Solutions for finding the differences between the two JSON files in Python

Table of Contents

Objective of this document **2**

Version History **2**

Problem Statement **2**

Approach 1 - Inefficient way **3**

Solution for Approach 1 **3**

Approach 2 - Efficient way **5**

Solution to Approach 2 **5**

Why Approach 2 is Efficient **7**

Bonus - Creating JSON files for testing using Python **7**

1. **Objective of the document**

The document contains the different solutions for finding the differences between the two JSON files in Python programming language.The document explains the efficient techniques to be used to solve such problems.

1. **Version History**

|  |  |  |  |
| --- | --- | --- | --- |
| Ver. | Description | Updated By | Updated Date |
| 0.1 | Initial version | Mukul Ramchandani | 11th Feb 2021 |
| 0.2 | Added Bonus Content | Mukul Ramchandani | 12th Feb 2021 |
|  |  |  |  |
|  |  |  |  |

# **Problem Statement**

Write a Python code that should be able to find the differences between the JSON files and write the differences in a CSV file.

The code should take care of following cases :

**Case #1 :** The elements that are modified in the second file w.r.t first file should be printed in the CSV.

**Case #2 :** The elements that exist only in the first file or only in the second file should also be printed in the CSV.

1. **Approach 1 -** Inefficient Way

Below points explains the pseudocode for the first approach :

1. Load the JSON files using json module

2. Create three lists - ***differenceElements*** *,* ***file1Users*** *,* ***file2Users***.

3. Create a function that takes two lists as parameters(***file1Users****,****file2Users***) and returns a list that contains elements which are either in the first file or second file.

4. Write a code that will populate all the unique identifiers of first file into ***file1Users*** and of second file into ***file2Users***.

5. Loop through the elements of the first file and write a nested loop that will loop through the elements of the second file.

6. Check for the particular identifier that elements are equal or not and if they are not equal, then find the difference between them and append it to ***differenceElements.***

7. After differences are populated, write a code that will call a function that we defined (refer step 3) and save the return value in some variable.

8. Loop through the elements of both the files and append the elements in the difference that are either in first file or in second file.

9. Create a list of unique keys that ***dict*** has.

10. Write the created ***list*** of ***dictionaries*** into a CSV file using the csv module of python.

1. **Solution to Approach 1**

Refer Link or Zoom-In into the picture attached.

[**Solution\_1.py**](https://github.com/mukulramchandani/learning-python/blob/main/solution_1.py)

****

1. **Approach 2 - Efficient Way**

Below points explains the pseudocode for second approach :

1. Load the JSON files using json module

2. Create four sets() - two for storing the dictionaries of lists in the form of tuples and other two for storing unique identifiers of both json files.

3. Add each element of the first file to the first set() in the form of tuple and add each element of the second file to the second set(). In the same loop also add the sets() where we need to add unique identifiers.

4. Find the difference between second set() and first set() and store in some variable.

5.Create a list - ***differenceElements***

6. Loop through all the elements of difference set() and convert tuple to dictionary and append that to ***differenceElements*** list.

7. Find the symmetric\_difference and store in some variable

8. Loop through the elements of both the files and append the elements in the difference that are either in first file or in second file.

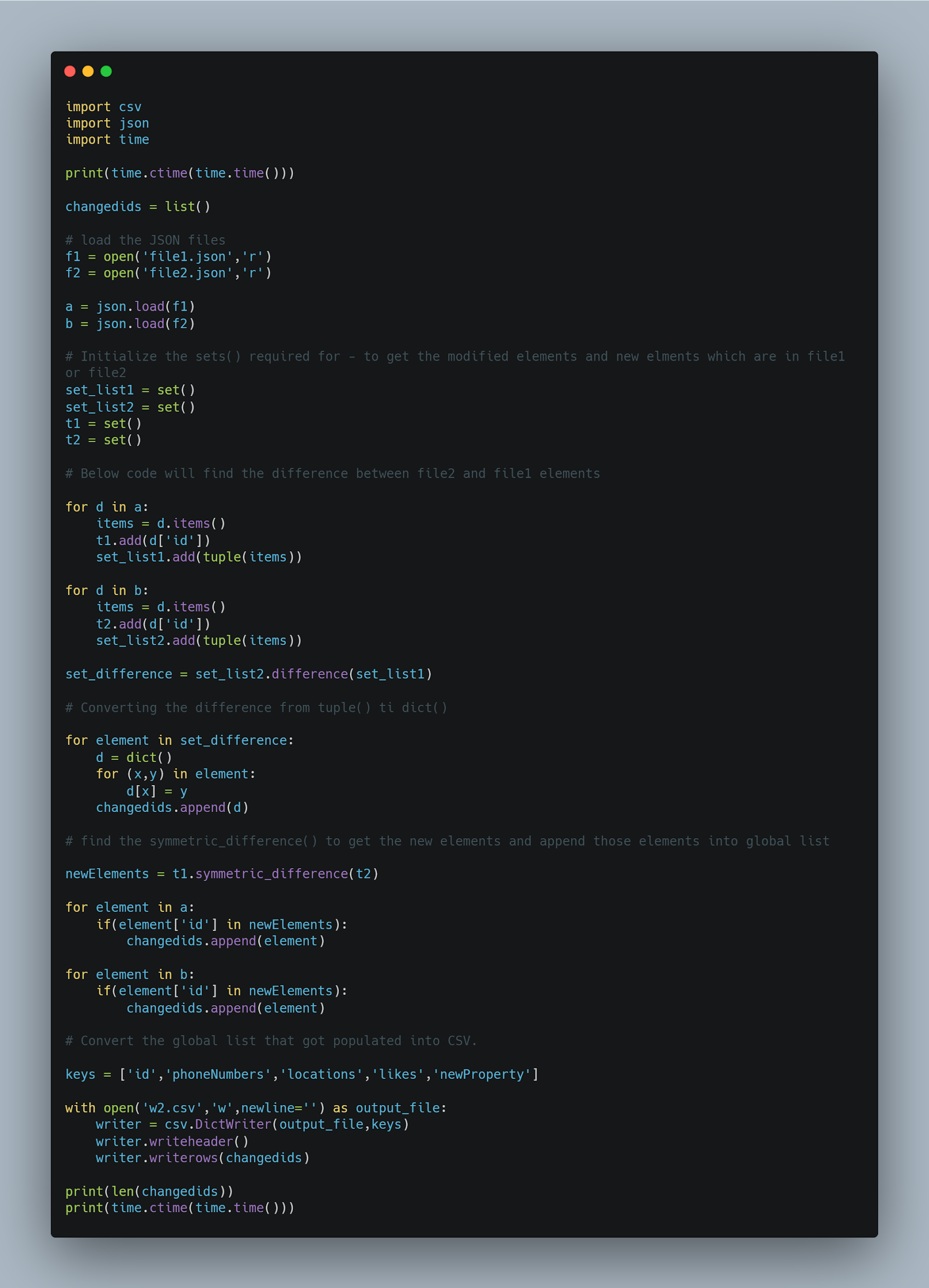
9. Create a list of unique keys that ***dict*** has.

10. Write the created ***list*** of ***dictionaries*** into a CSV file using the csv module of python.

1. **Solution of Approach 2**

Refer Link or Zoom-In into the picture attached.

[**Solution\_2.py**](https://github.com/mukulramchandani/learning-python/blob/main/solution_2.py)

****

1. **Why Approach 2 is efficient :**

1**.** It uses sets and tuples which are more efficient than lists and dictionaries in python. Operations performed on sets are faster in python.

2. It does not use nested loops.

1. **Bonus - Creating JSON files for testing :**

To create the two JSON files needed for testing, refer to [**this link**](https://github.com/mukulramchandani/learning-python/blob/main/createF1.py) or code below.

In the code you can change range value in loop to increase and decrease the users for both the files.

