

Java DSA Type Conversion Cheat Sheet

int[] <-> Integer[]

```
// int[] <-> Integer[]
int[] intArr = {1, 2, 3};
Integer[] integerArr = Arrays.stream(intArr).boxed().toArray(Integer[]::new);

// Integer[] <-> int[]
Integer[] integerArr = {1, 2, 3};
int[] intArr = Arrays.stream(integerArr).mapToInt(i -> i).toArray();
```

double[] <-> Double[]

```
// double[] <-> Double[]
double[] doubleArr = {1.1, 2.2};
Double[] boxedArr = Arrays.stream(doubleArr).boxed().toArray(Double[]::new);

// Double[] <-> double[]
Double[] boxedArr = {1.1, 2.2};
double[] doubleArr = Arrays.stream(boxedArr).mapToDouble(d -> d).toArray();
```

String[] <-> List<String>

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

// List<String> <-> String[]
List<String> list = List.of("x", "y");
String[] array = list.toArray(new String[0]);
```

int[] <-> List<Integer>

```
// int[] <-> List<Integer>
int[] arr = {1, 2, 3};
List<Integer> list = Arrays.stream(arr).boxed().collect(Collectors.toList());

// List<Integer> <-> int[]
List<Integer> list = List.of(4, 5, 6);
int[] arr = list.stream().mapToInt(i -> i).toArray();
```

Integer[] <-> List<Integer>

```
// Integer[] <-> List<Integer>
Integer[] arr = {1, 2, 3};
```

Java DSA Type Conversion Cheat Sheet

```
List<Integer> list = Arrays.asList(arr);
```

```
// List<Integer> <-> Integer[]
```

```
List<Integer> list = List.of(7, 8, 9);
```

```
Integer[] arr = list.toArray(new Integer[0]);
```

char[] <-> String

```
// char[] <-> String
```

```
char[] chars = {'h', 'e', 'y'};
```

```
String s = new String(chars);
```

```
// String <-> char[]
```

```
String s = "hello";
```

```
char[] chars = s.toCharArray();
```

String <-> List<Character>

```
// String <-> List<Character>
```

```
String s = "java";
```

```
List<Character> list = s.chars().mapToObj(c -> (char) c).collect(Collectors.toList());
```

```
// List<Character> <-> String
```

```
List<Character> list = List.of('j', 'a', 'v', 'a');
```

```
String s = list.stream().map(String::valueOf).collect(Collectors.joining());
```

List<Integer> <-> Set<Integer>

```
// List <-> Set
```

```
List<Integer> list = List.of(1, 2, 2, 3);
```

```
Set<Integer> set = new HashSet<>(list);
```

```
// Set <-> List
```

```
List<Integer> listAgain = new ArrayList<>(set);
```

2D int[][] <-> List<List<Integer>>

```
// 2D array <-> List of Lists
```

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

```
List<List<Integer>> list = Arrays.stream(matrix)
```

```
    .map(row -> Arrays.stream(row).boxed().collect(Collectors.toList()))
```

```
    .collect(Collectors.toList());
```

Java DSA Type Conversion Cheat Sheet

```
// List of Lists <-> 2D array
int[][] matrixBack = list.stream()
    .map(l -> l.stream().mapToInt(i -> i).toArray())
    .toArray(int[][]::new);
```

String (space-separated numbers) <-> int[] / List<Integer>

```
// "1 2 3" <-> int[]
String s = "1 2 3";
int[] arr = Arrays.stream(s.split(" ")).mapToInt(Integer::parseInt).toArray();
```

```
// "1 2 3" <-> List<Integer>
List<Integer> list = Arrays.stream(s.split(" "))
    .map(Integer::parseInt).collect(Collectors.toList());
```