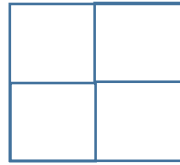


◇地図タイルとは

「レイヤ番号/X/Y」, レイヤ番号=0 のときは**全世界**, レイヤ番号が**1 増える**と, 面積は **1/4**



○画像の地図の場合には, 1 個の地図タイル = 256 × 256 の画像が標準

※ゲームの「マップ」でもタイル地図と同じ技術が使われていることも

- ・オンラインの地図システムでは, ダウンロードを早くするために, **地図データは地図タイル化されていることが多い**.
- ・最も荒いものが**レベル0**. レベルが増えることに, **範囲は狭く, 情報は精密**になる.

前準備

①Anaconda プロンプトを管理者として開き, 次のコマンドを実行する.

```
conda install pandas
conda install folium
pip install overpy
```

```
管理者: Anaconda Prompt
(C:\ProgramData\Anaconda3) C:\Windows\System32>conda install pandas
Fetching package metadata .....
Solving package specifications: .

Package plan for installation in environment C:\ProgramData\Anaconda3:

The following packages will be UPDATED:
  conda: 4.3.25-py36_0 --> 4.3.27-py36hcbae3bd_0

Proceed ([y]/n) y
conda-4.3.27-p 100% |#####| Time: 0:00:00 7.38 MB/s
(C:\ProgramData\Anaconda3) C:\Windows\System32>_
```

```
管理者: Anaconda Prompt
(C:\ProgramData\Anaconda3) C:\Windows\System32>conda install folium
Fetching package metadata .....
Solving package specifications: .

Package plan for installation in environment C:\ProgramData\Anaconda3:

The following packages will be UPDATED:

conda:         4.3.27-py36hcbae3bd_0 --> 4.3.28-py36h9daa44c_0
conda-env:     2.6.0-0 --> 2.6.0-h36134e3_1

Proceed ([y]/n) y
conda-env-2.6. 100% |#####| Time: 0:00:00 895.84 kB/s
conda-4.3.28-p 100% |#####| Time: 0:00:00 5.94 MB/s

(C:\ProgramData\Anaconda3) C:\Windows\System32>
```

```
(C:\ProgramData\Anaconda3) C:\Windows\System32>pip install overpy
Collecting overpy
  Downloading overpy-0.4.tar.gz (41kB)
    100% |#####| 51kB 819kB/s
Building wheels for collected packages: overpy
  Running setup.py bdist_wheel for overpy ... done
  Stored in directory: C:\Users\kaneko\AppData\Local\pip\Cache\wheels\ee\68\03\428ee05
Successfully built overpy
Installing collected packages: overpy
Successfully installed overpy-0.4

(C:\ProgramData\Anaconda3) C:\Windows\System32>
```

◇ Python を使って、OpenStreetMap のタイル地図を表示させてみる

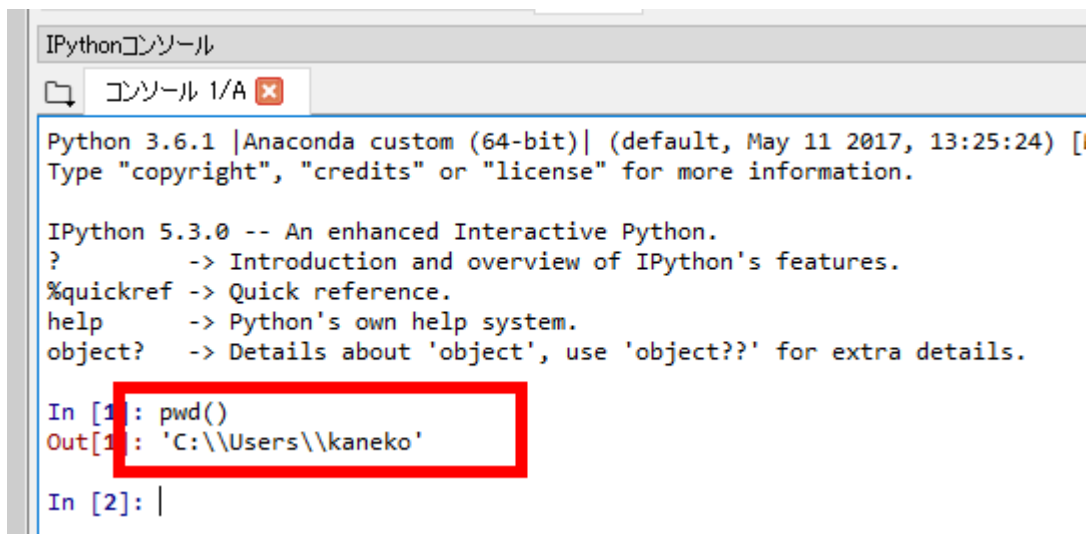
① spyder の起動

spyder は Windows のスタートメニューで簡単に実行できる。

② spyder のコンソールで、pwd() を実行して、カレントディレクトリを確認

あとで、ファイルを作成する。ファイルはカレントディレクトリに作成される。

(カレントディレクトリが分らないと、あとで、ファイルがどこにあるか分からなくなる)。



```
IPythonコンソール
Python 3.6.1 |Anaconda custom (64-bit)| (default, May 11 2017, 13:25:24) [
Type "copyright", "credits" or "license" for more information.

IPython 5.3.0 -- An enhanced Interactive Python.
?          -> Introduction and overview of IPython's features.
%quickref  -> Quick reference.
help       -> Python's own help system.
object?    -> Details about 'object', use 'object??' for extra details.

In [1]: pwd()
Out[1]: 'C:\\Users\\kaneko'

In [2]: |
```

③ 「OpenStreetMap のタイル地図をダウンロードするプログラム」の作成

folium パッケージを使い、プログラムを、簡単なコマンドで作成できる。

緯度経度： **34.4461, 133.2315**

レイヤ番号： **12** 番

OpenStreetMap を使い、緯度経度が **34.4461, 133.2315** のところを含むタイル地図で、レイヤ番号 **12** 番のものを表示したい。そんな表示を行う HTML ファイル **a.html** が生成される。

```
import folium
m = folium.Map(location=[34.4461, 133.2315], zoom_start=12)
m.save("a.html")
```

