Python Exercises

Dictionary

1. Go back to ex_if_else_bool exercise 2, implement the same logic, but using dict instead of if/elif/else. Test the program to make sure it works as it should.
2. Make a program that:
    a. Take an input by the user in console, using the format “key”;”value”. Hence an input with the two strings key and value split using “;”.
    b. Split the input on “;” and add the key value pair in a dictionary.
    c. Encapsulate the logic in a while loop, such that the program will ask for new inputs, until the input “revert” is given
    d. If a user write “revert” as input, first print the dictionary at hand, then add the revert key value pair in a new dictionary.
    eksample {‘hello’:’world’}, should be reverted to {‘world’: ‘hello’}.
    and print the new dictionary.
3. Write a Python script to concatenate following dictionaries to create a new one
    a. Sample Dictionary :
    dic1={1:10, 2:20}
    dic2={3:30, 4:40}
    dic3={5:50,6:60}

Expected Result : {1: 10, 2: 20, 3: 30, 4: 40, 5: 50, 6: 60}

b. Write a function called “unfold_dict” that takes a single dictionary as input. And return an unfolded version of the dictionary

example:

```
input_dict = {
   ‘a’:{‘d’:1, ‘e’:2},
   ‘b’:{‘f’:3},
   ‘c’:{‘g’:4, ‘h’:5}
}
```

Expected Result : {‘ad’:1, ‘ae’:2, ‘bf’:3, ‘cg’:4, ‘ch’:5}

4. Write a function called “set_by_path(input_dict, path, value)”, which takes an input dictionary, a path in list format (of any length), and a value, and returns an updated dictionary.

   Example:

   set_by_path({‘a’:1}, [‘b’], 2) -> {‘a’:1, ‘b’:2}

   set_by_path({‘a’:1}, [‘a’,’b’], 2) -> {‘a’:{‘b’:2}}

   set_by_path({‘a’:1}, [‘b’,’c’,’d’], 2) -> {‘a’:1, ‘b’:{‘c’:{‘d’2}}}

5. Extra challenge:

   Go to:

   https://mixedanalytics.com/blog/list-actually-free-open-no-auth-needed-apis/

   and find some suitable free API, and make a mini-project out of the information you can get from there.

   I suggest you go for the 7timer, read the documentation to

understand how to call, and how to use the output.
A possible 7timer project is to call the API for you location every 3 hours, and compare the temperature predictions for you location and last prediction and calculate the difference (is it predicting hotter or colder weather than previous).
This gives the possibilities to check the hypotheses if it is possible to use this information to help predicting the weather based on the weather forecasting over time.
Here is a starter code to get you started:

```
import requests
import json

url = r'http://www.7timer.info/bin/api.pl?lon=10.750&lat=59.925&unit=Metric&produ
ct=civil&output=json'

response = requests.get(url)

# test if the respons was ok
assert response.status_code==200

print(json.dumps(response.json()))
```