.core.QgsFeature object at 0x000000009BDE620>
<qgis.core.QgsFeature object at 0x000000009BDE6A8>
<qgis.core.QgsFeature object at 0x000000009BDE620>
<qgis.core.QgsFeature object at 0x000000009BDE6A8>
<qgis.core.QgsFeature object at 0x000000009BDE620>
<qgis.core.QgsFeature object at 0x000000009BDE6A8>
<qgis.core.QgsFeature object at 0x000000009BDE620>
<qgis.core.QgsFeature object at 0x000000009BDE6A8>
<qgis.core.QgsFeature object at 0x000000009BDE620>
<qgis.core.QgsFeature object at 0x000000009BDE6A8>
<qgis.core.QgsFeature object at 0x000000009BDE620>
```

```
>>>
```

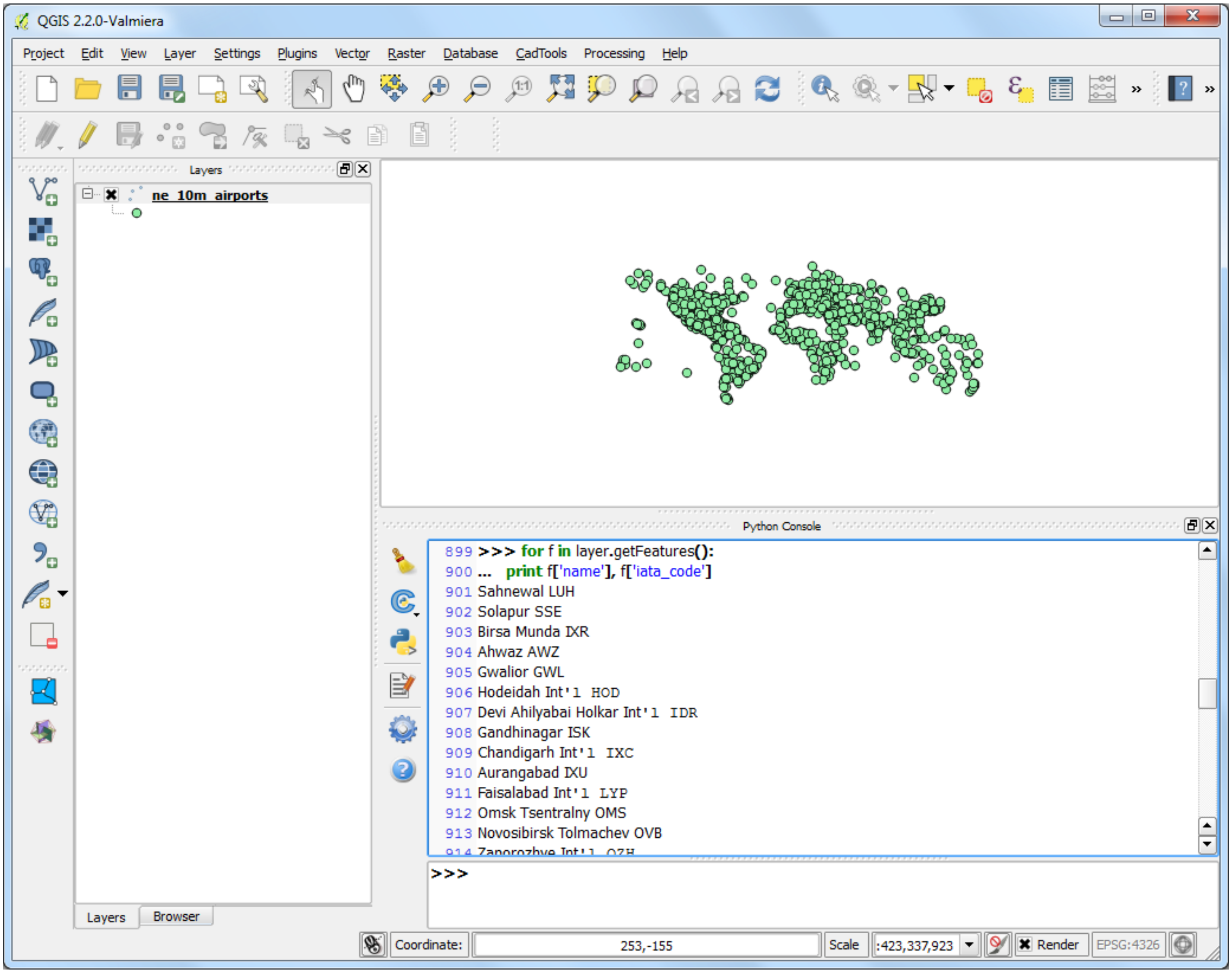
Coordinate: -369,97

Scale :423,337,923

EPSG:4326

8. 결과에서 보게되는 것 처럼 각 입력줄은 레이어 내의 객체의 레퍼런스를 포함하고 있습니다. 객체의 레퍼런스는 ``f`` 변수에 저장됩니다. 각 객체의 속성에 접근하기 위해 ``f`` 변수를 사용할 수 있습니다. 각 공항 객체에 대한 ``name`` 과 ``iata_code`` 를 출력하기 위해 다음과 같이 입력합니다.

