

# JavaScript Assignment - 07

Name – Soumitra Anil Kode

Roll no. – 42134 , BE- 06, Batch - P6

DOP :

**Git Repo Link :**

**Source Code :**

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1" />
  <title>Experiment 7: Array Object - Creative Style</title>
  <style>
    @import
url('https://fonts.googleapis.com/css2?family=Rubik:wght@400;600&display=sw
ap');
    body {
      margin: 0;
      padding: 20px;
      font-family: 'Rubik', Arial, sans-serif;
      background: linear-gradient(135deg, #f4e8f8, #d5e9f7);
      display: flex;
      justify-content: center;
      min-height: 100vh;
      align-items: center;
      user-select: none;
    }
    .container {
      width: 420px;
      background: rgba(255, 255, 255, 0.82);
      border-radius: 18px;
      padding: 30px 28px 32px 28px;
      box-shadow: 0 12px 36px rgba(148, 128, 187, 0.28);
      text-align: center;
      backdrop-filter: saturate(180%) blur(12px);
      border: 1.8px solid #b5abd4;
      animation: slideFadeIn 0.7s ease forwards;
    }
    @keyframes slideFadeIn {
      from {opacity: 0; transform: translateY(35px);}
      to {opacity: 1; transform: translateY(0);}
    }
    .header {
      background: #b796d8;
      color: #fff;
      border-radius: 9px;
```

```

padding: 21px 14px;
margin-bottom: 22px;
font-size: 1.15em;
font-weight: 600;
letter-spacing: 0.055em;
box-shadow: 0 4px 13px #a25fcc99;
user-select: text;
line-height: 1.4;
}
h2 {
font-size: 1.4em;
font-weight: 700;
margin-bottom: 20px;
color: #593f8a;
letter-spacing: 0.025em;
}
.row {
margin-top: 18px;
}
input[type="number"],
input[type="text"] {
width: 100%;
padding: 11px 14px;
border-radius: 12px;
border: 2px solid #bdb2db;
font-weight: 500;
font-size: 1.06em;
transition: border-color 0.3s, background-color 0.3s;
color: #483b74;
font-family: 'Rubik', Arial, sans-serif;
outline-offset: 2px;
outline-color: #a496c4cc;
user-select: text;
}
input[type="number"]:focus,
input[type="text"]:focus {
border-color: #7e67c0;
background-color: #eae2ff;
outline-color: #7461bbaa;
outline-style: solid;
outline-width: 1.8px;
}
button {
margin-top: 12px;
font-weight: 700;
font-size: 1.07em;
background: linear-gradient(90deg, #8462e1 10%, #6e4dcc 90%);

```

```

        color: #f8f7ff;
        padding: 14px 10px;
        border-radius: 16px;
        border: none;
        box-shadow: 0 4px 24px #6246b670;
        cursor: pointer;
        transition: background-color 0.25s ease, box-shadow 0.35s ease, transform
0.17s ease;
        width: 100%;
        user-select: none;
    }
    button:hover, button:focus {
        background: linear-gradient(90deg, #a07bf4 5%, #805dce 95%);
        box-shadow: 0 6px 29px #715acf92;
        outline: none;
        transform: scale(1.04);
    }
    .hint {
        font-size: 0.84em;
        color: #7869a7;
        margin-top: 6px;
        font-style: italic;
        text-align: left;
        user-select: text;
    }
    .output {
        margin-top: 22px;
        background: #f7f5ff;
        border: 1.5px solid #c4b7e2;
        border-radius: 14px;
        height: 168px;
        overflow-y: auto;
        text-align: left;
        padding: 15px 18px;
        font-family: Consolas, monospace;
        line-height: 1.55;
        color: #5a4d81;
        white-space: pre-wrap;
        user-select: text;
        box-shadow: inset 0 3px 12px #ddd8f6;
    }
</style>
</head>
<body>

```

```

    <main class="container" role="main" aria-label="Array Object Operations
Interface">

```

```
<section class="header" aria-label="Student Information">
  <h3>Soumitra Kode</h3>
  <p>Roll No: 42134 | Div: BE-06 | Batch: P6</p>
</section>
```

```
<h2>Experiment 7: Array Object</h2>
```

```
<section class="row" aria-labelledby="set-size-label">
  <input type="number" id="sizeInput" aria-describedby="size-hint"
placeholder="Enter array size (e.g., 6)" />
  <button onclick="setSize()" aria-live="polite" aria-label="Set array size">Set
Size</button>
  <div id="size-hint" class="hint">Enter a positive numeric size for the
array.</div>
</section>
```

