JavaScript

String Methods

String Methods are primitive datatypes and
immutable(not changable)
All string methods produce a new string without altering the original string.
There are various types of string methods given below:

1) length :It gives gives length of string elements
.length property in JavaScript counts from 1.
It also counts the spaces.

```
  1)Example : with String
     let userName = "Sam William01";
     console.log("Username",userName.length);

  Output: Username 13
 ----------------------------------------------------------------------
  2) Example : with for loop

  let word = "coding";
  for (let i = 0; i < word.length; i++) {
   console.log(`Character at ${i}:`, word[i]);
 }

 Output :
 Character at 0: c
 Character at 1: o
 Character at 2: d
 Character at 3: i
 Character at 4: n
 Character at 5: g
 ----------------------------------------------------------------------
  3) Example: with empty string

  let emptyStr = "";
     if (emptyStr.length === 0) {
     console.log("String is empty");
 }

 Output :
 String is empty
 -----------------------------------------------------------------------
```

2).charAt() : method in JavaScript returns the character at a specific index in a string.

```
 1) Example :
 let name = "JavaScript";

console.log(name.charAt(0));   // "J"
console.log(name.charAt(4));   // "S"
console.log(name.charAt(12)); // "" (emty space)

 Output:
 J
 S

-----------------------------------------------------------------------

 2) Example : string with []
 let str = "john";
 console.log(str.charAt(1));
 console.log(str[1]);

Output:
j
```

```
j
-----------------------------------------------------------------------
```

3) Example : get last character of string

```
 let str = "sam";
 let lastChar = str.charAt(str.length - 1);
 console.log("Last character:", lastChar);
```

Output:

```
 Last character: m
```

4) (i) Example with function and for loop

this output will give you count

```
 function countVowels(str) {
  let count = 0;
  let vowels = "aeiouAEIOU";

  for (let i = 0; i < str.length; i++) {
    if (vowels.includes(str.charAt(i))) {
      count++;
    }
  }

  return count;
}

console.log(countVowels("Hello World"));
```

Output:

```
3
```

(ii) This will give you vowels letters with count

```
function countVowels(str) {
  let count = 0;
  let vowels = "aeiouAEIOU";
  let found = [];

  for (let i = 0; i < str.length; i++) {
    let char = str.charAt(i);
    if (vowels.includes(char)) {
      count++;
      found.push(char);
    }
  }

  console.log("Vowels found:", found.join(", "));
  return count;
}

console.log("Vowel count:", countVowels("HEllo WOrld"));

Output:
Vowels found: E, o, O
Vowel count: 3
```

3) .toUpperCase() : String is converted to upper case

```
1)Example

let text = "Hello world";
let upperText = text.toUpperCase();

console.log(upperText);
Output: "HELLO WORLD"
```

4) .toLowerCase() : String is converted to lower case

```
1)Example:
let text = "HELLO WORLD";
let upperText = text.LowerCase();
```

```
console.log(upperText);
Output: "hello world"
```

5).contact() : Joins two or more strings

```
Example :
var text1 = "sam " + "john " + "alan ";
var text = text1.concat("shawn ", "chris ");
console.log(text);

Output:
sam john alan shawn chris
```

6) .repeat() : Method returns a string with a number of copies of a string.

```
1) Example:
let str = "ha";

console.log(str.repeat(3));   // "hahaha"
console.log("abc".repeat(2)); // "abcabc"
console.log("!".repeat(5));   // "!!!!!"
console.log("repeat".repeat(0)); // "" (empty string)
```

7) .replace() : method replaces a specified value with another value in a string
it will replace the 1st sentence that matches

```
1)Example :
var text ="quickxpert quick";
var repTxt = text.replace("quick","instant");
console.log(repTxt);
```

```
Output:
instantxpert quick
```

8).replaceAll() : Method is used to replace all occurrences of a substring or pattern in a string with another string.

```
1)Example :
var text ="quickxpert quick";
var repTxt = text.replaceAll("quick","instant");
console.log(repTxt);

Output:
instantxpert instant
```