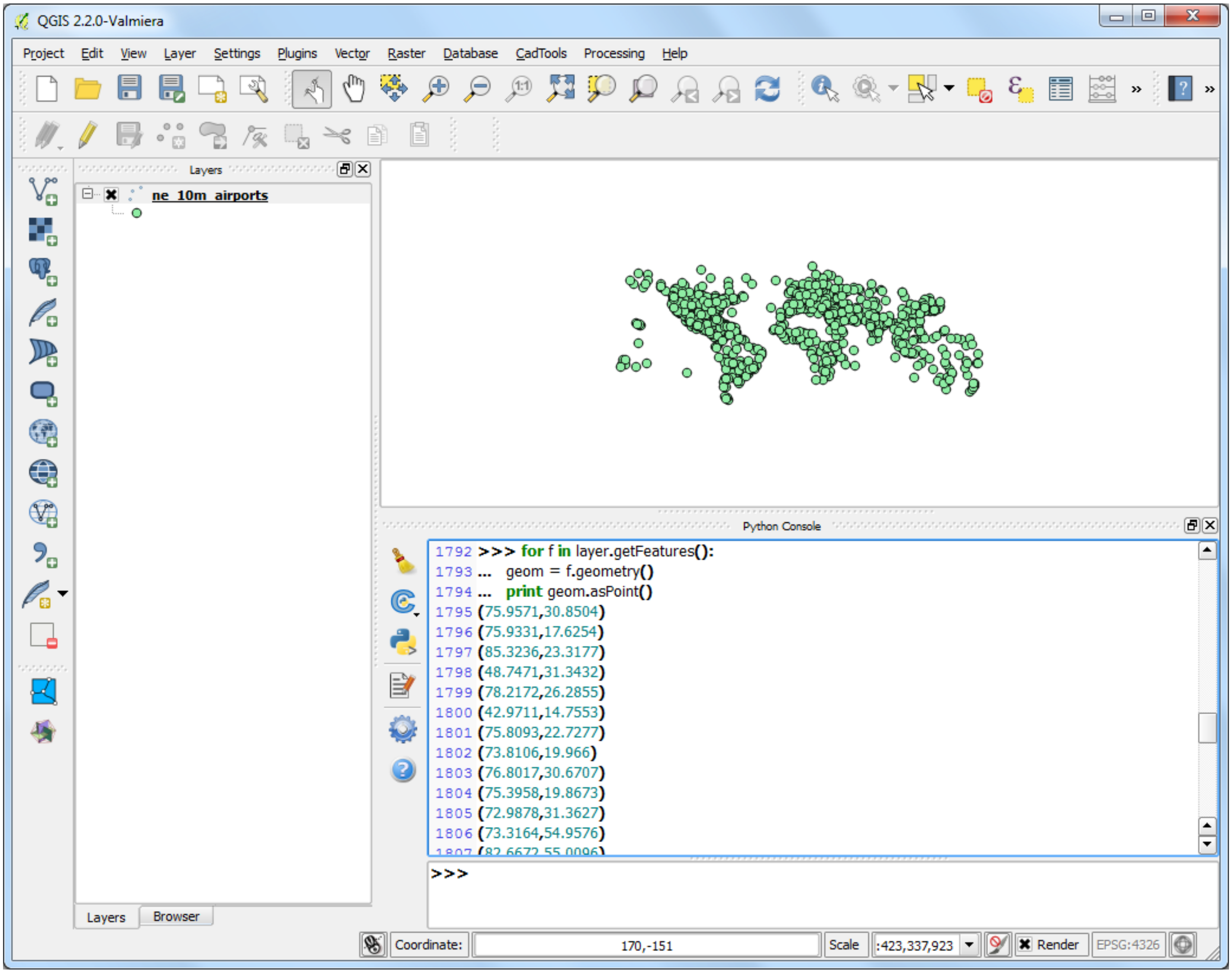
```
for f in layer.getFeatures():
  print f['name'], f['iata_code']
```


QGIS 2.2.0-Valmiera
Project Edit View Layer Settings Plugins Vector Raster Database CadTools Processing Help
Layers
ne_10m_airports
Python Console
Layers Browser
Coordinate: 253,-155
Scale :423,337,923
Render
EPSG:4326

```
>>> for f in layer.getFeatures():
...     print f['name'], f['iata_code']
Sahnewal LUH
Solapur SSE
Birsa Munda IXR
Ahwaz AWZ
Gwalior GWL
Hodeidah Int'l HOD
Devi Ahilyabai Holkar Int'l IDR
Gandhinagar ISK
Chandigarh Int'l IXC
Aurangabad IXU
Faisalabad Int'l LYP
Omsk Tsentralny OMS
