Serialized Weather

Our Weather server currently formats the data and sends the display text to the client.
In this exercise we'll change the server to send the raw data, and leave the display formatting code for the client.
The entire communication protocol will be based on passing JSON messages between the client and server, which will make it much easier to extend later.

Instructions

1. Use attached Weather server/client files `serializedweather_server.py` and `serializedweather_client.py`.
2. Assume each recv() call returns an entire message from the other host, and that this message is a complete JSON.
3. Implement the protocol messages as follows

Message #1: Client -> Server, weather request

For example, when the user requests weather data for 'London', **the client program will send the server** the following JSON: `{"request": "weather", "city": "London"}`

Message #2: Server -> Client, weather response

1. In case of an unsupported request (e.g. `{"request": "abcd"}`; basically a `request` value different from `weather`), server should send back the following JSON: `{"result": "unsupported_request"}`

2. In case of a `weather` request for a known city, the server should send back the following JSON: `{"result": "success", "city": "London", "temperature": 18, "humidity": 60, "description": "Partly cloudy"}`.
   Client program output for this example (notice that "London" is the user's input):

```
Enter the name of a city:
London
== Weather information for London ==
Temperature: 18°C
Humidity: 60%
Description: Partly cloudy
```

3. In case of a `weather` request for an unknown city (for example "springfield"), the server should send the following JSON: `{"result": "unknown_city", "city": "springfield"}`.
   Client program output for this example (notice that "springfield" is the user's input):

```
Enter the name of a city: springfield
** Server has no data on springfield **
```

Notice

Follow the instructions carefully!
Even small typos aren't acceptable, such as sending "temprature" instead of "temperature".
The protocol must be precisely as described.
The client output must be precisely as described.

Hint

You will need to use the JSON python library to parse the data, useful links:

- JSON Library Documentation
- w3school Python JSON Tutorial

To submit

Submit files `serializedweather_server.py` and `serializedweather_client.py`.