

Assume you are expert in java programming then provide straight forward all the string methods to cover

Got it 👍 Since you want a **straightforward list of all Java `String` methods** to cover as an expert, here's the complete set you should learn (from `java.lang.String`).

✅ All Important `String` Methods in Java

1. Basic Information

- `int length()` – returns length of string
 - `boolean isEmpty()` – checks if string length is 0
 - `boolean isBlank()` – checks if string is empty or only whitespace (Java 11+)
-

2. Character Access

- `char charAt(int index)` – returns character at index
 - `int codePointAt(int index)` – Unicode code point at index
 - `int codePointBefore(int index)` – Unicode code point before index
 - `int codePointCount(int beginIndex, int endIndex)` – count Unicode code points
 - `int offsetByCodePoints(int index, int codePointOffset)` – index after moving offset code points
-

3. Comparison

- `boolean equals(Object another)`
 - `boolean equalsIgnoreCase(String another)`
 - `int compareTo(String another)`
 - `int compareToIgnoreCase(String another)`
 - `boolean contentEquals(CharSequence cs)`
 - `boolean contentEquals(StringBuffer sb)`
 - `boolean matches(String regex)`
-

4. Searching

- `boolean contains(CharSequence seq)`
- `boolean startsWith(String prefix)`
- `boolean startsWith(String prefix, int toffset)`
- `boolean endsWith(String suffix)`
- `int indexOf(int ch)`
- `int indexOf(int ch, int fromIndex)`
- `int indexOf(String str)`
- `int indexOf(String str, int fromIndex)`
- `int lastIndexOf(int ch)`
- `int lastIndexOf(int ch, int fromIndex)`

- `int lastIndexOf(String str)`
 - `int lastIndexOf(String str, int fromIndex)`
-

5. Substring

- `String substring(int beginIndex)`
 - `String substring(int beginIndex, int endIndex)`
 - `CharSequence subSequence(int beginIndex, int endIndex)`
-

6. Modification (Immutable → Returns New String)

- `String concat(String str)`
- `String replace(char oldChar, char newChar)`
- `String replace(CharSequence target, CharSequence replacement)`
- `String replaceAll(String regex, String replacement)`
- `String replaceFirst(String regex, String replacement)`
- `String strip()` (Java 11+)
- `String stripLeading()` (Java 11+)
- `String stripTrailing()` (Java 11+)
- `String trim()`
- `String repeat(int count)` (Java 11+)

7. Splitting & Joining

- `String[] split(String regex)`
- `String[] split(String regex, int limit)`
- `static String join(CharSequence delimiter, CharSequence... elements)`
- `static String join(CharSequence delimiter, Iterable<? extends CharSequence> elements)`

8. Case Conversion

- `String toLowerCase()`
- `String toLowerCase(Locale locale)`
- `String toUpperCase()`
- `String toUpperCase(Locale locale)`

9. Conversion

- `byte[] getBytes()`
- `byte[] getBytes(Charset charset)`
- `byte[] getBytes(String charsetName)`
- `void getChars(int srcBegin, int srcEnd, char[] dst, int dstBegin)`

- `char[] toCharArray()`
 - `static String valueOf(char[] data)`
 - `static String valueOf(char[] data, int offset, int count)`
 - `static String valueOf(boolean b)`
 - `static String valueOf(char c)`
 - `static String valueOf(int i)`
 - `static String valueOf(long l)`
 - `static String valueOf(float f)`
 - `static String valueOf(double d)`
 - `static String valueOf(Object obj)`
 - `static String copyValueOf(char[] data)`
 - `static String copyValueOf(char[] data, int offset, int count)`
-

10. Interning

- `String intern()` – returns canonical representation (string pool)
-

11. Format & Utility

- `static String format(String format, Object... args)`
- `static String format(Locale l, String format, Object... args)`

12. Lines & Chars (Java 11+)

- `Stream<String> lines()` – returns stream of lines
- `IntStream chars()` – returns stream of int values of chars
- `IntStream codePoints()` – returns Unicode code points

🔥 That's the **full coverage of `String` methods** in Java (up to Java 17).

👉 As an expert, you should practice these with `String`, `StringBuilder`, and `StringBuffer` differences too.

Do you want me to **create a Java notebook/project with small examples for each method** (like we did for other classes)? That way you'll have a ready reference with code + outputs.