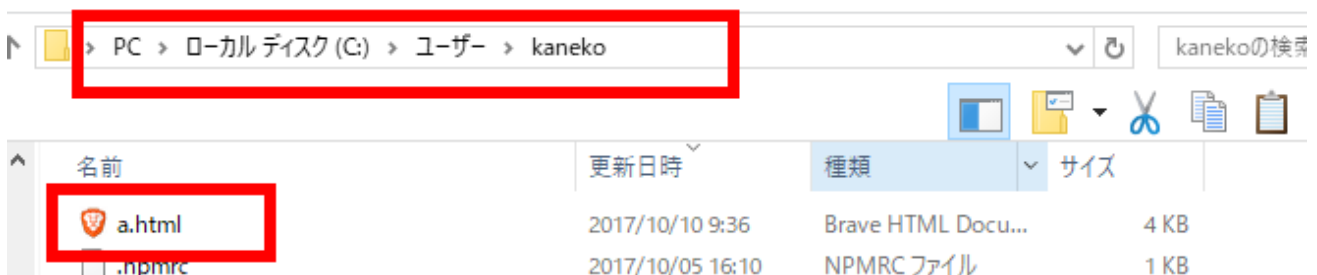
```
IPythonコンソール
コンソール 1/A
Python 3.6.1 |Anaconda custom (64-bit)| (default, May 11 2017, 13:25:24) [MS
Type "copyright", "credits" or "license" for more information.

IPython 5.3.0 -- An enhanced Interactive Python.
?          -> Introduction and overview of IPython's features.
%quickref  -> Quick reference.
help       -> Python's own help system.
object?    -> Details about 'object', use 'object??' for extra details.

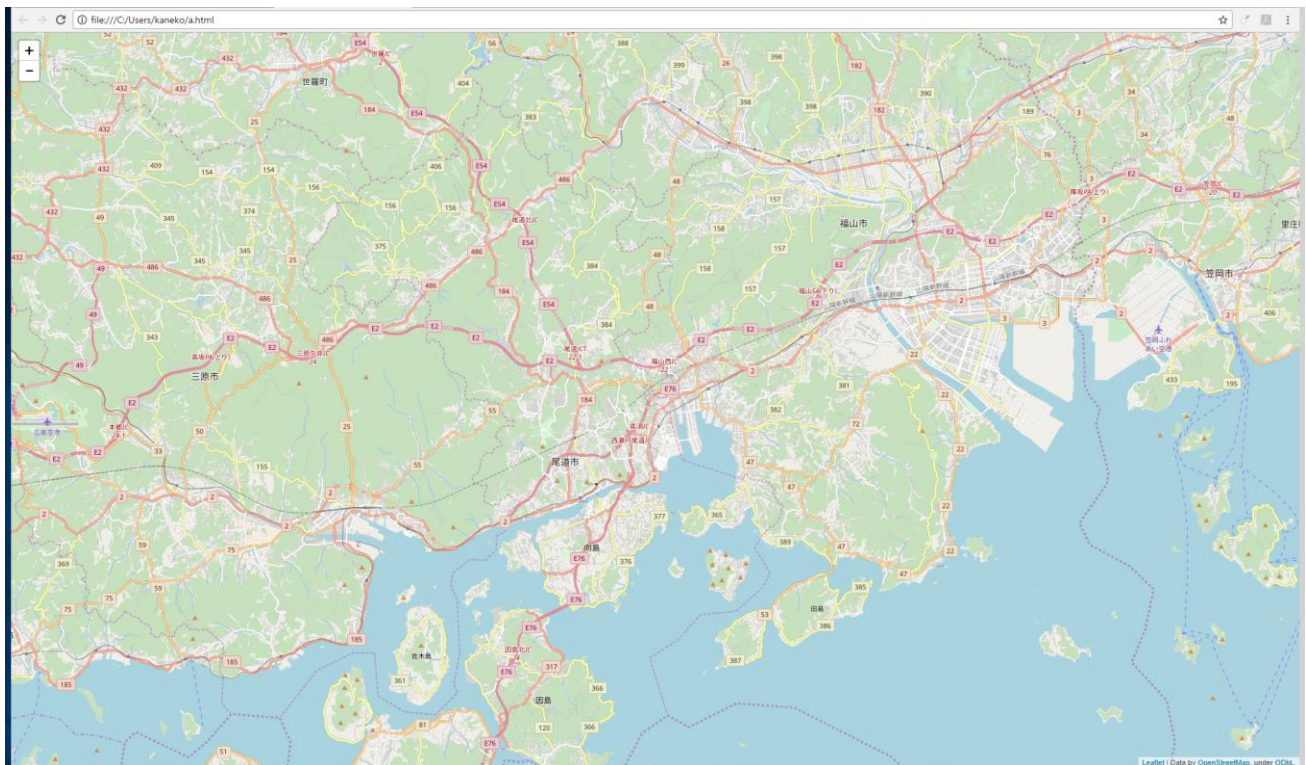
In [1]: pwd()
Out[1]: 'C:\\Users\\kaneko'

In [2]: import folium
...: m = folium.Map(location=[34.4461, 133.2315], zoom_start=12)
...: m.save("a.html")
...:
```

④ 先ほど調べた「カレントディレクトリ」にHTMLファイル **a.html** ができている



⑤ a.html をウェブブラウザで開いてみる



a.html の中身は次のようになっている

```
<!DOCTYPE html>
<head>
  <meta http-equiv="content-type" content="text/html; charset=UTF-8" />
  <script>L_PREFER_CANVAS = false; L_NO_TOUCH = false; L_DISABLE_3D =
false;</script>
  <script src="https://unpkg.com/leaflet@1.0.1/dist/leaflet.js"></script>
  <script src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.1/jquery.min.js"></script>
  <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.2.0/js/bootstrap.min.js">
</script>
  <script src="https://cdnjs.cloudflare.com/ajax/libs/Leaflet.awesome-
markers/2.0.2/leaflet.awesome-markers.js"></script>
  <script
src="https://cdnjs.cloudflare.com/ajax/libs/leaflet.markercluster/1.0.0/leaflet.markercluster-
src.js"></script>
  <script
src="https://cdnjs.cloudflare.com/ajax/libs/leaflet.markercluster/1.0.0/leaflet.markercluster.j
s"></script>
  <link rel="stylesheet" href="https://unpkg.com/leaflet@1.0.1/dist/leaflet.css" />
  <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.2.0/css/bootstrap.min.css" />
  <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.2.0/css/bootstrap-theme.min.css" />
  <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/font-
awesome/4.6.3/css/font-awesome.min.css" />
  <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/Leaflet.awesome-
markers/2.0.2/leaflet.awesome-markers.css" />
  <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/leaflet.markercluster/1.0.0/MarkerCluster.Default
t.css" />
  <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/leaflet.markercluster/1.0.0/MarkerCluster.css"
/>
  <link rel="stylesheet" href="https://rawgit.com/python-
visualization/foium/master/foium/templates/leaflet.awesome.rotate.css" />
  <style>html, body {width: 100%;height: 100%;margin: 0;padding: 0;}</style>
  <style>#map {position: absolute; top: 0; bottom: 0; right: 0; left: 0;}</style>

  <style> #map_822d3e2d493146b6881b1e88e8dc66d4 {
    position : relative;
    width : 100.0%;
    height: 100.0%;
    left: 0.0%;
    top: 0.0%;
  }
</style>

</head>
<body>

  <div class="foium-map" id="map_822d3e2d493146b6881b1e88e8dc66d4"></div>

</body>
<script>

  var southWest = L.latLng(-90, -180);
  var northEast = L.latLng(90, 180);
  var bounds = L.latLngBounds(southWest, northEast);

  var map_822d3e2d493146b6881b1e88e8dc66d4 = L.map(
    'map_822d3e2d493146b6881b1e88e8dc66d4',
    {center: [34.4461,133.2315],
    zoom: 12,
    maxBounds: bounds,
    layers: [],
    worldCopyJump: false,
    crs: L.CRS.EPSG3857
  });

  var tile_layer_b0c1e3360aa24ce59db20a51ca4cc098 = L.tileLayer(
    'https://{s}.tile.openstreetmap.org/{z}/{x}/{y}.png',
    {
      maxZoom: 18,
      minZoom: 1,
      continuousWorld: false,
      noWrap: false,
      attribution: 'Data by <a href="http://openstreetmap.org">OpenStreetMap</a>,
under <a href="http://www.openstreetmap.org/copyright">ODbL</a>.',
      detectRetina: false
    }
  ).addTo(map_822d3e2d493146b6881b1e88e8dc66d4);

</script>
```

