

Q.2 Explain with examples the remaining methods of String and Array.

Ans:-

String methods:

charAt(index) Method :

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

Eg.

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

concat(str) Method :

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

Eg.

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

indexOf(str) Method

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

Eg.

```
<script>
var s1="javascript from javatpoint indexof";
var n=s1.indexOf("from");
document.write(n);
</script>
```

lastIndexOf(str) Method

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

Eg.

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

toLowerCase() Method

The JavaScript String `toLowerCase()` method returns the given string in lowercase letters.

Eg.

```
<script>
var s1="JavaScript toLowerCase Example";
var s2=s1.toLowerCase();
document.write(s2);
</script>
```

toUpperCase() Method

The JavaScript String `toUpperCase()` method returns the given string in uppercase letters.

Eg.

```
<script>
var s1="JavaScript toUpperCase Example";
var s2=s1.toUpperCase();
document.write(s2);
</script>
```

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.

Eg.

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

String trim() Method

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

Eg.

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

String split() Method

Eg.

```
<script>
var str="This is JavaTpoint website";
document.write(str.split(" ")); //splits the given string.
</script>
```

Array Methods.

concat() Method

The JavaScript array `concat()` method combines two or more arrays and returns a new string. This method doesn't make any change in the original array.

Syntax

```
array.concat(arr1,arr2,,,,arrn)
```

eg.

```
<script>
var arr1=["C","C++","Python"];
var arr2=["Java","JavaScript","Android"];
var result=arr1.concat(arr2);
document.writeln(result);
</script>
```

copyWithin() method

The JavaScript array `copyWithin()` method copies the part of the given array with its own elements and returns the modified array. This method doesn't change the length of the modified array.

Syntax

```
array.copyWithin(target, start, end)
```

Eg.

```
<script>
var arr=["AngularJS","Node.js","jQuery","Bootstrap"]
// place at 0th position, the element between 1st and 2nd position.
var result=arr.copyWithin(0,1,2);
document.writeln(result);
```

</script>

entries() Method

The entries() method creates a new iterator object of an array, holding the key/value pairs for every value in the array. A key represents the index number carrying an item as its value. It does not affect the original array.

Syntax

array.entries()

eg.

<html>

<head> <h5> Array Methods </h5>

<body>

 <script>

 var arr=['John','Michael','Embrose','Herry','Lewis'];

 var itr=arr.entries();

 document.write("After applying the entries method:"+"
");

 for(var e of itr) //for loop using var.

 {

 document.write(e+"</br>");

 }

 </script>

</body>

</head>

</html>

every() method

The JavaScript array `every()` method checks whether all the given elements in an array are satisfying the provided condition. It returns `true` when each given array element satisfying the condition otherwise `false`.

Syntax

```
array.every(callback(currentvalue,index,arr),thisArg)
```

eg.

```
<script>
```

```
var marks=[50,40,45,37,20];
```

```
function check(value)
```

```
{
```

```
    return value>30; //return false, as marks[4]=20
```

```
}
```

```
document.writeln(marks.every(check));
```

```
</script>
```

flat() Method

The `flat()` method is an inbuilt array method that flattens a given array into a newly created one-dimensional array. It concatenates all the elements of the given multidimensional array, and flats upto the specified depth. We can specify the depth limit to where we need to flatten the array. By default, the depth limit is 1.

Syntax

```
var newArr=arr.flat(<depth>);
```

eg.

```
<html>
```

```
<head> <h5> Array Methods </h5> </head>
```

```
<body>
<script>
var arr=['a','b',['c','d']]; //given 2D array
var newArr=arr.flat(); //using flat() method
document.write("After flattening the array: "+newArr);
</script>
</body>
</html>
```

flatMap() Method

The flatMap() method is a combination of flat() and map() methods. This method initially maps each array element through a mapping function, then flatten up the array with depth value as 1.

Syntax

```
var newArr=arr.flatMap(function callback(currentValue[ , index[ , array]])
{
return element
} [ , thisArg])
```

Eg.

```
<html>
<head> <h5> Javascript Array Methods </h5> </head>
<body>
<script>
var arr=[2,4,6,8];
document.write(arr.flatMap(x=>[[x/2]]));
</script>
</body>
```

</html>

fill() method

The JavaScript array fill() method fills the elements of the given array with the specified static values. This method modifies the original array. It returns undefined, if no element satisfies the condition.

