<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<meta http-equiv="X-UA-Compatible" content="ie=edge">

<title>Document</title>

<style>

[id\*="demo"] { …

}

.test { …

}

</style>

</head>

<body>

<!-- .test container -->

<div> …

</div>

<!-- [id\*="demo"] container -->

<div>

<div id="demo1"> demo1 : </div>

<div id="demo2"> demo2 : </div>

<div id="demo3"> demo3 : </div>

<div id="demo4"> demo4 : </div>

<div id="demo5"> demo5 : </div>

<div id="demo6"> demo6 : </div>

<div id="demo7"> demo7 : </div>

<div id="demo8"> demo8 : </div>

<div id="demo9"> demo9 : </div>

<div id="demo10"> demo10 : </div>

<div id="demo11"> demo11 : </div>

<div id="demo12"> demo12 : </div>

<div id="demo13"> demo13 : </div>

<div id="demo14"> demo14 : </div>

</div>

<script>

var str = "Hello World Of The World!";

// =================> indexOf() & lastIndexOf()

// indexOf()

var indexOf\_met = str.indexOf('World');

document.getElementById("demo1").innerHTML += indexOf\_met +

"<code> str.indexOf('World'); </code> ";

// lastIndexOf()

var lastIndexOf\_met = str.lastIndexOf("World");

document.getElementById("demo2").innerHTML += lastIndexOf\_met +

"<code> str.lastIndexOf('World'); </code> ";

// second parameter as starting position argument

var x = str.indexOf("World", 10);

document.querySelector("#demo3").innerHTML += x +

" <code>str.indexOf('World',10);</code>";

// ==============> search()

var y = str.search(/wOrlD/gi); // search() accepts RegExp

document.getElementById("demo4").innerHTML += y +

" <code>str.search(/wOrlD/gi);</code> ";

//search method can't take second parameter

// search method accepts RegExp but indexOf can't take RegExp

var z = str.search("Hello");

document.getElementById("demo5").innerHTML += z +

"<code> str.search('Hello');</code>";

//==========> Extracting String Parts

// slice(start, end);

// substring(strat, end);

// substr(start, length);

//==========> slice(start, end);

var res = str.slice(0, 5); // [start, end)

document.getElementById("demo6").innerHTML += res +

' <code>str.slice(0, 5);</code>' ;

//==========> substring(start, end);

// substring() method is same as slice() method, but it can't accept

// negative values.

var substring\_method = str.substring(5, 12);

document.getElementById("demo7").innerHTML += substring\_method +

' <code>str.substring(5, 12);</code>';

// '<code></code>'

//=========> substr(start, length);

// similar to slice(), but the second parameter specifies the length of

the extracted part

var substr\_met = str.substr(12, 6);

document.getElementById("demo8").innerHTML += substr\_met +

' <code>str.substr(12, 6);</code>';

//==========> replace("lastValue", "newValue");

var replace\_met = str.replace(/wOrLd/gi, "boss");

document.getElementById("demo9").innerHTML += replace\_met +

' <code>str.replace(/wOrLd/gi, "boss");</code>';

//=========> toUpperCase() & toLowerCase();

document.getElementById("demo10").innerHTML += str.toUpperCase() +

' <code>str.toUpperCase();</code>';

document.getElementById("demo11").innerHTML += str.toLowerCase() +

' <code>str.toLowerCase();</code>';

//=========> charAt()

var charAt\_met = str.charAt(0);

document.getElementById("demo12").innerHTML += charAt\_met +

' <code>str.charAt(0);</code>';

// var charCodeAt()

// returns the unicode of character at a specified index

var charCodeAt\_met = str.charCodeAt("0");

document.getElementById("demo13").innerHTML += charCodeAt\_met +

' <code>str.charCodeAt("0");</code>';

//=========> split();

var split\_met = str.split(" ");

document.getElementById("demo14").innerHTML += split\_met +

' <code>str.split(" ");</code>' ;

</script>

</body>

</html>