◇folium パッケージが扱える地図の種類（データソースの種類）

- ・ OpenStreetMap ※ 既定（デフォルト）
- ・ Stamen Terrain
- ・ Stamen Toner

など

先ほどは、「OpenStreetMap のタイル地図をダウンロードするプログラム」だった。 他の地図も試してみる。

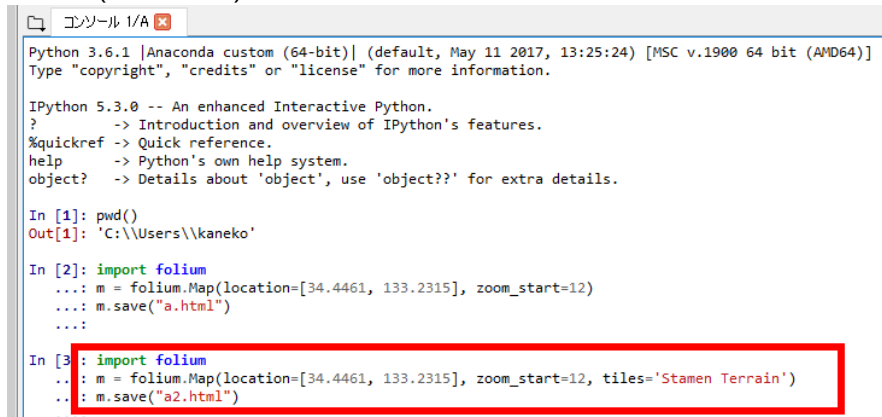
① Stamen Terrain

spyder のコンソールで次を実行してみる。

```
import folium
```

```
m = folium.Map(location=[34.4461, 133.2315], zoom_start=12, tiles='Stamen Terrain')
```

```
m.save("a2.html")
```



```
Python 3.6.1 |Anaconda custom (64-bit)| (default, May 11 2017, 13:25:24) [MSC v.1900 64 bit (AMD64)]
Type "copyright", "credits" or "license" for more information.

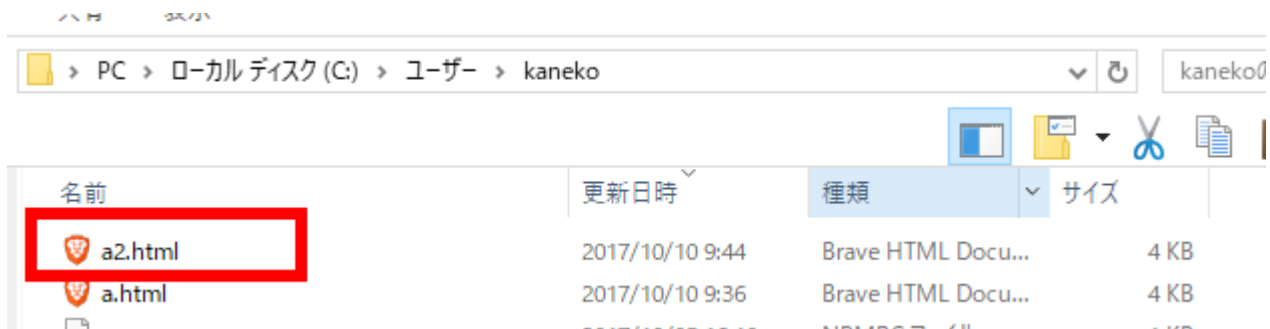
IPython 5.3.0 -- An enhanced Interactive Python.
?                -> Introduction and overview of IPython's features.
%quickref        -> Quick reference.
help             -> Python's own help system.
object?          -> Details about 'object', use 'object??' for extra details.

In [1]: pwd()
Out[1]: 'C:\\Users\\kaneko'

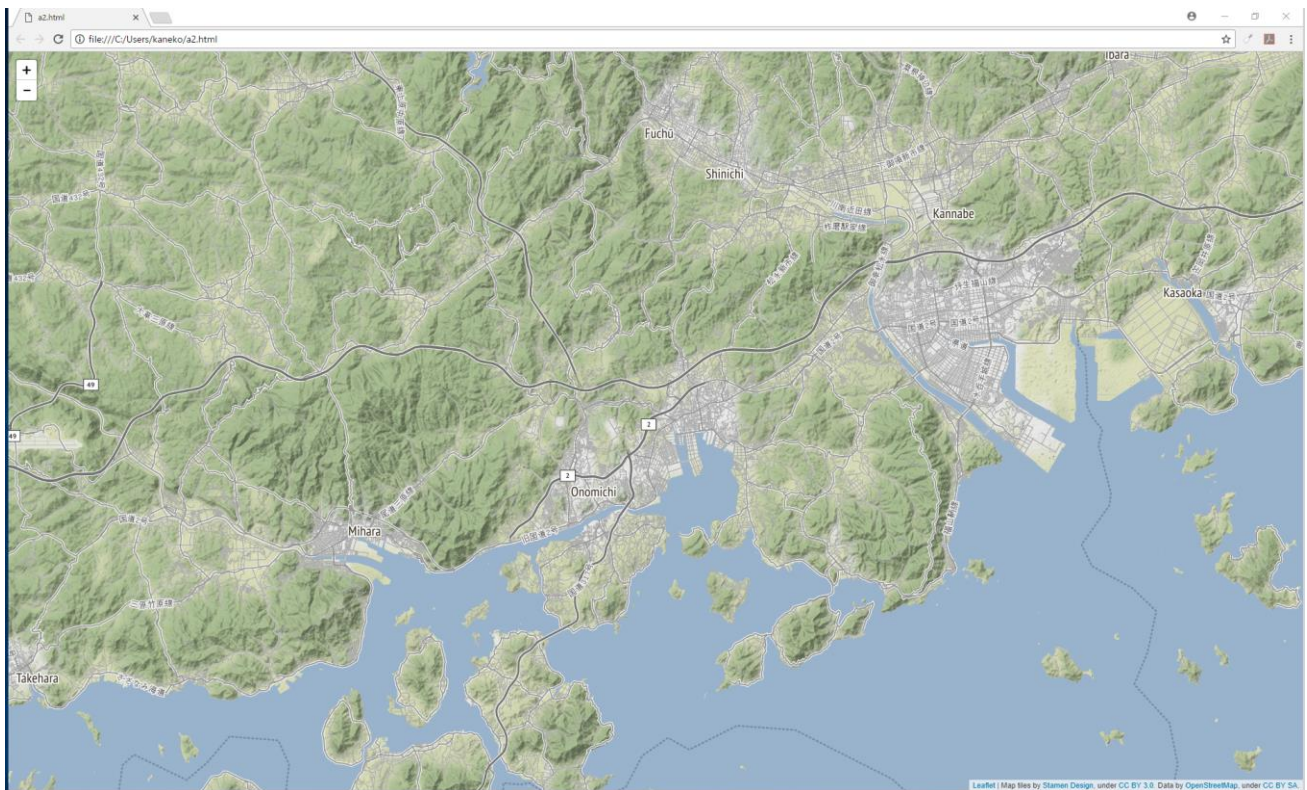
In [2]: import folium
...: m = folium.Map(location=[34.4461, 133.2315], zoom_start=12)
...: m.save("a.html")
...:

In [3]: import folium
...: m = folium.Map(location=[34.4461, 133.2315], zoom_start=12, tiles='Stamen Terrain')
...: m.save("a2.html")
...:
```