Novosibirsk Tolmachev OVB
Zaporozhye Int'l OZH
>>>
```

떻게 체계적으로 접근하는지 알게됩니다. 이제 객체의 좌표에 어떻게 접근
[illegible] [illegible] ``geom`` [illegible] [illegible] [illegible] [illegible]. [illegible] *x, y*[illegible] [illegible] [illegible] [illegible] [illegible] [illegible] ``*asPoint()*
롬프트에 다음의 코드를 입력하고 엔터 :kbd:`Enter` 를 누릅니다.

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

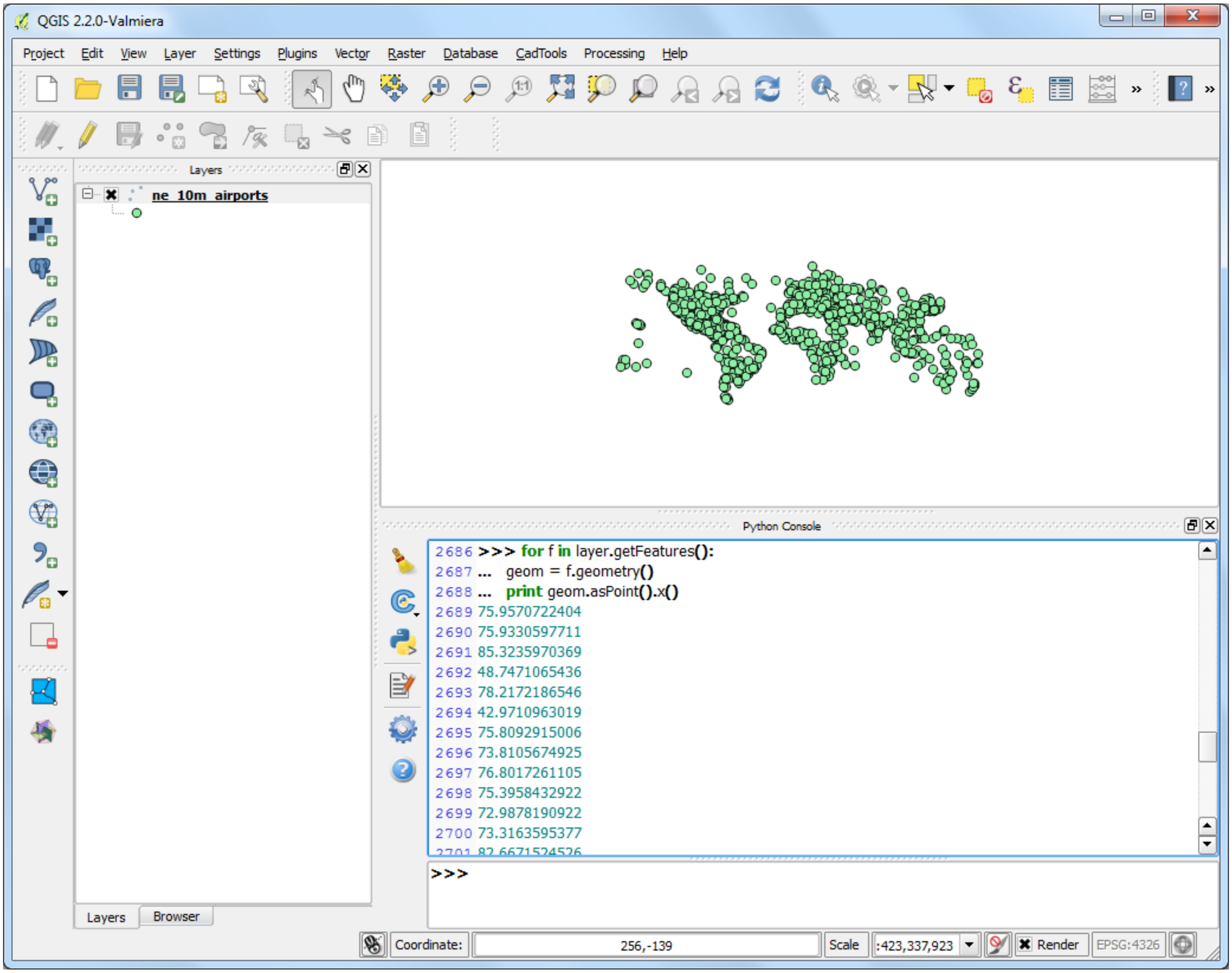

QGIS 2.2.0-Valmiera
Project Edit View Layer Settings Plugins Vector Raster Database CadTools Processing Help
Layers
ne_10m_airports
Python Console
Layers Browser
Render

```
