Array Type

- Arrays in computer programming are used to **reduce overhead and complexity**.
- JavaScript **Array can store different types of values** in sequential order.
- It can reduce overhead by storing values in sequential order.
- It can reduce complexity by storing multiple values under one name.
- **Array size can be changed dynamically.**
- Array in JavaScript have the behaviour of collections like stack, queue, hash table.

Declaring Array:

- Array can be declared by using
    - Array Meta Character “[]”
    - Array Constructor “Array()”

Ex:

```
<script>
   function f1(){
```

```
    var categories = [];
    var products = new Array();
  }
  f1();
</script>
```

Initialize values into Array:

```
<script>
  function f1(){
    var categories = ["Electronics","Footwear"];
    var products = new Array("Speaker","Nike Causals");
  }
  f1();
</script>
```

Assign Values by using Array Property

- Property is used to map with index number in memory.
- So you can use property to access and or send value into memory.

Ex:

```
<script>
   function f1(){
     var categories = [];
     categories["0"] = "Electronics";
     categories["1"] = "Footwear";
     for(var property in categories) {
        document.write(`${property} : [${typeof property}]<br>`);
     }
   }
   f1();
</script>
```

Array Manipulation

Read Array Elements:

Method	Description
toString()	Returns array elements separated with comma. Ex:

	<script>
	function f1(){
	var products = ["TV", "Mobile", "Shoe"];
	document.write(products.toString());
	}
	f1();
	</script>

join()	Returns array elements separated with custom delimiter.
	
	Ex:
	<script>
	function f1(){
	var products = ["TV", "Mobile", "Shoe"];
	document.write(products.join("-->"));
	}
	f1();
	</script>

slice()	Return array element between specified index.
	
	Ex:
	<script>
	function f1(){
	var products = ["TV", "Mobile", "Shoe"];
	document.write(products.slice(1,2));
	}
	f1();
	</script>

for..of	It reads and return all array elements in sequential order.
	
	Ex:
	<script>
	function f1(){
	var products = ["TV", "Mobile", "Shoe"];
	for(var item of products) {

	document.write(item + "
	");
	}
	}
	f1();
	</script>

for..in	It reads and return all array properties.
	
	Ex:
	<script>
	function f1(){
	var products = ["TV", "Mobile", "Shoe"];
	for(var item in products) {
	document.write(item + "
	");
	}
	}
	f1();
	</script>

for	It uses a loop to read all elements by using initialization, condition and counter.

	Syntax:
	for(initializer, condition, iterator) {
	}
	
	Ex:
	<script>
	function f1(){
	var products = ["TV", "Mobile", "Shoe"];
	for(var i=0; i<products.length; i++) {
	document.write(products[i] + "
	");
	}
	}
	f1();
	</script>

Add Array Elements into HTML Page to present as DOM elements:

- To Add any element you have first create element by using the method document.createElement(“elementName”)
- You can add element by using the method append(), appendChild()

Ex:

```
<!DOCTYPE html>
<html>
  <head>
    <title>Array</title>
    <link rel="stylesheet" href="../node_modules/bootstrap/dist/css/bootstrap.css">
    <script>
      var categories = ["All", "Electronics", "Footwear", "Fashion", "Accessories"];
      function bodyload(){
        var lstCategories = document.getElementById("lstCategories");
```

```
            var optCategories =
document.getElementById("optCategories");
            for(var item of categories) {
               var li = document.createElement("li");
               li.innerHTML = item;
               lstCategories.appendChild(li);

               var option =
document.createElement("option");
               option.text = item;
               option.value = item;
               optCategories.appendChild(option);
            }

         }
      </script>
   </head>
   <body onload="bodyload()" class="container-fluid">
      <div class="form-group">
       <h3>Select a Category</h3>
       <ol id="lstCategories">
```

```
        </ol>
      </div>
      <div class="form-group">
        <h3>Select Category</h3>
        <select class="form-control"
id="optCategories">


        </select>
      </div>
    </body>
</html>
```

Adding and Removing Elements from Array:

Method	Description
push()	Add new elements as last item.
unshift()	Add new elements as first item.
pop()	Remove and return last item.
shift()	Remove and return first item.

splice()	It is used to add or remove item at any specific index. Syntax: splice(startIndex, removeCount, NewItems…)

