using zip to combine data

In [1]:
```
countries=['Hong Kong','Hubei','Hunan','Inner Mongolia','Jiangsu','Jiangxi','Jilin','Liaoning']
```

In [2]:
```
confirmed=['0','444','4','0','1','2','0','2']
```

In [3]:
```
zipped_lists=zip(countries,confirmed)
```

In [4]:
```
result_dict=dict(zipped_lists)
```

In [5]:
```
print(result_dict)
```

```
{'Hong Kong': '0', 'Hubei': '444', 'Hunan': '4', 'Inner Mongolia': '0', 'Jiangsu': '1', 'Jiangxi': '2', 'Jilin': '0', 'Liaoning': '2'}
```

using zip to combine data using function

In [6]:
```
def lists_to_dict(list1,list2):
    zipped_lists=zip(list1,list2)
    rs_dict=dict(zipped_lists)
    return(rs_dict)
rs_fun=lists_to_dict(countries,confirmed)
print(rs_fun)
```

```
{'Hong Kong': '0', 'Hubei': '444', 'Hunan': '4', 'Inner Mongolia': '0', 'Jiangsu': '1', 'Jiangxi': '2', 'Jilin': '0', 'Liaoning': '2'}
```

In [7]:
```
header_names=['Province/State','Country/Region','Last Update','Confirmed_cases']
```

In [8]:
```
row_lists=[['Hong Kong','Hong Kong','1/22/2020 17:00','0'],
['Hubei','Mainland China','1/22/2020 17:00','444'],
['Hunan','Mainland China','1/22/2020 17:00','4'],
['Inner Mongolia','Mainland China','1/22/2020 17:00','0'],
['Jiangsu','Mainland China','1/22/2020 17:00','1'],
['Jiangxi','Mainland China','1/22/2020 17:00','2'],
['Jilin','Mainland China','1/22/2020 17:00','0' ],
['Liaoning','Mainland China','1/22/2020 17:00','2']]
```

In [9]:
```
list_of_dicts=[lists_to_dict(header_names,sublist) for sublist in row_lists]
```

In [10]:
```
print(list_of_dicts)
```

```
[{'Province/State': 'Hong Kong', 'Country/Region': 'Hong Kong', 'Last Update': '1/22/2020 17:00', 'Confirmed_cases': '0'}, {'Province/State': 'Hubei', 'Countr
y/Region': 'Mainland China', 'Last Update': '1/22/2020 17:00', 'Confirmed_cases': '444'}, {'Province/State': 'Hunan', 'Country/Region': 'Mainland China', 'Las
t Update': '1/22/2020 17:00', 'Confirmed_cases': '4'}, {'Province/State': 'Inner Mongolia', 'Country/Region': 'Mainland China', 'Last Update': '1/22/2020 17:0
0', 'Confirmed_cases': '0'}, {'Province/State': 'Jiangsu', 'Country/Region': 'Mainland China', 'Last Update': '1/22/2020 17:00', 'Confirmed_cases': '1'}, {'Pr
ovince/State': 'Jiangxi', 'Country/Region': 'Mainland China', 'Last Update': '1/22/2020 17:00', 'Confirmed_cases': '2'}, {'Province/State': 'Jilin', 'Country/
Region': 'Mainland China', 'Last Update': '1/22/2020 17:00', 'Confirmed_cases': '0'}, {'Province/State': 'Liaoning', 'Country/Region': 'Mainland China', 'Last
Update': '1/22/2020 17:00', 'Confirmed_cases': '2'}]
```

In [12]:
```
import pandas as pd
df=pd.DataFrame(list_of_dicts)
print(df)
```

```
   Province/State  Country/Region      Last Update Confirmed_cases
0       Hong Kong       Hong Kong  1/22/2020 17:00               0
1           Hubei  Mainland China  1/22/2020 17:00             444
2           Hunan  Mainland China  1/22/2020 17:00               4
3  Inner Mongolia  Mainland China  1/22/2020 17:00               0
4         Jiangsu  Mainland China  1/22/2020 17:00               1
5         Jiangxi  Mainland China  1/22/2020 17:00               2
6           Jilin  Mainland China  1/22/2020 17:00               0
7        Liaoning  Mainland China  1/22/2020 17:00               2
```

GENERATORS

In [14]:
```
#generate values from 0 to n
def num_seq(n):
    i=0
    while i<n:
        yield i
        i+=1
gen_file=num_seq(10)
print(next(gen_file))
print(next(gen_file))
print(next(gen_file))
print(next(gen_file))
print(next(gen_file))
```

```
0
1
2
3
4
```

