Array Methods

Method : Array.prototype.at()

Definition: The `at()` method takes an integer value and returns the item at that array index.

Example:

```
var arr1 = [0,10,3,5,6,8,9,7,90,80,20];

     function returnLast(arr){
         return arr.at(-1); //output: 20
     }
     var item1 = returnLast(arr1);
     console.log(item1);
```

Method : Array.prototype.concat()

Definition: The `concat()` method is used to merge two or more arrays.

Example:

```
        var arr1 = ['ball'];
        var arr2 = ['bat'];
        var arr3 = arr1.concat(arr2);
        console.log(arr3);//output: ball,bat
```

Method: Array.prototype.copyWithin():

The `copyWithin()` method copies part of an array to another location in same array and returns it without modifying its length.

Example:

```
var arr1 = ['ball','bat','tenis','cricket'];
        console.log(arr1.copyWithin(1,2));
```

Method: Array.prototype.entries()

This method returns an Arrayobject with key/value pairs but does not change the original array

Example:

```
<p id="demo"></p>
    <script>
        var vehicles = ['suzuki','royal enfield','tvs','activa'];
        var f = vehicles.entries();
        for(let x of f){
            document.getElementById("demo").innerHTML += x;
        }
    </script>
```

Method: Array.prototype.every()

Definition: this method executes a function for each array element and returns true or false

Example:

```
<p id="demo"></p>
   <script>
       var ages1 = [32,33,40,21];
       document.getElementById("demo").innerHTML = ages1.every(checkAge);
       function checkAge(age){
           return age>18;
       }
   </script>
```

Method: Array.prototype.fill()

definition: The fill() method fills specified elements in an array with a value and overwrites the original array

Example:

```
   <script>
       var fruits = ['apple','banana','orange','grapes'];
       console.log(fruits.fill('kiwi',2));
   </script>
```

Method: Array.prototype.filter()

Definition: The filter() method **creates a new array** with all elements that pass the test implemented by the provided function.

Example:

```
<script>
       var fruits = ['apple','banana','orange','grapes'];
       var result = fruits.filter(word => word.length > 5);
       console.log(result);
   </script>
```

Method: Array.prototype.find()

Definition: The find() method returns value of the first element that passes a function

Example:

```
<script>
        var fruits = ['apple','banana','orange','grapes'];
        var result = fruits.find(word => word.length > 5);
        console.log(result);
    </script>
```

Method:Array.prototype.findIndex()

Definition: The findIndex() method executes a function for each array element.

Example:

```
<script>
        var fruits = ['apple','banana','orange','grapes'];
        var result = fruits.findIndex(word => word.length > 5);
        console.log(result);
    </script>
```

Method:Array.prototype.flat()

Definition: The flat() method creates a new array with all sub-array elements concatenated into it recursively up to the specified depth.

Example:

```
<script>
        var fruits = [1,2,3,[12,13]];
        console.log(fruits.flat());
    </script>
```

Method:Array.prototype.flatMap()

Definition: The flatMap() method returns a new array formed by applying a given callback function to each element of the array, and then flattening the result by one level.

Example:

```
    <script>

        var arr1 = [1,2,3,12,13];
        console.log(arr1.flatMap(x => [x*2]));
    </script>
```

Method:Array.prototype.forEach()

Definition: The forEach() method executes a provided function once for each array element

Example:

```
<script>
        var arr1 = ['p','o','o','j','a'];
        arr1.forEach(element => console.log(element));
    </script>
```

Method:Array.prototype.includes()

Definition: The includes() method determines whether an array includes a certain value among its entries, returning true or false as appropriate

Example:

```
<script>
        var fruits = ['apple','banana','orange','grapes'];
        console.log(fruits.includes('apple'));
    </script>
```

Array.prototype.indexOf()

Definition: The indexOf() method returns the first index at which a given element can be found in the array, or -1 if it is not present

Example:

```
<script>
        var fruits = ['apple','banana','orange','grapes'];
        console.log(fruits.indexOf('apple'));
    </script>
```

Array.prototype.join()

Definition: The join() method creates and returns a new string by concatenating all of the elements in an array

Example:

```
<script>
        var arr1 = ['p','o','o','j','a'];
        console.log(arr1.join('.'))
    </script>
```

Method: Array.prototype.keys()

Definition: The keys() method returns an Array object with the keys of an array.

