



## What is String?

String is an **immutable** class in Java. Once created, its value cannot be changed. Any modification (like concatenation) actually creates a **new String object** in memory.<sup>[1]</sup> <sup>[2]</sup>

## Differences: String vs StringBuffer vs StringBuilder

Feature	String	StringBuffer	StringBuilder
<b>Mutability</b>	<b>Immutable</b> (Cannot change)	<b>Mutable</b> (Can change in-place)	<b>Mutable</b> (Can change in-place)
<b>Thread Safety</b>	Thread-safe (by nature of immutability)	<b>Thread-safe</b> (Synchronized methods)	<b>Not</b> thread-safe
<b>Performance</b>	Slow for heavy modifications (creates many objects)	Slower than StringBuilder (due to sync overhead)	<b>Fastest</b> for modifications
<b>Introduced</b>	JDK 1.0	JDK 1.0	JDK 1.5

## Use Cases (When to use which?)

### 1. Use String when:

- The value is **constant** or rarely changes (e.g., config keys, error messages).
- You need to use it as a **Map Key** (immutability guarantees hashCode stability).
- Security is a concern (e.g., database connection URLs, usernames).

### 2. Use StringBuilder when:

- You are manipulating strings in a loop (e.g., building a large CSV or JSON string).
- You are in a **single-threaded** environment (e.g., inside a method).
- *This is the default choice for string manipulation.*<sup>[3]</sup>

### 3. Use StringBuffer when:

- You need to modify a string **shared between multiple threads** concurrently (legacy code or specific concurrency needs).
- Note: In modern development, StringBuffer is rarely used. Even for concurrency, using AtomicReference or locks is often preferred.<sup>[4]</sup> <sup>[5]</sup>



1. <https://www.youtube.com/watch?v=KcthwrborUQ>

2. <https://outcomeschool.com/blog/string-vs-stringbuffer-vs-stringbuilder>
3. <https://www.shiksha.com/online-courses/articles/stringbuilder-vs-string-buffer/>
4. <https://www.digitalocean.com/community/tutorials/string-vs-stringbuffer-vs-stringbuilder>
5. <https://stackoverflow.com/questions/355089/difference-between-stringbuilder-and-stringbuffer>
6. <https://ppl-ai-file-upload.s3.amazonaws.com/web/direct-files/attachments/images/81815274/429036f4-3bc3-4a7a-8c6f-90246e73b9b9/image.jpg>
7. <https://www.teravisiontech.com/blog/java-string-vs-string-builder-vs-string-buffer>
8. <https://stackoverflow.com/questions/2971315/string-stringbuffer-and-stringbuilder>
9. <https://www.baeldung.com/java-string-builder-string-buffer>
10. <https://ppl-ai-file-upload.s3.amazonaws.com/web/direct-files/attachments/images/81815274/24f18c0a-b56e-4862-a619-e2959036a5c2/image.jpg>
11. <https://ppl-ai-file-upload.s3.amazonaws.com/web/direct-files/attachments/images/81815274/97e9ed86-ed11-4035-a18b-1eb05c1bab4e/image.jpg>
12. <https://ppl-ai-file-upload.s3.amazonaws.com/web/direct-files/attachments/images/81815274/87f62423-96f2-4071-9802-8f6699e0ecd8/image.jpg>
13. <https://www.geeksforgeeks.org/java/stringbuffer-vs-stringbuilder/>
14. <https://www.geeksforgeeks.org/java/string-vs-stringbuilder-vs-stringbuffer-in-java/>