

"JAVASCRIPT"

INTRODUCTION :- JavaScript is a lightweight, interpreted programming language. It is designed for creating network-centric applications. It is complimentary to and integrated with Java. JavaScript is very easy to implement because it is integrated with HTML. It is open and cross-platform. JavaScript is the programming language of HTML and the Web. JavaScript is easy to learn.

CASE SENSITIVITY :- JavaScript is a case-sensitive language. This means that the language keywords, variables, function names, and any other identifiers must always be typed with a consistent capitalization of letters.

SYNTAX :- `<script ...>`

JavaScript code

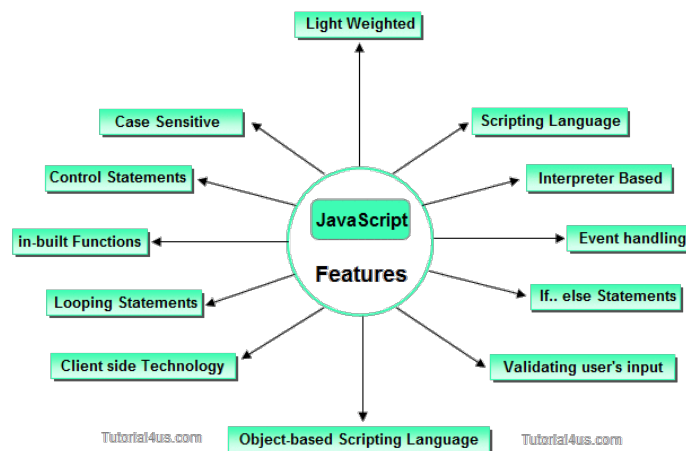
`</script>`

JavaScript can be implemented using JavaScript statements that are placed within the `<script>... </script>` HTML tags in a web page.

You can place the `<script>` tags, containing your JavaScript, anywhere within your web page, but it is normally recommended that you should keep it within the `<head>` tags.

The `<script>` tag alerts the browser program to start interpreting all the text between these tags as a script.

"FEATURES OF JAVASCRIPT"



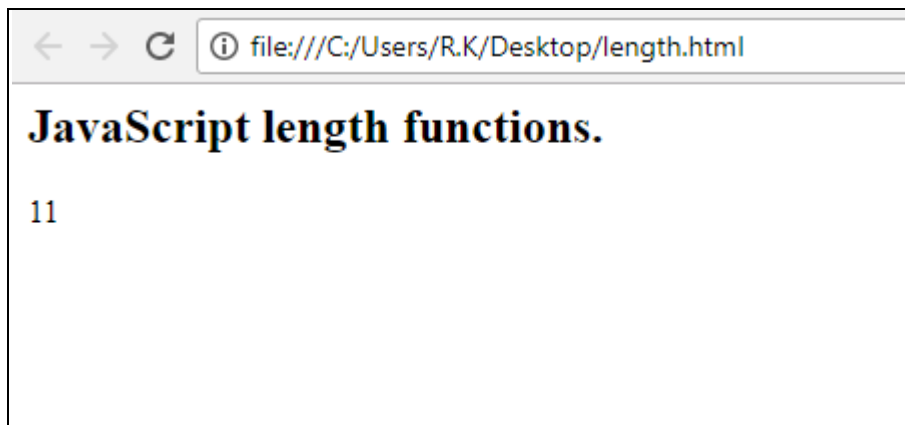
- JavaScript is a object-based scripting language.
- Giving the user more control over the browser.
- It Handling dates and time.
- It Detecting the user's browser and OS,
- It is light weighted.
- JavaScript is a scripting language and it is not java.
- JavaScript is interpreter based scripting language.
- JavaScript is case sensitive.
- JavaScript is object based language as it provides predefined objects.
- Every statement in JavaScript must be terminated with semicolon (;).
- Most of the JavaScript control statements syntax is same as syntax of control statements in C language.
- An important part of JavaScript is the ability to create new functions within scripts. Declare a function in JavaScript using **function** keyword.

WORKING WITH STRING FUNCTIONS.

STRING LENGTH

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

```
<html>
<body>
<h2>JavaScript length functions.</h2>
<p id="demo"></p>
<script>
var txt = "hello world";
document.getElementById("demo").innerHTML = txt.length;
</script>
</body>
</html>
```