Example:

```
<script>
        var arr = ["A", "B", "C"];
        var iterator = arr.keys();
        for(var key of iterator){
        console.log(key);
        }
    </script>
```

Method: Array.prototype.lastIndexOf()

Definition: The lastIndexOf() method returns the last index (position) of a specified value.

Example:

```
<script>
         var fruits = ['apple','banana','orange','grapes'];
         console.log(fruits.lastIndexOf('orange'));
    </script>
```

Method: Array.prototype.map()

Definition: creates a new array from calling a function for every array element.

Example:

```
    <script>
        var array1 = [2, 4, 5, 10];
        var  map1 = array1.map(x => x * 3);
        console.log(map1);
    </script>
```

Method: Array.prototype.pop()

Definition: method removes the last element of an array

Examples:

```
<script>
         var fruits = ['apple','banana','orange','grapes'];
         console.log(fruits);
         var popped = fruits.pop();
```

```
        console.log(popped);
    </script>
```

Methods: Array.prototype.push()

Definition: method adds new items to the end of an array.

Example:

```
    <script>
         var fruits = ['apple','banana','orange','grapes'];
         console.log(fruits);
         var pushed = fruits.push('kiwi');
         console.log(pushed);
    </script>
```

Methods: Array.prototype.reduce()

Definition: method executes a reducer function for array element and gves sum.

Example:

```
    <script>
        var numbers = [20, 25];
        var initialValue = 0;
        var sumW = numbers.reduce((previousValue, currentValue) => previousValue +
currentValue,initialValue);
        console.log(sumW);
    </script>
```

Array.prototype.reduceRight():

Definition: method executes a reducer function for each array element.

Example:

```
<script>
        var numbers = [[20, 25], [0,1],[6,7]];
        var initialValue = 0;
        var sumW = numbers.reduce((accumulator, currentValue) =>
accumulator.concat(currentValue));
        console.log(sumW);
    </script>
```

Method:Array.prototype.reverse():

Definition: method reverses the order of the elements in an array.

Example:

```
<script>
         var fruits = ['apple','banana','orange','grapes'];
         console.log(fruits.reverse());
    </script>
```

Method:Array.prototype.shift()

Definition: method removes the first item of an array

Example:

```
    <script>
         var fruits = ['apple','banana','orange','grapes'];
         console.log(fruits.shift());
    </script>
```

Method: Array.prototype.slice()

Definition: method returns selected elements in an array, as a new array.

Example:

```
    <script>
         var fruits = ['apple','banana','orange','grapes'];
         console.log(fruits.slice(2));
    </script>
```

Method:Array.prototype.some()

Definition: method checks if any array elements pass a function

Example:

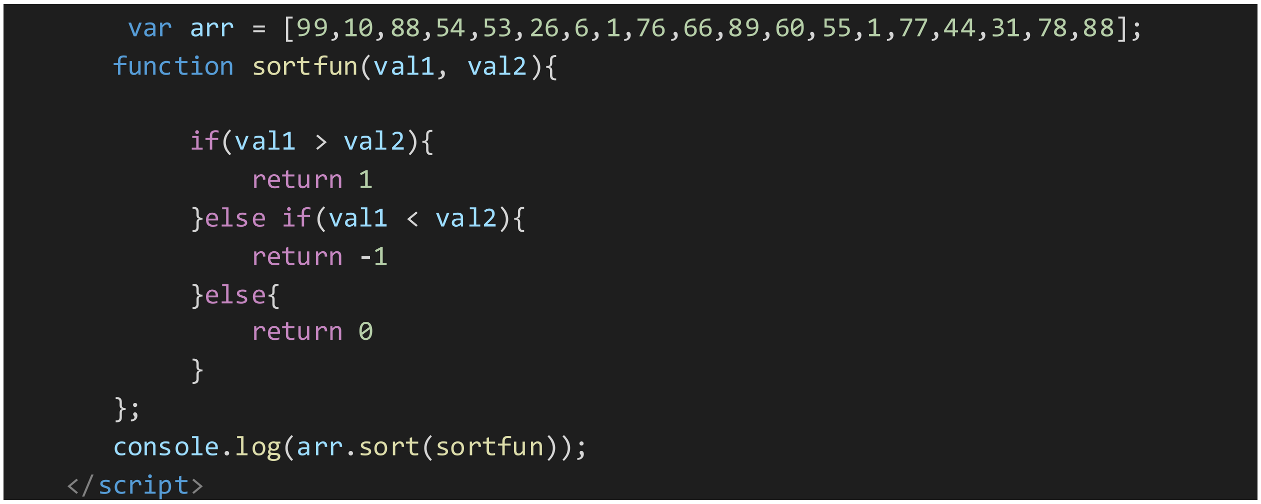
```
    <script>
         var farr1 = [20,10];
         var odd = (element) => element % 2 !== 0;
         console.log(farr1.some(odd));
    </script>
```

