

Le **array** (o "matrici") in JavaScript sono oggetti speciali utilizzati per immagazzinare più valori in una singola variabile. A differenza di una variabile normale, che può contenere un solo valore alla volta, un array può contenere molti valori, che possono essere di tipo diverso (stringhe, numeri, oggetti, altre array, ecc.).

## Caratteristiche principali delle array in JavaScript:

1. **Indice numerico:** gli elementi di un array sono accessibili tramite un indice numerico, che parte da 0 (zero).
2. **Possono contenere valori di tipo diverso:** a differenza di altri linguaggi di programmazione, un array in JavaScript può contenere sia numeri che stringhe o altri tipi di dati.
3. **Lunghezza variabile:** puoi aggiungere o rimuovere elementi a un array in qualsiasi momento, e la sua lunghezza si aggiornerà automaticamente.

## Creazione di un Array in JavaScript:

In JavaScript, puoi creare un array in vari modi:

### 1. Array con valori definiti

```
let numeri = [1, 2, 3, 4, 5];  
console.log(numeri); // [1, 2, 3, 4, 5]
```

### 2. Array vuoto

```
let vuoto = [];  
console.log(vuoto); // []
```

### 3. Array con diversi tipi di dati

```
let miscela = [1, 'due', true, [1, 2], {nome: 'Mario'}];  
console.log(miscela); // [1, "due", true, [1, 2], {nome: "Mario"}]
```

## Accesso agli Elementi di un Array:

Gli elementi in un array si possono accedere usando l'indice. Ricorda che l'indice parte da **0**.

```
let frutta = ['mela', 'banana', 'ciliegia'];  
console.log(frutta[0]); // "mela"  
console.log(frutta[1]); // "banana"  
console.log(frutta[2]); // "ciliegia"
```

Se provi a usare un indice che non esiste, come `frutta[3]`, JavaScript restituirà `undefined`.

## Modifica degli Elementi di un Array:

Puoi modificare un elemento in un array assegnando un nuovo valore all'indice desiderato.

```
let colori = ['rosso', 'verde', 'blu'];  
colori[1] = 'giallo'; // Modifico l'elemento all'indice 1  
console.log(colori); // ['rosso', 'giallo', 'blu']
```

## Lunghezza di un Array:

Ogni array ha una proprietà `length`, che ti indica il numero di elementi che contiene.

```
let numeri = [1, 2, 3, 4];  
console.log(numeri.length); // 4
```

## Aggiungere e Rimuovere Elementi da un Array:

JavaScript fornisce vari metodi per aggiungere e rimuovere elementi dagli array.

### 1. `push()` - Aggiunge uno o più elementi alla fine dell'array

```
let frutta = ['mela', 'banana'];  
frutta.push('ciliegia'); // Aggiunge "ciliegia" alla fine  
console.log(frutta); // ["mela", "banana", "ciliegia"]
```

### 2. `pop()` - Rimuove l'ultimo elemento dell'array

```
let frutta = ['mela', 'banana', 'ciliegia'];  
frutta.pop(); // Rimuove l'ultimo elemento ("ciliegia")  
console.log(frutta); // ["mela", "banana"]
```

### 3. `shift()` - Rimuove il primo elemento dell'array

```
let frutta = ['mela', 'banana', 'ciliegia'];  
frutta.shift(); // Rimuove il primo elemento ("mela")  
console.log(frutta); // ["banana", "ciliegia"]
```

### 4. `unshift()` - Aggiunge uno o più elementi all'inizio dell'array

```
let frutta = ['banana', 'ciliegia'];  
frutta.unshift('mela'); // Aggiunge "mela" all'inizio  
console.log(frutta); // ["mela", "banana", "ciliegia"]
```

## Iterazione su un Array:

Puoi iterare (scorrere) gli elementi di un array usando diversi metodi come `for`, `forEach`, `map`, etc.

### 1. Ciclo `for` tradizionale

```
let frutta = ['mela', 'banana', 'ciliegia'];  
for (let i = 0; i < frutta.length; i++) {  
    console.log(frutta[i]);  
}  
// "mela"  
// "banana"  
// "ciliegia"
```

### 2. `forEach()`

```
let frutta = ['mela', 'banana', 'ciliegia'];
```

```
frutta.forEach(function(elemento) {  
    console.log(elemento);  
});  
// "mela"  
// "banana"  
// "ciliegia"
```

### 3. map() - Crea un nuovo array applicando una funzione a ciascun elemento

```
let numeri = [1, 2, 3, 4];  
let numeriQuadrato = numeri.map(function(numero) {  
    return numero * numero;  
});  
console.log(numeriQuadrato); // [1, 4, 9, 16]
```

