List<String> arraysAsList = Arrays.asList("ONE", "TWO", "THREE");

**String** commaSeparatedString = String.join(",", arraysAsList); assertThat(commaSeparatedString).isEqualTo("ONE,TWO,THREE");

**Second, we'll use the [StringJoiner](https://www.baeldung.com/java-string-joiner) class, which has a constructor that accepts a CharSequence delimiter as a parameter:**

**StringJoiner** stringJoiner = **new** **StringJoiner**(",");

arraysAsList.stream()

.forEach(v -> stringJoiner.add(v));

**String** commaSeparatedString = stringJoiner.toString();

assertThat(commaSeparatedString).isEqualTo("ONE,TWO,THREE");

**String** commaSeparatedUsingCollect = arraysAsList.stream() .collect(Collectors.joining(",")); assertThat(commaSeparatedUsingCollect).isEqualTo("ONE,TWO,THREE");

**String** commaSeparatedObjectToString = arraysAsList.stream() .map(Object::toString) .collect(Collectors.joining(",")); assertThat(commaSeparatedObjectToString).isEqualTo("ONE,TWO,THREE");

**String** commaSeparatedStringValueOf = arraysAsList.stream() .map(String::valueOf) .collect(Collectors.joining(",")); assertThat(commaSeparatedStringValueOf).isEqualTo("ONE,TWO,THREE");

**String** commaSeparatedStringValueOfWithDelimiterPrefixSuffix = arraysAsList.stream() .map(String::valueOf) .collect(Collectors.joining(",", "[", "]"));

**String** commaSeparatedUsingReduce = arraysAsList.stream() .reduce((x, y) -> x + "," + y) .get();