29/03/2017

Datasheet

Map_generator.py, V1.0

python

Etienne BOURDEAU
LABORATOIRE DE PHYSIQUE CORPUSCULAIRE, AUBIERE

Summary:

Program Introduction

This program was created by Mr. BOURDEAU Etienne, on internship in Aubière's the Particle Physics Laboratory. It was coded in Python 3.6, and responds to the demand of the laboratory, which needed of a way to print on a base map the information gathered by Decodeb64.py (for more information about this program, please read the related datasheet).

Map_generator.py relies on folium, an open-source leaflet application to Python. You can download Folium from their GitHub: https://github.com/python-visualization/folium

If you are using an IDE to use this program, like PyCharm or Visual Studio Code, PyCharm can be downloaded automatically.

Program Analysis

Imports

`import folium` ➔ Folium is the only module to be imported here. It uses leaflet.js (see Sources for info about leaflet.js) and applies it with Python commands. It allows to create interactive maps with a lot of options, though this program only uses basic markers. For more information about Folium, please see Sources).

Map Creation

Using Folium allows you to create a local map in html, using OpenStreetMaps.
Creating a map with Folium takes these 2 lines:

1- First, create a variable corresponding to GPS coordinates you want to focus your map on.

```
Var1 = (YourLatitude, YourLongitude)
```

2- Then, create another variable with the value:

```
Var2 = folium.Map(location=Abovevar, zoom_start=17)
```

Now your map is registered in Python memory, though yet it won't create an html map. We will see at the end of the program how to save your map, since we first need to apply markers on the map.

Global Variables

```
file = './out_file.txt' # Name of file to read
LAT = 'Latitude'
LNG = 'Longitude'
TMP = 'Temperature'
LSNR = 'Signal to Noise Ratio'
RSSI = 'Received Signal Strength Indicator (dBm)'
DAT = 'Date'
```

Here, we set the variables we will need to use in the main program. First, the path of the file to read, then all patterns you will want to extract information from in the file.

Main Program

Environment variables setting

```
with open(file, 'r') as f:
    lat = None
    lng = None
    tmp = None  # Initializing all values to False, so you can check their presence later.
    rssi = None
    dat = None
    lsnr = None
```

We first need to open the previously set file (see 'Global Variables'). To call it again more simply, we will nickname it 'f'.

Then, we're setting every environment variable no 'None', which is the same thing than 'False', so we can check if we find them later.

Variable presence checking

```
for line in f:
    kv = line.split(' = ')  # Creates a table, splitting columns from '=': 1st element is the key, 2nd is the value.
    if kv[0] == TMP:  # Checks if each line contains the wanted pattern (patterns are set in Global Variables above.
        value = kv[1].strip('°C\n')
        tmp = float(value)
    elif kv[0] == LAT:
        value = kv[1].strip('°\n')
        lat = float(value)
    elif kv[0] == LNG:
        value = kv[1].strip('°\n')
        lng = float(value)
    elif kv[0] == LSNR:
        value = kv[1].strip('\n')
        lsnr = float(value)
    elif kv[0] == RSSI:
        value = kv[1].strip('\n')
        rssi = int(value)
    elif kv[0] == DAT:
        dat = kv[1].strip('\n')
```

This loop is long, but quite simple:

1 – First, we set a loop 'for line in f:', which means we are going to check the file line by line (Reminder: line is a Python variable which automatically values one line of a file)

2 – Then, we are going to split the file. The out_file generated by 'Decodeb64.py' is always of the form 'a key' '=' 'a value'. Splitting the file from the character '=' will create a table, here named 'kv' (for key value) whose first column (kv[0]) will be the key, and the second column (kv[1]) will be the value.

3 – For each value you want to add to your map, you will have 3 lines:

```
If/elif kv[0] == YourGlobVar:
    value = kv[1].strip('charactersafterthevalue')
    lat = float(value)
```

The 'strip' part allows not to take in count characters after the value you want to take, like '°C'.

Map Saving

Now, your map is registered and your markers are set. You only need this line to save it as an html file:

```
Var2.save('./nameofyourmap.html')
```

Sources

Folium

Leaflet.js official page:
http://leafletjs.com/

Introduction to Folium (French):
http://www.portailsig.org/content/python-leaflet-folium-ou-comment-creer-des-cartes-interactives-simplement

Folium Official Page:
https://pypi.python.org/pypi/folium