Removing an element from a HashSet

Below are the ways that we can remove an element from the **HashSet**.

Using the `remove(Object o)` method

We can use the `remove(Object o)` method to remove an element from **HashSet**. This method takes an object that needs to be removed as a parameter. If the element is removed, then this method returns `true`. If the element is not present, then it returns `false`.

Using the `clear()` method

We can use the `clear()` method to remove all the elements from a **HashSet**.

```java
import java.util.HashSet;
import java.util.Set;

public class HashSetDemo {
    public static void main(String args[]) {
        Set<Integer> set = new HashSet<>();

        set.add(23);
        set.add(34);
        set.add(56);

        set.remove(23);

        System.out.println("HashSet after removing one element" + set);

        set.clear();

        System.out.println("HashSet after removing all elements" + set);

    }
}
```

Run Save Reset

Checking if the HashSet is empty

We can check if the **HashSet** is empty using the `isEmpty()` method. This method returns `true` if the Set does not have any elements and returns `false` if the Set has some elements.

```java
import java.util.HashSet;
import java.util.Set;

public class HashSetDemo {
    public static void main(String args[]) {
        Set<Integer> set = new HashSet<>();

        set.add(23);
        set.add(34);
        set.add(56);

        System.out.println(set.isEmpty());
    }
}
```

Run Save Reset