Ex:

```
<!DOCTYPE html>
<html>
   <head>
      <title>Array</title>
      <link rel="stylesheet" href="../node_modules/bootstrap/dist/css/bootstrap.css">
      <script>
         var categories = ["All", "Electronics", "Footwear"];
         function bodyload(){
            var lstCategories = document.getElementById("lstCategories");
```

```
            var optCategories =
document.getElementById("optCategories");
            lstCategories.innerHTML="";
            optCategories.innerHTML="";
            for(var item of categories) {
               var li = document.createElement("li");
               li.innerHTML = item;
               lstCategories.appendChild(li);

               var option =
document.createElement("option");
               option.text = item;
               option.value = item;
               optCategories.appendChild(option);
            }
         }
         function AddClick(){
            var txtName =
document.getElementById("txtName");
            categories.splice(1,0,txtName.value);
            alert("Item Added");
```

```
            txtName.value="";
            bodyload();
        }
        function RemoveClick(){
            var item = categories.shift();
            alert(`${item} Removed`);
            bodyload();
        }
        function RemoveSelected(){
            var selectedItem =
document.getElementById("optCategories").value;
            var selectedIndex =
categories.indexOf(selectedItem);
            var c = confirm("Are you Sure Want to
Delete?");
            if(c==true) {
                categories.splice(selectedIndex,1);
                bodyload();
            }
        }
    </script>
```

```
    </head>
    <body onload="bodyload()" class="container-fluid">
        <div class="form-group">
            <label>Add Category</label>
            <div>
                <input id="txtName" type="text">
                <button onclick="AddClick()">Add</button>
            </div>
        </div>
        <div class="form-group">
          <h3>Select a Category</h3>
          <ol id="lstCategories">


          </ol>
          <div>
              <button
onclick="RemoveClick()">Remove</button>
          </div>
        </div>
        <div class="form-group">
```

```
        <h3>Select Category</h3>
        <select size="3" class="form-control"
id="optCategories">


        </select>
        <div>
            <button onclick="RemoveSelected()">Remove
Selected</button>
        </div>
      </div>
   </body>
</html>
```

Searching for Elements in Array

indexOf()	It can search for element in array based on given string and returns the “index” number.
lastIndexOf()	It returns the last occurrence index number.

find()	It finds and returns the first occurrence element that matches the given condition.
	
	Ex:
	<script>
	function f1(){
	var sales = [34500, 20000, 45000, 12000, 30000];
	var result = sales.find(function(val){
	return val>30000;
	});
	document.write(result);
	}
	f1();
	</script>

filter()	It finds and returns all elements that matches the given condition. Ex: <script>

	function f1(){
	var sales = [34500, 20000, 45000, 12000, 30000];
	var result = sales.filter(function(val){
	return val<=30000;
	});
	document.write(result.toString());
	}
	f1();
	</script>

Ex:

```
<script>
    function f1(){
        var sales = [34500, 20000, 45000, 12000, 30000];
        function search(val){
            return val>=30000;
        }
        var result = sales.find(search);
        document.write(result);
```

```
    }
    f1();
</script>
```

Sort Array Elements

- sort() arranges elements in ascending order.
- reverse() arranges elements in reverse order [bottom to top]

Ex:

```
<script>
    function f1(){
        var sales = [34500, 20000, 45000, 12000, 30000];
        function search(val){
            return val>=30000;
        }
        var result = sales.filter(search);
        result.sort();
        result.reverse();
        document.write(result.toString());
    }
```

```
    f1();
</script>
```

FAQ:

1. What type of values we can store in array?

You can store any type of value.

2. We store function in Array?

Yes.

Ex:

```
 <script>
    function f1(){
      var methods = [function(){return "Hello
!"}, function(a, b){return a + b}];
      document.write(methods[0]()  + "<br>");
      document.write(methods[1](10,20));
    }
    f1();
</script>
```

3. What is Array Destruction?

It is a technique used to access array elements and store in individual memory references.

Ex:

```
<script>
   function f1(){
      var methods = [function(){return "Hello
!"}, function(a, b){return a + b}];
      //Without Destruction
      var m1 = methods[0];
      var m2 = methods[1];
      document.write(m1() + "<br>");
      document.write(m2(10,30) + "<br>");
      // With Destruction
      var [x1, x2] = methods;
      document.write(x1() + "<br>");
      document.write(x2(10,20));
   }
   f1();
</script>
```

4. **Can we define Array inside Array [Multi Dimension]?**

   Yes.

   Ex:

```
<script>
   function f1(){
      var values = [[10,20],["A","B"]];
      document.write(values[0][1]);
```

```
    }
    f1();
</script>
```