a2.html ができる



a2.html をウェブブラウザで開いてみる



② Stamen Toner

spyder のコンソールで次を実行してみる。

```
import folium
m = folium.Map(location=[34.4461, 133.2315], zoom_start=12, tiles='Stamen Toner')
m.save("a3.html")
```

```
IPythonコンソール
コンソール I/A

Python 3.6.1 |Anaconda custom (64-bit)| (default, May 11 2017, 13:25:24) [MSC v.1900 64 bit (AMD64)]
Type "copyright", "credits" or "license()" for more information.

IPython 5.3.0 -- An enhanced Interactive Python.
?                -> Introduction and overview of IPython's features.
%quickref        -> Quick reference.
help             -> Python's own help system.
object?         -> Details about 'object', use 'object??' for extra details.

In [1]: pwd()
Out[1]: 'C:\\Users\\kaneko'

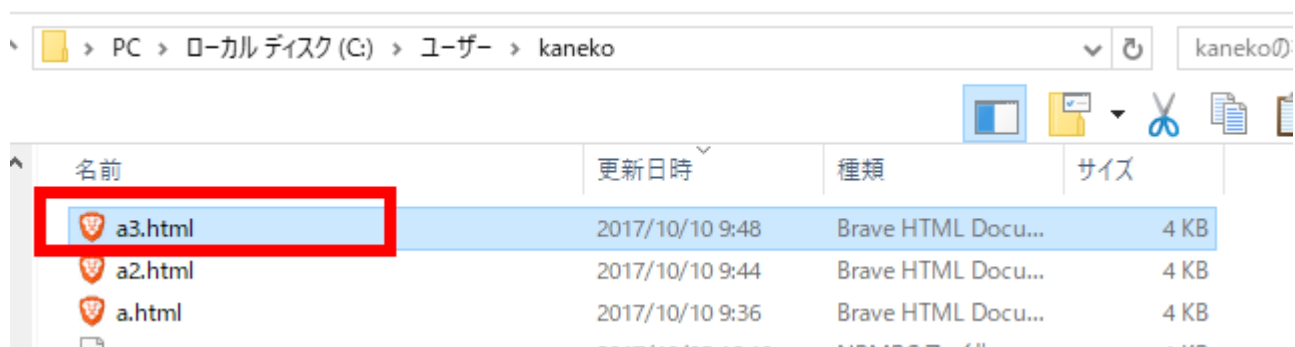
In [2]: import folium
...: m = folium.Map(location=[34.4461, 133.2315], zoom_start=12)
...: m.save("a.html")
...:

In [3]: import folium
...: m = folium.Map(location=[34.4461, 133.2315], zoom_start=12, tiles='Stamen Terrain')
...: m.save("a2.html")
...:

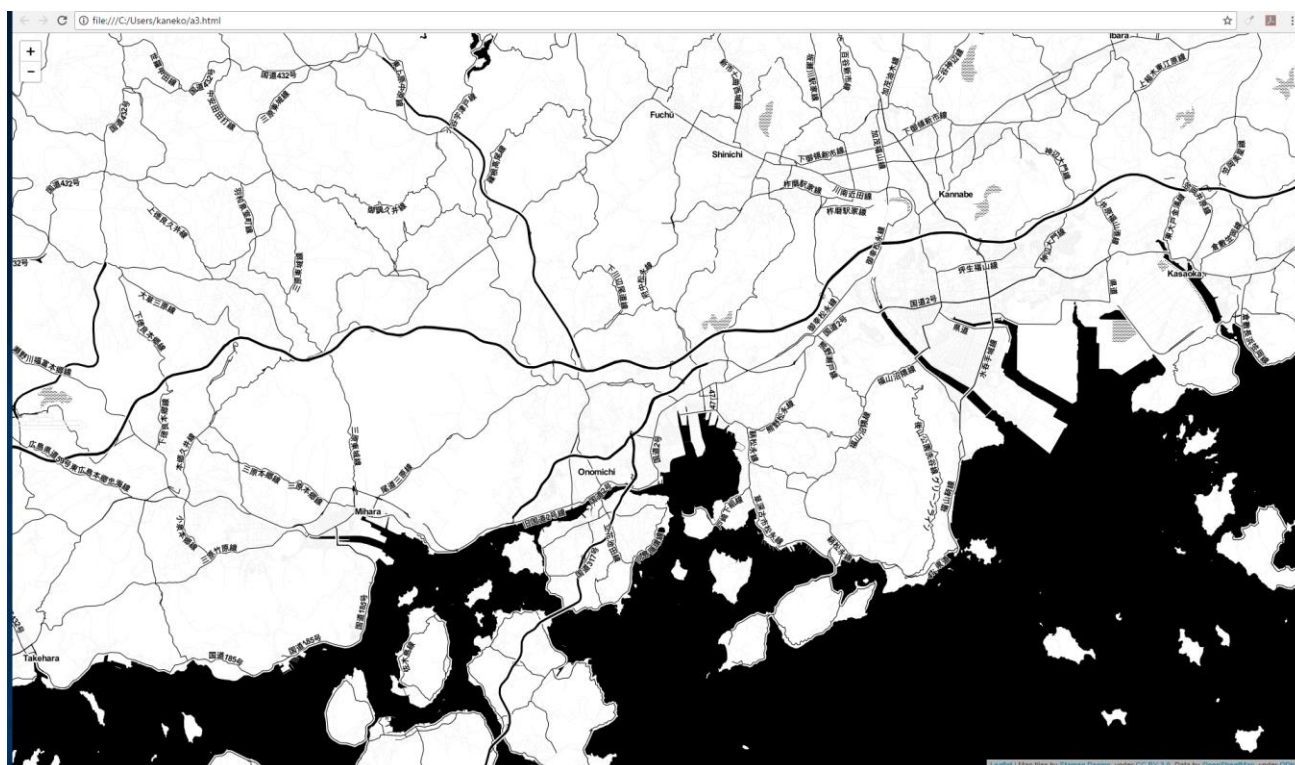
In [4]: import folium
...: m = folium.Map(location=[34.4461, 133.2315], zoom_start=12, tiles='Stamen Toner')
...: m.save("a3.html")
...:

In [5]: |
```

a3.html ができる



a3.html をウェブブラウザで開いてみる



◇ マーカー

<https://media.readthedocs.org/pdf/folium/latest/folium.pdf>

に記載のサンプルプログラム（下に引用）を, **spyder のコンソール**で動かしてみる

```
map_1 = folium.Map(location=[45.372, -121.6972],
zoom_start=12,
tiles='Stamen Terrain')
folium.Marker([45.3288, -121.6625], popup='Mt. Hood Meadows').add_to(map_1)
folium.Marker([45.3311, -121.7113], popup='Timberline Lodge').add_to(map_1)
map_1
```

```
In [5]: map_1 = folium.Map(location=[45.372, -121.6972],
...: zoom_start=12,
...: tiles='Stamen Terrain')
...: folium.Marker([45.3288, -121.6625], popup='Mt. Hood Meadows').add_to(map_1)
...: folium.Marker([45.3311, -121.7113], popup='Timberline Lodge').add_to(map_1)
...: map_1
...:
Out[5]: <folium.folium.Map at 0x201b9e734a8>

In [6]:
```

