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.

2. 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

3. 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.

4. **Approach 1 -** Inefficient Way

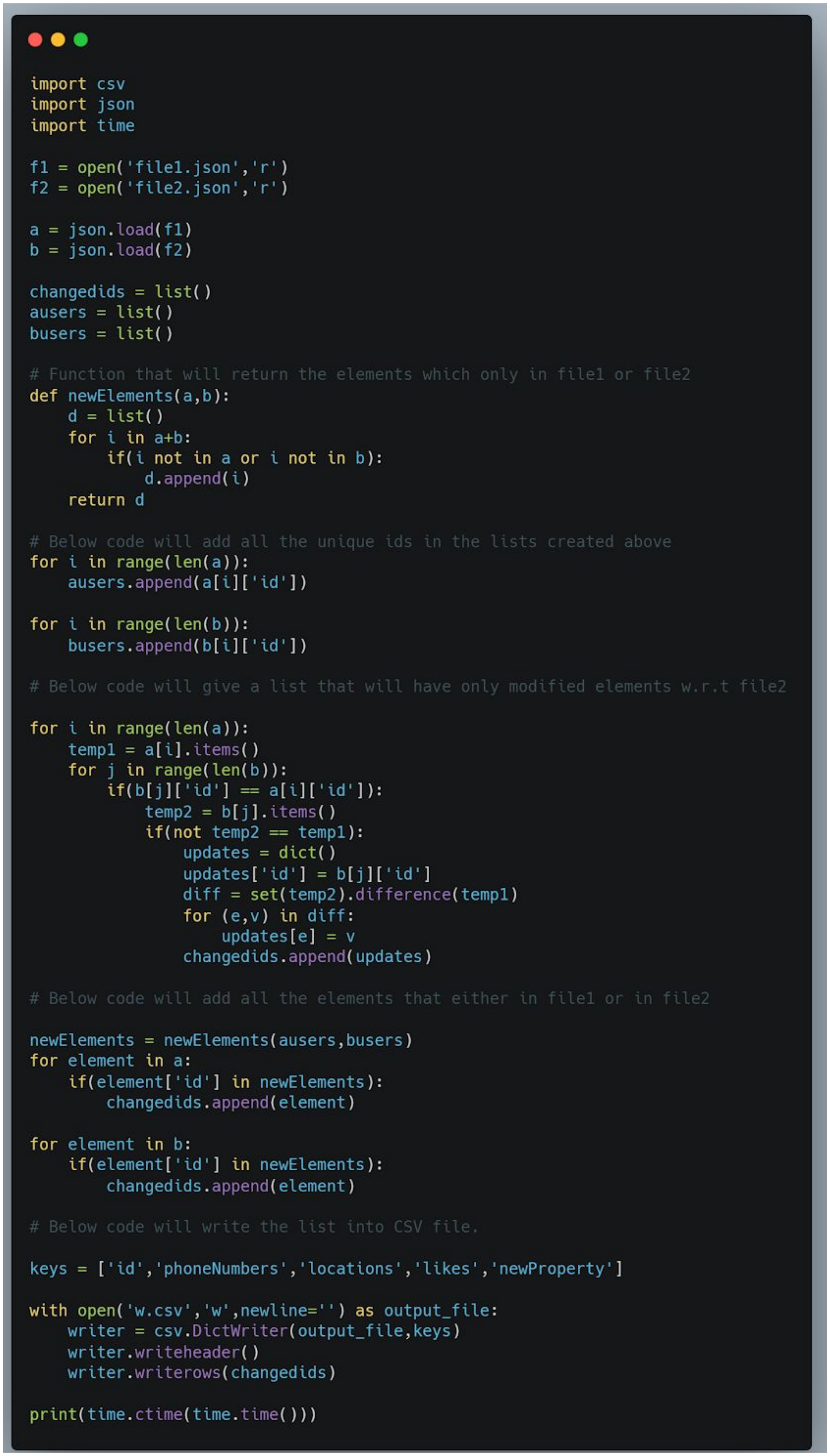
Below points explains the pseudocode for the first approach :

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.

5. Solution to Approach 1

Refer Link or Zoom-In into the picture attached.

