Java Script String Methods:

1. **`length`**: Returns the length of the string.
   ```javascript
   let str = "Hello, world!";
   console.log(str.length); // Output: 13
   ```

2. **`charAt(index)`**: Returns the character at the specified index in the string.
   ```javascript
   let str = "Hello, world!";
   console.log(str.charAt(1)); // Output: e
   ```

3. **`charCodeAt(index)`**: Returns the Unicode value (ASCII code) of the character at the specified index.
   ```javascript
   let str = "Hello, world!";
   console.log(str.charCodeAt(0)); // Output: 72 (Unicode for 'H')
   ```

4. **`concat(...strings)`**: Concatenates the given strings and returns a new string.
   ```javascript
   let str1 = "Hello";
   let str2 = "world!";
   let result = str1.concat(", ", str2);
   console.log(result); // Output: Hello, world!
   ```

5. **`endsWith(searchString, length)`**: Checks whether the string ends with the specified search string.
   ```javascript
   let str = "Hello, world!";
   console.log(str.endsWith("world!")); // Output: true
   ```

6. **`includes(searchString, position)`**: Checks whether the string contains the specified search string.
   ```javascript
   let str = "Hello, world!";
   console.log(str.includes("world")); // Output: true
   ```

7. **`indexOf(searchString, fromIndex)`**: Returns the index of the first occurrence of the specified search string.
   ```javascript
   let str = "Hello, world!";
   console.log(str.indexOf("world")); // Output: 7
   ```

8. **`lastIndexOf(searchString, fromIndex)`**: Returns the index of the last occurrence of the specified search string.
   ```javascript
   let str = "Hello, world! Hello!";
   console.log(str.lastIndexOf("Hello")); // Output: 14
   ```

9. **`match(regex)`**: Searches the string for a match using a regular expression and returns the matches.
   ```javascript
   let str = "Hello, world!";
   let regex = /o/g;
   console.log(str.match(regex)); // Output: ["o", "o"]
   ```

10. **`matchAll(regex)`**: Returns an iterator of all results matching the regular expression.
    ```javascript
    let str = "Hello, world!";
    let regex = /o/g;
    let matches = str.matchAll(regex);
    for (const match of matches) {
        console.log(match);
    }
    // Output: ["o"], ["o"]
    ```

11. **`normalize([form])`**: Returns a Unicode normalized string according to the specified form.
    ```javascript
    let str = "\u0041\u0301"; // "A" + combining acute accent
    console.log(str.normalize("NFC")); // Output: Á
    ```

12. **`padEnd(targetLength, padString)`**: Pads the end of the string with another string up to the target length.
    ```javascript
    let str = "Hello";
    console.log(str.padEnd(10, "!")); // Output: Hello!!!!!
    ```

13. **`padStart(targetLength, padString)`**: Pads the start of the string with another string up to the target length.
   ```javascript
   let str = "Hello";
   console.log(str.padStart(10, "!")); // Output: !!!!!Hello
   ```

14. **`repeat(count)`**: Repeats the string a specified number of times.
   ```javascript
   let str = "Hi";
   console.log(str.repeat(3)); // Output: HiHiHi
   ```

15. **`replace(searchValue, replaceValue)`**: Replaces occurrences of the search value with the replace value.
   ```javascript
   let str = "Hello, world!";
   console.log(str.replace("world", "everyone")); // Output: Hello, everyone!
   ```

16. **`replaceAll(searchValue, replaceValue)`**: Replaces all occurrences of the search value with the replace value.
   ```javascript
   let str = "Hello, world! Hello!";
   console.log(str.replaceAll("Hello", "Hi")); // Output: Hi, world! Hi!
   ```

17. **`search(regex)`**: Searches the string for a match using a regular expression and returns the index of the first match.
   ```javascript
   let str = "Hello, world!";
   console.log(str.search(/world/)); // Output: 7
   ```

18. **`slice(beginIndex, endIndex)`**: Extracts a section of the string and returns it as a new string.
   ```javascript
   let str = "Hello, world!";
   console.log(str.slice(7, 12)); // Output: world
   ```

19. **`split(separator, limit)`**: Splits the string into an array using the specified separator.

```javascript
   let str = "Hello, world!";
   console.log(str.split(", ")); // Output: ["Hello", "world!"]
   ```

20. **`startsWith(searchString, position)`**: Checks whether the string starts with the specified search string.
   ```javascript
   let str = "Hello, world!";
   console.log(str.startsWith("Hello")); // Output: true
   ```

21. **`substring(indexStart, indexEnd)`**: Extracts a substring from the string.
   ```javascript
   let str = "Hello, world!";
   console.log(str.substring(7, 12)); // Output: world
   ```

22. **`toLowerCase()`**: Converts the string to lowercase.
   ```javascript
   let str = "Hello, World!";
   console.log(str.toLowerCase()); // Output: hello, world!
   ```

23. **`toUpperCase()`**: Converts the string to uppercase.
   ```javascript
   let str = "Hello, World!";
   console.log(str.toUpperCase()); // Output: HELLO, WORLD!
   ```

24. **`toString()`**: Returns the string representation of the string object.
   ```javascript
   let strObj = new String("Hello, world!");
   console.log(strObj.toString()); // Output: Hello, world!
   ```

25. **`trim()`**: Removes whitespace from both ends of the string.
   ```javascript
   let str = "  Hello, world!  ";
   console.log(str.trim()); // Output: Hello, world!
   ```

26. **`trimStart()`**: Removes whitespace from the start of the string.
   ```javascript
   ```

   let str = " Hello, world!";
   console.log(str.trimStart()); // Output: Hello, world!

27. **`trimEnd()`**: Removes whitespace from the end of the string.
   ```javascript
   let str = "Hello, world!  ";
   console.log(str.trimEnd()); // Output: Hello, world!
   ```

28. **`valueOf()`**: Returns the primitive value of the string object.
   ```javascript
   let strObj = new String("Hello, world!");
   console.log(strObj.valueOf()); // Output: Hello, world!
   ```