

<https://zhuanlan.zhihu.com/p/25108871>

HashMap在遍历的时候，一般使用entrySet()，其原因是entry中包含key和value，一次取出。看懂这个entrySet()源码也是费了一番功夫的，并不是它的代码有多难，而是有一些细小的点上要明白。先看一下JDK的源码，看看具体的实现是怎样的。

```
public Set<Map.Entry<K,V>> entrySet() {  
    return entrySet0();  
}  
  
private Set<Map.Entry<K,V>> entrySet0() {  
    Set<Map.Entry<K,V>> es = entrySet;  
    return es != null ? es : (entrySet = new EntrySet());  
}  
  
private final class EntrySet extends  
AbstractSet<Map.Entry<K,V>> {  
    public Iterator<Map.Entry<K,V>> iterator() {  
        return newEntryIterator();  
    }  
    public boolean contains(Object o) {  
        if (!(o instanceof Map.Entry))  
            return false;  
        Map.Entry<K,V> e = (Map.Entry<K,V>) o;  
        Entry<K,V> candidate = getEntry(e.getKey());  
        return candidate != null && candidate.equals(e);  
    }  
    public boolean remove(Object o) {  
        return removeMapping(o) != null;  
    }  
    public int size() {  
        return size;  
    }  
    public void clear() {  
        HashMap.this.clear();  
    }  
}
```

一般人有疑问的地方是在

```
return es != null ? es : (entrySet = new EntrySet());
```

这行代码，好像没有看到对entrySet变量有过赋值的操作，怎么就有数据了呢。

回答这个问题，先得明白"在debug时看对象时，其实是调用了toString()方法"。本身这个变量是没有值的，在访问的时候，才会去调用iterator()方法，那么这个iterator()方法在什么时候回调用的呢？在父类AbstractCollection的toString()中会有答案。

```
public String toString() {
    Iterator<E> it = iterator();
    if (! it.hasNext())
        return "[]";

    StringBuilder sb = new StringBuilder();
    sb.append('[');
    for (;;) {
        E e = it.next();
        sb.append(e == this ? "(this Collection)" : e);
        if (! it.hasNext())
            return sb.append(']').toString();
        sb.append(', ').append(' ');
    }
}
```

看到没有，这里有一个iterator()方法的调用。

自己可以写一个简单的例子来测试一下，尽管toString()方法是有内容输出，但debug是不会进入的！

```
class TT{
    private String name;
    public TT(String name){
        this.name = name;
    }
}
```

```
/**
 * 测试debug查看对象时是否被调用
 */
@Override
public String toString() {
    System.out.println("输出了...");
    return super.toString();
}
}
public class Test {
    public static void main(String args[]) {
        System.out.println("begin");
        TT tt = new TT("aaa");
        System.out.println("exit");
    }
}
```

debug进入到 System.out.println("exit");这行时，每次把鼠标放到tt这个对象时，console就会输出“输出了...”。