9) .split() method is used to convert into array.

```
1) Example:
const fruits = "banana,apple,cherry";
//var fruitArray = fruits.split(","); // split on commas
//var fruitArray = fruits.split(" "); // split on spaces
//var fruitArray = fruits.split("");  // it will split each letters
var fruitArray = fruits.split("|");  // it will merge in single array
console.log(fruitArray);

Output:
['banana', 'apple', 'cherry']  commas

['banana,apple,cherry'] spaces are gone

['b', 'a', 'n', 'a', 'n', 'a', ',', 'a', 'p', 'p', 'l', 'e', ',', 'c', 'h', 'e', 'r', 'r', 'y'] split each letters

['banana,apple,cherry'] merge in single array
---------------------------------------------------------------------------------------------------
```

10) .trim() : Method removes whitespaces from both side of strings

```
Example: spaces from start and end
let text1 = "     Hello World!     ";
let text2 = text1.trim();
console.log(text2);

Output:
Hello World!
-------------------------------------------------------------------------------------------------
// https://www.w3resource.com/javascript-exercises/javascript-regexp-exercise-6.php
Example:If you want to remove spaces from middle

let text1 = "     Hello              World!     ";
let text2 = text1.trim().replace(/\s+/g, ' ');
// \s+ --> matches one or more whitespace characters
// g --> global it replaces all matches, not just the first
// , ' ' is the replacement string in the .replace()



console.log(text1.length) // 35
console.log(text2.length); // 12
console.log(text2);// Hello World!
```

11)There are 3types for extracting string

- slice(start, end) Counts from the end
- substring(start, end) Treats them as 0
- substr(start, length)

```
» slice() : extracts a part of a string and returns the extracted part in a new string.
  If start >= end, the result is an empty string

 Example1: with positive values
```

```
let text = "HELLO WORLD";
console.log(text.slice(1,8));// +ve take count from start i.e 0
```

Output:

```
ELLO WO
```

Example 2 : 1st value is -ve, 2nd value is +ve

```
let str = "HELLO WORLD";
console.log(text.slice(-1,8)); // as 1st value is less then 2nd, result empty string
```

Output:

Example 3 : 1st value is +ve 2nd value is -ve. -ve value start from end

```
let str = "HELLO WORLD";
console.log(text.slice(4,-6));// start E ,end will start from D i.e -1,-2...
```

Output:

```
EL
```

Example 4 : Both -ve values

```
let text = "HELLO WORLD";
console.log(text.slice(-11,-6)); // here 1st value is greater than 2nd so no empty
```

Output:

```
HELLO
```

» substring() is similar to slice()

difference is that start and end values less than 0 are treated as 0

Example 1: with +ve values

```
let text = "HELLO WORLD";
console.log(text.substring(0,5));
```

Output:

```
HELLO
```

Example 2: 1st value -ve, 2nd value +ve

```
let text = "HELLO WORLD";
console.log(str.substring(-2, 3));
// substring(-2, 3) => substring(0, 3) => "Hel"
```

Output:

```
Hel
```

Example 3: 1st value is +ve 2nd value is -ve

```
let text = "HELLO WORLD";
console.log(str.substring(3,-2));
// substring(3, -2) => substring(0, 3) --> it swaps the values
```

Output:

```
Hel
```

Example 4: Both -ve values result will be empty

```
let text = "HELLO WORLD";
console.log(str.substring(-3,-2));
```

Output:

» substr() is similar to slice()

difference is that the second parameter specifies the length of the extracted part.

substr() method is removed (deprecated) in the latest JavaScript

Use substring() or slice() instead.

Example1:substr with +ve values

```
let text = "HELLO WORLD";
```

```
console.log(text.substr(1,8));
```

Output:

```
ELLO WOR
```

Example 2: 1st value is -ve 2nd value is +ve

```
let text = "HELLO WORLD";
console.log(text.substr(-3,9));  // total length is 11 i.e 11-3 = 8, start from 8 till end
```

Output:

```
RLD
```