```
<section class="row" id="addSection" style="display:none;" aria-label="Add
Element Section" aria-live="polite">
  <input type="text" id="valueInput" aria-label="Input value to add to array"
placeholder='Enter any object value (e.g:[1,2], {"a":10})' />
  <button onclick="appendElement()" aria-label="Add element to array">Add
Element</button>
  <div class="hint">
    Use valid JSON for arrays/objects. Keys must be enclosed in double quotes.
  </div>
</section>
```

```
<section class="row" id="checkSection" style="display:none;" aria-
live="polite">
  <input type="number" id="indexInput" aria-label="Enter index to check if
element is an array" placeholder="Enter index to check" />
  <button onclick="checkIfArray()" aria-label="Check if element at index is
array">Check If Array</button>
</section>
```

```
<section class="output" id="output" aria-live="polite" aria-
atomic="true"></section>
</main>
```

```
<script>
let myarr = [];
let maxSize = 0;

function setSize() {
  const n = parseInt(document.getElementById("sizeInput").value);
  if (isNaN(n) || n <= 0) {
    showOutput("Please enter a valid positive size!");
  }
}
```

```

        return;
    }
    maxSize = n;
    myarr = [];
    document.getElementById("addSection").style.display = "block";
    document.getElementById("checkSection").style.display = "block";
    showOutput("Size limit = " + maxSize);
    showOutput("Current Array: " + JSON.stringify(myarr));
}

function appendElement() {
    if (maxSize === 0) {
        showOutput("Set the array size first!");
        return;
    }
    if (myarr.length >= maxSize) {
        showOutput("Overflow! Cannot add more than " + maxSize + " elements.");
        return;
    }

    const raw = document.getElementById("valueInput").value.trim();
    if (!raw) {
        showOutput("Please enter a value!");
        return;
    }

    let value;
    try {
        value = JSON.parse(raw);
    } catch {
        value = raw;
    }

    myarr.push(value);
    showOutput("Element added: " + JSON.stringify(value));
    showOutput("Current Array (" + myarr.length + "/" + maxSize + "): " +
JSON.stringify(myarr));
    document.getElementById("valueInput").value = "";
}

function checkIfArray() {
    const idx = parseInt(document.getElementById("indexInput").value);
    if (isNaN(idx) || idx < 0 || idx >= myarr.length) {
        showOutput("Invalid index!");
        return;
    }
    const el = myarr[idx];

```

```
    if (Array.isArray(el)) {
        showOutput("Element at index " + idx + " IS an array: " + JSON.stringify(el));
    } else {
        showOutput("Element at index " + idx + " is NOT an array (" + typeof el + ").
Value: " + JSON.stringify(el));
    }
}
```

```
function showOutput(msg) {
    const out = document.getElementById("output");
    out.innerHTML += msg + "\n\n";
    out.scrollTop = out.scrollHeight;
}
</script>
</body>
</html>
```

Output –

A. Initial Landing page –

The initial landing page features a purple header with the name "Soumitra Kode" and identification details "Roll No: 42134 | Div: BE-06 | Batch: P6". Below this, the title "Experiment 7: Array Object" is displayed. A text input field prompts the user to "Enter array size (e.g., 6)". A prominent purple button labeled "Set Size" is positioned below the input field. A small instructional note, "Enter a positive numeric size for the array.", is located just below the button. The bottom half of the page is occupied by a large, empty light purple rectangular area, likely intended for displaying the array's contents or related visualizations.

B. Adding Element to the array –

a. Input Element.

This section is used for adding a new element to the array. It contains a text input field with the JSON object `{"firstName": "Soumitra", "lastName": "Kode"}` entered. Below the input field is a purple button labeled "Add Element". A note at the bottom states, "Use valid JSON for arrays/objects. Keys must be enclosed in double quotes."

b. Current array.

This section displays the current state of the array. It shows the "Element added:" as `{"firstName": "Soumitra", "lastName": "Kode"}`. Below this, it indicates the "Current Array (1/3):" and shows the array contents as `[{"firstName": "Soumitra", "lastName": "Kode"}]`. The text is presented in a light purple box with a scrollbar on the right side.

C. Getting element at a index –

1

**Check If Array**

```
{ "firstName": "Ishan", "lastName": "Dhotre" },  
{ "firstName": "Lionel", "lastName": "Messi" }]
```

Element at index 1 is NOT an array  
(object). Value:  
{ "firstName": "Ishan", "lastName": "Dhotre" }