引き続いて, 次を実行

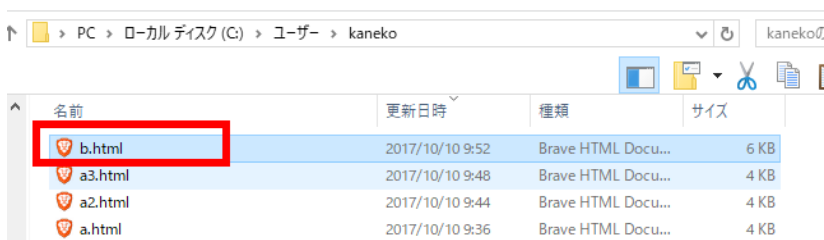
map_1.save("b.html")

```
In [5]: map_1 = folium.Map(location=[45.372, -121.6972],
...: zoom_start=12,
...: tiles='Stamen Terrain')
...: folium.Marker([45.3288, -121.6625], popup='Mt. Hood Meadows').add_to(map_1)
...: folium.Marker([45.3311, -121.7113], popup='Timberline Lodge').add_to(map_1)
...: map_1
...:
Out[5]: <folium.folium.Map at 0x201b9e734a8>

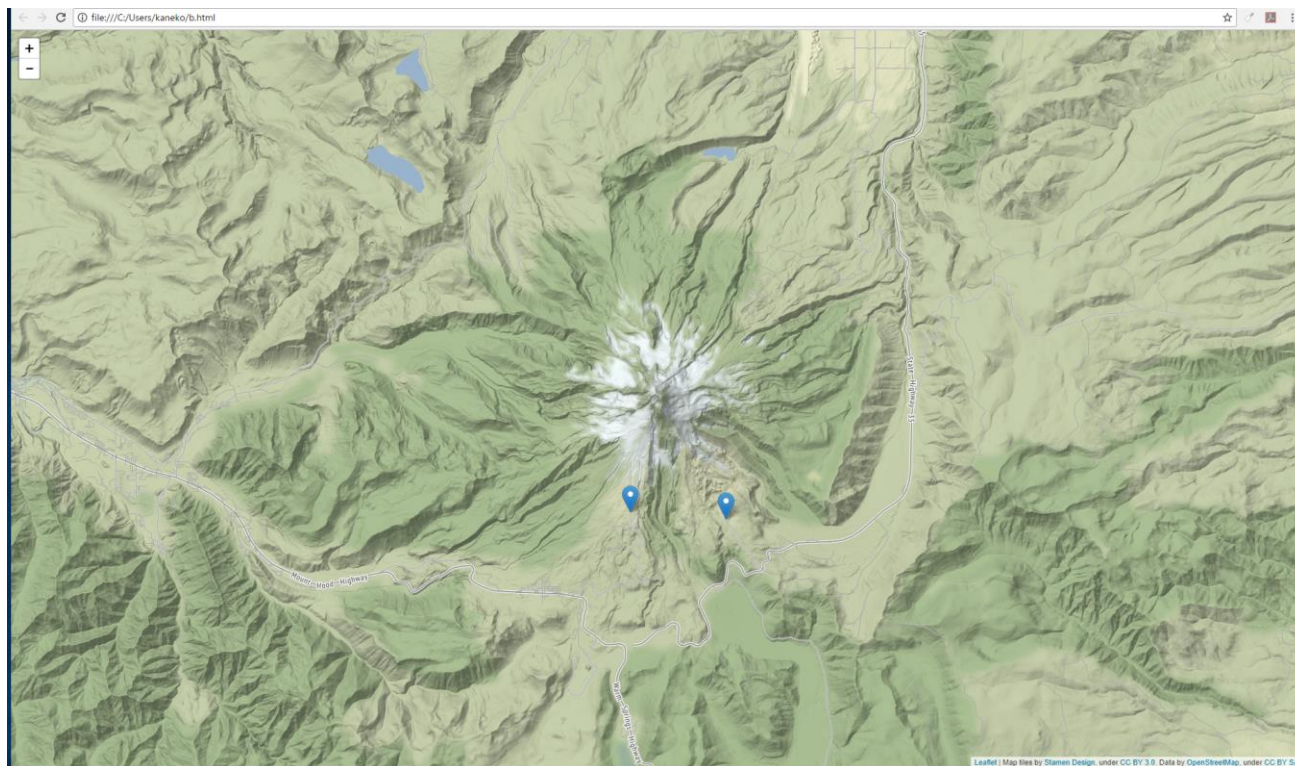
In [6]: map_1.save("b.html")

In [7]: |
```

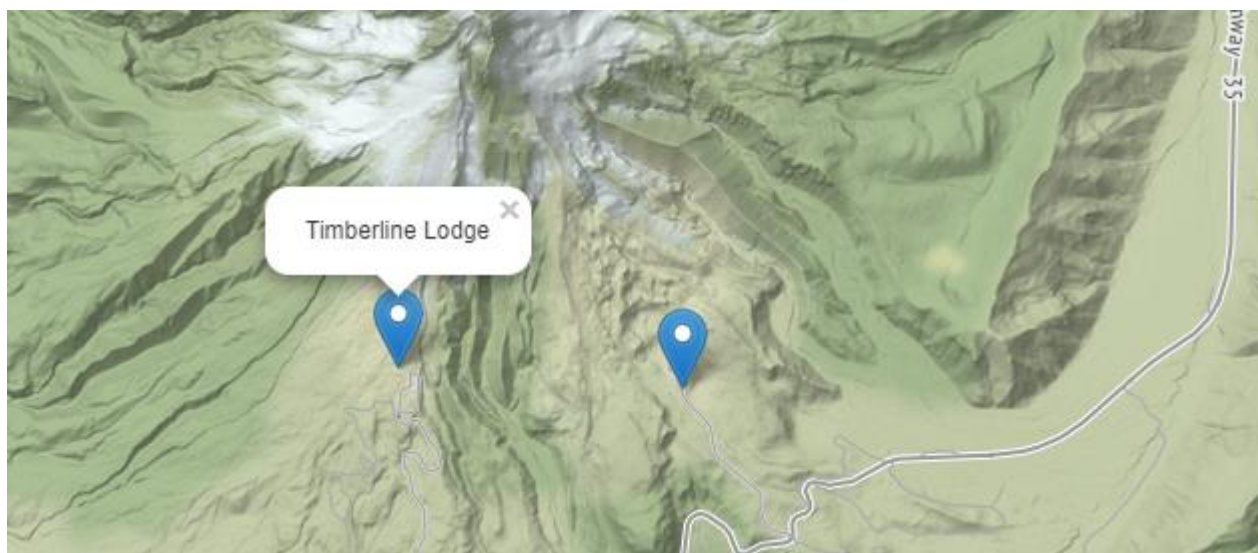
b.html ができる



できたファイル b.html をウェブブラウザで開いてみる。 **マーカーが2個**あることを確認。



マーカーをクリックするとポップアップ表示が出る



<まとめ>

- ・ マーカーは, プログラムを使って, **自動で配置**できる
- ・ マーカーをクリックすると**ポップアップ表示**, のようなことも簡単にできる.