In [26]:
```
import pandas as pd
df1=pd.read_csv('C:/Users/USER/Downloads/covid_19_data (1).csv')
print(df1.head)
```

```
<bound method NDFrame.head of            SNo ObservationDate     Province/State  Country/Region  \
0            1      01/22/2020              Anhui  Mainland China   
1            2      01/22/2020            Beijing  Mainland China   
2            3      01/22/2020          Chongqing  Mainland China   
3            4      01/22/2020             Fujian  Mainland China   
4            5      01/22/2020              Gansu  Mainland China   
...        ...             ...                ...             ...   
306424  306425      05/29/2021  Zaporizhia Oblast         Ukraine   
306425  306426      05/29/2021            Zeeland     Netherlands   
306426  306427      05/29/2021           Zhejiang  Mainland China   
306427  306428      05/29/2021    Zhytomyr Oblast         Ukraine   
306428  306429      05/29/2021       Zuid-Holland     Netherlands   

                Last Update  Confirmed  Deaths  Recovered  
0           1/22/2020 17:00        1.0     0.0        0.0  
1           1/22/2020 17:00       14.0     0.0        0.0  
2           1/22/2020 17:00        6.0     0.0        0.0  
3           1/22/2020 17:00        1.0     0.0        0.0  
4           1/22/2020 17:00        0.0     0.0        0.0  
...                     ...        ...     ...        ...  
306424  2021-05-30 04:20:55   102641.0  2335.0    95289.0  
306425  2021-05-30 04:20:55    29147.0   245.0        0.0  
306426  2021-05-30 04:20:55     1364.0     1.0     1324.0  
306427  2021-05-30 04:20:55    87550.0  1738.0    83790.0  
306428  2021-05-30 04:20:55   391559.0  4252.0        0.0  

[306429 rows x 8 columns]>
```

In [31]:
```
 import pandas as pd
df1=pd.read_csv('C:/Users/USER/Downloads/covid_19_data (1).csv',chunksize=10) #to read first 10 rows
print(next(df1))
```

```
   SNo ObservationDate Province/State  Country/Region      Last Update  \
0    1      01/22/2020          Anhui  Mainland China  1/22/2020 17:00   
1    2      01/22/2020        Beijing  Mainland China  1/22/2020 17:00   
2    3      01/22/2020      Chongqing  Mainland China  1/22/2020 17:00   
3    4      01/22/2020         Fujian  Mainland China  1/22/2020 17:00   
4    5      01/22/2020          Gansu  Mainland China  1/22/2020 17:00   
5    6      01/22/2020      Guangdong  Mainland China  1/22/2020 17:00   
6    7      01/22/2020        Guangxi  Mainland China  1/22/2020 17:00   
7    8      01/22/2020        Guizhou  Mainland China  1/22/2020 17:00   
8    9      01/22/2020         Hainan  Mainland China  1/22/2020 17:00   
9   10      01/22/2020          Hebei  Mainland China  1/22/2020 17:00   

   Confirmed  Deaths  Recovered  
0        1.0     0.0        0.0  
1       14.0     0.0        0.0  
2        6.0     0.0        0.0  
3        1.0     0.0        0.0  
4        0.0     0.0        0.0  
5       26.0     0.0        0.0  
6        2.0     0.0        0.0  
7        1.0     0.0        0.0  
8        4.0     0.0        0.0  
9        1.0     0.0        0.0  
```

In [33]:
```
print(next(df1))
print(next(df1))
```

```
    SNo ObservationDate Province/State  Country/Region      Last Update  \
30   31      01/22/2020          Tibet  Mainland China  1/22/2020 17:00   
31   32      01/22/2020     Washington              US  1/22/2020 17:00   
32   33      01/22/2020       Xinjiang  Mainland China  1/22/2020 17:00   
33   34      01/22/2020         Yunnan  Mainland China  1/22/2020 17:00   
34   35      01/22/2020       Zhejiang  Mainland China  1/22/2020 17:00   
35   36      01/22/2020            NaN           Japan  1/22/2020 17:00   
36   37      01/22/2020            NaN        Thailand  1/22/2020 17:00   
37   38      01/22/2020            NaN     South Korea  1/22/2020 17:00   
38   39      01/22/2020        Unknown           China  1/22/2020 17:00   
39   40      01/22/2020            NaN        Kiribati  1/22/2020 17:00   

    Confirmed  Deaths  Recovered  
30        0.0     0.0        0.0  
31        1.0     0.0        0.0  
32        0.0     0.0        0.0  
33        1.0     0.0        0.0  
34       10.0     0.0        0.0  
35        2.0     0.0        0.0  
36        4.0     0.0        2.0  
37        1.0     0.0        0.0  
38        0.0     0.0        0.0  
39        0.0     0.0        0.0  
    SNo ObservationDate Province/State  Country/Region    Last Update  \
40   41      01/23/2020          Anhui  Mainland China  1/23/20 17:00   
41   42      01/23/2020        Beijing  Mainland China  1/23/20 17:00   
42   43      01/23/2020      Chongqing  Mainland China  1/23/20 17:00   
43   44      01/23/2020         Fujian  Mainland China  1/23/20 17:00   
44   45      01/23/2020          Gansu  Mainland China  1/23/20 17:00   
45   46      01/23/2020      Guangdong  Mainland China  1/23/20 17:00   
46   47      01/23/2020        Guangxi  Mainland China  1/23/20 17:00   
47   48      01/23/2020        Guizhou  Mainland China  1/23/20 17:00   
48   49      01/23/2020         Hainan  Mainland China  1/23/20 17:00   
49   50      01/23/2020          Hubei  Mainland China  1/23/20 17:00   

    Confirmed  Deaths  Recovered  
40        9.0     0.0        0.0  
41       22.0     0.0        0.0  
42        9.0     0.0        0.0  
43        5.0     0.0        0.0  
44        2.0     0.0        0.0  
45       32.0     0.0        2.0  
46        5.0     0.0        0.0  
47        3.0     0.0        0.0  
48        5.0     0.0        0.0  
49      444.0    17.0       28.0  
```

In [ ]: