

List <-> Array ? ???

1. List ? Array ???

List 类型	转换方法
List<String>	list.toArray(new String[0])
List<Integer>	list.toArray(new Integer[0])
List<Integer> → int[]	list.stream().mapToInt().toArray()

1) List ? Array (???, ? : Integer, String ?)

- toArray() 方法返回一个 Object 类型的数组。

```
List<String> list = Arrays.asList("a", "b", "c");
String[] arr = list.toArray(new String[0]);
```

- list.toArray(new String[0]): 返回一个 Object 类型的数组。
- new String[0] 表示一个长度为 0 的 String 数组。

```
List<Integer> list = Arrays.asList(1, 2, 3);
Integer[] arr = list.toArray(new Integer[0]);
```

2) List ? Array (???, ? : int)

- Java 中 List<Integer> 不能直接转换为 int[] 数组。
- 需要通过 stream() 方法将 List 转换为 Stream，再使用 mapToInt() 方法转换为 int 流，最后使用 toArray() 方法转换为 int 数组。

```
List<Integer> list = Arrays.asList(1, 2, 3);
int[] arr = new int[list.size()];

for (int i = 0; i < list.size(); i++) {
    arr[i] = list.get(i); // 手动将 List 中的元素复制到数组中
}
```

2. Array ? List ???

	Array	List
<code>String[]</code>		<code>Arrays.asList(arr)</code>
<code>Integer[]</code>		<code>Arrays.asList(arr)</code>
<code>int[]</code>		<code>Arrays.stream(arr).boxed().collect(Collectors.toList())</code>

1) ??? ?? ? List (?: String[], Integer[] ?)

```
String[] arr = {"a", "b", "c"};
List<String> list = Arrays.asList(arr);
```

- `Arrays.asList()`은 Array를 List로 변환해주는 메서드이다.
- List / Array의 차이 → List는 크기가 변할 수 있다.

```
List<String> modifiableList = new ArrayList<>(Arrays.asList(arr));
```

2) ??? ?? ? List (?: int[], double[] ?)

- `int[]`을 `List<Integer>`로 변환하는 방법은?
- `Stream + boxed()`를 사용하면 된다.

```
int[] arr = {1, 2, 3, 4};

List<Integer> list = Arrays.stream(arr) // IntStream 생성
                          .boxed()     // int → Integer (Boxing)
                          .collect(Collectors.toList());
```