

Write a Python script to sort (ascending and descending) a dictionary by value.

```
# Sample dictionary
```

```
sample_dict = {'apple': 50, 'banana': 30, 'orange': 20, 'grape': 40}
```

```
# Sorting in ascending order by value
```

```
sorted_dict_asc = {k: v for k, v in sorted(sample_dict.items(), key=lambda item: item[1])}
```

```
print("Sorted dictionary in ascending order by value:", sorted_dict_asc)
```

```
# Sorting in descending order by value
```

```
sorted_dict_desc = {k: v for k, v in sorted(sample_dict.items(), key=lambda item: item[1], reverse=True)}
```

```
print("Sorted dictionary in descending order by value:", sorted_dict_desc)
```

output:

Sorted dictionary in ascending order by value: {'orange': 20, 'banana': 30, 'grape': 40, 'apple': 50}

Sorted dictionary in descending order by value: {'apple': 50, 'grape': 40, 'banana': 30, 'orange': 20}

Write a Python script to add a key to a dictionary.

Sample Dictionary : {0: 10, 1: 20}

Expected Result : {0: 10, 1: 20, 2: 30}

```
# Sample dictionary
```

```
sample_dict = {0: 10, 1: 20}
```

```
# Adding a new key-value pair to the dictionary
```

```
sample_dict[2] = 30
```

```
# Printing the updated dictionary
```

```
print("Updated dictionary:", sample_dict)
```

Write a Python script to concatenate following dictionaries to create a new one.

Sample Dictionary :

dic1={1:10, 2:20}

dic2={3:30, 4:40}

dic3={5:50,6:60}

Expected Result : {1: 10, 2: 20, 3: 30, 4: 40, 5: 50, 6: 60}

```
# Sample dictionaries
```

```
dic1 = {1: 10, 2: 20}
```

```
dic2 = {3: 30, 4: 40}
```

```
dic3 = {5: 50, 6: 60}
```

```
# Concatenating the dictionaries into a new dictionary
```

```
new_dict = {**dic1, **dic2, **dic3}
```

```
# Printing the new dictionary
```

```
print("New dictionary:", new_dict)
```

output:

New dictionary: {1: 10, 2: 20, 3: 30, 4: 40, 5: 50, 6: 60}

Write a Python script to check if a given key already exists in a dictionary

```
# Sample dictionary
```

```
sample_dict = {'apple': 50, 'banana': 30, 'orange': 20, 'grape': 40}
```

```
# Key to check
```

```
key_to_check = 'banana'
```

```
# Checking if the key exists in the dictionary
```

```
if key_to_check in sample_dict:
```

```
    print(key_to_check, "exists in the dictionary")
```

```
else:
```

```
    print(key_to_check, "does not exist in the dictionary")
```

output:

banana exists in the dictionary

Write a Python program to iterate over dictionaries using for loops

Sample dictionary

```
sample_dict = {'apple': 50, 'banana': 30, 'orange': 20, 'grape': 40}
```

Iterating over keys using for loop

```
print("Keys:")
```

```
for key in sample_dict:
```

```
    print(key)
```

Iterating over values using for loop

```
print("\nValues:")
```

```
for value in sample_dict.values():
```

```
    print(value)
```

Iterating over key-value pairs using for loop

```
print("\nKey-Value Pairs:")
```

```
for key, value in sample_dict.items():
```

```
    print(key, ":", value)
```

output:

Keys:

apple

banana

orange

grape

Values:

50

30

20

40

Key-Value Pairs:

apple : 50

banana : 30

orange : 20

grape : 40

Write a Python script to generate and print a dictionary that contains a number (between 1 and n) in the form (x, x*x).