>>> for f in layer.getFeatures():
...   geom = f.geometry()
...   print geom.asPoint()
(75.9571,30.8504)
(75.9331,17.6254)
(85.3236,23.3177)
(48.7471,31.3432)
(78.2172,26.2855)
(42.9711,14.7553)
(75.8093,22.7277)
(73.8106,19.966)
(76.8017,30.6707)
(75.3958,19.8673)
(72.9878,31.3627)
(73.3164,54.9576)
(82.6672,55.0096)
>>>
```

Coordinate: 170,-151
Scale :423,337,923
EPSG:4326

10. 객체의 *x* 좌표만 얻고자 한다면 어떨까요 ? 점 객체에 대해 ``x()`` 함수를 불러 x 좌표을 얻을 수 있습니다 .

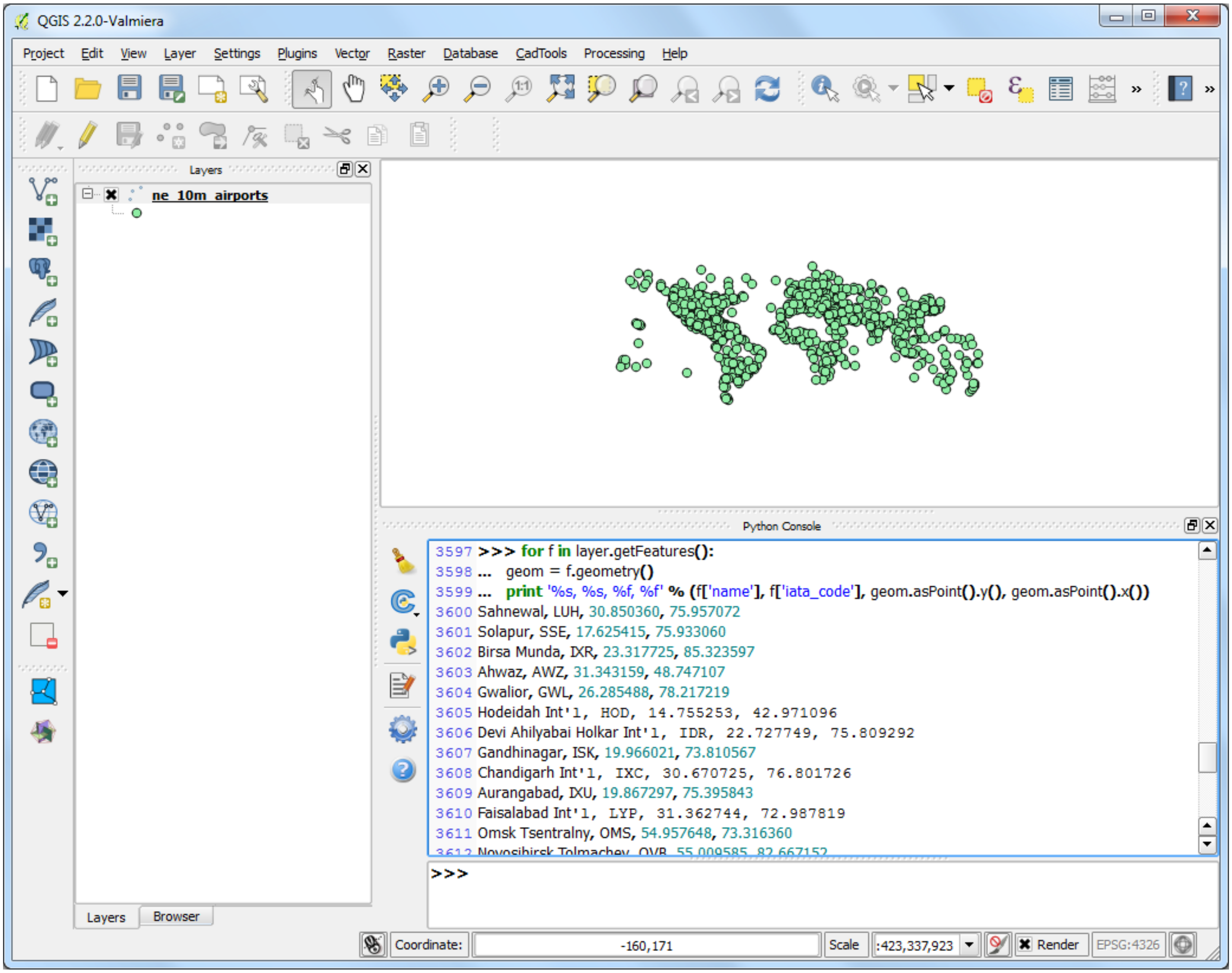
```
for f in layer.getFeatures():
  geom = f.geometry()
  print geom.asPoint().x()
```


QGIS 2.2.0-Valmiera
Project
Edit
View
Layer
Settings
Plugins
Vector
Raster
Database
CadTools
Processing
Help
Layers
ne_10m_airports
Layers
Browser
Python Console
Coordinate:
256,-139
Scale
:423,337,923
Render
EPSG:4326

```
>>> for f in layer.getFeatures():
...   geom = f.geometry()
...   print geom.asPoint().x()
75.9570722404
75.9330597711
85.3235970369
48.7471065436
78.2172186546
42.9710963019
75.8092915006
73.8105674925
76.8017261105
75.3958432922
72.9878190922
73.3163595377
82.6671524526
>>>
```

11. 이제 원하는 결과를 만들어 내기 위해 함께 조합을 할 수 있는 모든 명령어 조각을 확보했습니다. 각 공항 객체의 이름, 공항코드, 위도 그리고 경도를 출력하려면 다음의 코드를 입력하십시오. ``%s`` 와 ``%f`` 표시는 문자와 숫자변수를 포맷하는 방법입니다.

