

Java String Important Questions with Solutions

1. What is the difference between String, StringBuilder, and StringBuffer?

- String is immutable.
- StringBuilder is mutable and not thread-safe.
- StringBuffer is mutable and thread-safe.

2. Are Strings immutable in Java? Why?

- Yes, Strings are immutable.
- Reason: Security, caching, synchronization, and class loading performance.

3. What is the String pool?

- A special memory region where Java stores string literals.
- Duplicate string literals refer to the same object in the pool.

4. What will be the output of this: "hello" == new String("hello")?

- Output: false
- Because new String("hello") creates a new object in heap memory, while "hello" is from the string pool.

5. Use of substring(), indexOf(), and charAt():

```
String str = "CodingThinker";
```

```
System.out.println(str.substring(0, 6)); // Output: Coding
```

```
System.out.println(str.indexOf('T'));    // Output: 6
```

```
System.out.println(str.charAt(3));       // Output: i
```

6. Count vowels in a string:

```
String input = "Rupesh Kumar";

int count = 0;

for (char c : input.toLowerCase().toCharArray()) {

    if ("aeiou".indexOf(c) != -1) {

        count++;

    }

}

System.out.println("Vowels: " + count); // Output: Vowels: 4
```

7. Check if a string is a palindrome:

```
String str = "racecar";

String rev = "";

for (int i = str.length() - 1; i >= 0; i--) {

    rev += str.charAt(i);

}

System.out.println(str.equals(rev)); // Output: true
```

8. Ways to compare Strings:

```
String s1 = "Java";

String s2 = "java";

System.out.println(s1.equals(s2));          // false

System.out.println(s1.equalsIgnoreCase(s2)); // true

System.out.println(s1.compareTo(s2));       // Negative value
```

9. Replace spaces with '-':

```
String s = "Java is fun";
```

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

10. Reverse a String manually:

```
String s = "Coding";  
  
String rev = "";  
  
for (int i = s.length() - 1; i >= 0; i--) {  
    rev += s.charAt(i);  
}  
  
System.out.println(rev); // Output: gnidoC
```

Prepared by: Rupesh Kumar

www.codingthinker.in