

# Acquisiamo informazioni sulle condizioni metereologiche con Python

## Il progetto “Weather Stations”

Nel 2016 la Raspberry foundation ha inviato un migliaio di mini stazioni metereologiche ad alcune scuole sparse per il mondo. Molte di queste stazioni continuano a pubblicare in tempo più o meno reale le informazioni acquisite e sono interrogabili tramite URL.

Per esempio per vedere la lista completa delle stazioni disponibili si può utilizzare il seguente URL:

<https://apex.oracle.com/pls/apex/raspberrypi/weatherstation/getallstations>

Questo URL però non ritorna una pagina vera e propria come quelle che siamo abituati a vedere, bensì una serie di dati in formato JSON che nascono per essere interpretati da un software.

## Otteniamo la lista di tutte le stazioni disponibili

Vediamo quindi un primo programma Python che acquisire la lista delle stazioni. Per fare questo utilizzeremo la libreria `requests`, che permette di inviare richieste HTTP in modo molto semplice. La libreria va installata nel PC usato per lo sviluppo tramite il package manager di Python `pip`:

```
python3 -m pip install -U requests
```

Ecco quindi il codice, da scrivere in un file di nome `fetch_stations.py`:

```
#!/usr/bin/env python3
from requests import get
import json
from pprint import pprint

url = 'https://apex.oracle.com/pls/apex/raspberrypi/weatherstation/getallstations'

stations = get(url).json()['items']
pprint(stations)
```

Come si può vedere questo programma stampa una serie di lista di stazioni e per ogni stazione stampa le seguenti informazioni:

- `weather_stn_id`: un numero che identifica univocamente la stazione
- `weather_stn_lat`: la latitudine del punto sul globo terrestre dove si trova la stazione
- `weather_stn_long`: la longitudine dello stesso punto
- `weather_stn_name`: una stringa che identifica la stazione (per esempio il nome della scuola)

## Leggiamo i dati da una stazione

E possibile interrogare una stazione tramite il suo ID. In questo caso l'URL da utilizzare è il seguente:

[https://apex.oracle.com/pls/apex/raspberrypi/weatherstation/getlatestmeasurements/<station\\_id>](https://apex.oracle.com/pls/apex/raspberrypi/weatherstation/getlatestmeasurements/<station_id>)

Dove `<station_id>` va sostituito con uno degli identificativi stampati dal programma precedente. Per esempio:

<https://apex.oracle.com/pls/apex/raspberrypi/weatherstation/getlatestmeasurements/511059>

Dopo aver provato sul browser, scriviamo il secondo programma che fa la stessa cosa. Salviamolo in un nuovo file che chiamiamo `fetch_weather.py`.

```
#!/usr/bin/env python3
from requests import get
import json
from pprint import pprint

ws_base_url='https://apex.oracle.com/pls/apex/raspberrypi/weatherstation'
station_id=511059

url=f"{ws_base_url}/getlatestmeasurements/{station_id}"

weather = get(url).json()['items']
pprint(weather)
```

## Posizione di un punto sul globo terrestre

Supponiamo di voler prelevare le informazioni dalla stazione che si trova alla minima distanza da noi.

Per prima cosa dobbiamo determinare la nostra posizione, ma per fare questo occorre prima sapere come si rappresenta la nostra posizione sul globo terrestre. Abbiamo visto che per rappresentare la posizione delle stazioni sono utilizzati due numeri detti **latitudine** e **longitudine**. Questi numeri detti anche coordinate geografiche rappresentano univocamente un punto sulla superficie terrestre.

Per scoprire le coordinate geografiche del luogo dove ci troviamo possiamo per esempio utilizzare Open-StreetMaps al link <https://www.openstreetmap.org/>.

Scegliete il punto sulla mappa e cliccando con il pulsante destro scegliere la voce “Show address”.

Per esempio le coordinate di Impact Hub sono le seguenti:

- Latitudine: 43.798135
- Longitudine: 11.238411

## Determiniamo la stazione alla minima distanza dalla nostra posizione

Per ottenere la stazione alla minima distanza da noi dobbiamo quindi calcolare la distanza sul globo terrestre tra il punto corrisponde alle coordinate geografiche di tutte le stazioni e il punto dove ci troviamo noi e determinare il minimo.

Ma come facciamo a calcolare questa distanza? Il calcolo è abbastanza complicato e utilizza la cosiddetta formula di Haversine. Noi ci semplifichiamo la vita utilizzando una libreria Python che, oltre a tantissime altre cose, fa anche questo: **GeoPy**.

Per installarla sul nostro PC utilizziamo il solito package manager **pip**:

```
python3 -m pip install -U geopy
```

Scriviamo quindi il codice che segue in un file di nome `search_nearest.py`.

```
#!/usr/bin/env python3
from requests import get
import sys
import json
import geopy.distance
from pprint import pprint

my_lat = 43.798135
```

```

my_lon = 11.238411

ws_base_url='https://apex.oracle.com/pls/apex/raspberrypi/weatherstation'
get_station_url = f"{ws_base_url}/getallstations"

stations = get(get_station_url).json()['items']

min_distance = sys.float_info.max
nearest_station_id = -1
nearest_station_lat = 9999
nearest_station_lon = 9999

for station in stations:
    dist = geopy.distance.distance(
        (station['weather_stn_lat'], station['weather_stn_long']),
        (my_lat, my_lon))
    if dist < min_distance:
        min_distance = dist
        nearest_station_id = station['weather_stn_id']
        nearest_station_lat = station['weather_stn_lat']
        nearest_station_lon = station['weather_stn_long']

print(f"Nearest station found at {min_distance} with ID = {nearest_station_id}")
print(f"Latitude = {nearest_station_lat} - Longitude = {nearest_station_lon}")
print("\nWeather data:")
fetch_weather_url = f"{ws_base_url}/getlatestmeasurements/{nearest_station_id}"
weather = get(fetch_weather_url).json()['items']
pprint(weather)

```

Il programma stamperà quindi le informazioni sulla stazione più vicina e successivamente gli ultimi dati raccolti.

## Approfondimenti

- La libreria Requests:
  - <https://requests.readthedocs.io/en/master/>
- Una guida per imparare a usare la libreria Requests:
  - <https://realpython.com/python-requests/>
- Representational State Transfer, ossia come interrogare il web via software:
  - [https://it.wikipedia.org/wiki/Representational\\_State\\_Transfer](https://it.wikipedia.org/wiki/Representational_State_Transfer)
- Il formato JSON, ossia il formato con cui i computer si scambiano dati sul web:
  - <https://www.json.org/json-en.html>
- Latitudine e longitudine:
  - <https://www.youmath.it/domande-a-risposte/view/6594-latitudine-longitudine.html>
- La formula di Haversine:
  - [https://en.wikipedia.org/wiki/Haversine\\_formula](https://en.wikipedia.org/wiki/Haversine_formula)
- GeoPy:
  - <https://geopy.readthedocs.io/en/stable/>

Infine, una versione molto più Cool di questo tutorial:

<https://projects.raspberrypi.org/en/projects/fetching-the-weather/>