The **StringJoiner** class in **Java 8** is a utility that simplifies the process of **joining strings** with a delimiter, and optionally with a prefix and suffix. It's part of the java.util package and is often used when building delimited strings (like CSV, logs, or human-readable lists).

**🔷 What is StringJoiner?**

Introduced in **Java 8**, StringJoiner is used to **construct a sequence of characters separated by a delimiter** and optionally **start with a prefix** and **end with a suffix**.

**✅ Constructor Variants**

// Only delimiter

StringJoiner joiner1 = new StringJoiner(","); // "A,B,C"

// Delimiter with prefix and suffix

StringJoiner joiner2 = new StringJoiner(",", "[", "]"); // "[A,B,C]"

**🧪 Basic Example**

StringJoiner joiner = new StringJoiner(", ");

joiner.add("Java");

joiner.add("Python");

joiner.add("C++");

System.out.println(joiner); // Output: Java, Python, C++

**🔰 With Prefix and Suffix**

StringJoiner joiner = new StringJoiner(", ", "[", "]");

joiner.add("Java");

joiner.add("Python");

System.out.println(joiner); // Output: [Java, Python]

**🔁 Merge Two StringJoiners**

StringJoiner joiner1 = new StringJoiner(", ");

joiner1.add("Apple").add("Banana");

StringJoiner joiner2 = new StringJoiner(", ");

joiner2.add("Mango").add("Grapes");

joiner1.merge(joiner2); // Adds all elements from joiner2 to joiner1

System.out.println(joiner1); // Output: Apple, Banana, Mango, Grapes

**🔍 Important Methods**

| **Method** | **Description** |
| --- | --- |
| add(CharSequence) | Adds a new element to the joiner |
| merge(StringJoiner) | Merges contents of another joiner |
| toString() | Returns the final joined string |
| setEmptyValue(String) | Sets value to return when no elements are added |
| length() | Returns current length of the string |

**🔃 setEmptyValue()**

StringJoiner joiner = new StringJoiner(", ");

joiner.setEmptyValue("Nothing added");

System.out.println(joiner); // Output: Nothing added

Once add() is called, it switches to normal behavior.

**📌 Practical Use: Manual CSV**

List<String> fields = Arrays.asList("ID", "Name", "Age");

StringJoiner csvLine = new StringJoiner(",");

fields.forEach(csvLine::add);

System.out.println(csvLine); // Output: ID,Name,Age

**🔁 Comparison: StringJoiner vs String.join()**

| **Feature** | **StringJoiner** | **String.join()** |
| --- | --- | --- |
| Type | Object-based | Static utility method |
| Prefix/suffix support | ✅ Yes | ❌ No |
| Customization | More flexible | Simpler for fixed joining |
| Mutability | Can add items dynamically | Requires all items beforehand |

**Example:**

String result = String.join(", ", "A", "B", "C"); // "A, B, C"

**✅ When to Use StringJoiner**

Use it when:

* You need **delimiter + prefix/suffix**
* You're **building a string dynamically**
* You're working inside a **loop** or with **streams**

**💡 Advanced: Using with Collectors.joining()**

List<String> names = Arrays.asList("Tom", "Jerry", "Spike");

String result = names.stream()

.collect(Collectors.joining(", ", "[", "]"));

System.out.println(result); // Output: [Tom, Jerry, Spike]

**🧠 Summary**

| **Feature** | **Detail** |
| --- | --- |
| Package | java.util |
| Introduced in | Java 8 |
| Core Use | Join strings with delimiter (optionally prefix/suffix) |
| Mutable? | ✅ Yes — use add() to build |
| Empty Value | Custom message for empty joiner |
| Similar to | StringBuilder, String.join() |