--Dictionary

Convert two lists into a dictionary.

In [1]:
```
keys = ['Ten', 'Twenty', 'Thirty']
values = [10, 20, 30]
dict={keys[i]:values[i] for i in range(len(values)) }
print(dict)
```

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

Merge two Python dictionaries into one

In [2]:
```
dict1 = {'Ten': 10, 'Twenty': 20, 'Thirty': 30}
dict2 = {'Thirty': 30, 'Fourty': 40, 'Fifty': 50}
dict3 = {**dict1, **dict2}
print(dict3)
```

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

Print the value of key ‘history’ from the below dict

In [3]:
```
a = { "class": { "student": { "name": "Mike","marks": { "physics": 70,"history":
print(a['class']['student']['marks']['history'])
```

```
80
```

Initialize dictionary with default values

In [4]:
```
employees = ['Kelly', 'Emma']
defaults = {"designation": 'Developer', "salary": 8000}
default1=dict.fromkeys(employees,defaults)
print(default1)
```

```
{'Kelly': {'designation': 'Developer', 'salary': 8000}, 'Emma': {'designatio
n': 'Developer', 'salary': 8000}}
```

Check if a value exists in a dictionary

In [7]:
```
sample_dict = {'a': 100, 'b': 200, 'c': 300}
if 200 in sample_dict.values():
    print("200 present in a dict")
else:
    print(" Nah!, It's not there... ")
```

```
200 present in a dict
```

Rename key of a dictionary

In [9]:
```
sample_dict = {
  "name": "Kelly",
  "age":25,
  "salary": 8000,
  "city": "New york"
}
sample_dict['location']=sample_dict.pop('city')
print(sample_dict)
```

```
{'name': 'Kelly', 'age': 25, 'salary': 8000, 'location': 'New york'}
```

Get the key of a minimum value from the following dictionary

In [10]:
```
sample_dict = {
  'Physics': 82,
  'Math': 65,
  'history': 75
}
print(min(sample_dict,key=sample_dict.get))
```

```
Math
```

Change value of a key in a nested dictionary, change Brad's salary to 8500

In [11]:
```
sample_dict = {
    'emp1': {'name': 'Jhon', 'salary': 7500},
    'emp2': {'name': 'Emma', 'salary': 8000},
    'emp3': {'name': 'Brad', 'salary': 500}
}
sample_dict['emp3']['salary']=8500
print(sample_dict)
```

```
{'emp1': {'name': 'Jhon', 'salary': 7500}, 'emp2': {'name': 'Emma', 'salar
y': 8000}, 'emp3': {'name': 'Brad', 'salary': 8500}}
```

--List

Reverse a list in Python

In [12]:
```
list1 = [100, 200, 300, 400, 500]
list1.reverse()
print(list1)
```

```
[500, 400, 300, 200, 100]
```

Concatenate two lists index-wise

In [13]:
```
list1 = ["M", "na", "i", "Ke"]
list2 = ["y", "me", "s", "lly"]
list3=[i+j for i,j in zip(list1,list2)]
print(list3)
```

```
['My', 'name', 'is', 'Kelly']
```

Turn every item of a list into its square

In [14]:
```
numbers = [1, 2, 3, 4, 5, 6, 7]
square= [numbers**2 for numbers in numbers]
print(square)
```

```
[1, 4, 9, 16, 25, 36, 49]
```

Concatenate two lists in the following order

In [15]:
```
list1 = ["Hello ", "take "]
list2 = ["Dear", "Sir"]
list3 = [l1+l2 for l1 in list1 for l2 in list2]
print(list3)
```

```
['Hello Dear', 'Hello Sir', 'take Dear', 'take Sir']
```

Iterate both lists simultaneously

In [16]:
```
list1 = [10, 20, 30, 40]
list2 = [100, 200, 300, 400]
for l1,l2 in zip(list1,list2[::-1]):
 print(l1,l2)
```

```
10 400
20 300
30 200
40 100
```

Remove empty strings from the list of strings

In [17]:
```
l1= ["Mike", "", "Emma", "Kelly", "", "Brad"]
for m in l1:
  if m=="":
    l1.remove(m)
print(l1)
```

```
['Mike', 'Emma', 'Kelly', 'Brad']
```

Add new item to list after a specified item

In [18]:
```
a1 = [10, 20, [300, 400, [5000, 6000], 500], 30, 40]
a2=[7000]
a1[2][2].extend(a2)
print(a1)
```

```
[10, 20, [300, 400, [5000, 6000, 7000], 500], 30, 40]
```

Extend nested list by adding the sublist

In [19]:
```
list1 = ["a", "b", ["c", ["d", "e", ["f", "g"], "k"], "l"], "m", "n"]
sub_list = ["h", "i", "j"]
list1[2][1][2].extend(sub_list)
print(list1)
```

```
['a', 'b', ['c', ['d', 'e', ['f', 'g', 'h', 'i', 'j'], 'k'], 'l'], 'm', 'n']
```

Replace list's item with new value if found

In [20]:
```
list1 = [5, 10, 15, 20, 25, 50, 20]
list1[list1.index(20)]=200
print(list1)
```

```
[5, 10, 15, 200, 25, 50, 20]
```

Remove all occurrences of a specific item from a list.