FINDING A STRING IN A STRING

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

```
<html>

<body>

<h2>Indexof method</h2>

<p id="demo"></p>

<script>

var str = "nature is very beautiful i love 'nature'";

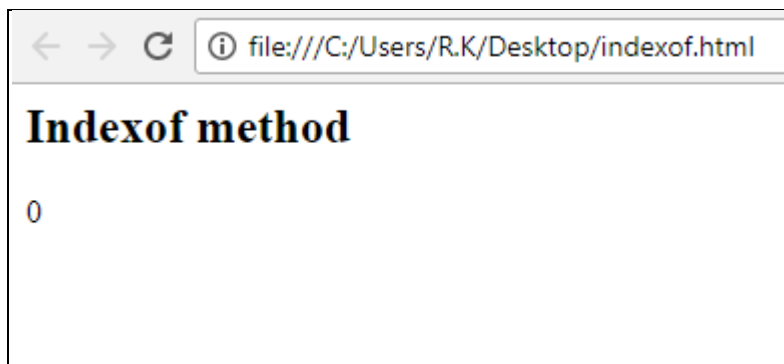
var pos = str.indexOf("nature");

document.getElementById("demo").innerHTML = pos;

</script>

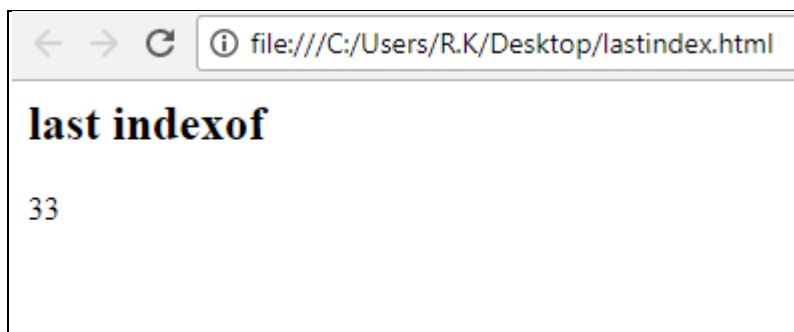
</body>

</html>
```



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

```
<html>
<body>
<h2>last indexof</h2>
<p id="demo"></p>
<script>
var str = "nature is very beautiful i love 'nature'";
var pos = str.lastIndexOf("nature");
document.getElementById("demo").innerHTML = pos;
</script>
</body>
</html>
```



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).

```
<html>

<body>

<h2>Slice function</h2>

<p id="demo"></p>

<script>

var str = "Mango, Apple, Litchi";

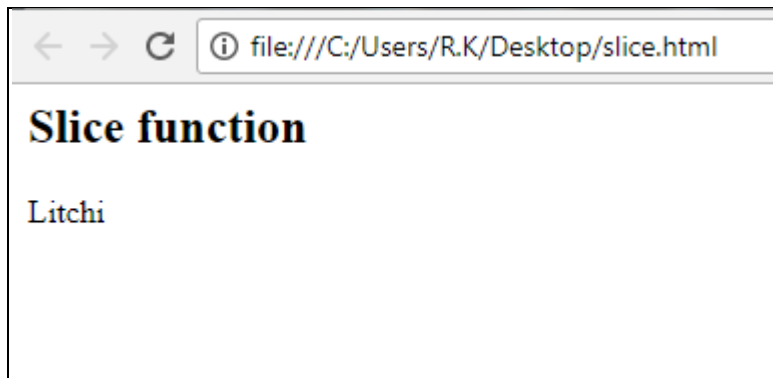
var result = str.slice(14, 20);

document.getElementById("demo").innerHTML = result;

</script>

</body>

</html>
```



The substring() Method

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

```
<html>
```

```
<body>
```

```
<h2>Substring function</h2>
```

```
<p id="demo"></p>
```

```
<script>
```

```
var str = "Mango, Apple, Litchi";
```

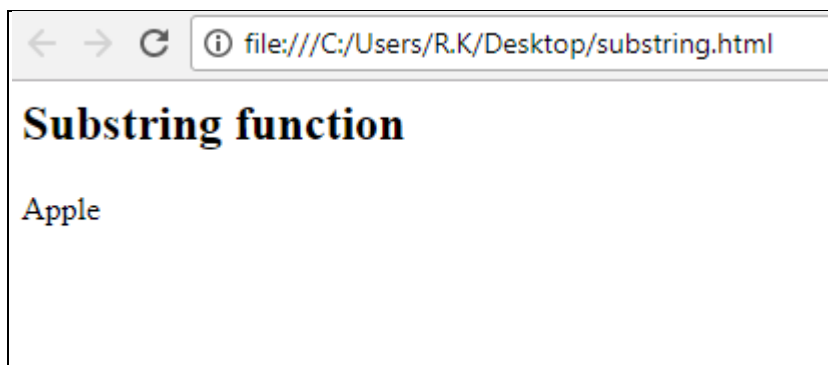
```
var result = str.Substring(7, 12);
```

```
document.getElementById("demo").innerHTML = result;
```

```
</script>
```

```
</body>
```

```
</html>
```



The substr() Method

substr() is similar to slice().

```
<html>
```

```
<body>
```

```
<h2>Substr function</h2>
```

```
<p id="demo"></p>
```

```
<script>
```

```
var str = "Mango, Apple, Litchi";
```

