

JavaScript String

The **JavaScript string** is an object that represents a sequence of characters.

There are 2 ways to create string in JavaScript

1. By string literal

The string literal is created using double quotes. The syntax of creating string using string literal is given below:

```
var stringname="string-value";
```

Let's see the simple example of creating string literal.

```
<script>
```

```
var str="This is my string";
```

```
document.write(str);
```

```
</script>
```

2. By string object (using new keyword)

The syntax of creating string object using new keyword is given below:

```
var stringname=new String("string literal");
```

Here, **new keyword** is used to create instance of string.

Let's see the example of creating string in JavaScript by new keyword.

```
<script>
```

```
var str=new String("hello javascript string");
```

```
document.write(str);
```

```
</script>
```

JavaScript String Methods

Let's see the list of JavaScript string methods with examples.

| Methods | Description |
|---------------|---|
| charAt() | It provides the char value present at the specified index. (index starts from 0) |
| charCodeAt() | It provides the Unicode value of a character present at the specified index. |
| concat() | It provides a combination of two or more strings. |
| indexOf() | It provides the position of a char value present in the given string. |
| lastIndexOf() | It provides the position of a char value present in the given string by searching a character from the last position. |
| search() | It searches a specified regular expression in a given string and returns its position if a match occurs. |
| match() | It searches a specified regular expression in a given string and returns that regular expression if a match occurs. |
| replace() | It replaces a given string with the specified replacement. |
| substr() | It is used to fetch the part of the given string on the basis of the specified starting position and length. |
| substring() | It is used to fetch the part of the given string on the basis of the specified index. |
| slice() | It is used to fetch the part of the given string. It allows us to assign positive as well negative index. |
| toLowerCase() | It converts the given string into lowercase letter. |

| | |
|---------------------|---|
| toLocaleLowerCase() | It converts the given string into lowercase letter on the basis of host's current locale. |
| toUpperCase() | It converts the given string into uppercase letter. |
| toLocaleUpperCase() | It converts the given string into uppercase letter on the basis of host's current locale. |
| toString() | It provides a string representing the particular object. |
| valueOf() | It provides the primitive value of string object. |
| split() | It splits a string into substring array, then returns that newly created array. |
| trim() | It trims the white space from the left and right side of the string. |

1. JavaScript String charAt(index) Method

The JavaScript String charAt() method returns the character at the given index.

```
<script>
var str="javascript";
document.write(str.charAt(2));
</script>
```

2. JavaScript String concat(str) Method

The JavaScript String concat(str) method concatenates or joins two strings.

```
<script>
var s1="javascript ";
var s2="concat example";
var s3=s1.concat(s2);
document.write(s3);
</script>
```

3. JavaScript String indexOf(str) Method

The JavaScript String indexOf(str) method returns the index position of the given string.

```
<script>
var s1="javascript function program for string functions";
var n=s1.indexOf("function");
document.write(n);
</script>
```

4. JavaScript String lastIndexOf(str) Method

The JavaScript String lastIndexOf(str) method returns the last index position of the given string.

```
<script>
var s1="javascript from javastring indexof function";
var n=s1.lastIndexOf("java");
document.write(n);
</script>
```

5. JavaScript String slice(beginIndex, endIndex) Method

The JavaScript String slice(beginIndex, endIndex) method returns the parts of string from given beginIndex to endIndex. In slice() method, beginIndex is inclusive and endIndex is exclusive.

```
<script>
var s1="abcdefgh";
var s2=s1.slice(2,5);
document.write(s2);
</script>
```

Output: cde

6. JavaScript String trim() Method

The JavaScript String trim() method removes leading and trailing whitespaces from the string.

```
<script>
var s1="  javascript trim  ";
var s2=s1.trim();
document.write(s2);
</script>
```

7. JavaScript Strings search() Method

search() searches a string for a value and returns the position of the first match:

```
<script>
```

```
var s1="Mr. Blue has a blue house";  
var pos1= s1.search("Blue");  
var pos2= s1.search("blue");  
var pos3= s1.search(/blue/i); // search with /str/ will be a regular expression search.  
document.write(pos1 + "<br />" + pos2 + "<br />" + pos3);
```

```
</script>
```

Output: ?

- The search() method matches a string against a regular expression **
- The search() method returns the index (position) of the first match.
- The search() method returns -1 if no match is found.
- The search() method is case sensitive.

8. JavaScript String match()

- The match() method matches a string against a regular expression **
- The match() method returns an array with the matches.
- The match() method returns *null* if no match is found.

```
<script>
```

```
var s1="Mr. Blue has a blue house";  
var res=s1.match(/ue/g); // global search. gi will be case-insensitive global search
```

```
</script>
```

9. JavaScript String replace()

- The replace() method searches a string for a value or a regular expression.
- The replace() method returns a new string with the value(s) replaced.
- The replace() method does not change the original string.

Syntax:

```
string.replace(searchValue, newValue)
```

<script>

```
var text = "Mr Blue has a blue house and a blue car";  
var result = text.replace(/blue/g, "red"); //global replacement. Try single replacement.  
document.write(result);
```

</script>

10. JavaScript String substr()

- The substr() method extracts a part of a string.
- The substr() method begins at a specified position, and returns a specified number of characters.
- The substr() method does not change the original string.
- To extract characters from the end of the string, use a negative start position.

<script>

```
var text = "Hello world!";  
var result1 = text.substr(1, 4);  
document.write(result1); //ello starting from 1 i.e e since H is 0 and 4 characters  
var result2=str.substr(2);  
document.write(result2); //llo world!  
var result3= result = str.substr(-6); // world!  
document.write(result3);
```

</script>

```

<html>
<head>
  <script>
    <!--
      function testVariable()
      {
        var str1 = document.getElementById("txtName").value;
        var str2 = document.getElementById("txtEmail").value;
        var str3=str1.concat(str2);
        document.getElementById("txtNameEmail").value = ""+str3;
        document.getElementById("p1").innerHTML = str1;
        document.getElementById("p2").innerHTML = str2;
      }
    -->
  </script>
</head>

<body>
  <form>

    Enter Name: <input type="text" name="Name" id="txtName">
    Email address <input type="text" name="Email" id="txtEmail">
    <br />
    <input type="button" value="Submit" id="btnSubmit" onclick="testVariable()">
    <br />
    <input type="text" name="NameEmail" id="txtNameEmail">
  </form>
  <p id="p1"></p>
  <p id="p2"></p>
</body>
</html>

```

Str3's value will be printed on textbox txtName_Email

str1 and str2 will be printed on p1 and p2 respectively in the body