◆ OpenStreetMap の地図データのダウンロード

ライン： 線分や折れ線

ノード： 点（緯度経度）

Python のプログラムを使い、OpenStreetMap 地図データをダウンロード

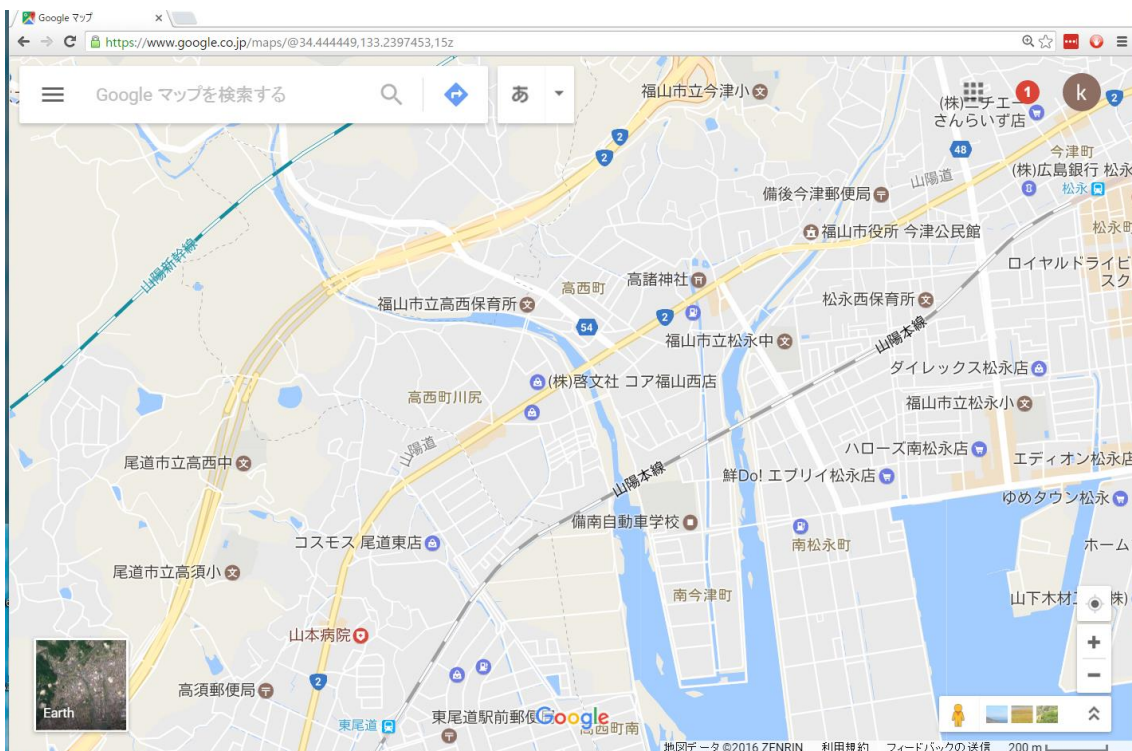
① Google Map を使って緯度・経度を調べてみる

緯度・経度はどこでもよい。（あとで使う）.

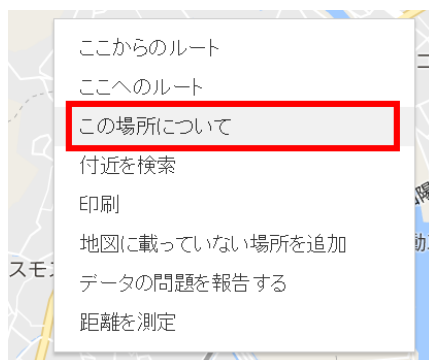
（１）まず Google Map を開く

<https://www.google.co.jp/maps>

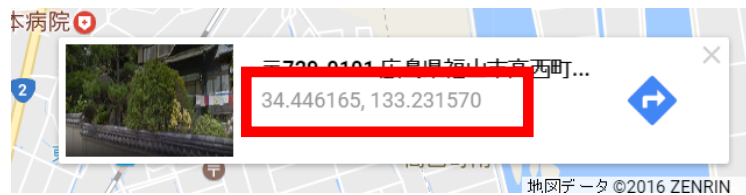
（２）Google Map で好きな場所に移動する



（３）右クリックして、「この場所について」を選ぶ



(4) 緯度, 経度が表示されるので, メモしておく



② spyder のコンソールで, 緯度と経度を設定

※さきほど調べた緯度と経度

(実行例)

lat = **34.4461**

lon = **133.2315**

```
In [52]: lat = 34.4461
...: lon = 133.2315
...:
```

③ 欲しいデータのタイプの設定

OpenStreetMap の「ライン」のデータのうち, 「highway」であるようなデータが欲しいとする.

OpenStreetMap には, ポイントやラインやエリアには, 「highway」のようなデータが付いている.

次のように実行

type = **'highway'**

```
In [53]: type = 'highway'
```

④ lat, lon, type の設定が終わったところで, **spyder のコンソール**で, 次のプログラムを実行し, **データのダウンロードと表示**を行う.

※ OpenStreetMap もタイル地図だが, 次のプログラムでは, 最下位のレイヤ (もっとも詳細なレイヤ) からダウンロードされる


```

import overpy
api = overpy.Overpass()
result = api.query("""
    way(%f, %f, %f, %f) [%s];
    (._>);
    out body;
    """ % (lat - 0.01, lon - 0.01, lat + 0.01, lon + 0.01, type) )

```

```

for way in result.ways:
    print("Name: %s" % way.tags.get("name", "n/a"))
    print("  Highway: %s" % way.tags.get("highway", "n/a"))
    print("  Nodes:")
    for node in way.nodes:
        print("    Lat: %f, Lon: %f" % (node.lat, node.lon))

```

```

In [4]: import overpy
...: api = overpy.Overpass()
...: result = api.query("""
...:     way(%f, %f, %f, %f) [%s];
...:     (._>);
...:     out body;
...:     """ % (lat - 0.01, lon - 0.01, lat + 0.01, lon + 0.01, type) )
...:
...: for way in result.ways:
...:     print("Name: %s" % way.tags.get("name", "n/a"))
...:     print("  Highway: %s" % way.tags.get("highway", "n/a"))
...:     print("  Nodes:")
...:     for node in way.nodes:
...:         print("    Lat: %f, Lon: %f" % (node.lat, node.lon))
...:
Name: n/a
Highway: motorway_link
Nodes:
  Lat: 34.454723, Lon: 133.244376
  Lat: 34.454571, Lon: 133.244554
  Lat: 34.454385, Lon: 133.244715
  Lat: 34.454225, Lon: 133.244780
  Lat: 34.454061, Lon: 133.244792
  Lat: 34.453887, Lon: 133.244768

