

1. Sort a dictionary by value (ascending and descending)

```
sample_dict = {0: 10, 1: 20, 2: 5}
```

```
ascending_sorted_dict = dict(sorted(sample_dict.items(), key=lambda item: item[1]))
```

```
descending_sorted_dict = dict(sorted(sample_dict.items(), key=lambda item: item[1], reverse=True))
```

```
print("Ascending sorted dictionary:", ascending_sorted_dict)
```

```
print("Descending sorted dictionary:", descending_sorted_dict)
```

2. Add a key to a dictionary

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

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

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

3. To Concatenate dictionaries to create a new one

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

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

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

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

```
print("Concatenated dictionary:", concatenated_dict)
```

4. To Check if a key exists in a dictionary

```
key_to_check = 2
```

```
if key_to_check in sample_dict:
```

```
    print(f"Key {key_to_check} exists in the dictionary.")
```

```
else:
```

```
    print(f"Key {key_to_check} does not exist in the dictionary.")
```

5. To Iterate over dictionaries using for loops

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

```
    print(f"Key: {key}, Value: {value}")
```

6. To Generate and print a dictionary of squares

```
n = 5  
squared_dict = {x: x*x for x in range(1, n+1)}  
print("Squared dictionary:", squared_dict)
```

7. Merge two dictionaries

```
d1 = {'a': 100, 'b': 200}  
d2 = {'c': 300, 'd': 400}  
merged_dict = {**d1, **d2}  
print("Merged dictionary:", merged_dict)
```

8. Sum all items in a dictionary

```
sum_of_items = sum(sample_dict.values())  
print("Sum of items:", sum_of_items)
```

9. Multiply all items in a dictionary

```
from functools import reduce  
product_of_items = reduce(lambda x, y: x * y, sample_dict.values())  
print("Product of items:", product_of_items)
```

10. Remove a key from a dictionary

```
key_to_remove = 1  
sample_dict.pop(key_to_remove, None)  
print("Updated dictionary after removing key:", sample_dict)
```

11. Sort a dictionary by key

```
sorted_by_key_dict = dict(sorted(sample_dict.items()))  
print("Dictionary sorted by key:", sorted_by_key_dict)
```

12. Get the maximum and minimum value in a dictionary

```
max_value = max(sample_dict.values())
min_value = min(sample_dict.values())
print("Maximum value:", max_value)
print("Minimum value:", min_value)
```

13. Remove duplicates from Dictionary

```
sample_dict = {0: 10, 1: 20, 2: 10, 3: 30}

no_duplicates_dict = {key: value for key, value in sample_dict.items() if
list(sample_dict.values()).count(value) == 1}

print("Dictionary without duplicates:", no_duplicates_dict)
```

14. Check if a dictionary is empty or not

```
if not sample_dict:
    print("Dictionary is empty.")
else:
    print("Dictionary is not empty.")
```

15 Combine two dictionaries adding values for common keys

```
from collections import Counter

d1 = {'a': 100, 'b': 200, 'c': 300}
d2 = {'a': 300, 'b': 200, 'd': 400}

combined_dict = Counter(d1) + Counter(d2)

print("Combined dictionary:", combined_dict)
```

16. Find the highest 3 values in a dictionary

```
highest_values = sorted(sample_dict.values(), reverse=True)[:3]

print("Highest 3 values:", highest_values)
```

17. Match key values in two dictionaries

```
dict1 = {'key1': 1, 'key2': 3, 'key3': 2}
dict2 = {'key1': 1, 'key2': 2}
common_keys = set(dict1.keys()) & set(dict2.keys())
for key in common_keys:
    if dict1[key] == dict2[key]:
        print(f"{key}: {dict1[key]} is present in both dict1 and dict2")
```

18. Check if dictionaries in a list are empty or not

```
list_of_dicts = [{}, {}, {}]
if all(not d for d in list_of_dicts):
    print("All dictionaries in the list are empty.")
else:
    print("Some dictionaries in the list are not empty.")
```

19. Remove duplicates from a list of lists

```
sample_list = [[10, 20], [40], [30, 56, 25], [10, 20], [33], [40]]
no_duplicates_list = [list(t) for t in {tuple(item) for item in sample_list}]
print("List without duplicates:", no_duplicates_list)
```

20. Extend a list without append

```
sample_data = [10, 20, 30]
extended_list = [40, 50, 60]
extended_list.extend(sample_data)
print("Extended list:", extended_list)
```