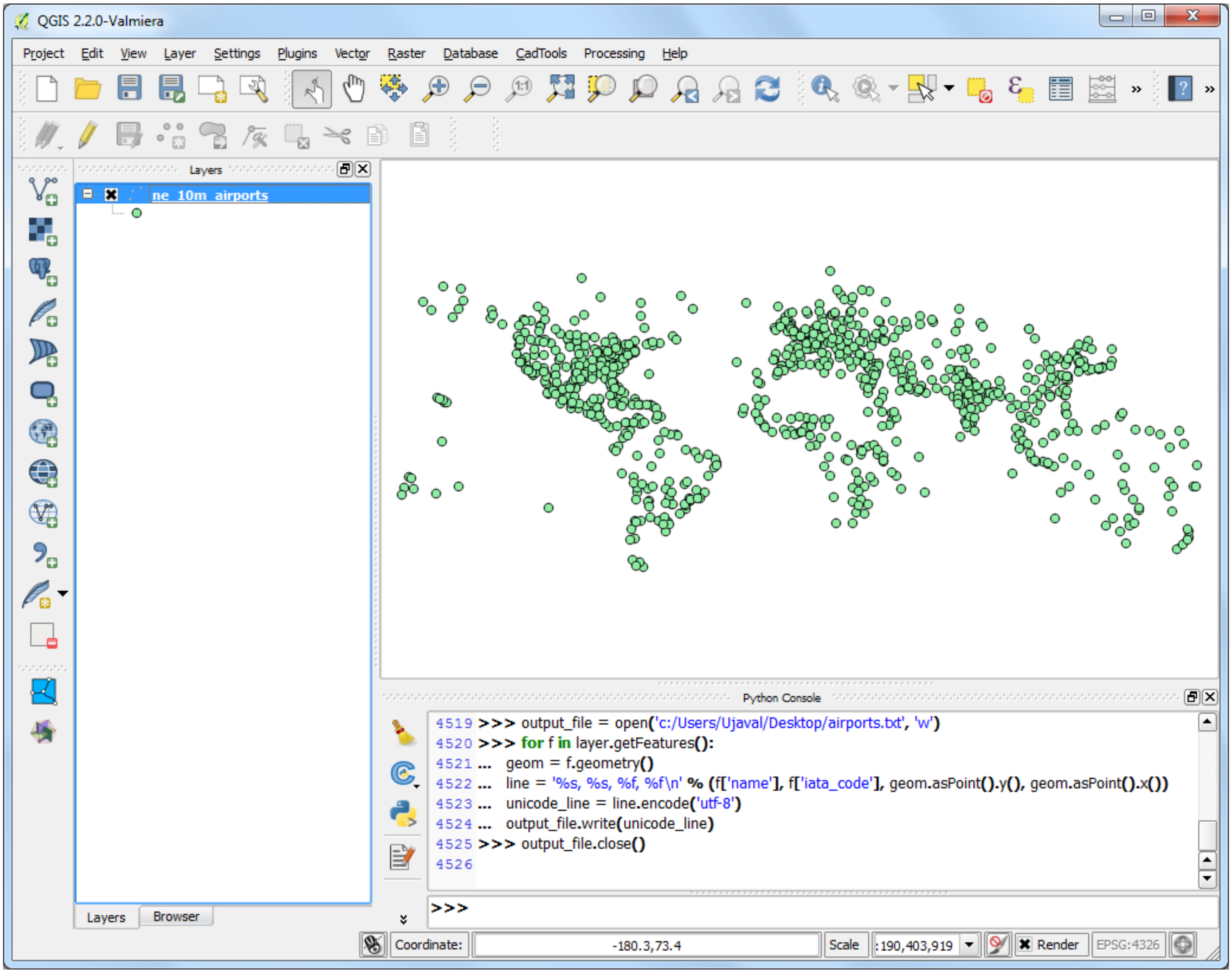
```
for f in layer.getFeatures():
  geom = f.geometry()
  print '%s, %s, %f, %f' % (f['name'], f['iata_code'],
         geom.asPoint().y(), geom.asPoint().x())
```


QGIS 2.2.0-Valmiera
Project Edit View Layer Settings Plugins Vector Raster Database CadTools Processing Help
Layers
ne_10m_airports
Layers Browser
Python Console
Coordinate: -160,171
Scale :423,337,923
Render
EPSG:4326

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

□ 볼 수 있을 것입니다 . 결과를 저장하는 보다 유용한 방법은 파일에 담는 것입니다 . 파일을 만들고 그 파일에 결과를 쓰기 위해서 다음과 같은 □
경로에 맞게 파일 경로를 교체하십시오. 줄을 만들기
██ ███████. ███ █ ███ ██ ████ ███ ██ ███ ██ █████. ██ *``unicode_line = line*
□ 이어가 유니코드를 가진 객체를 포함하고 있으므로 단순하게 텍스트 파일에 그것을 기록할 수 없습니다 . UTF-8 엔코딩을 사용해서 텍스트를 □

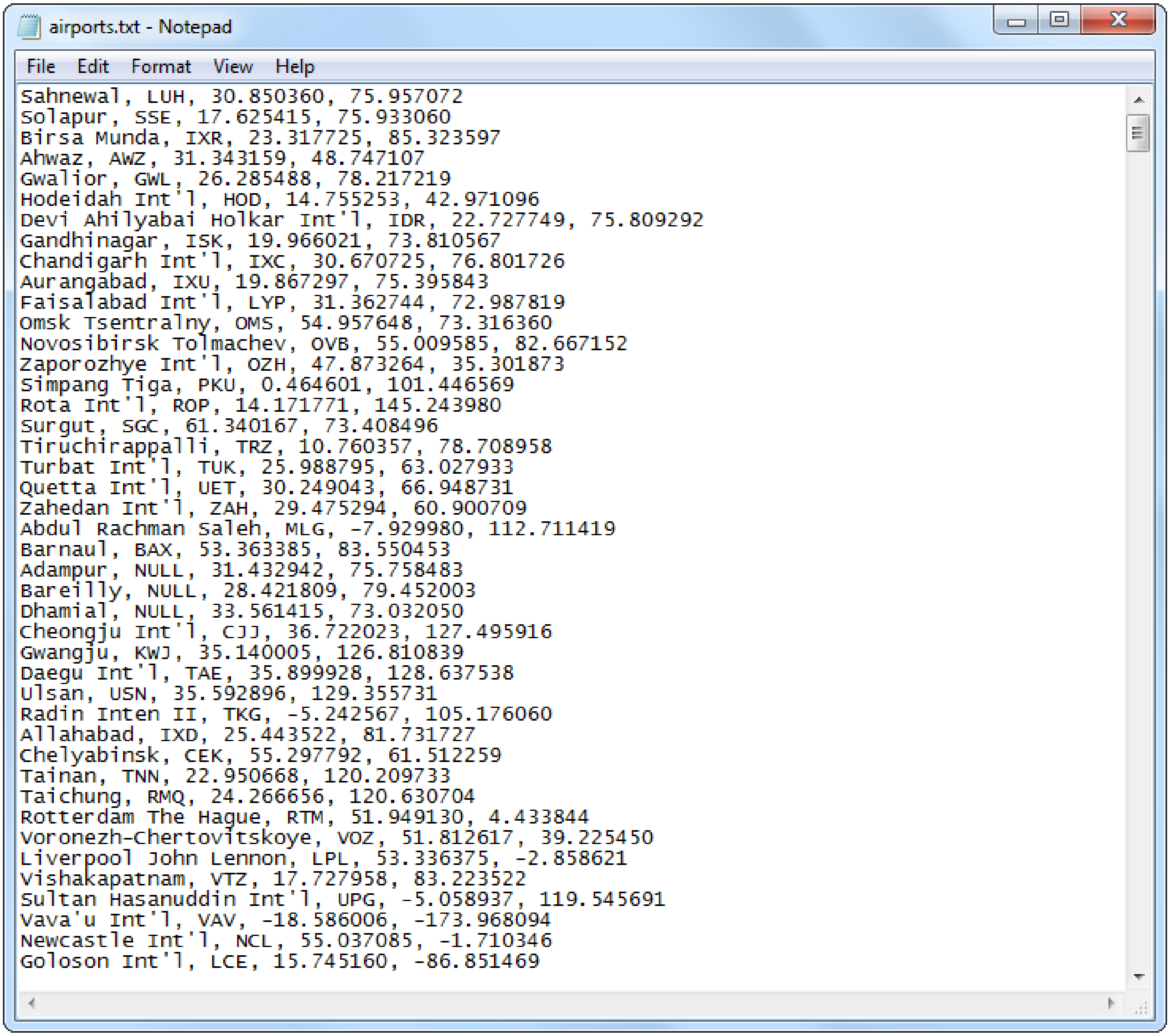
```
output_file = open('c:/Users/Ujaval/Desktop/airports.txt', 'w')
for f in layer.getFeatures():
  geom = f.geometry()
  line = '%s, %s, %f, %f\n' % (f['name'], f['iata_code'],
          geom.asPoint().y(), geom.asPoint().x())
  unicode_line = line.encode('utf-8')
  output_file.write(unicode_line)
output_file.close()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


QGIS 2.2.0-Valmiera
Project
Edit
View
Layer
Settings
Plugins
Vector
Raster
Database
CadTools
Processing
Help
Layers
ne_10m_airports
Python Console
Layers
Browser
Coordinate:
-180.3,73.4
Scale
:190,403,919
Render
EPSG:4326

```
>>> 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. 명시한 위치의 결과 파일로 가서 텍스트 파일을 열 수 있습니다. 파이썬 스크립팅을 이용해서 공항 shapefile에서 추출한 데이터를 볼 수 있을 것입니다.



```
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
```