**Common String Methods in Java**

1. **toLowerCase()**
   * **Definition**: Converts all characters in the string to lowercase.
   * **Example**:

java

Copy code

String s = "HELLO";

System.out.println(s.toLowerCase()); // Output: hello

1. **toUpperCase()**
   * **Definition**: Converts all characters in the string to uppercase.
   * **Example**:

java

Copy code

String s = "hello";

System.out.println(s.toUpperCase()); // Output: HELLO

1. **substring(int startIndex, int endIndex)**
   * **Definition**: Returns a new string that is a substring of the original string.
   * **Example**:

java

Copy code

String s = "hello world";

System.out.println(s.substring(0, 5)); // Output: hello

1. **trim()**
   * **Definition**: Removes leading and trailing whitespace from the string.
   * **Example**:

java

Copy code

String s = " hello ";

System.out.println(s.trim()); // Output: hello

1. **split(String regex)**
   * **Definition**: Splits the string into an array of substrings based on the specified regular expression.
   * **Example**:

java

Copy code

String s = "hello world";

String[] parts = s.split(" ");

System.out.println(Arrays.toString(parts)); // Output: [hello, world]

1. **join(CharSequence delimiter, CharSequence... elements)**
   * **Definition**: Joins elements of a collection into a single string with a specified delimiter.
   * **Example**:

java

Copy code

String[] words = {"hello", "world"};

String result = String.join(" ", words);

System.out.println(result); // Output: hello world

1. **replace(CharSequence target, CharSequence replacement)**
   * **Definition**: Replaces all occurrences of a specified substring with another substring.
   * **Example**:

java

Copy code

String s = "hello world";

System.out.println(s.replace("world", "Java")); // Output: hello Java

1. **indexOf(String str)**
   * **Definition**: Returns the index of the first occurrence of a specified substring, or -1 if not found.
   * **Example**:

java

Copy code

String s = "hello world";

System.out.println(s.indexOf("world")); // Output: 6

1. **lastIndexOf(String str)**
   * **Definition**: Returns the index of the last occurrence of a specified substring, or -1 if not found.
   * **Example**:

java

Copy code

String s = "hello world world";

System.out.println(s.lastIndexOf("world")); // Output: 12

1. **contains(CharSequence sequence)**
   * **Definition**: Checks if the string contains the specified sequence of characters.
   * **Example**:

java

Copy code

String s = "hello world";

System.out.println(s.contains("world")); // Output: true

1. **startsWith(String prefix)**
   * **Definition**: Checks if the string starts with the specified prefix.
   * **Example**:

java

Copy code

String s = "hello world";

System.out.println(s.startsWith("hello")); // Output: true

1. **endsWith(String suffix)**
   * **Definition**: Checks if the string ends with the specified suffix.
   * **Example**:

java

Copy code

String s = "hello world";

System.out.println(s.endsWith("world")); // Output: true

1. **equals(Object anObject)**
   * **Definition**: Compares the string to another object for equality.
   * **Example**:

java

Copy code

String s1 = "hello";

String s2 = "hello";

System.out.println(s1.equals(s2)); // Output: true

1. **isEmpty()**
   * **Definition**: Checks if the string is empty (i.e., its length is 0).
   * **Example**:

java

Copy code

String s = "";

System.out.println(s.isEmpty()); // Output: true

1. **length()**
   * **Definition**: Returns the length of the string.
   * **Example**:

java

Copy code

String s = "hello";

System.out.println(s.length()); // Output: 5

**Concepts**

* **String Immutability**
  + **Definition**: Strings in Java are immutable, meaning once a string is created, it cannot be changed.
  + **Example**:

java

Copy code

String s = "hello";

// s[0] = 'H'; // This will cause a compile-time error.

* **Slicing**
  + **Definition**: You can extract parts of a string using the substring() method.
  + **Example**:

java

Copy code

String s = "hello world";

System.out.println(s.substring(6)); // Output: world

* **Concatenation**
  + **Definition**: Joining two or more strings together.
  + **Example**:

java

Copy code

String s1 = "hello";

String s2 = "world";

System.out.println(s1 + " " + s2); // Output: hello world

* **String Formatting**
  + **Definition**: Creating formatted strings using String.format() or System.out.printf().
  + **Example**:

java

Copy code

String name = "Alice";

int age = 30;

System.out.printf("%s is %d years old.\n", name, age); // Output: Alice is 30 years old.

* **Escaping Characters**
  + **Definition**: Using backslashes to include special characters in strings.
  + **Example**:

java

Copy code

String s = "He said, \"Hello!\"";

System.out.println(s); // Output: He said, "Hello!"

* **Raw Strings (Text Blocks)**
  + **Definition**: Multi-line strings introduced in Java 13, which allow for easier formatting of string literals.
  + **Example**:

java

Copy code

String s = """

This is a

multiline string.

""";

System.out.println(s);