Sample Dictionary (n = 5) :

Expected Output : {1: 1, 2: 4, 3: 9, 4: 16, 5: 25}

```
# Value of n
```

```
n = 5
```

```
# Generating the dictionary using a dictionary comprehension
```

```
result_dict = {x: x*x for x in range(1, n+1)}
```

```
# Printing the dictionary
```

```
print(result_dict)
```

```
output:
```

```
{1: 1, 2: 4, 3: 9, 4: 16, 5: 25}
```

Write a Python script to merge two Python dictionaries

```
# Sample dictionaries
```

```
dict1 = {'apple': 50, 'banana': 30, 'orange': 20}
```

```
dict2 = {'grape': 40, 'kiwi': 10}
```

```
# Merging the dictionaries using the update() method
```

```
dict1.update(dict2)
```

```
# Printing the merged dictionary
```

```
print(dict1)
```

```
output:
```

```
{'apple': 50, 'banana': 30, 'orange': 20, 'grape': 40, 'kiwi': 10}
```

Write a Python program to sum all the items in a dictionary

```
# Sample dictionary
```

```
sample_dict = {'apple': 50, 'banana': 30, 'orange': 20, 'grape': 40}
```

```
# Using the sum() function to calculate the sum of all the values in the dictionary
```

```
total_sum = sum(sample_dict.values())
```

```
# Printing the total sum
```

```
print("Total sum of values in the dictionary:", total_sum)
```

```
output:
```

```
Total sum of values in the dictionary: 140
```

Write a Python program to multiply all the items in a dictionary

Sample dictionary

```
sample_dict = {'apple': 2, 'banana': 3, 'orange': 4, 'grape': 5}
```

Initializing the product to 1

```
product = 1
```

Looping through all the values in the dictionary and multiplying them with the running product

```
for value in sample_dict.values():
```

```
    product *= value
```

Printing the final product

```
print("Product of all values in the dictionary:", product)
```

output:

Product of all values in the dictionary: 120

Write a Python program to remove a key from a dictionary

Sample dictionary

```
sample_dict = {'apple': 50, 'banana': 30, 'orange': 20, 'grape': 40}
```

Removing the key 'banana' using the del keyword

```
del sample_dict['banana']
```

Printing the updated dictionary

```
print(sample_dict)
```

output:

```
{'apple': 50, 'orange': 20, 'grape': 40}
```

Write a Python program to sort a dictionary by key

Sample dictionary

```
sample_dict = {'apple': 50, 'banana': 30, 'orange': 20, 'grape': 40}
```

Sorting the dictionary by key using the sorted() function

```
sorted_dict = dict(sorted(sample_dict.items()))
```

Printing the sorted dictionary

```
print(sorted_dict)
```

output:

```
{'apple': 50, 'banana': 30, 'grape': 40, 'orange': 20}
```

Write a Python program to get the maximum and minimum value in a dictionary

```
# Sample dictionary
sample_dict = {'apple': 50, 'banana': 30, 'orange': 20, 'grape': 40}
# Getting the maximum value in the dictionary using the max() function
max_value = max(sample_dict.values())
# Getting the key associated with the maximum value in the dictionary
max_key = max(sample_dict, key=sample_dict.get)
# Getting the minimum value in the dictionary using the min() function
min_value = min(sample_dict.values())
# Getting the key associated with the minimum value in the dictionary
min_key = min(sample_dict, key=sample_dict.get)
# Printing the maximum and minimum values and their respective keys
print(f"Maximum value: {max_value} (Key: {max_key})")
print(f"Minimum value: {min_value} (Key: {min_key})")
output:
Maximum value: 50 (Key: apple)
Minimum value: 20 (Key: orange)
```

Write a Python program to remove duplicates from Dictionary

```
# Sample dictionary with duplicates
sample_dict = {'apple': 50, 'banana': 30, 'orange': 20, 'grape': 40, 'banana': 10}
# Creating a new dictionary without duplicates
unique_dict = {}
# Looping through the key-value pairs in the original dictionary
for key, value in sample_dict.items():
    # Checking if the value is already in the new dictionary
    if value not in unique_dict.values():
        # If not, add the key-value pair to the new dictionary
        unique_dict[key] = value
# Printing the new dictionary without duplicates
print(unique_dict)
output:
{'apple': 50, 'banana': 10, 'orange': 20, 'grape': 40}
```