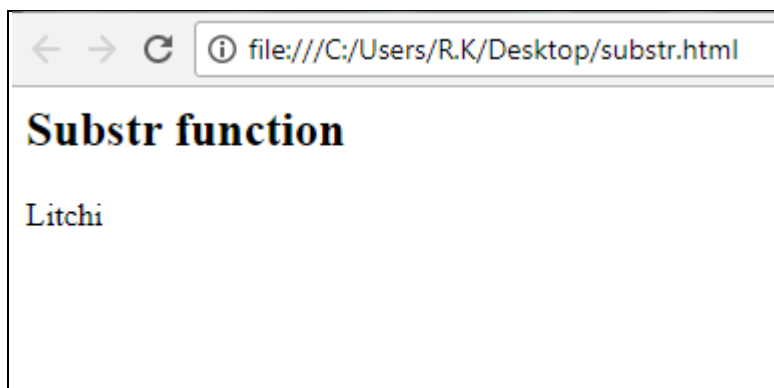
```
var result = str.substr(14, 20);
```

```
document.getElementById("demo").innerHTML = result;
```

```
</script>
```

```
</body>
```

```
</html>
```



Replacing String Content

The `replace()` method replaces a specified value with another value in a string:

```
<html>

<body>

<h2>Replace function.</h2>

<button onclick="myFunction()">Click it</button>

<p id="demo">Please visit my Site.</p>

<script>

function myFunction() {

    var str = document.getElementById("demo").innerHTML;

    var txt = str.replace("Site","Office");

    document.getElementById("demo").innerHTML = txt; }

</script>

</body>

</html>
```



Converting to Upper and Lower Case

A string is converted to upper case with `toUpperCase()`:

```
<html>
<body>
<p>Convert string to upper case:</p>
<button onclick="myFunction()">Try it</button>
<p id="demo">welcome to javascript.</p>
<script>
function myFunction() {
    var text = document.getElementById("demo").innerHTML;
    document.getElementById("demo").innerHTML = text.toUpperCase();
}
</script>
</body>
</html>
```



A string is converted to lower case with `toLowerCase()`:

```
<html>
<body>
<p>Convert string to lower case:</p>
<button onclick="myFunction()">Try it</button>
<p id="demo">WELCOME.</p>
<script>
function myFunction() {
    var text = document.getElementById("demo").innerHTML;
    document.getElementById("demo").innerHTML = text.toLowerCase();
}
</script>
</body>
</html>
```



The concat() Method

concat() **joins two or more strings:**

```
<html>
```

```
<body>
```

```
<h2>Concat function.</h2>
```

```
<p id="demo"></p>
```

```
<script>
```

```
var text1 = "Hello";
```

```
var text2 = "How are you?";
```

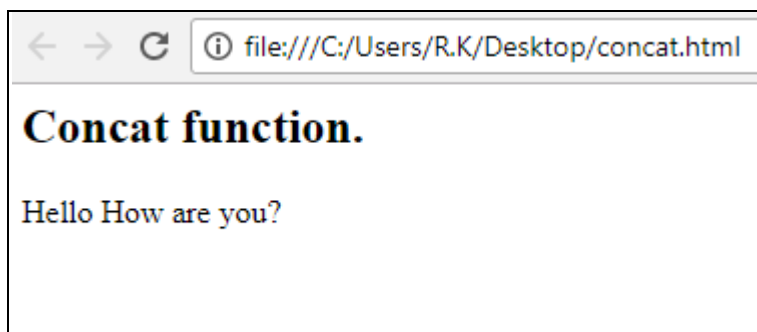
```
var text3 = text1.concat(" ",text2);
```

```
document.getElementById("demo").innerHTML = text3;
```

```
</script>
```

```
</body>
```

```
</html>
```



WORKING WITH DATE FUNCTIONS.

“Displaying Dates”

The Date object lets you work with dates (years, months, days, hours, minutes, seconds, and milliseconds)

```
<html>
<body>
<p id="demo"></p>
<script>
document.getElementById("demo").innerHTML = Date();
</script>
</body>
</html>
```



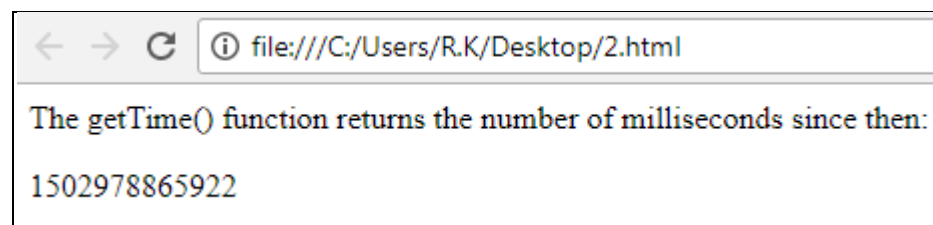
“Date Get Methods”

Get methods are used for getting a part of a date.

