JavaScript String Methods

String Length

- The **length** property returns the length of a string:

Example:

```
<html>
<body>
<script>
var txt = "ABC";
var res = txt.length;
document.write(res);

</script></body> </html>
```

Output: 3

concat

- The concat() Method
- **concat()** joins two or more strings:
- **Example**

```
var text1 = "Hello";
var text2 = "World";
var text3 = text1.concat(" ", text2);
document.write(text3);
```

Output:

```
Hello World!
```

Other method

```
var text = "Hello" + " " + "World!";
document.write(text);
```

- Extracting String Characters

  2 methods:

1. charAt(position)
2. charCodeAt(position)

The charAt() Method

- The **charAt()** method returns the character at a specified index (position) in a string:
- Example

-
  ```
  var str = "HELLO WORLD";
  str.charAt(0);          // returns H
  ```

The charCodeAt() Method

- The **charCodeAt()** method returns the unicode of the character at a specified index in a string:
- Example
- var str = "HELLO WORLD";

  str.charCodeAt(0);   // returns 72

Replace method

- Replacing String Content
- the replace() function replaces **only the first** match
- It is case sensitive
- The **replace()** method replaces a specified value with another value in a string:
- Example
- str = “Gud Morning!";
  var n = str.replace("Morning", “afternoon");

example

```
<html>
<body>

<script>
str = "Gud Morning!"
var n = str.replace("Morning", "afternoon");
document.write(n);
</script>

</body>
</html>
```

Output:

```
Gud afternoon!
```

toUpperCase():

- Converting to Upper and Lower Case
- A string is converted to upper case

example

```
<html>
<body>

<script>
var text1 = "Hello World!";
var text2 = text1.toUpperCase();
document.write(text2);
</script>
</body>
</html>
```

Output:

```
HELLO WORLD!
```

toLowerCase()

```
var text1 = "Hello World!";
 var text2 = text1.toLowerCase();
```

Split()

The split() method is used to split a string into an array of substrings, and returns the new array.

```
<html>
<body>
<script>
    var str = "How are you doing today?";
    var res = str.split(" ");
    document.write(res);
</script></body></html>
```

Output:

```
How,are,you,doing,today?
```

indexOf()

- The **indexOf()** method returns the index of (the position of) the **first** occurrence of a specified text in a string:

Example

```
<script>
var str = "hello world. welcome to this world of
   programming";
var result = str.indexOf("world");
document.write(result);
</script>
```

Output:

```
6
```

- The **lastIndexOf()** method returns the index of the **last** occurrence of a specified text in a string.

```
<script>
var str = "hello world. welcome to this world of programming";
var res = str.lastIndexOf("world");
document.write(res);
</script>
```

Output:

```
29
```

Both the indexOf(), and the lastIndexOf() methods return -1 if the text is not found.

Both methods accept a second parameter as the starting position for the search:

```
<script>
var str = "hello world world";
var position = str.indexOf("world",11);
document.write(position);
</script>
```

Output: 12

Searching for a String in a String

- The **search()** method searches a string for a specified value and returns the position of the match

```
<script>
var str = "hello world";
var res = str.search("world");
document.write(res);
</script>
```

Output: 6

Difference between search and indexof()

The search() method cannot take a second start position argument.

- **Extracting String Parts**
- There are 3 methods for extracting a part of a string:

1. slice(start, end)
2. substring(start, end)
3. substr(start, length)

The slice() Method

- **slice()** extracts a part of a string and returns the extracted part in a new string.
- The method takes 2 parameters: the starting index (position), and the ending index (position).
- This example slices out a portion of a string from position 7 to position 13:
- Example

```
var str = "Apple, Banana, Kiwi";
 var res = str.slice(7, 13);
```

- The output of res will be:
- Banana

- If a parameter is negative, the position is counted from the end of the string.

```
var str = "Apple, Banana, Kiwi";
var res = str.slice(-3);
```

Output

```
iwi
```

Using 1 parameter: slice

```
<script>
var str = "Apple, Banana, Kiwi";
var res = str.slice(7);
document.write(res);
</script>
```

Output:

```
Banana, Kiwi
```

The substring() Method

- **substring()** is similar to slice().
- The difference is that substring() cannot accept negative indexes.

example

```
var str = "Apple, Banana, Kiwi";
   var res = str.substring(7, 13);
```

The result of *res* will be:

Banana

The substr() Method

- **substr()** is similar to slice().
- The difference is that the second parameter specifies the **length** of the extracted part.

Example

```
var str = "Apple, Banana, Kiwi";
var res = str.substr(7, 6);
```

The result of res will be:

Banana