Exercise 1: Convert two lists into a dictionary

Below are the two lists. Write a Python program to convert them into a dictionary in a way that item from list1 is the key and item from list2 is the value

```
keys = ['Ten', 'Twenty', 'Thirty']
values = [10, 20, 30]
```

Expected output:

```
{'Ten': 10, 'Twenty': 20, 'Thirty': 30}
```

In [1]:

```python
keys=["Ten","Twenty","Thirty"]
values=[10,20,30]
#-----------------
result_dict = {} # or use dict()
for i in range(len(keys)):
    result_dict.update({keys[i]:values[i]})
#Explanation ==> we update an empty dictionnary by adding an inside dic each time (ke
print(result_dict)
######################"""""" solution 2
#the zip(list1,list2) function is a predefined function that aggregate two functions
dictio = dict(zip(keys,values))
print(dictio)
```

```
{'Ten': 10, 'Twenty': 20, 'Thirty': 30}
{'Ten': 10, 'Twenty': 20, 'Thirty': 30}
```

Exercise 2: Merge two Python dictionaries into one

```
dict1 = {'Ten': 10, 'Twenty': 20, 'Thirty': 30}
dict2 = {'Thirty': 30, 'Fourty': 40, 'Fifty': 50}
```

Expected output:

```
{'Ten': 10, 'Twenty': 20, 'Thirty': 30, 'Fourty': 40, 'Fifty': 50}
```

In [2]:

```python
dict1 = {'Ten': 10, 'Twenty': 20, 'Thirty': 30}
dict2 = {'Thirty': 30, 'Fourty': 40, 'Fifty': 50}
##""
empty_dict = dict()
empty_dict=dict1.copy() #Copying first dic in an empty dict
empty_dict.update(dict2) # Update our dictionnary by adding the second dic
print(empty_dict)
#################Solution 2 using python 3.5++
dict3={**dict1,**dict2}
print(dict3)
```

```
{'Ten': 10, 'Twenty': 20, 'Thirty': 30, 'Fourty': 40, 'Fifty': 50}
{'Ten': 10, 'Twenty': 20, 'Thirty': 30, 'Fourty': 40, 'Fifty': 50}
```

Exercise 3: Print the value of key 'history' from the below dict

```
sampleDict = {
    "class": {
        "student": {
            "name": "Mike",
            "marks": {
                "physics": 70,
                "history": 80
            }
        }
    }
}
```

Expected output:

80

In [11]:

```python
sampleDict = {
    "class": {
        "student": {
            "name": "Mike",
            "marks": {
                "physics": 70,
                "history": 80
            }
        }
    }
}
print(sampleDict)
#Access each nested dictionnary just like a list (list[0][1])...
print(sampleDict["class"])
print(sampleDict["class"]["student"])
print(sampleDict["class"]["student"]["marks"])
print(sampleDict["class"]["student"]["marks"]["history"])
```

```
{'class': {'student': {'name': 'Mike', 'marks': {'physics': 70, 'history': 80}}}}
{'student': {'name': 'Mike', 'marks': {'physics': 70, 'history': 80}}}
{'name': 'Mike', 'marks': {'physics': 70, 'history': 80}}
{'physics': 70, 'history': 80}
80
```

Exercise 4: Delete a list of keys from a dictionary

Given:

```
sample_dict = {
    "name": "Kelly",
    "age": 25,
    "salary": 8000,
    "city": "New york"
}

# Keys to remove
keys = ["name", "salary"]
```

Expected output:

```
{'city': 'New york', 'age': 25}
```

In [12]:

```python
sample_dict = {
    "name": "Kelly",
    "age": 25,
    "salary": 8000,
    "city": "New york"
}

# Keys to remove
keys = ["name", "salary"]
for key in keys:
    sample_dict.pop(key)
print(sample_dict)
```

```
{'age': 25, 'city': 'New york'}
```

Exercise 5: Check if a value exists in a dictionary

We know how to check if the key exists in a dictionary. Sometimes it is required to check if the given value is present.

Write a Python program to check if value 200 exists in the following dictionary.

Given:

```
sample_dict = {'a': 100, 'b': 200, 'c': 300}
```

Expected output:

```
200 present in a dict
```

In [13]:

```python
sample_dict = {'a': 100, 'b': 200, 'c': 300}
if 200 in sample_dict.values():
    print("200 in the dict")
else:
    print("200 is not in the dict")
```

```
200 in the dict
```

EXERCISE 6 - List-to-Series Conversion

Given a list, output the corresponding pandas series

Sample Solution

```
given_list = [2, 4, 5, 6, 9]
```

Corresponding Output

```
0    2
1    4
2    5
3    6
4    9
dtype: int64
```

In [1]:

```python
import pandas as pd #as pd is used to simply each time we need to call pandas
given_list=[2,4,5,6,9]
serie = pd.Series(given_list)
print(serie)
```

```
0    2
1    4
2    5
3    6
4    9
dtype: int64
```

EXERCISE 7 - Dictionary-to-Dataframe Conversion

Given a dictionary, convert it into corresponding dataframe and display it

Given :

```
dictionary = {'name': ['Vinay', 'Kushal', 'Aman'],
              'age' : [22, 25, 24],
              'occ' : ['engineer', 'doctor', 'accountant']}
```

Corresponding Output

```
     name  age         occ
0   Vinay   22    engineer
1  Kushal   25      doctor
2    Aman   24  accountant
```