```

表示を確認。(道路がないようなところを緯度経度で指定していると表示は空になる)。

道路のデータである。緯度経度が並んでいる。


```

  Lat: 34.437859, Lon: 133.221665
  Lat: 34.437902, Lon: 133.221733
  Lat: 34.438003, Lon: 133.221932
  Lat: 34.438244, Lon: 133.222129
  Lat: 34.438286, Lon: 133.222158
Name: n/a
  Highway: service
  Nodes:
    Lat: 34.455673, Lon: 133.230545
    Lat: 34.455523, Lon: 133.230355
    Lat: 34.455427, Lon: 133.230202
    Lat: 34.455460, Lon: 133.229356
Name: n/a
  Highway: service
  Nodes:
    Lat: 34.456395, Lon: 133.229164
    Lat: 34.456007, Lon: 133.229084
Name: 尾道福山自動車道(国道2号松永道路)
  Highway: motorway
  Nodes:
    Lat: 34.452081, Lon: 133.241236
    Lat: 34.451651, Lon: 133.240666
    Lat: 34.450738, Lon: 133.239259
    Lat: 34.450044, Lon: 133.237732
    Lat: 34.449549, Lon: 133.236213
Name: 尾道福山自動車道(国道2号松永道路)
  Highway: motorway
  Nodes:
    Lat: 34.435400, Lon: 133.223180
    Lat: 34.436358, Lon: 133.224048
    Lat: 34.437556, Lon: 133.224908
    Lat: 34.438587, Lon: 133.225484
    Lat: 34.439311, Lon: 133.225837
    Lat: 34.440041, Lon: 133.226122
    Lat: 34.440976, Lon: 133.226428
    Lat: 34.441911, Lon: 133.226688
Name: 尾道福山自動車道(国道2号松永道路)
  Highway: motorway
  Nodes:
    Lat: 34.441869, Lon: 133.227017
    Lat: 34.439991, Lon: 133.226316
    Lat: 34.438519, Lon: 133.225641
    Lat: 34.437544, Lon: 133.225080
    Lat: 34.436934, Lon: 133.224661
    Lat: 34.436343, Lon: 133.224216
    Lat: 34.435358, Lon: 133.223341

In [11]:
```

⑤ 「ライン」のデータの確認

ラインが線分のとき： 端のポイントが2個

ラインが折れ線のとき： 端のポイントが2個で、途中のポイントがある

spyder のコンソールで次を実行

result.ways

```
In [11]: result.ways
Out[11]:
<overpy.way id=38133366 nodes=[448806426, 448806434, 448806436, 1296938301, 448806439, 1296938309, 448806447, 448806449, 448806452, 448806455, 448806458, 448806460, 448806462, 448806465, 1296938230, 448806467]>,
<overpy.way id=59777925 nodes=[741996432, 3906196137, 3906196155, 3906196160, 3906196169, 741996467, 741996471, 3906196184, 741996515, 741996409]>,
<overpy.way id=59777929 nodes=[741996515, 3906196190, 3906196189, 3139493453, 741996460]>,
<overpy.way id=60213187 nodes=[1425799298, 262075713, 1425799318, 1374332103]>,
<overpy.way id=60213188 nodes=[262075710, 1425799086]>,
<overpy.way id=60213189 nodes=[1425799323, 1425799314, 2007461575, 448807597, 1425799292]>,
<overpy.way id=62102411 nodes=[741990697, 3139493456, 1374438074, 744443549, 1374438054, 1374438048, 3139493446, 1374438032, 3139493437, 1374437990, 774443550, 774443552, 774443546, 774443541, 774443536, 774443554, 774443548, 774443543, 1297855054, 1298832993, 1297855133, 774443528, 774443558, 774443545, 1297855059, 1374437610, 1374437604, 1374437593, 1374437584, 1374437581, 1374437571, 1296923539, 774443540, 774443535, 1298154277, 1374437415, 774443529, 3906865526, 1297855085, 1297855061, 1297855097, 774443560, 1298142809, 774443556, 774443551, 1374437259, 1374381921]>,
<overpy.way id=62102420 nodes=[774443704, 774443729]>,
<overpy.way id=62102421 nodes=[687052062, 774443723, 774443704, 3906865853, 774443718]>,
<overpy.way id=62102450 nodes=[774444219, 774444198, 1374438096, 774444221, 3906865814, 774444205, 1374438082, 3139493456]>,
<overpy.way id=62102451 nodes=[1374438081, 3914056430, 3914056428, 3914056427, 3914056429, 774444192, 774444198]>,
<overpy.way id=62102452 nodes=[1297855067, 1267979765, 1425799037, 1425799038, 1425799042, 1425799043, 1425799047, 774444218, 1425799050, 1374437929, 1374437972, 1374437999, 1374438028, 774444215, 1425799059, 3139493439, 3139493451, 3906865802, 1296951782]>,
<overpy.way id=62102453 nodes=[687052062, 1296951559, 3139493448, 1374438114, 774444196, 1374438093, 774444194, 774444188, 774444216, 774444208, 774444201, 774444203, 774444192, 3906865795, 3906865780, 774444197, 3132692925]>,
<overpy.way id=71465411 nodes=[1374332207, 3912501406, 1374332233, 1374332253, 3912501440, 850347707, 850347681, 3912501582, 850347651, 1374332482, 3912501633, 850347577, 1377255897, 3912501733, 850347688, 3912501875, 1302922105, 1296938265, 850347631, 850347602, 850347589, 850347569, 1302922147, 1302922036, 850347695, 850347674, 850347644, 850347619, 850347579, 850347553, 850347689, 850347662, 850347633, 1302922153, 850347603, 850347590, 1302909952, 850347571, 850347697, 850347677, 850347646, 850347621, 850347596, 3910417560, 1377256083, 850347583, 850347555, 1377256195, 850347664, 839667531]>,
<overpy.way id=86641022 nodes=[1374332335, 850347599, 3912501560, 1161394548, 1374332366, 3912501574, 1374332378, 3912501583, 1161395061, 3912501585, 3912501592, 1161394775, 1374332447, 1161394539, 1161395051, 1374332544, 3912501674, 1161394590, 1374332665, 1161395108, 1161394895, 1161394663, 1374332950, 3912815763, 1161395025, 3912815765, 1161394790, 1374332947, 1161395079, 1161394866, 1374332914, 1374332911, 1161394407, 1161394847, 3913347065, 1161394619, 1161395138, 1161394930, 1161394689]>,
<overpy.way id=114502956 nodes=[687052062, 774444200, 1296951633]>,
<overpy.way id=114725938 nodes=[1297855087, 1298833034, 1298833061, 1298833015, 1298833092, 1298833002, 1298833008, 1298832982, 1298833048, 1298833036, 1298833097, 1298833069, 1298833013, 1298832987, 1298833044, 1298833030, 1298833103, 1298833046, 1298833047, 1298832989, 1298833071, 1298833006, 1298833004, 1298833077, 1298832997]>,
<overpy.way id=114725943 nodes=[1298833094, 1298833081, 1298833063, 1298833101, 1298833083, 1298833026, 1298833000, 1298832995, 1298832977, 1298833085, 1298833087, 1298833091, 1298833089, 1298833019, 1298833017, 1298833068, 1298833028, 1298833099, 1298833002, 774443552]>,
<overpy.way id=114725947 nodes=[1298833048, 1298833099, 1298833015]>,
<overpy.way id=114725948 nodes=[1298833013, 1298833028, 1298833061]>,
<overpy.way id=115244624 nodes=[741996477, 1425799381, 741996548, 3906866119, 3906866118, 741996428, 741996442, 741996487, 741996483, 741996546, 741996391, 741996438, 3906866100, 3906866097, 3906866093, 1302926822, 741996519, 741996398, 741996536, 741996418, 3906866069, 741996492, 741996549, 741996396, 3906866075, 741996446, 741996440, 741996485, 741996497, 741996394, 741996400, 741996456, 1302922079, 3912501456, 741996466, 1425799371, 3912501449, 1425799370, 741996522, 741996550, 3912501435, 741996434, 3912501404, 1425799338, 1425799307, 850347623, 741996516, 3912501327, 741996411, 3906865938]>,
<overpy.way id=123161341 nodes=[448807593, 3912501593, 3912501572, 1374332207, 2007461584, 850347575]>,
<overpy.way id=123161366 nodes=[1374332914, 1374332904, 1374332892, 1374332884, 1374332861, 1374332837, 1374332767, 1374332736, 1374332725, 1374332705, 1374332696, 1374332688, 1374332687, 1374332690, 1374332701, 1374332719, 1374332753, 1374332783, 1374332820, 1374332913, 1374332937, 1374332950]>,
<overpy.way id=123161397 nodes=[1374330781, 1374330807, 1374330815, 1374330820, 1374330847, 1374330863, 1374330890, 1374330913, 1374330924, 1374330955]>,
<overpy.way id=123161400 nodes=[1374331004, 1374331002, 1374330994, 1374330985, 1374330921]>,
<overpy.way id=123161405 nodes=[1374330955, 1374330992, 1374331013, 3906196180, 1374331030, 1374331050]>,
<overpy.way id=123161409 nodes=[1374330432, 1374330523, 1374330542, 1374330561, 1374330614, 1374330674, 1374330737]>,
<overpy.way id=123161418 nodes=[1374330870, 1374330892, 1374330895, 1374330899, 1374330902, 1374330903, 1374330918, 1374330936]>,
<overpy.way id=123161421 nodes=[1374331108, 1374331177, 1374331200, 1374331226, 1374331246, 1374331269, 1374331291, 1374331330, 1374331352, 1374331415, 1374331433]>,
```

