

Java Arrays.sort() ??

1. ??

- `Arrays.sort()` 是 Java 中 对 数组 进行 排序 的 方法。
- 它 对 数组 进行 排序 (in-place) 且 返回 值 为 `void`。

2. ?? ??

```
Arrays.sort(arr);
```

参数	说明
<code>arr</code>	要 排序 的 数组 (类型: <code>int[]</code> , <code>char[]</code> , <code>String[]</code>)

3. void? ??

用法	说明
<code>Arrays.sort(arr)</code>	对 数组 <code>arr</code> 进行 排序
<code>Arrays.sort(arr, fromIndex, toIndex)</code>	对 数组 <code>arr</code> 的 子 数组 进行 排序
<code>Arrays.sort(arr, Comparator)</code>	使用 自定义 比较器 对 数组 <code>arr</code> 进行 排序
<code>int[] sorted = Arrays.sort(arr);</code>	返回 排序 后 的 数组 副本

4. ??

```
int[] numbers = {3, 1, 4};
Arrays.sort(numbers); // numbers 现在 是 [1, 3, 4]

System.out.println(Arrays.toString(numbers)); // [1, 3, 4]
```

5. ?? ?? ?? ?

```
int[] sorted = Arrays.sort(numbers); // 返回已排序的副本: void 返回 int[] 副本 不 修改
```

6. ??: ??? ??? ?? ??? ?????

```
int[] original = {3, 1, 2};
int[] copy = Arrays.copyOf(original, original.length);
Arrays.sort(copy);
```

7. ?? ???

方法	说明
Arrays.sort(arr)	对 arr 进行原地排序
Arrays.sort(arr, Comparator)	对 arr 进行原地排序，使用 Comparator 比较器 (Comparator 接口)
Collections.sort(list)	对 list 进行原地排序 (list 实现 Comparable 接口)