In [23]:
```
list1= [5, 20, 15, 20, 25, 50, 20]
for i in list1:
    if i==20:
        list1.remove(i)
print(list1)
```

```
[5, 15, 25, 50]
```

--SET

Add a list of elements to a set

In [24]:
```
sample_set = {"Yellow", "Orange", "Black"}
sample_list = ["Blue", "Green", "Red"]
sample_set.update(sample_list)
print(sample_set)
```

```
{'Red', 'Blue', 'Black', 'Yellow', 'Orange', 'Green'}
```

Return a new set of identical items from two sets

In [25]:
```
set1 = {10, 20, 30, 40, 50}
set2 = {30, 40, 50, 60, 70}
new_set=set1.intersection(set2)
print(new_set)
```

```
{40, 50, 30}
```

Get Only unique items from two sets

In [26]:
```
set1 = {10, 20, 30, 40, 50}
set2 = {30, 40, 50, 60, 70}
new_set=set1.union(set2)
print(new_set)
```

```
{70, 40, 10, 50, 20, 60, 30}
```

Update the first set with items that don’t exist in the second set

In [27]:
```
set1 = {10, 20, 30}
set2 = {20, 40, 50}
set1.difference_update(set2)
print(set1)
```

```
{10, 30}
```

Remove items from the set at once

In [28]:
```
set1 = {10, 20, 30, 40, 50}
set1.difference_update({10,20,30})
print(set1)
```

```
{40, 50}
```

Return a set of elements present in Set A or B, but not both

In [36]:
```
set1 = {10, 20, 30, 40, 50}
set2 = {30, 40, 50, 60, 70}
print(set1.symmetric_difference(set2))
```

```
{20, 70, 10, 60}
```

Check if two sets have any elements in common. If yes, display the common elements

In [30]:
```
set1 = {10, 20, 30, 40, 50}
set2 = {60, 70, 80, 90, 10}
if set1.intersection(set2):
  print(set1.intersection(set2))
else:
  print("No common elements")
```

```
{10}
```

Update set1 by adding items from set2, except common items

In [35]:
```
set1 = {10, 20, 30, 40, 50}
set2 = {30, 40, 50, 60, 70}
new_set=set1.difference(set2).union(set2.difference(set1))
print(new_set)
```

```
{70, 10, 20, 60}
```

Return a new set of identical items from two sets

In [37]:
```
set1 = {10, 20, 30, 40, 50}
set2 = {30, 40, 50, 60, 70}
new_set=set1.intersection(set2)
print(new_set)
```

```
{40, 50, 30}
```

--Tuple

Reverse the tuple

In [38]:
```
tuple1 = (10, 20, 30, 40, 50)
tuple1 = tuple1[::-1]
print(tuple1)
```

```
(50, 40, 30, 20, 10)
```

Reverse the tuple

In [39]:
```
tuple1 = ("Orange", [10, 20, 30], (5, 15, 25))
print(tuple1[1][1])
```

```
20
```

Create a tuple with single item 50

In [40]:
```
tuple1= (50, )
print(tuple1)
```

```
(50,)
```

Unpack the tuple into 4 variables

In [41]:
```
tuple1 = (10, 20, 30, 40)
a, b, c, d = tuple1
print(a)
print(b)
print(c)
print(d)
```

```
10
20
30
40
```

Swap two tuples in Python

In [42]:
```
tuple1 = (11, 22)
tuple2 = (99, 88)
tuple1, tuple2 = tuple2, tuple1
print(tuple2)
print(tuple1)
```

```
(11, 22)
(99, 88)
```

Copy specific elements from one tuple to a new tuple

In [43]:
```
tuple1 = (11, 22, 33, 44, 55, 66)
tuple2 = tuple1[3:-1]
print(tuple2)
```

```
(44, 55)
```

Modify the tuple

In [44]:
```
tuple1 = (11, [22, 33], 44, 55)
tuple1[1][0] = 222
print(tuple1)
```

```
(11, [222, 33], 44, 55)
```

Sort a tuple of tuples by 2nd item

In [45]:
```
tuple1 = (('a', 23), ('b', 37), ('c', 11), ('d', 29))
tuple1 = tuple(sorted(list(tuple1), key=lambda x: x[1]))
print(tuple1)
```

```
(('c', 11), ('a', 23), ('d', 29), ('b', 37))
```

Counts the number of occurrences of item 50 from a tuple

In [46]:
```
tuple1 = (50, 10, 60, 70, 50)
print(tuple1.count(50))
```

```
2
```

Check if all items in the tuple are the same

In [47]:
```
def check(t):
    return all(i == t[0] for i in t)
tuple1 = (45, 45, 45, 45)
print(check(tuple1))
```

```
True
```