Solution_1.py



```python
import csv
import json
import time

f1 = open('file1.json','r')
f2 = open('file2.json','r')

a = json.load(f1)
b = json.load(f2)

changedids = list()
ausers = list()
busers = list()

# Function that will return the elements which only in file1 or file2
def newElements(a,b):
    d = list()
    for i in a+b:
        if(i not in a or i not in b):
            d.append(i)
    return d

# Below code will add all the unique ids in the lists created above
for i in range(len(a)):
    ausers.append(a[i]['id'])

for i in range(len(b)):
    busers.append(b[i]['id'])

# Below code will give a list that will have only modified elements w.r.t file2

for i in range(len(a)):
    temp1 = a[i].items()
    for j in range(len(b)):
        if(b[j]['id'] == a[i]['id']):
            temp2 = b[j].items()
            if(not temp2 == temp1):
                updates = dict()
                updates['id'] = b[j]['id']
                diff = set(temp2).difference(temp1)
                for (e,v) in diff:
                    updates[e] = v
                changedids.append(updates)

# Below code will add all the elements that either in file1 or in file2

newElements = newElements(ausers,busers)
for element in a:
    if(element['id'] in newElements):
        changedids.append(element)

for element in b:
    if(element['id'] in newElements):
        changedids.append(element)

# Below code will write the list into CSV file.

keys = ['id','phoneNumbers','locations','likes','newProperty']

with open('w.csv','w',newline='') as output_file:
    writer = csv.DictWriter(output_file,keys)
    writer.writeheader()
    writer.writerows(changedids)

print(time.ctime(time.time()))
```

6. Approach 2 - Efficient Way

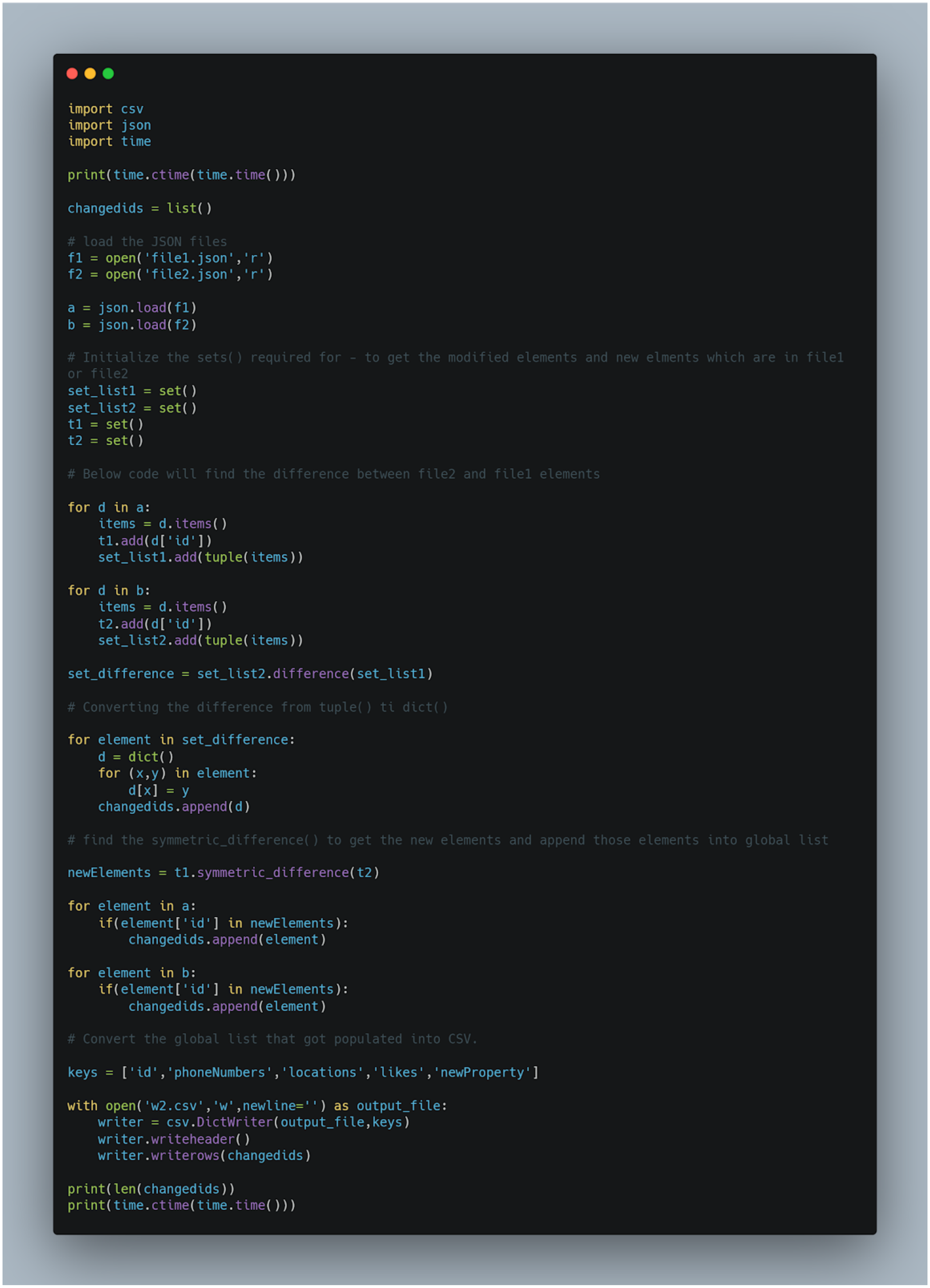
Below points explains the pseudocode for second approach :

