**String class in Java**

In Java, the String class is a **built-in class** that represents a **sequence of characters**.

It is part of the “java.lang” package, so we don’t need to import it explicitly.

**Key Features of the String Class**

* **Immutable**: Once a String object is created, its value **cannot be changed**.
* **Stored in String pool**: Java maintains a pool for strings to optimize memory usage.
* **Final class**: Cannot be extended (i.e., public final class String).
* **Implements**: Serializable, Comparable<String>, and CharSequence.

**Creating Strings**

String s1 = "Hello"; // using string literal

String s2 = new String("Hello"); // using new keyword

Both create strings, but only the first uses the **string pool**.

**Common String Methods**

| **Method** | **Description** | **Example** |
| --- | --- | --- |
| length() | Returns string length | s.length() |
| charAt(index) | Returns character at index | s.charAt(0) → 'H' |
| substring(start, end) | Returns a substring | s.substring(0, 4) → "Hell" |
| equals(str) | Compares content | s.equals("Hello") |
| equalsIgnoreCase(str) | Case-insensitive comparison | s.equalsIgnoreCase("hello") |
| compareTo(str) | Lexicographic comparison | s.compareTo("World") |
| contains(seq) | Checks for substring | s.contains("ell") → true |
| toUpperCase() | Converts to uppercase | s.toUpperCase() |
| toLowerCase() | Converts to lowercase | s.toLowerCase() |
| trim() | Removes whitespace | " hello ".trim() → "hello" |
| replace(a, b) | Replaces characters | s.replace('H', 'J') → "Jello" |
| split(delimiter) | Splits string into array | "a,b,c".split(",") → ["a","b","c"] |
| indexOf(char) | Finds index of first occurrence | s.indexOf('e') |
| isEmpty() | Checks if string is empty | "".isEmpty() → true |

**Immutability Example**

String s = "Hello";

s.concat(" World");

System.out.println(s); // Output: Hello (not Hello World)

//concat() does not change s, because strings are **immutable**.

To change it:

s = s.concat(" World");

System.out.println(s); // Output: Hello World

**String Comparison**

String a = "Java";

//String b= “Java”;

String b = new String("Java");

System.out.println(a == b); // false (different objects)

System.out.println(a.equals(b)); // true (same content)

**String Pool Concept**

String s1 = "Java";

String s2 = "Java";

System.out.println(s1 == s2); // true (same pool reference)

Literal strings are stored in the **string pool** and reused.

**String Class Declaration (Simplified)**

public final class String implements java.io.Serializable,

Comparable<String>,

CharSequence {

// many methods and fields

}