



```
1 # Create a list of fruits
2 fruits = ["apple", "banana", "orange"]
3
4 # Print the list
5 print("Original list:", fruits)
6
7 # Add an element (grape) to the end of the list
8 fruits.append("grape")
9 print("List after adding grape:", fruits)
10
11 # Remove an element (banana) by value
12 fruits.remove("banana")
13 print("List after removing banana:", fruits)
14
15 # Modify an element (change orange to mango) by index
16 fruits[1] = "mango"
17 print("List after modifying orange to mango:", fruits)
18
19 # Get the length of the list
20 list_length = len(fruits)
21 print("Length of the list:", list_length)
22
```

```
Original list: ['apple', 'banana', 'orange']
List after adding grape: ['apple', 'banana', 'orange', 'grape']
List after removing banana: ['apple', 'orange', 'grape']
List after modifying orange to mango: ['apple', 'mango', 'grape']
Length of the list: 3
```

```
=== Code Execution Successful ===
```

main.py



Share

Run

Output

Clear

```
1 # Create a dictionary to store information about students
2 students = {
3     "1": {"name": "Alice", "age": 20, "course": "Computer Science"},
4     "2": {"name": "Bob", "age": 22, "course": "Mathematics"}
5 }
6
7 # Print the dictionary
8 print("Original dictionary:", students)
9
10 # Add a new student (ID: 3)
11 students["3"] = {"name": "Charlie", "age": 19, "course": "Physics"}
12 print("Dictionary after adding Charlie:", students)
13
14 # Remove a student (ID: 2) by key
15 del students["2"]
16 print("Dictionary after removing Bob:", students)
17
18 # Modify a student's information (change Alice's course to
    Engineering)
19 students["1"]["course"] = "Engineering"
20 print("Dictionary after modifying Alice's course:", students)
21
22 # Check if a key exists (check for ID: 4)
23 if "4" in students:
24     print("Student with ID 4 exists")
25 else:
```

```
Original dictionary: {'1': {'name': 'Alice', 'age': 20, 'course': 'Computer
Science'}, '2': {'name': 'Bob', 'age': 22, 'course': 'Mathematics'}}
Dictionary after adding Charlie: {'1': {'name': 'Alice', 'age': 20, 'course'
: 'Computer Science'}, '2': {'name': 'Bob', 'age': 22, 'course':
'Mathematics'}, '3': {'name': 'Charlie', 'age': 19, 'course':
'Physics'}}
Dictionary after removing Bob: {'1': {'name': 'Alice', 'age': 20, 'course':
'Computer Science'}, '3': {'name': 'Charlie', 'age': 19, 'course':
'Physics'}}
Dictionary after modifying Alice's course: {'1': {'name': 'Alice', 'age': 20
, 'course': 'Engineering'}, '3': {'name': 'Charlie', 'age': 19, 'course'
: 'Physics'}}
Student with ID 4 does not exist

=== Code Execution Successful ===
```



main.py



Share

Run

Output

Clear

```
1 # Create a set of unique characters
2 letters = set("apple")
3
4 # Print the set
5 print("Original set:", letters)
6
7 # Adding elements (duplicate 'p' will be ignored)
8 letters.add('p')
9 letters.add('p')
10 print("Set after adding 'p':", letters)
11
12 # Removing an element (if it exists)
13 letters.remove('e') # Throws an error if 'e' doesn't exist
14 print("Set after removing 'e':", letters)
15
16 # Modifying elements (not possible in sets as elements are immutable)
17 # Instead, we can create a new set with the desired changes
18
19 modified_letters = letters.union({'b', 'a'}) # Add 'b' and 'a' (if
    not already present)
20 print("Modified set (original cannot be modified):",
    modified_letters)
21
22 # Check if an element exists
23 if 'l' in letters:
```

```
Original set: {'l', 'e', 'p', 'a'}
Set after adding 'p': {'l', 'e', 'p', 'a'}
Set after removing 'e': {'l', 'p', 'a'}
Modified set (original cannot be modified): {'l', 'b', 'p', 'a'}
'l' exists in the set
```

```
=== Code Execution Successful ===
```