StringBuffer & StringBuilder

Swipe ▷▷

🧩Introduction to StringBuffer & StringBuilder

In Java, **StringBuffer** and **StringBuilder** are classes used to create mutable (modifiable) strings.

Unlike **String**, they allow you to change the value of strings without creating a new object.

🧩 Why Not String?

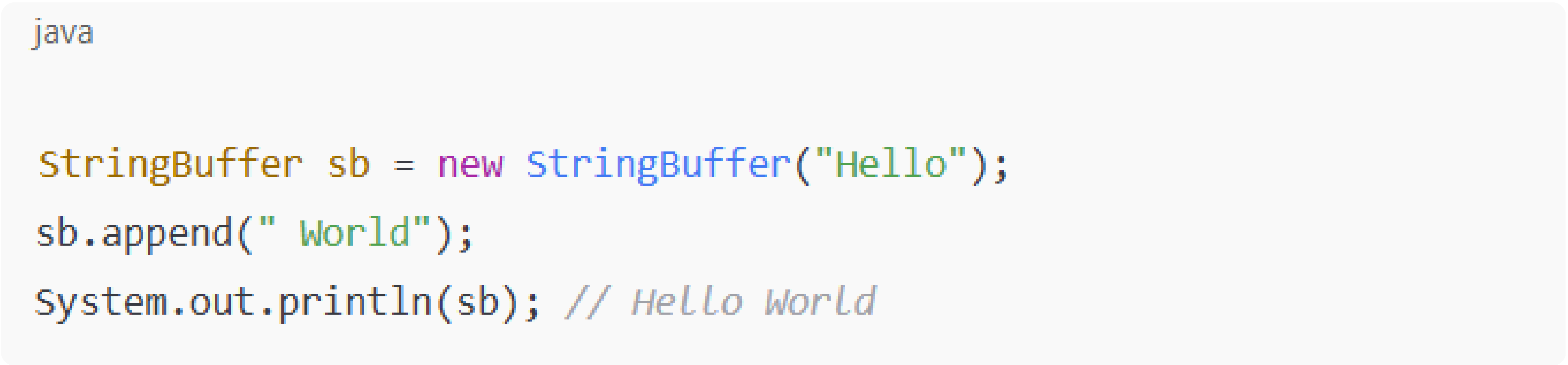
- Strings in Java are immutable.
- Every modification creates a new object in memory.
- Inefficient when you perform frequent modifications like append, insert, delete, etc.

🧩 StringBuffer

✅ Thread-safe (synchronized)

✅ Good for multi-threaded environments

✅ Slower than StringBuilder because of synchronization overhead.



java

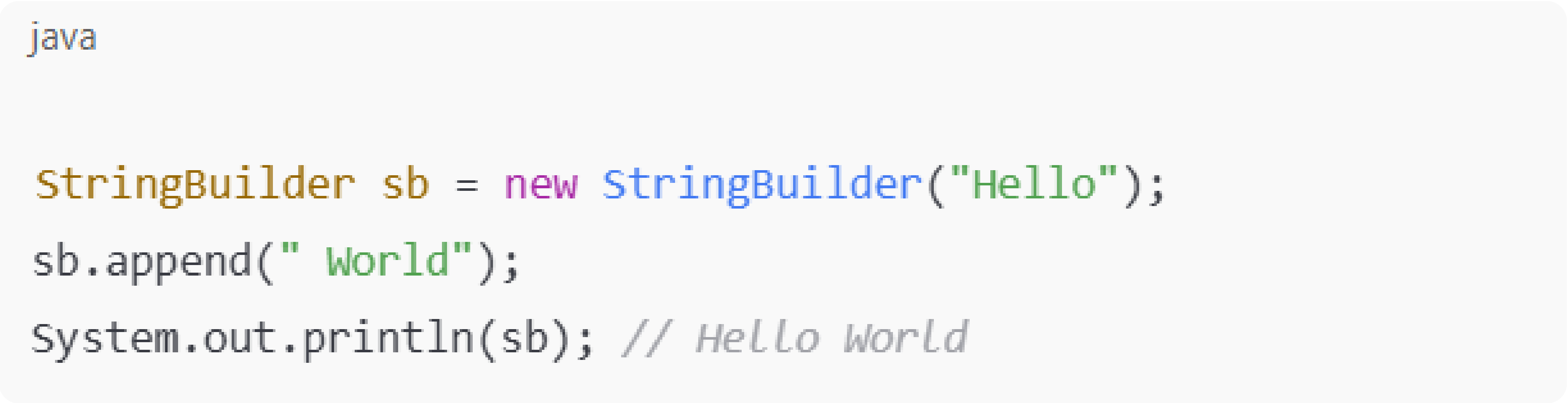
```java
StringBuffer sb = new StringBuffer("Hello");
sb.append(" World");
System.out.println(sb); // Hello World
```

🧩StringBuilder

⚡ Not thread-safe (no synchronization).

⚡ Faster than StringBuffer

⚡ Best choice for single-threaded applications



java

```java
StringBuilder sb = new StringBuilder("Hello");
sb.append(" World");
System.out.println(sb); // Hello World
```

🧩 Key Differences

Feature	**StringBuffer**	**StringBuilder**
Thread Safety	Yes (Synchronized)	No
Performance	Slower	Faster
Use Case	Multi-threaded env.	Single-threaded env.

🧩 When to Use?

✅ Use StringBuffer if working in multi-threaded applications.

✅ Use StringBuilder for better performance in single-threaded apps.

Thank You