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.

7. Solution of Approach 2

Refer Link or Zoom-In into the picture attached.

Solution_2.py



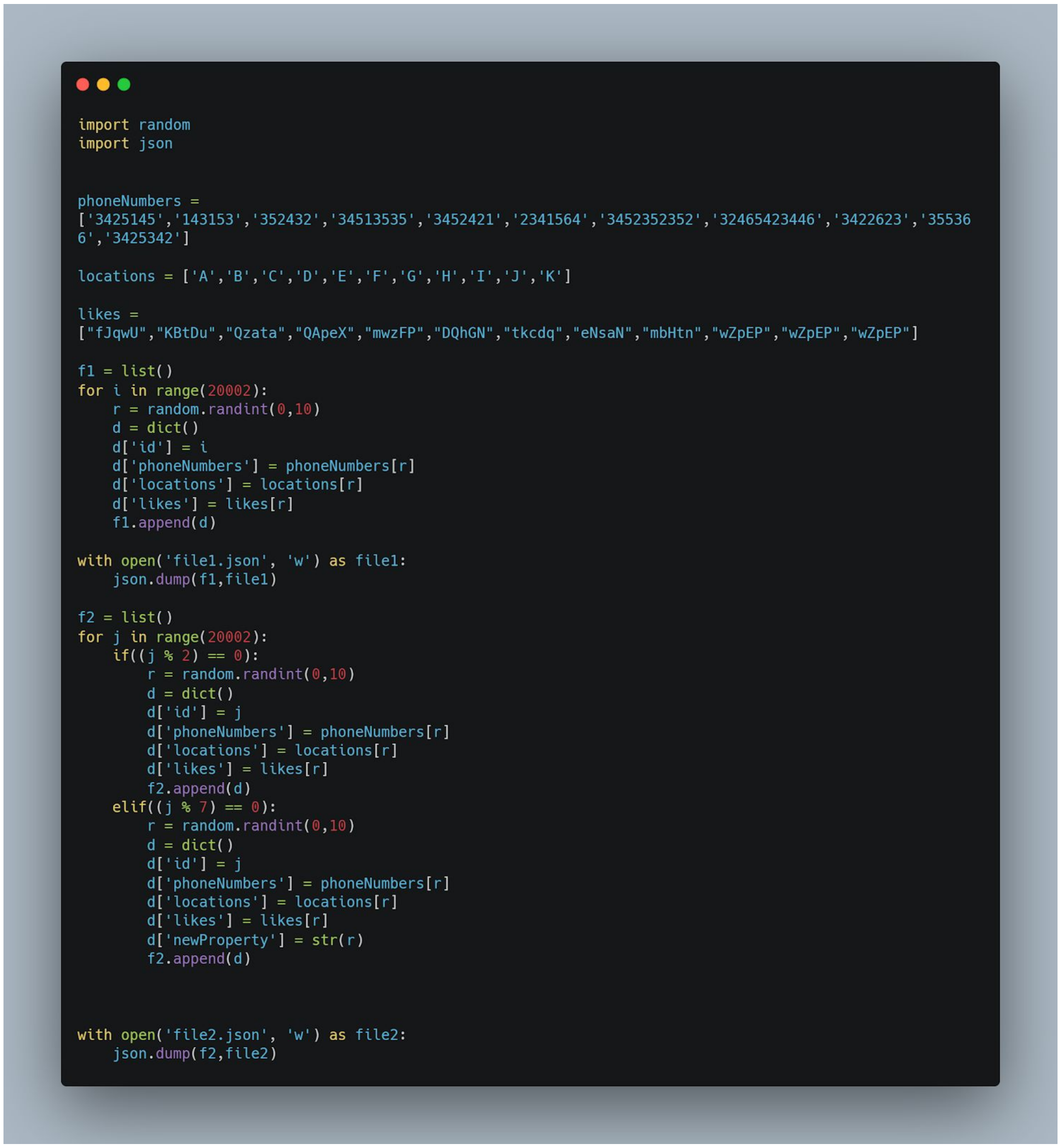
```python
import csv
import json
import time

print(time.ctime(time.time()))

changedids = list()

# load the JSON files
f1 = open('file1.json','r')
f2 = open('file2.json','r')

a = json.load(f1)
b = json.load(f2)

# Initialize the sets() required for - to get the modified elements and new elments which are in file1
or file2
set_list1 = set()
set_list2 = set()
t1 = set()
t2 = set()

# Below code will find the difference between file2 and file1 elements

for d in a:
    items = d.items()
    t1.add(d['id'])
    set_list1.add(tuple(items))

for d in b:
    items = d.items()
    t2.add(d['id'])
    set_list2.add(tuple(items))

set_difference = set_list2.difference(set_list1)

# Converting the difference from tuple() ti dict()

for element in set_difference:
    d = dict()
    for (x,y) in element:
        d[x] = y
    changedids.append(d)

# find the symmetric_difference() to get the new elements and append those elements into global list

newElements = t1.symmetric_difference(t2)

for element in a:
    if(element['id'] in newElements):
        changedids.append(element)

for element in b:
    if(element['id'] in newElements):
        changedids.append(element)

# Convert the global list that got populated into CSV.

keys = ['id','phoneNumbers','locations','likes','newProperty']

with open('w2.csv','w',newline='') as output_file:
    writer = csv.DictWriter(output_file,keys)
    writer.writeheader()
    writer.writerows(changedids)

print(len(changedids))
print(time.ctime(time.time()))
```

