Iterating a **HashMap** is a bit challenging compared to a **List** or a **Set** as it contains pairs of elements. In a map, each key-value pair is called **Entry**. The `entrySet()` method returns the **Set** view of the mapped elements. We can iterate over this Set using any of the below approaches.

Iterating using a `for` loop

We can easily iterate the **EntrySet** returned by the `entrySet()` method using an enhanced for loop. The **Entry** class contains two methods: `getKey()` and `getValue()`, which can be used to get the key and value respectively.

```
import java.util.HashMap;
import java.util.Map;
import java.util.Map.Entry;
import java.util.Set;

public class HashMapDemo {

    public static void main(String args[]) {

        Map<String, Integer> stockPrice = new HashMap<>();

        stockPrice.put("Oracle", 56);
        stockPrice.put("Fiserv", 117);
        stockPrice.put("BMW", 73);
        stockPrice.put("Microsoft", 213);

        Set<Entry<String, Integer>> entrySet = stockPrice.entrySet(); // Returns a Set of Entries

        for (Entry<String, Integer> entry : entrySet) {
            System.out.println("Company Name: " + entry.getKey() + " Stock Price: " + entry.getValue());
        }
    }
}
```

Run Save Reset

Iterating using an `iterator`

Instead of using a for loop, we can get the iterator on the **EntrySet** and then use it to iterate the HashMap. If we remove an element from the **EntrySet**, then it is also removed from the original Map.

```
import java.util.HashMap;
import java.util.Iterator;
import java.util.Map;
import java.util.Map.Entry;
import java.util.Set;

public class HashMapDemo {

    public static void main(String args[]) {

        Map<String, Integer> stockPrice = new HashMap<>();

        stockPrice.put("Oracle", 56);
        stockPrice.put("Fiserv", 117);
        stockPrice.put("BMW", 73);
        stockPrice.put("Microsoft", 213);

        Set<Entry<String, Integer>> entrySet = stockPrice.entrySet(); // Returns a Set of Entries

        Iterator<Entry<String, Integer>> itr = entrySet.iterator(); //Getting the iterator

        while (itr.hasNext()) {
            Entry<String,Integer> entry = itr.next();
            System.out.println("Company Name: " + entry.getKey() + " Stock Price: " + entry.getValue());

            if(entry.getKey().equals("Oracle")) {
                itr.remove();
            }
```

Run Save Reset

Iterating using the `forEach()` method

The `forEach(BiConsumer<? super K, ? super V> action)` method is a default method that was introduced in Java 8. It takes a `BiConsumer` as a parameter. The below example shows how we can use the `forEach` method to print the key-value pairs.

```
import java.util.HashMap;
import java.util.Map;

public class HashMapDemo {

    public static void main(String args[]) {

        Map<String, Integer> stockPrice = new HashMap<>();

        stockPrice.put("Oracle", 56);
        stockPrice.put("Fiserv", 117);
        stockPrice.put("BMW", 73);
        stockPrice.put("Microsoft", 213);

        stockPrice.forEach((key, value) -> System.out
                .println("Company Name: " + key + " Stock Price: " + value));

    }
}
```

Run Save Reset