Important String Library Functions in Java

1. `length()`

- **Description:** Returns the length of the string.

Example:
java
Copy code
```
String str = "Hello";
System.out.println(str.length()); // Output: 5
```

-

2. `charAt(int index)`

- **Description:** Returns the character at the specified index.

Example:
java
Copy code
```
String str = "Hello";
System.out.println(str.charAt(1)); // Output: e
```

-

3. `substring(int beginIndex, int endIndex)`

- **Description:** Returns a substring from `beginIndex` (inclusive) to `endIndex` (exclusive).

Example:
java
Copy code
```
String str = "Hello World";
System.out.println(str.substring(0, 5)); // Output: Hello
```

-

4. `contains(CharSequence sequence)`

- **Description:** Checks if the string contains the specified sequence.

Example:
java
Copy code
```
String str = "Hello World";
System.out.println(str.contains("World")); // Output: true
```

-

5. `toLowerCase()` and `toUpperCase()`

- **Description:** Converts the string to lowercase or uppercase.

Example:
java
Copy code
```
String str = "Hello World";
System.out.println(str.toLowerCase()); // Output: hello world
System.out.println(str.toUpperCase()); // Output: HELLO WORLD
```

-

6. `trim()`

- **Description:** Removes leading and trailing whitespace from the string.

Example:
java
Copy code
```
String str = "   Hello World   ";
System.out.println(str.trim()); // Output: Hello World
```

-

7. `equals()` and `equalsIgnoreCase()`

- **Description:** Compares two strings for equality, case-sensitive (`equals()`) or case-insensitive (`equalsIgnoreCase()`).

Example:
java
Copy code
```
String str1 = "Hello";
String str2 = "hello";
System.out.println(str1.equals(str2)); // Output: false
```

```
System.out.println(str1.equalsIgnoreCase(str2)); // Output: true
```

-

8. `replace(char oldChar, char newChar)`

- **Description:** Replaces occurrences of a character with another character.

Example:
java
Copy code
```
String str = "Hello World";
System.out.println(str.replace('o', 'a')); // Output: Hella Warld
```

-

9. `split(String regex)`

- **Description:** Splits the string around matches of the regex.

Example:
java
Copy code
```
String str = "Java is fun";
String[] words = str.split(" ");
for (String word : words) {
    System.out.println(word);
}
// Output:
// Java
// is
// fun
```

-

10. `indexOf()` and `lastIndexOf()`

- **Description:** Finds the first or last occurrence of a character or substring.

Example:
java
Copy code
```
String str = "Hello World";
System.out.println(str.indexOf('o')); // Output: 4
```

```
System.out.println(str.lastIndexOf('o')); // Output: 7
```

-

Frequently Asked String Interview Questions

1. Reverse a String

Problem: Write a program to reverse a string without using the reverse function. **Solution:**

java
Copy code

```
public class ReverseString {
    public static void main(String[] args) {
        String str = "Interview";
        StringBuilder reversed = new StringBuilder();
        for (int i = str.length() - 1; i >= 0; i--) {
            reversed.append(str.charAt(i));
        }
        System.out.println("Reversed String: " + reversed);
    }
}
```

---

2. Check if Two Strings are Anagrams

Problem: Write a program to check if two strings are anagrams.
Solution:

java
Copy code

```
import java.util.Arrays;

public class AnagramCheck {
    public static void main(String[] args) {
        String str1 = "listen";
        String str2 = "silent";

        char[] arr1 = str1.toCharArray();
        char[] arr2 = str2.toCharArray();
```

```
        Arrays.sort(arr1);
        Arrays.sort(arr2);

        if (Arrays.equals(arr1, arr2)) {
            System.out.println("The strings are anagrams.");
        } else {
            System.out.println("The strings are not anagrams.");
        }
    }
}
```

---

3. Count the Frequency of Characters in a String

Problem: Write a program to count the frequency of each character in a string.
Solution:

java
Copy code

```
import java.util.HashMap;

public class CharFrequency {
    public static void main(String[] args) {
        String str = "hello world";
        HashMap<Character, Integer> freq = new HashMap<>();

        for (char c : str.toCharArray()) {
            freq.put(c, freq.getOrDefault(c, 0) + 1);
        }

        System.out.println(freq);
    }
}
```

---

4. Check if a String is a Palindrome

Problem: Write a program to check if a string is a palindrome.
Solution:

java
Copy code

```
public class PalindromeCheck {
    public static void main(String[] args) {
        String str = "madam";
        String reversed = new StringBuilder(str).reverse().toString();

        if (str.equals(reversed)) {
            System.out.println("The string is a palindrome.");
        } else {
            System.out.println("The string is not a palindrome.");
        }
    }
}
```

---

5. Find the Longest Palindromic Substring

Problem: Write a program to find the longest palindromic substring in a string.
Solution:

java
Copy code

```
public class LongestPalindrome {
    public static void main(String[] args) {
        String str = "babad";
        String result = "";

        for (int i = 0; i < str.length(); i++) {
            for (int j = i; j < str.length(); j++) {
                String sub = str.substring(i, j + 1);
                if (isPalindrome(sub) && sub.length() >
result.length()) {
                    result = sub;
                }
            }
        }
```

```
        System.out.println("Longest Palindromic Substring: " +
result);
    }

    public static boolean isPalindrome(String s) {
        return s.equals(new StringBuilder(s).reverse().toString());
    }
}
```

---

6. Count Vowels and Consonants

Problem: Write a program to count vowels and consonants in a string.
Solution:

java
Copy code

```
public class VowelConsonantCount {
    public static void main(String[] args) {
        String str = "Java Programming";
        int vowels = 0, consonants = 0;

        for (char c : str.toLowerCase().toCharArray()) {
            if (c >= 'a' && c <= 'z') {
                if ("aeiou".indexOf(c) != -1) {
                    vowels++;
                } else {
                    consonants++;
                }
            }
        }

        System.out.println("Vowels: " + vowels);
        System.out.println("Consonants: " + consonants);
    }
}
```