String in Java

A **String** in Java is a sequence of characters, represented by the `String` class in `java.lang` package. Strings in Java are **immutable**, meaning their values cannot be changed after they are created.

Types of Strings in Java

There are two main types of Strings in Java:

1. **String (Immutable)**
2. **StringBuilder and StringBuffer (Mutable)**

Ways to Create String Objects in Java

You can create strings in Java in two ways:

Using String Literals

java
Copy code

```java
String str1 = "Hello";
```

1.
    - Stored in the **String Constant Pool**.
    - If a string with the same value exists, it is reused.

Using `new` Keyword

java
Copy code

```java
String str2 = new String("Hello");
```

2.
    - Stored in the **Heap Memory**.
    - Even if a string with the same value exists in the pool, a new object is created.

String Constant Pool

The **String Constant Pool** is a special area in the Java heap memory where string literals are stored.

- When you create a string using literals, Java first checks the pool for an existing instance.

- If it exists, the reference is returned; otherwise, a new string is created.

Mutable and Immutable Objects

Immutable Objects: Once created, their values **cannot be changed**.
Example: `String`
java
Copy code

```
String s = "Hello";
s.concat(" World");  // This does not change 's'
```

-

Mutable Objects: Can be modified after creation.
Examples: `StringBuilder` and `StringBuffer`
java
Copy code

```
StringBuilder sb = new StringBuilder("Hello");
sb.append(" World");  // Modifies 'sb'
```

-

Location of String Constant Pool in Memory

The **String Constant Pool** is part of the **Heap Memory**.

- In Java versions **before Java 7**, it was located in **Permanent Generation (PermGen)**.
- From **Java 7 onwards**, it was moved to the **Heap Space (Metaspace in Java 8+)** for better memory management.