Method	Description
getTime()	Get the time (milliseconds since January 1, 1970)
getDay()	Get the weekday as a number (0-6)
getFullYear()	Get the four digit year (yyyy)

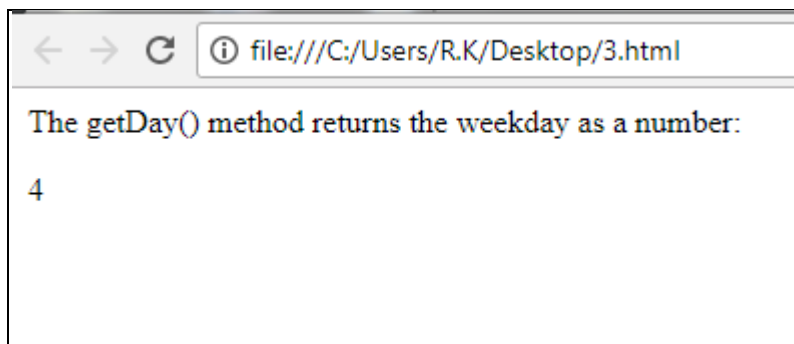
“getTime() method”

```
<html>
<body>
<p>The getTime() function returns the number of milliseconds since then:</p>
<p id="demo"></p>
<script>
var d = new Date();
document.getElementById("demo").innerHTML = d.getTime();
</script>
</body>
</html>
```



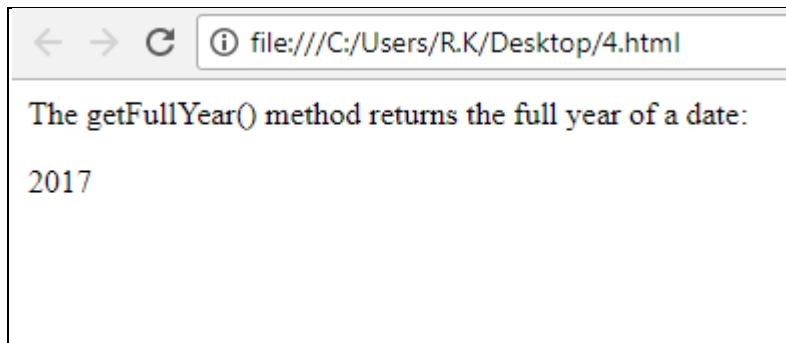
“getDay() method”

```
<html>
<body>
<p>The getDay() method returns the weekday as a number:</p>
<p id="demo"></p>
<script>
var d = new Date();
document.getElementById("demo").innerHTML = d.getDay();
</script>
</body>
</html>
```



“getFullYear() method”

```
<html>
<body>
<p>The getFullYear() method returns the full year of a date:</p>
<p id="demo"></p>
<script>
var d = new Date();
document.getElementById("demo").innerHTML = d.getFullYear();
</script>
</body>
</html>
```



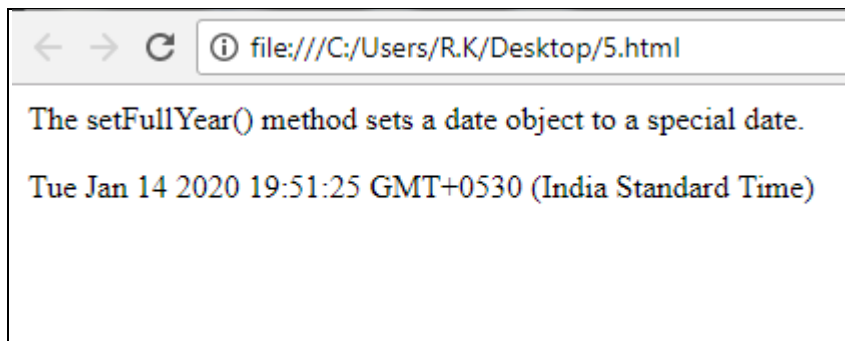
“Date Set Methods”

Set methods are used for setting a part of a date.

“setFullYear() Method”

setFullYear() sets a date object to a specific date.

```
<html>
<body>
<p>The setFullYear() method sets a date object to a special date.</p>
<p id="demo"></p>
<script>
var d = new Date();
d.setFullYear(2020, 0, 14);
document.getElementById("demo").innerHTML = d;
</script>
</body>
</html>
```



“setDate() Method”

setDate() sets the day of the month (1-31):

```
<html>
<body>
<p>The setDate() method sets the date of a month.</p>
<p id="demo"></p>
<script>
var d = new Date();
d.setDate(15);
document.getElementById("demo").innerHTML = d;
</script>
</body>
</html>
```