表示の1行： 1個の「ライン」オブジェクト

各行の nodes = [] の中身： 端のポイントや途中のポイントのID（点番号）

⑥ 「ノード」のデータの確認

spyder のコンソールで次を実行

result.nodes

```
コンソール I/O
<overpy.Node id=1374331025 lat=34.4422180 lon=133.2211940>,
<overpy.Node id=1374331026 lat=34.4422190 lon=133.2264430>,
<overpy.Node id=1374331030 lat=34.4421339 lon=133.2222686>,
<overpy.Node id=1374331031 lat=34.4422980 lon=133.2229480>,
<overpy.Node id=1374331032 lat=34.4423030 lon=133.2263140>,
<overpy.Node id=1374331034 lat=34.4423150 lon=133.2286450>,
<overpy.Node id=1374331038 lat=34.4423930 lon=133.2280460>,
<overpy.Node id=1374331039 lat=34.4423990 lon=133.2281010>,
<overpy.Node id=1374331041 lat=34.4424040 lon=133.2262390>,
<overpy.Node id=1374331042 lat=34.4424100 lon=133.2280120>,
<overpy.Node id=1374331043 lat=34.4423723 lon=133.2227349>,
<overpy.Node id=1374331044 lat=34.4424270 lon=133.2286580>,
<overpy.Node id=1374331047 lat=34.4424500 lon=133.2262250>,
<overpy.Node id=1374331050 lat=34.4424110 lon=133.2226317>,
<overpy.Node id=1374331054 lat=34.4424720 lon=133.2279440>,
<overpy.Node id=1374331056 lat=34.4424820 lon=133.2212010>,
<overpy.Node id=1374331057 lat=34.4424890 lon=133.2282710>,
<overpy.Node id=1374331058 lat=34.4424950 lon=133.2286240>,
<overpy.Node id=1374331059 lat=34.4425120 lon=133.2279310>,
<overpy.Node id=1374331063 lat=34.4425450 lon=133.2283390>,
<overpy.Node id=1374331064 lat=34.4425910 lon=133.2284130>,
<overpy.Node id=1374331065 lat=34.4425960 lon=133.2262250>,
<overpy.Node id=1374331066 lat=34.4425960 lon=133.2285430>,
<overpy.Node id=1374331076 lat=34.4427310 lon=133.2262590>,
<overpy.Node id=1374331078 lat=34.4427530 lon=133.2212340>,
<overpy.Node id=1374331084 lat=34.4427070 lon=133.2228174>,
<overpy.Node id=1374331086 lat=34.4428830 lon=133.2263140>,
<overpy.Node id=1374331089 lat=34.4429167 lon=133.2228870>,
<overpy.Node id=1374331095 lat=34.4429840 lon=133.2263750>,
<overpy.Node id=1374331103 lat=34.4429769 lon=133.2213485>,
<overpy.Node id=1374331108 lat=34.4431146 lon=133.2213032>,
<overpy.Node id=1374331114 lat=34.4431073 lon=133.2231022>,
<overpy.Node id=1374331123 lat=34.4432777 lon=133.2215717>,
<overpy.Node id=1374331124 lat=34.4432500 lon=133.2266260>,
<overpy.Node id=1374331130 lat=34.4433829 lon=133.2321868>,
<overpy.Node id=1374331132 lat=34.4433077 lon=133.2230418>,
<overpy.Node id=1374331140 lat=34.4433090 lon=133.2227273>,
<overpy.Node id=1374331152 lat=34.4432930 lon=133.2231511>,
<overpy.Node id=1374331160 lat=34.4433781 lon=133.2234223>,
<overpy.Node id=1374331176 lat=34.4435140 lon=133.2268230>,
<overpy.Node id=1374331177 lat=34.4435370 lon=133.2210100>,
<overpy.Node id=1374331200 lat=34.4436820 lon=133.2209560>,
<overpy.Node id=1374331226 lat=34.4439200 lon=133.2209350>,
<overpy.Node id=1374331246 lat=34.4440550 lon=133.2209420>,
<overpy.Node id=1374331251 lat=34.4440720 lon=133.2271830>,
<overpy.Node id=1374331254 lat=34.4439773 lon=133.2215449>,
<overpy.Node id=1374331262 lat=34.4441170 lon=133.2272780>,
<overpy.Node id=1374331265 lat=34.4441330 lon=133.2274550>,
<overpy.Node id=1374331269 lat=34.4441570 lon=133.2209560>,
<overpy.Node id=1374331274 lat=34.4441900 lon=133.2275440>.
```

表示の**1行**： 1個の「**ポイント**」オブジェクト

各行の**中身**： **ID（点番号）**，**緯度**，**経度**

まとめ

- ・**ライン**のデータ： **ポイントのID（点番号）**が**2個以降並んでいる**
- ・**ポイント**のデータ： **ID（点番号）**，**緯度**，**経度**