

## 1. Create a Set and Display Elements

# Coding Python

Auto saved at 12:51:28

```
1 def create_display_set():
2     my_set = {10, 20, 30}
3     print("Set elements:", my_set)
4
5 create_display_set()
```

## Compile Result

Set elements: {10, 20, 30}

[Process completed - press Enter]

## 2. Add an Element to a Set

# Coding Python

Auto saved at 12:56:35

```
1 def add_element():
2     my_set = {1, 2, 3}
3     my_set.add(4)
4     print("Updated set:", my_set)
5 add_element()
```

## Compile Result

Updated set: {1, 2, 3, 4}

[Process completed - press Enter]

### 3. Remove an Element from a Set

# Coding Python

Auto saved at 12:56:40

```
1 def remove_element():
2     my_set = {5, 6, 7}
3     my_set.remove(6)
4     print("After removal:", my_set)
5 remove_element()
```

### Compile Result

After removal= {5, 7}

[Process completed - press Enter]

#### 4. Clear All Elements from a Set

# Coding Python

Auto saved at 13:00:22

```
1 def clear_set():
2     my_set = {1, 2, 3}
3     my_set.clear()
4     print("Cleared set:", my_set)
5 clear_set()
```

## Compile Result

Cleared set: set()

[Process completed - press Enter]

## 5. Copy a Set to Another Set

# Coding Python

Auto saved at 13:02:48

```
1 def copy_set():
2     original = {100, 200, 300}
3     new_set = original.copy()
4     print("Copied set:", new_set)
5 copy_set()
```

## Compile Result

Copied set: {200, 100, 300}

[Process completed - press Enter]

## 6. Check if an Element Exists in a Set

# Coding Python

Auto saved at 13:06:55

```
1 def check_element():
2     my_set = {10, 20, 30}
3     if 20 in my_set:
4         print("20 exists in the set")
5     else:
6         print("20 not in set")
7 check_element()
```

## Compile Result

20 exists in the set

[Process completed - press Enter]

## 7. Find the Length of a Set

# Coding Python

Auto saved at 13:08:52

```
1 def length_of_set():
2     my_set = {1, 2, 3, 4, 5}
3     print("Length of set:", len(my_set))
4 length_of_set()
```

## Compile Result

Length of set= 5

[Process completed - press Enter]

## 8. Iterate Through a Set Using Loop

# Coding Python

Auto saved at 13:11:14

```
1 def iterate_set():
2     my_set = {"apple", "banana", "cherry"}
3     for item in my_set:
4         print(item)
5 iterate_set()
```

## Compile Result

banana  
cherry  
apple

[Process completed - press Enter]

## 9. Find the Union of Two Sets

# Coding Python

Auto saved at 13:14:18

```
1 def union_sets():
2     a = {1, 2, 3}
3     b = {3, 4, 5}
4     print("Union:", a | b)
5 union_sets()
```

## Compile Result

Union= {1, 2, 3, 4, 5}

[Process completed - press Enter]

## 10. Find the Intersection of Two Sets

# Coding Python

Auto saved at 13:17:12

```
1 def intersection_sets():
2     a = {1, 2, 3}
3     b = {2, 3, 4}
4     print("Intersection:", a & b)
5 intersection_sets()
```

## Compile Result

Intersection={2, 3}

[Process completed - press Enter]

## 11. Find the Difference Between Two Sets

# Coding Python

Auto saved at 13:21:26

```
1 def difference_sets():
2     a = {1, 2, 3, 4}
3     b = {3, 4, 5}
4     print("Difference (a - b):", a - b)
5 difference_sets()
```

## Compile Result

Difference (a - b): {1, 2}

[Process completed - press Enter]

## 12. Find Symmetric Difference Between Sets

# Coding Python

Auto saved at 13:24:14

```
def symmetric_difference_sets():
    a = {1, 2, 3}
    b = {3, 4, 5}
    print("Symmetric Difference:", a ^ b)

symmetric_difference_sets()
```

# Compile Result

```
Symmetric Difference= {1, 2, 4, 5}
[Process completed - press Enter]
```

## 13. Check if One Set is Subset of Another

# Coding Python

Auto saved at 13:24:24

```
1 def check_subset():
2     a = {1, 2}
3     b = {1, 2, 3}
4     print("Is a subset:", a.issubset(b))
5 check_subset()
```

## Compile Result

Is a subset: True

[Process completed - press Enter]

## 14. Check if One Set is Superset of Another

# Coding Python

Auto saved at 13:26:45

```
1 def check_superset():
2     a = {1, 2, 3}
3     b = {1, 2}
4     print("Is a superset:", a.issuperset(b))
5 check_superset()
```

## Compile Result

Is a superset: True

[Process completed - press Enter]

## 15. Check if Two Sets are Disjoint

# Coding Python

Auto saved at 13:30:00

```
def check_disjoint():
    a = {1, 2}
    b = {3, 4}
    print("Are disjoint:", a.isdisjoint(b))
check_disjoint()
```

## Compile Result

Are disjoint: True

[Process completed - press Enter]

## 16. Remove Duplicates from a List Using Set

# Coding Python

Auto saved at 13:32:46

```
1 def remove_duplicates():
2     my_list = [1, 2, 2, 3, 4, 4, 5]
3     no_duplicates = set(my_list)
4     print("List without duplicates:", no_duplicates)
5 remove_duplicates()
```

## Compile Result

List without duplicates: {1, 2, 3, 4, 5}

[Process completed - press Enter]

## 17. Convert List to Set and Back to List

# Coding Python

Auto saved at 13:34:45

```
1 def list_set_conversion():
2     my_list = [1, 2, 2, 3]
3     my_set = set(my_list)
4     new_list = list(my_set)
5     print("Converted list:", new_list)
6 list_set_conversion()
```

## Compile Result

Converted list: [1, 2, 3]

[Process completed - press Enter]

## 18. Find Common Elements in Two Lists

# Coding Python

Auto saved at 13:36:48

```
1 def common_elements():
2     a = [1, 2, 3]
3     b = [2, 3, 4]
4     common = set(a) & set(b)
5     print("Common elements:", common)
6 common_elements()
```

## Compile Result

Common elements: {2, 3}

[Process completed - press Enter]

## 19. Elements in One List but Not in Another

# Coding Python

Auto saved at 13:40:36

```
1 def unique_elements():
2     a = [1, 2, 3]
3     b = [3, 4, 5]
4     difference = set(a) - set(b)
5     print("Elements in a not in b:", difference)
6 unique_elements()
```

## Compile Result

Elements in a not in b: {1, 2}

[Process completed - press Enter]

## 20. Set Comprehension – Generate Squares

# Coding Python

Auto saved at 13:42:59

```
1 def square_set():
2     squares = {x**2 for x in range(1, 6)}
3     print("Squares using set comprehension:", squares)
4 square_set()
```

# Compile Result

Squares using set comprehension= {1, 4, 9  
, 16, 25}

[Process completed - press Enter]

## 21. Create and Print a Frozenset

# Coding Python

Auto saved at 13:44:41

```
1 def frozen_set_demo():
2     fs = frozenset([1, 2, 3, 4])
3     print("Frozenset:", fs)
4 frozen_set_demo()
```

## Compile Result

```
Frozenset= frozenset({1, 2, 3, 4})  
[Process completed - press Enter]
```