1. Python | Convert a list of Tuples into Dictionary

In [1]:
```
def Convert(tup, di):
    for a, b in tup:
        di.setdefault(a, []).append(b)
    return di

tups = [("akash", 10), ("gaurav", 12), ("anand", 14),
     ("suraj", 20), ("akhil", 25), ("ashish", 30)]
dictionary = {}
print (Convert(tups, dictionary))
```

```
{'akash': [10], 'gaurav': [12], 'anand': [14], 'suraj': [20], 'akhil': [25], 'ashish': [30]}
```

2.Python counter and dictionary intersection example (Make a string using deletion and rearrangement)

In [2]:
```
from collections import Counter

def makeString(str1,str2):

    dict1 = Counter(str1)
    dict2 = Counter(str2)

    result = dict1 & dict2

    return result == dict1

if __name__ == "__main__":
    str1 = 'ABHISHEKsinGH'
    str2 = 'gfhfBHkooIHnfndSHEKsiAnG'
    if (makeString(str1,str2)==True):
        print("Possible")
    else:
        print("Not Possible")
```

```
Possible
```

3.Python dictionary, set and counter to check if frequencies can become same

In [5]:
```
from collections import Counter

def allSame(input):

    dict=Counter(input)

    same = list(set(dict.values()))

    if len(same)>2:
        print('No')
    elif len (same)==2 and same[1]-same[0]>1:
        print('No')
    else:
        print('Yes')

if __name__ == "__main__":
    input = 'xxxyyzzt'
    allSame(input)
```

```
No
```

4. Scraping And Finding Ordered Words In A Dictionary using Python

In [15]:
```
import requests
def scrapeWords():

  scrape_url = "https://raw.githubusercontent.com/dwyl/english-words/master/words_alpha.txt"
  scrapeData = requests.get(scrape_url)
  listofwords = scrapeData.content
  listofwords = listofwords.decode("utf-8").split()
  return listofwords
def isOrdered():
  collection = scrapeWords()
  collection = collection[:100]
  word = ''
  for word in collection:
    result = 'Word is ordered'
    i = 0
    l = len(word) - 1
    if (len(word) < 3):
      continue
    while i < l:
      if (ord(word[i]) > ord(word[i+1])):
        result = 'Word is not ordered'
        break
      else:
        i += 1
    if (result == 'Word is ordered'):
      print(word,': ',result)
if __name__ == '__main__':
  isOrdered()
```

```
aaa :  Word is ordered
aah :  Word is ordered
aahs :  Word is ordered
aal :  Word is ordered
aals :  Word is ordered
aam :  Word is ordered
aaru :  Word is ordered
aas :  Word is ordered
```

5. Possible Words using given characters in Python

In [16]:
```
def charCount(word):
    dict = {}
    for i in word:
        dict[i] = dict.get(i, 0) + 1
    return dict


def possible_words(lwords, charSet):
    for word in lwords:
        flag = 1
        chars = charCount(word)
        for key in chars:
            if key not in charSet:
                flag = 0
            else:
                if charSet.count(key) != chars[key]:
                    flag = 0
        if flag == 1:
            print(word)

if __name__ == "__main__":
    input = ['goo', 'bat', 'me', 'eat', 'goal', 'boy', 'run']
    charSet = ['e', 'o', 'b', 'a', 'm', 'g', 'l']
    possible_words(input, charSet)
```

```
me
goal
```

6. Python – Keys associated with Values in Dictionary

In [18]:
```
test_dict = {'god' : [4, 5], 'is' : [8], 'best' : [10, 12]}

print("The original dictionary : " + str(test_dict))

val_list = [5, 10]

temp = {}
for key, vals in test_dict.items():
    for val in vals:
        temp[val] = key
res = [temp[ele] for ele in val_list]

print("The keys mapped to " + str(val_list) + " are : " + str(res))
```

```
The original dictionary : {'god': [4, 5], 'is': [8], 'best': [10, 12]}
The keys mapped to [5, 10] are : ['god', 'best']
```

7. Python program to Find the size of a Tuple

In [20]:
```
import sys

Tuple1 = ("A", 1, "B", 2, "C", 3)
Tuple2 = ("Geti", "Raju", "Get2", "Nikhil", "Get3", "Deepanshu")
Tuple3 = ((1, "Lion"), ( 2, "Tiger"), (3, "Fox"), (4, "Wolf"))

print("Size of Tuple1: " + str(sys.getsizeof(Tuple1)) + "bytes")
print("Size of Tuple2: " + str(sys.getsizeof(Tuple2)) + "bytes")
print("Size of Tuple3: " + str(sys.getsizeof(Tuple3)) + "bytes")
```

```
Size of Tuple1: 96bytes
Size of Tuple2: 96bytes
Size of Tuple3: 80bytes
```

8. Python – Maximum and Minimum K elements in Tuple

In [21]:
```
test_tup = (5, 20, 3, 7, 6, 8)

print("The original tuple is : " + str(test_tup))

K = 2

res = []
test_tup = list(sorted(test_tup))

for idx, val in enumerate(test_tup):
    if idx < K or idx >= len(test_tup) - K:
        res.append(val)
res = tuple(res)

print("The extracted values : " + str(res))
```

```
The original tuple is : (5, 20, 3, 7, 6, 8)
The extracted values : (3, 5, 8, 20)
```

9. Create a list of tuples from given list having number and its cube in each tuple

In [22]:
```
list1 = [1, 2, 5, 6]

res = [(val, pow(val, 3)) for val in list1]

print(res)
```

```
[(1, 1), (2, 8), (5, 125), (6, 216)]
```

10. Python – Adding Tuple to List and vice – versa

In [23]:
```
test_list = [5, 6, 7]

print("The original list is : " + str(test_list))

test_tup = (9, 10)

test_list += test_tup

print("The container after addition : " + str(test_list))
```

```
The original list is : [5, 6, 7]
The container after addition : [5, 6, 7, 9, 10]
```

11. Python – Closest Pair to Kth index element in Tuple

In [24]:
```
test_list = [(3, 4), (78, 76), (2, 3), (9, 8), (19, 23)]

print("The original list is : " + str(test_list))

tup = (17, 23)

K = 1

min_dif, res = 999999999, None
for idx, val in enumerate(test_list):
    dif = abs(tup[K - 1] - val[K - 1])
    if dif < min_dif:
        min_dif, res = dif, idx

print("The nearest tuple to Kth index element is : " + str(test_list[res]))
```

```
The original list is : [(3, 4), (78, 76), (2, 3), (9, 8), (19, 23)]
The nearest tuple to Kth index element is : (19, 23)
```

In [ ]:
```

```