

ASSIGNMENT- 01

MCA-C201: Responsive Web Applications

Name: VIVEK KUMAR MCA(B)

Roll No.: 64

Q1 Demonstrate different types of arrays in JavaScript.

Ans. There are some types of array in javascript.

- 1. **Numeric Arrays:** These are the most common type of arrays in JavaScript, where each element is accessed by its numeric index starting from 0. Numeric arrays can have any number of elements, and the elements can be of any data type

Ex.

```
// Numeric Array

const numericArray = [1, 2, 3, 4, 5]; // An array of numbers

console.log(numericArray[0]); // Output: 1

console.log(numericArray.length); // Output: 5
```

- 1. **String Arrays:** These arrays store elements as strings and can be accessed by their numeric index just like numeric arrays.

Ex.

```
// String Array

const stringArray = ['apple', 'banana', 'cherry', 'date']; // An array of strings

console.log(stringArray[2]); // Output: cherry

console.log(stringArray.length); // Output: 4
```

- 1. **Multi-dimensional Arrays:** These arrays contain arrays as their elements, forming a matrix or a table-like structure.

Ex.

```
// Multi-dimensional Array

const multiDimArray = [

[1, 2, 3],

[4, 5, 6],

[7, 8, 9]

]; // A 2D array of numbers

console.log(multiDimArray[1][2]); // Output: 6
```

- 1. **Array of Objects:** These arrays store objects as their elements, making it possible to create collections of related data.

```
// Array of Objects

const arrayOfObjects = [
```

```
{ name: 'John', age: 25 },  
  
{ name: 'Jane', age: 30 },  
  
{ name: 'Bob', age: 40 }  
]; // An array of objects
```

```
console.log(arrayOfObjects[0].name); // Output: John
```

Q2. Describe and implement DOM and various Java script functions to access DOM elements.

Ans.

### JavaScript - HTML DOM Methods

- HTML DOM methods are actions you can perform (on HTML Elements).
- HTML DOM properties are values (of HTML Elements) that you can set or change.

### The DOM Programming Interface

- The HTML DOM can be accessed with JavaScript (and with other programming languages).
- In the DOM, all HTML elements are defined as objects.
- The programming interface is the properties and methods of each object.
- A property is a value that you can get or set (like changing the content of an HTML element).
- A method is an action you can do (like add or deleting an HTML element).

### Example.

The following example changes the content (the inner HTML) of the <p> element with id="demo":

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

### The getElementById Method

- The most common way to access an HTML element is to use the id of the element.
- In the example above the getElementById method used id="demo" to find the element.

### The inner HTML Property

- The easiest way to get the content of an element is by using the inner HTML property.
- The inner HTML property is useful for getting or replacing the content of HTML elements.
- The inner HTML property can be used to get or change any HTML element, including and

Q3. Design a webpage to implement the following:

1. **Change the background color of BODY on the button click.**

Ans.

```
<!DOCTYPE html>

<html>

<head>

<title>Change Background Color</title>

<script>

function changeColor() {

document.body.style.backgroundColor = 'blue';

}

function changeColor2() {

document.body.style.backgroundColor = 'red';

}

</script>

</head>

<body>

<h1>Change Background Color</h1>

<button onclick="changeColor()">Click to Change Background Color(Blue)</button>

<button onclick="changeColor2()">Click to Change Background Color(Red)</button>

</body>

</html>
```

1. **Create a form with checkboxes and do write Java Script to manipulate checkboxes on checked and Unchecked.**

Ans.

```
<!DOCTYPE html>

<html>

<head>

<title>Checkbox Manipulation</title>

<script>

function onCheckboxChange(checkboxId) {

var checkbox = document.getElementById(checkboxId);

if(checkbox.checked) {
```

```

console.log('Checkbox with ID ' + checkboxId + ' is checked.');
```

```

} else {

console.log('Checkbox with ID ' + checkboxId + ' is unchecked.');
```

```

}

}

</script>

</head>

<body>

<h1>Checkbox Manipulation</h1>

<form>

<input type="checkbox" id="checkbox1" onchange="onCheckboxChange('checkbox1')">

<label for="checkbox1">Checkbox 1</label><br>

<input type="checkbox" id="checkbox2" onchange="onCheckboxChange('checkbox2')">

<label for="checkbox2">Checkbox 2</label><br>

<input type="checkbox" id="checkbox3" onchange="onCheckboxChange('checkbox3')">

<label for="checkbox3">Checkbox 3</label><br>

</form>

</body>

</html>

```

Q4.) Design a Responsive registration form (User Name, Email ID, City (dropdown), Hobbies (Checkbox) and do client-side validation.

Ans.

```

//Index.html

<!DOCTYPE html>

<html>

<head>

<title>Registration Form</title>

<link href="css/style.css" rel="stylesheet" >

<script src="js/val.js"></script>

</head>

<body>

<h1>Registration Form</h1>

```

```
<form id="registrationForm">

<label for="username">User Name:</label>

<input type="text" id="username" name="username">

<label for="email">Email ID:</label>

<input type="email" id="email" name="email">

<label for="city">City:</label>

<select id="city" name="city">

<option value="">--Select City--</option>

<option value="New York">New York</option>

<option value="London">London</option>

<option value="Tokyo">Tokyo</option>

</select>

<label for="hobbies">Hobbies:</label>

<input type="checkbox" id="hobby1" name="hobbies" value="hobby1">Hobby 1

<input type="checkbox" id="hobby2" name="hobbies" value="hobby2">Hobby 2

<input type="checkbox" id="hobby3" name="hobbies" value="hobby3">Hobby 3

<input type="submit" value="Submit">

</form>
```

```
<script src="script.js"></script>
```

```
</body>
```

```
</html>
```

//Style.css file

```
body {

font-family: Arial, sans-serif;

margin: 0;

padding: 20px;

}

form {

max-width: 400px;

margin: 0 auto;

}
```

```
label,

input,

select {

display: block;

margin-bottom: 10px;

}

input[type="submit"] {

background-color: #007bff;

color: #fff;

padding: 10px 15px;

border: none;

cursor: pointer;

}

input[type="submit"]:hover {

background-color: #0069d9;

}
```

//Script.js file

```
document.getElementById('registrationForm').addEventListener('submit', function(event) {

event.preventDefault();

var username = document.getElementById('username').value;

var email = document.getElementById('email').value;

var city = document.getElementById('city').value;

var hobbies = document.getElementsByName('hobbies');

var selectedHobbies = [];

// Loop through checkboxes to get selected hobbies

for (var i = 0; i < hobbies.length; i++) {

if(hobbies[i].checked) {

selectedHobbies.push(hobbies[i].value);

}

}

}

// Perform client-side validation
```

```
if(!username || !email || !city || selectedHobbies.length === 0) {  
    alert('Please fill in all fields and select at least one hobby.');
```

return;

```
}  
  
// Submit the form if validation passes  
  
alert('Registration form submitted successfully!');
```

// Add code here to submit the form data to the server

```
});
```