In [6]:

```python
import pandas as pd #as pd is used to simply each time we need to call pandas
dictionary = {"name":['Vinay',"Kushal","Aman"],
              "age":[22,25,24],
              "occ":["Engineer","Doctor","Accountant"]}
df = pd.DataFrame(dictionary)
print(df)
print("-------------------------------------")
print("Note that all arrays in the dictionnary values must be same length or an errow
dictionary = {"name":['Vinay',"Kushal"],
              "age":[22,25,24],
              "occ":["Engineer","Doctor","Accountant"]}
df = pd.DataFrame(dictionary)
print(df)
```

```
     name  age         occ
0   Vinay   22    Engineer
1  Kushal   25      Doctor
2    Aman   24  Accountant
-------------------------------------
Note that all arrays in the dictionnary values must be same length or an errow will be
thrown
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
<ipython-input-6-874bdeccb298> in <module>
     10               "age":[22,25,24],
     11               "occ":["Engineer","Doctor","Accountant"]}
---> 12 df = pd.DataFrame(dictionary)
     13 print(df)

C:\ProgramData\Anaconda3\lib\site-packages\pandas\core\frame.py in __init__(self, dat
a, index, columns, dtype, copy)
    466
    467         elif isinstance(data, dict):
--> 468             mgr = init_dict(data, index, columns, dtype=dtype)
    469         elif isinstance(data, ma.MaskedArray):
    470             import numpy.ma.mrecords as mrecords

C:\ProgramData\Anaconda3\lib\site-packages\pandas\core\internals\construction.py in in
it_dict(data, index, columns, dtype)
    281             arr if not is_datetime64tz_dtype(arr) else arr.copy() for arr in a
rrays
    282         ]
--> 283     return arrays_to_mgr(arrays, data_names, index, columns, dtype=dtype)
    284
    285

C:\ProgramData\Anaconda3\lib\site-packages\pandas\core\internals\construction.py in ar
rays_to_mgr(arrays, arr_names, index, columns, dtype, verify_integrity)
     76         # figure out the index, if necessary
     77         if index is None:
---> 78             index = extract_index(arrays)
     79         else:
     80             index = ensure_index(index)

C:\ProgramData\Anaconda3\lib\site-packages\pandas\core\internals\construction.py in ex
tract_index(data)
    395             lengths = list(set(raw_lengths))
    396             if len(lengths) > 1:
--> 397                 raise ValueError("arrays must all be same length")
    398
    399             if have_dicts:

ValueError: arrays must all be same length
```

Exercise 8: From the given dataset print the first and last five rows

Expected Output:

	index	company	body-style	wheel-base	length	engine-type	num-of-cylinders	horsepower	average-mileage	price
0	0	alfa-romero	convertible	88.6	168.8	dohc	four	111	21	13495.0
1	1	alfa-romero	convertible	88.6	168.8	dohc	four	111	21	16500.0
2	2	alfa-romero	hatchback	94.5	171.2	ohcv	six	154	19	16500.0
3	3	audi	sedan	99.8	176.6	ohc	four	102	24	13950.0
4	4	audi	sedan	99.4	176.6	ohc	five	115	18	17450.0

Python Pandas printing first 5 rows

	index	company	body-style	wheel-base	length	engine-type	num-of-cylinders	horsepower	average-mileage	price
56	81	volkswagen	sedan	97.3	171.7	ohc	four	85	27	7975.0
57	82	volkswagen	sedan	97.3	171.7	ohc	four	52	37	7995.0
58	86	volkswagen	sedan	97.3	171.7	ohc	four	100	26	9995.0
59	87	volvo	sedan	104.3	188.8	ohc	four	114	23	12940.0
60	88	volvo	wagon	104.3	188.8	ohc	four	114	23	13415.0

In [10]:

```python
#The file Automobile_data.csv
import pandas as pd
df = pd.read_csv("Automobile_Data.csv")
print(df) # Read our csv and print it
print("Now print the 5 first lines and 5 last lines")
print("First five")
print(df.head(5))# note that df.head() with no arguments will return the first five by
print("Last Five")
print(df.tail(5))# note that df.tail() with no arguments will return the last five by
```

```
        company  wheel-base engine-type num-of-cylinders  horsepower  \
0   alfa-romero        88.6        dohc             four         111
1   alfa-romero        88.6        dohc             four         111
2   alfa-romero        94.5        ohcv              six         154
3          audi        99.8         ohc             four         102
4          audi        99.4         ohc             five         115
..          ...         ...         ...              ...         ...
56   volkswagen        97.3         ohc             four          85
57   volkswagen        97.3         ohc             four          52
58   volkswagen        97.3         ohc             four         100
59        volvo       104.3         ohc             four         114
60        volvo       104.3         ohc             four         114

    average-mileage    price
0                21  13495.0
1                21  16500.0
2                19  16500.0
3                24  13950.0
4                18  17450.0
..              ...      ...
56               27   7975.0
57               37   7995.0
58               26   9995.0
59               23  12940.0
60               23  13415.0

[61 rows x 7 columns]
Now print the 5 first lines and 5 last lines
First five
       company  wheel-base engine-type num-of-cylinders  horsepower  \
0  alfa-romero        88.6        dohc             four         111
1  alfa-romero        88.6        dohc             four         111
2  alfa-romero        94.5        ohcv              six         154
3         audi        99.8         ohc             four         102
4         audi        99.4         ohc             five         115

   average-mileage    price
0               21  13495.0
1               21  16500.0
2               19  16500.0
3               24  13950.0
4               18  17450.0
Last Five
       company  wheel-base engine-type num-of-cylinders  horsepower  \
56  volkswagen        97.3         ohc             four          85
57  volkswagen        97.3         ohc             four          52
58  volkswagen        97.3         ohc             four         100
59       volvo       104.3         ohc             four         114
60       volvo       104.3         ohc             four         114

    average-mileage    price
56               27   7975.0
57               37   7995.0
58               26   9995.0
59               23  12940.0
60               23  13415.0
```

In [ ]: