The `String` class in Java is *one of the most commonly used classes*, and it comes with **dozens of built-in methods** for manipulating text data.

Let's go step by step, organized by **categories** so it's easier to remember and understand.

1. Basic String Information Methods

Method	Description	Example
`length()`	Returns the number of characters in the string.	`"Hello".length()` → `5`
`charAt(int index)`	Returns the character at the specified index (0-based).	`"Java".charAt(2)` → `'v'`
`isEmpty()`	Checks if the string is empty (`length() == 0`).	`"".isEmpty()` → `true`
`isBlank()` (Java 11+)	Checks if a string is empty or only whitespace.	`" ".isBlank()` → `true`

2. String Comparison Methods

Method	Description	Example
`equals(Object another)`	Compares contents (case-sensitive).	`"Java".equals("java")` → `false`
`equalsIgnoreCase(String another)`	Compares strings, ignoring case.	`"Java".equalsIgnoreCase("java")` → `true`
`compareTo(String another)`	Lexicographically compares two strings (returns int).	`"A".compareTo("B")` → negative
`compareToIgnoreCase(String another)`	Same as above but ignores case.	`"a".compareToIgnoreCase("A")` → `0`

3. Searching Methods

Method	Description	Example
`contains(CharSequence s)`	Checks if substring exists.	`"Hello".contains("ell")` → `true`
`startsWith(String prefix)`	Checks if string starts with prefix.	`"Hello".startsWith("He")` → `true`
`endsWith(String suffix)`	Checks if string ends with suffix.	`"Hello".endsWith("lo")` → `true`
`indexOf(String str)`	Returns index of first occurrence.	`"Hello".indexOf("l")` → 2
`indexOf(String str, int fromIndex)`	Searches from a specific position.	`"Hello".indexOf("l", 3)` → 3
`lastIndexOf(String str)`	Returns index of last occurrence.	`"Hello".lastIndexOf("l")` → 3

4. Case Conversion Methods

Method	Description	Example
`toUpperCase()`	Converts all letters to uppercase.	`"java".toUpperCase()` → `"JAVA"`
`toLowerCase()`	Converts all letters to lowercase.	`"JAVA".toLowerCase()` → `"java"`

5. Substring Methods

Method	Description	Example
`substring(int beginIndex)`	Returns substring from index to end.	`"Hello".substring(2)` → `"llo"`
`substring(int beginIndex, int endIndex)`	Returns substring between indices.	`"Hello".substring(1, 4)` → `"ell"`

6. Concatenation & Joining Methods

Method	Description	Example
`concat(String str)`	Joins two strings.	`"Hello".concat("World")` → `"HelloWorld"`
`join(CharSequence delimiter, CharSequence... elements)`	Joins multiple strings with a delimiter.	`String.join("-", "A", "B", "C")` → `"A-B-C"`
`repeat(int count)` (Java 11+)	Repeats string n times.	`"Hi".repeat(3)` → `"HiHiHi"`

7. Trimming & Whitespace Handling

Method	Description	Example
`trim()`	Removes leading and trailing spaces (but not inner).	`" Hello ".trim()` → `"Hello"`
`strip()` (Java 11+)	Similar to `trim()` but Unicode-aware.	`" Hello ".strip()` → `"Hello"`
`stripLeading()` (Java 11+)	Removes only leading spaces.	`" Hi".stripLeading()` → `"Hi"`
`stripTrailing()` (Java 11+)	Removes only trailing spaces.	`"Hi ".stripTrailing()` → `"Hi"`

8. Replacement Methods

Method	Description	Example
`replace(char oldChar, char newChar)`	Replaces all old chars with new chars.	`"banana".replace('a', 'o')` → `"bonono"`
`replace(CharSequence oldStr, CharSequence newStr)`	Replaces substring.	`"hello".replace("he", "we")` → `"wello"`
`replaceAll(String regex, String replacement)`	Replaces all substrings matching regex.	`"a1b2c3".replaceAll("\\d", "")` → `"abc"`
`replaceFirst(String regex, String replacement)`	Replaces first occurrence matching regex.	`"a1b2c3".replaceFirst("\\d", "")` → `"ab2c3"`

9. Splitting & Joining

Method	Description	Example
`split(String regex)`	Splits string into an array using regex.	`"a,b,c".split(",")` → `["a", "b", "c"]`
`split(String regex, int limit)`	Same as above with limit on parts.	`"a,b,c".split(",", 2)` → `["a", "b,c"]`

10. Conversion Methods

Method	Description	Example
`toCharArray()`	Converts string into character array.	`"Hi".toCharArray()` → `['H','i']`
`valueOf(...)`	Converts other data types to string.	`String.valueOf(123)` → `"123"`
`format(String format, Object... args)`	Creates formatted string.	`String.format("Age: %d", 25)` → `"Age: 25"`

11. Matching Methods (Regex-based)

Method	Description	Example
`matches(String regex)`	Checks if string matches regex pattern.	`"abc123".matches("[a-z]+\\d+")` → `true`
`regionMatches(...)`	Compares regions of two strings.	`"Hello".regionMatches(0, "Heaven", 0, 2)` → `true`

12. Miscellaneous Methods

Method	Description	Example
`intern()`	Returns the canonical (pooled) version of a string.	`new String("Java").intern()`
`hashCode()`	Returns hash code (used in collections).	`"Java".hashCode()`

getBytes()	Converts to byte array using platform encoding.	"ABC".getBytes()
getBytes(Charset charset)	Converts to byte array in specific encoding.	"ABC".getBytes(StandardCharsets.UTF_8)
codePointAt(int index)	Returns Unicode code point at given index.	"A".codePointAt(0) → 65