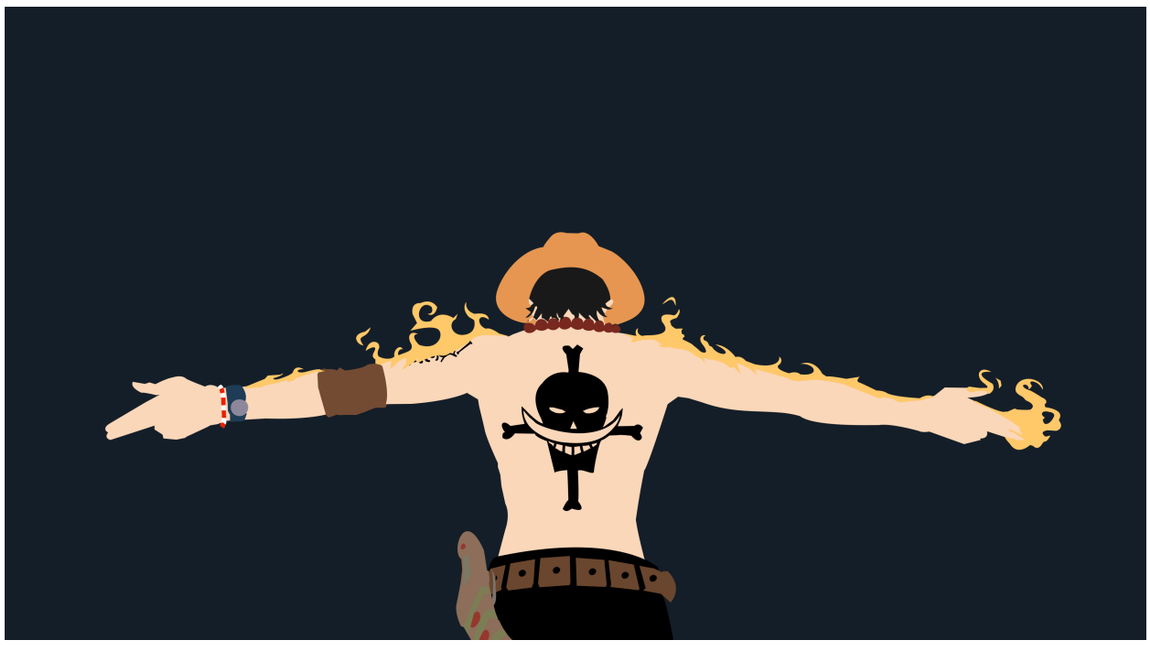

Java - String Methods – List with Uses and Examples

What is String Methods ?

In Java, `String` is a class in `java.lang` package that provides many built-in methods to perform operations on text.

Java String Methods – Quick Reference Table

Method	Use	Example	Output
`length()`	Returns length of string	`"Hello".length()`	`5`
`charAt(i)`	Returns character at index	`"Java".charAt(2)`	`v`
`substring(i)`	Part of string from index	`"Programming".substring(3)`	`gramming`
`substring(i, j)`	Part between indices	`"Programming".substring(0, 6)`	`Progra`
`equals()`	Compares 2 strings (case-sensitive)	`"Java".equals("java")`	`false`
`equalsIgnoreCase()`	Compares ignoring case	`"Java".equalsIgnoreCase("java")`	`true`
`compareTo()`	Lexicographic comparison	`"Apple".compareTo("Banana")`	`-1`
`concat()`	Joins 2 strings	`"Hello".concat(" World")`	`Hello World`

Method	Use	Example	Output
`toUpperCase()`	Converts to uppercase	`"java".toUpperCase()`	`JAVA`
`toLowerCase()`	Converts to lowercase	`"JAVA".toLowerCase()`	`java`
`trim()`	Removes spaces from ends	`" Hello ".trim()`	`Hello`
`replace()`	Replaces character/substring	`"Java".replace('a','@')`	`J@v@`
`contains()`	Checks substring present	`"Hello Java".contains("Java")`	`true`
`startsWith()`	Checks beginning	`"Hello".startsWith("He")`	`true`
`endsWith()`	Checks ending	`"Hello".endsWith("lo")`	`true`
`indexOf()`	First occurrence position	`"programming".indexOf("g")`	`3`
`lastIndexOf()`	Last occurrence position	`"programming".lastIndexOf("g")`	`10`
`isEmpty()`	Checks empty string	`"".isEmpty()`	`true`
`isBlank()` (Java 11+)	Empty or only spaces	`" ".isBlank()`	`true`
`split()`	Splits string by regex	`"a,b,c".split(",")`	`["a","b","c"]`
`valueOf()`	Converts to string	`String.valueOf(100)`	`"100"`
`toCharArray()`	Converts to char array	`"Hi".toCharArray()`	`['H','i']`
`format()`	Returns formatted string	`String.format("Age: %d", 20)`	`Age: 20`