Method:Array.prototype.sort()

Definition: sorts the element of an array

Example:

```
    <script>
```



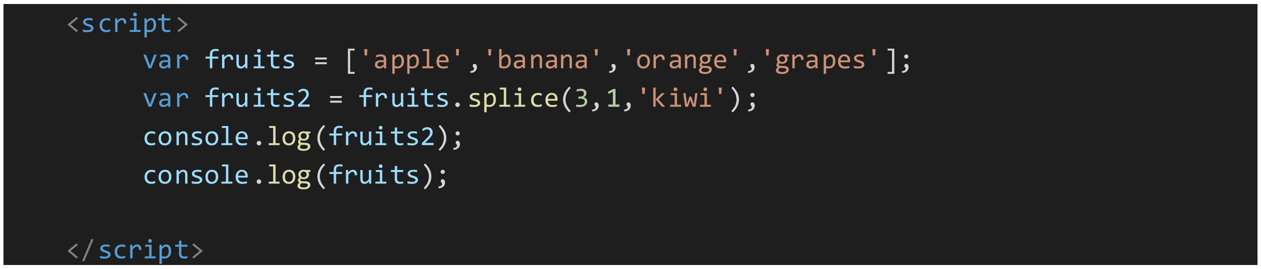
```
    var arr = [99,10,88,54,53,26,6,1,76,66,89,60,55,1,77,44,31,78,88];
   function sortfun(val1, val2){

        if(val1 > val2){
            return 1
        }else if(val1 < val2){
            return -1
        }else{
            return 0
        }
   };
   console.log(arr.sort(sortfun));
</script>
```

Method:Array.prototype.splice()

Definition: The splice() method adds and/or removes array elements.

Example:



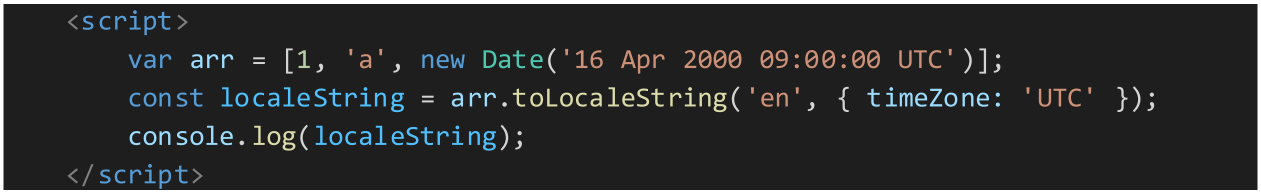
```
<script>
     var fruits = ['apple','banana','orange','grapes'];
     var fruits2 = fruits.splice(3,1,'kiwi');
     console.log(fruits2);
     console.log(fruits);

</script>
```

Array.prototype.toLocaleString()

Definition: The toLocaleString() method returns a string representing the elements of the array.

Example:



```
<script>
    var arr = [1, 'a', new Date('16 Apr 2000 09:00:00 UTC')];
    const localeString = arr.toLocaleString('en', { timeZone: 'UTC' });
    console.log(localeString);
</script>
```

Method:Array.prototype.toString()

Definition: The toString() method returns a string representing the specified array and its elements

Example:

```
    <script>
        const array1 = ['p','o','o','j','a',1];
        console.log(array1.toString());
    </script>
```

Method: Array.prototype.unshift()

Definition: The unshift() method adds one or more elements to the beginning of an array and returns the new length of the array.

Example:

```
    <script>
        var array1 = [1, 2, 3];
        console.log(array1.unshift(4, 5));
        console.log(array1);
    </script>
```

Methods:Array.prototype.values()

Definition: The values() method returns a new *array* object that contains the values for each index in the array

Examples:

```
    <script>
        const array1 = ['apple','banana','orange', ,'grapes'];
        const iterator = array1.values();

        for (const value of iterator) {
        console.log(value);
        }
    </script>
```