8. 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.

9. Bonus - Creating JSON files for testing :

To create the two JSON files needed for testing, refer to [this link] or code below. In the code you can change range value in loop to increase and decrease the users for both the files.



```
import random
import json


phoneNumbers =
['3425145','143153','352432','34513535','3452421','2341564','3452352352','32465423446','3422623','35536
6','3425342']

locations = ['A','B','C','D','E','F','G','H','I','J','K']

likes =
["fJqwU","KBtDu","Qzata","QApeX","mwzFP","DQhGN","tkcdq","eNsaN","mbHtn","wZpEP","wZpEP","wZpEP"]

f1 = list()
for i in range(20002):
    r = random.randint(0,10)
    d = dict()
    d['id'] = i
    d['phoneNumbers'] = phoneNumbers[r]
    d['locations'] = locations[r]
    d['likes'] = likes[r]
    f1.append(d)

with open('file1.json', 'w') as file1:
    json.dump(f1,file1)

f2 = list()
for j in range(20002):
    if((j % 2) == 0):
        r = random.randint(0,10)
        d = dict()
        d['id'] = j
        d['phoneNumbers'] = phoneNumbers[r]
        d['locations'] = locations[r]
        d['likes'] = likes[r]
        f2.append(d)
    elif((j % 7) == 0):
        r = random.randint(0,10)
        d = dict()
        d['id'] = j
        d['phoneNumbers'] = phoneNumbers[r]
        d['locations'] = locations[r]
        d['likes'] = likes[r]
        d['newProperty'] = str(r)
        f2.append(d)



with open('file2.json', 'w') as file2:
    json.dump(f2,file2)
```