Java String Methods – Uses, Examples & Outputs

1. length()

- **Use**: Returns number of characters.

```
String s = "Hello";
System.out.println(s.length());
// Output: 5
```

2. charAt(int index)

- **Use**: Returns character at given index.

```
String s = "Java";
System.out.println(s.charAt(2));
// Output: v
```

---

3. substring(int beginIndex)

- **Use**: Extracts substring from index to end.

```
String s = "Programming";
System.out.println(s.substring(3));
// Output: gramming
```

---

4. substring(int beginIndex, int endIndex)

- **Use**: Extracts substring between two indices.

```
String s = "Programming";
System.out.println(s.substring(0, 6));
// Output: Progra
```

---

5. equals(Object another)

- **Use**: Checks equality (case-sensitive).

```
String s1 = "Java", s2 = "java";
System.out.println(s1.equals(s2));
// Output: false
```

---

6. equalsIgnoreCase(String another)

- **Use**: Checks equality ignoring case.

```
String s1 = "Java", s2 = "java";
System.out.println(s1.equalsIgnoreCase(s2));
// Output: true
```

---

7. compareTo(String another)

- **Use**: Compares lexicographically.

```
System.out.println("Apple".compareTo("Banana"));
// Output: -1 (because Apple < Banana)
```

---

8. concat(String str)

- **Use**: Joins two strings.

```
String s1 = "Hello", s2 = "World";
System.out.println(s1.concat(" " + s2));
// Output: Hello World
```

---

9. toUpperCase()

- **Use**: Converts to uppercase.

```
String s = "java";
System.out.println(s.toUpperCase());
// Output: JAVA
```

---

10. toLowerCase()

- **Use**: Converts to lowercase.

```
String s = "JAVA";
System.out.println(s.toLowerCase());
// Output: java
```

---

11. trim()

- **Use**: Removes spaces from start and end.

```
String s = "   Hello Java   ";
System.out.println(s.trim());
// Output: Hello Java
```

---

12. replace(char old, char new)

- **Use**: Replaces characters.

```
String s = "Java is fun";
System.out.println(s.replace('a', '@'));
// Output: J@v@ is fun
```

---

13. replace(CharSequence target, CharSequence replacement)

- **Use**: Replaces substrings.

```
String s = "Java is fun";
System.out.println(s.replace("fun", "easy"));
// Output: Java is easy
```

---

14. contains(CharSequence seq)

- **Use**: Checks substring presence.

```
String s = "Learning Java";
System.out.println(s.contains("Java"));
// Output: true
```

---

15. startsWith(String prefix)

- **Use**: Checks if string starts with prefix.

```
String s = "HelloWorld";
System.out.println(s.startsWith("Hello"));
// Output: true
```

---

16. endsWith(String suffix)

- **Use**: Checks if string ends with suffix.

```
String s = "HelloWorld";
System.out.println(s.endsWith("World"));
// Output: true
```

---

17. indexOf(String str)

- **Use**: Finds first occurrence of substring.

```
String s = "programming";
System.out.println(s.indexOf("g"));
// Output: 3
```

---

18. lastIndexOf(String str)

- **Use**: Finds last occurrence of substring.

```
String s = "programming";
System.out.println(s.lastIndexOf("g"));
// Output: 10
```

---

19. isEmpty()

- **Use**: Checks if string length is 0.

```
String s = "";
System.out.println(s.isEmpty());
// Output: true
```

---

20. isBlank() (Java 11+)

- **Use**: Checks if empty or whitespace only.

```
String s = "   ";
System.out.println(s.isBlank());
// Output: true
```

---

21. split(String regex)

- **Use**: Splits string into parts.

```
String s = "apple,banana,grapes";
String[] arr = s.split(",");
for(String fruit : arr) {
    System.out.println(fruit);
```

```
}
```

Output:

```
apple
banana
grapes
```

---

22. valueOf()

- **Use**: Converts value to string.

```
int num = 100;
String s = String.valueOf(num);
System.out.println(s + 50);
// Output: 10050
```

---

23. toCharArray()

- **Use**: Converts to character array.

```
String s = "Hello";
char[] arr = s.toCharArray();
for(char c : arr) {
    System.out.print(c + " ");
}
```

Output:

```
H e l l o
```