Syntax

```
arr.fill(value[, start[, end]])
```

eg.

```
<script>
```

```
var arr=["AngularJS","Node.js","jQuery"];
```

```
var result=arr.fill("Bootstrap");
```

```
document.writeln(arr);
```

```
</script>
```

from() Method

The from() method creates a new array that holds the shallow copy from an array or iterable object. When applied to a string, each word gets converted to an array element in the new array.

Syntax

```
Array.from(object,map_fun,thisArg);
```

Eg.

```
<html>
```

```
<head> <h5> JavaScript Array Methods </h5> </head>
```

```
<body>
```

```
<script>
```



```
var arr=Array.from("You are viewing an example of string"); //The string will get converted to an array.
```

```
document.write("The resultant is: <br>" +arr);
```

```
</script>
```

```
</body>
```

```
</html>
```

filter() method

The JavaScript array filter() method filter and extract the element of an array that satisfying the provided condition. It doesn't change the original array.

Syntax

```
array.filter(callback(currentvalue,index,arr),thisArg)
```

eg.

```
<script>
```

```
var marks=[50,40,45,37,20];
```

```
function check(value)
```

```
{
```

```
    return value>30;
```

```
}
```

```
document.writeln(marks.filter(check));
```

```
</script>
```

find() method

The JavaScript array `find()` method returns the first element of the given array that satisfies the provided function condition.

Syntax

```
array.find(callback(value,index,arr),thisArg)
```

eg.

```
<script>
```

```
var arr=[5,22,19,25,34];
```

```
var result=arr.find(x=>x>20);
```

```
document.writeln(result)
```

```
</script>
```

findIndex() method

The JavaScript array `findIndex()` method returns the index of first element of the given array that satisfies the provided function condition. It returns -1, if no element satisfies the condition.

Syntax

```
array.findIndex(callback(value,index,arr),thisArg)
```

eg.

```
<script>
```

```
var arr=[5,22,19,25,34];
```

```
var result=arr.findIndex(x=>x>20);
```

```
document.writeln(result)
```

```
</script>
```

forEach() method

The JavaScript array `forEach()` method is used to invoke the specified function once for each array element.

Syntax

`array.forEach(callback(currentvalue,index,arr),thisArg)`

eg.

```
<script>
```

```
var arr = ["C", "C++", "Python"];
```

```
arr.forEach(function(fetch) {
```

```
    document.writeln(fetch);
```

```
});
```

```
</script>
```

forEach() method

The JavaScript array `forEach()` method is used to invoke the specified function once for each array element.

Syntax

`array.forEach(callback(currentvalue,index,arr),thisArg)`

eg.

```
<script>
```

```
var arr = ["C", "C++", "Python"];
```

```
arr.forEach(function(fetch) {
```

```
    document.writeln(fetch);
```

```
});
```

```
</script>
```

includes() method

The JavaScript array includes() method checks whether the given array contains the specified element. It returns true if an array contains the element, otherwise false.

Syntax

```
array.includes(element,start)
```

eg.

```
<script>
var arr=["AngularJS","Node.js","jQuery"]
var result=arr.includes("AngularJS");
document.writeln(result);
</script>
```

indexOf() method

The JavaScript array indexOf() method is used to search the position of a particular element in a given array. This method is case-sensitive.

The index position of first element in an array is always start with zero. If an element is not present in an array, it returns -1.

```
<script>
var arr=["C","C++","Python","C++","Java"];
var result= arr.indexOf("C++");
document.writeln(result);
</script>
```

toString() Method

The toString() method is used for converting and representing an array into string form. It returns the string containing the specified array elements. Commas separate these elements, and the string does not affect the original array.

Syntax

`array.toString()`

eg.

```
<!DOCTYPE html>
```

```
<html>
```

```
<head> <h3>Array Methods</h3> </br>
```

```
</head>
```

```
<body>
```

```
<script>
```

```
    var arr=['j','a','v','a','T','p','o','i','n','t']; //array elements
```

```
    var str=arr.toString(); //toString() method implementation
```

```
    document.write("After converting into string: "+str);
```

```
</script>
```

```
</body>
```

```
</html>
```

toLocaleString() Method

The `toLocaleString()` method creates a string that represents the elements of an array. It converts the array elements into the string.

Syntax

`array.toLocaleString();`

eg.

```
<html>
```

```
<head> <h5> Javascript Array Methods </h5> </head>
```

```
<body>
```

```
<script>
```

```
var arr=["ram","shyam","Heera"]; // an array is defined.
```

```
var str=arr.toLocaleString(); //using array toLocaleString() method
```

```
document.write("The array is represented in the string form as: "+str); // This will return the  
array elements in the form of string.
```

```
</script>
```

```
</body>
```

```
</html>
```