Write a Python program to check a dictionary is empty or not

```
# Sample dictionary
sample_dict = {}
# Check if the dictionary is empty
if not bool(sample_dict):
    print("The dictionary is empty.")
else:
    print("The dictionary is not empty.")
output:
The dictionary is empty.
```

Write a Python program to combine two dictionary adding values for common keys.

```
d1 = {'a': 100, 'b': 200, 'c': 300}
```

```
d2 = {'a': 300, 'b': 200, 'd': 400}
```

Sample output: Counter({'a': 400, 'b': 400, 'd': 400, 'c': 300})

```
from collections import Counter
```

```
# Sample dictionaries
```

```
d1 = {'a': 100, 'b': 200, 'c': 300}
```

```
d2 = {'a': 300, 'b': 200, 'd': 400}
```

```
# Combine dictionaries using Counter
```

```
result = Counter(d1) + Counter(d2)
```

```
# Print the combined dictionary
```

```
print(result)
```

```
output:
```

```
Counter({'a': 400, 'b': 400, 'd': 400, 'c': 300})
```

Write a Python program to find the highest 3 values in a dictionary

```
import heapq
# Sample dictionary
sample_dict = {'a': 100, 'b': 200, 'c': 300, 'd': 400, 'e': 500}
# Find the highest 3 values using heapq.nlargest()
highest_values = heapq.nlargest(3, sample_dict.values())
# Print the highest 3 values
print("The highest 3 values are:", highest_values)
```

output:

The highest 3 values are: [500, 400, 300]

Write a Python program to match key values in two dictionaries.

Sample dictionary: {'key1': 1, 'key2': 3, 'key3': 2}, {'key1': 1, 'key2': 2}

Expected output: key1: 1 is present in both x and y

```
x = {'key1': 1, 'key2': 3, 'key3': 2}
y = {'key1': 1, 'key2': 2}
for key in x:
    if key in y and x[key] == y[key]:
        print(key + ": " + str(x[key]) + " is present in both x and y")
```

output:

key1: 1 is present in both x and y

Write a Python program to check if all dictionaries in a list are empty or not.

Sample list : [{}, {}, {}]

Return value : True

Sample list : [{1,2}, {}, {}]

Return value : False

```
def check_empty_dicts(lst):
    for d in lst:
        if bool(d):
            return False
    return True
# Example usage
lst1 = [{}, {}, {}]
lst2 = [{1,2}, {}, {}]
print(check_empty_dicts(lst1)) # Output: True
print(check_empty_dicts(lst2)) # Output: False
```


Write a Python program to remove duplicates from a list of lists.

Sample list : `[[10, 20], [40], [30, 56, 25], [10, 20], [33], [40]]`

New List : `[[10, 20], [30, 56, 25], [33], [40]]`

```
def remove_duplicates(lst):  
    return [list(x) for x in set(tuple(x) for x in lst)]  
  
# Example usage  
lst = [[10, 20], [40], [30, 56, 25], [10, 20], [33], [40]]  
new_lst = remove_duplicates(lst)  
print(new_lst)  
Output: [[33], [40], [10, 20], [30, 56, 25]]
```

Write a Python program to extend a list without append.

Sample data: `[10, 20, 30]`

`[40, 50, 60]`

Expected output : `[40, 50, 60, 10, 20, 30]`

```
def extend_list(lst1, lst2):  
    for elem in lst2:  
        lst1.insert(0, elem)  
    return lst1  
  
# Example usage  
lst1 = [10, 20, 30]  
lst2 = [40, 50, 60]  
new_lst = extend_list(lst1, lst2)  
print(new_lst)  
Output: [40, 50, 60, 10, 20, 30]
```