### Array Multidimensionali:

Un array può anche contenere altri array, creando così array multidimensionali. Ad esempio, puoi avere una matrice 2D (un array di array).

```
let matrice = [  
    [1, 2, 3],  
    [4, 5, 6],  
    [7, 8, 9]  
];  
console.log(matrice[0][1]); // 2 (primo array, secondo elemento)  
console.log(matrice[2][2]); // 9 (terzo array, terzo elemento)
```

### Esempio Completo di Programma con Array in JavaScript:

```
// Creazione di un array di numeri  
let numeri = [10, 20, 30, 40, 50];  
  
// Aggiungere un elemento alla fine dell'array  
numeri.push(60);  
  
// Rimuovere l'ultimo elemento dell'array  
numeri.pop();  
  
// Aggiungere un elemento all'inizio  
numeri.unshift(5);  
  
// Rimuovere il primo elemento  
numeri.shift();  
  
// Iterazione su tutti gli elementi dell'array  
numeri.forEach(function(numero) {  
    console.log("Numero: " + numero);  
});  
  
// Creare un nuovo array con i quadrati dei numeri  
let quadrati = numeri.map(function(numero) {  
    return numero * numero;  
});
```

```
});  
  
console.log("Array dei quadrati:", quadrati);  
  
// Accesso a un elemento specifico  
console.log("Primo numero:", numeri[0]);  
console.log("Ultimo numero:", numeri[numeri.length - 1]);
```

## Output del Programma:

```
yaml  
Copia codice  
Numero: 5  
Numero: 10  
Numero: 20  
Numero: 30  
Numero: 40  
Numero: 50  
Array dei quadrati: [25, 100, 400, 900, 1600]  
Primo numero: 5  
Ultimo numero: 50
```

## Conclusioni:

Le array in JavaScript sono una delle strutture dati più potenti e flessibili, utilizzate in una vasta gamma di applicazioni. Le operazioni comuni su array includono aggiungere, rimuovere, modificare elementi e iterare attraverso gli array con cicli o metodi come `forEach` e `map`.

Esempio completo di una pagina web che utilizza JavaScript per dimostrare l'utilizzo delle array. In questo esempio, la pagina contiene un modulo in cui l'utente può aggiungere elementi a una lista e visualizzare un array in tempo reale, utilizzando diverse operazioni su array (aggiungere, rimuovere e visualizzare).

## Esempio Completo di Pagina Web con JavaScript per l'Utilizzo delle Array

```
<!DOCTYPE html>  
<html lang="it">  
<head>  
  <meta charset="UTF-8">  
  <meta name="viewport" content="width=device-width, initial-scale=1.0">  
  <title>Dimostrazione delle Array in JavaScript</title>  
  <style>  
    body {  
      font-family: Arial, sans-serif;  
      padding: 20px;  
    }  
    h2 {  
      color: #4CAF50;  
    }  
    .array-list {  
      list-style-type: none;  
      padding: 0;  
    }  
  </style>  
</head>  
<body>  
  <div>  
    <h2>Dimostrazione delle Array</h2>  
    <div>  
      <div>  
        <input type="text" value="Nuovo numero:" />  
        <button type="button" value="Aggiungi" />  
      </div>  
      <div>  
        <ul>  
          <li>5</li>  
          <li>10</li>  
          <li>20</li>  
          <li>30</li>  
          <li>40</li>  
          <li>50</li>  
        </ul>  
      </div>  
    </div>  
  </div>  
</body>  
</html>
```

```

        .array-list li {
            padding: 5px;
            background-color: #f1f1f1;
            margin-bottom: 5px;
            border-radius: 3px;
        }
    </style>
</head>
<body>

    <h1>Dimostrazione dell'Utilizzo delle Array in JavaScript</h1>

    <h2>Gestisci una Lista di Elementi</h2>
    <label for="elemento">Aggiungi un elemento alla lista:</label>
    <input type="text" id="elemento" placeholder="Scrivi un elemento...">
    <button onclick="aggiungiElemento()">Aggiungi</button>
    <button onclick="rimuoviUltimoElemento()">Rimuovi Ultimo</button>
    <button onclick="rimuoviPrimoElemento()">Rimuovi Primo</button>

    <h3>Array Attuale:</h3>
    <ul id="listaElementi" class="array-list">
        <!-- Gli elementi dell'array verranno visualizzati qui -->
    </ul>

    <h3>Array dopo l'operazione:</h3>
    <p id="arrayOutput"></p>

    <script>
        // Creiamo un array vuoto per tenere traccia degli elementi
        let lista = [];

        // Funzione per aggiungere un elemento all'array
        function aggiungiElemento() {
            let input = document.getElementById("elemento").value;
            if(input) {
                lista.push(input); // Aggiunge l'elemento alla fine dell'array
                document.getElementById("elemento").value = ""; // Svuota il campo di input
                aggiornaLista();
            } else {
                alert("Inserisci un valore valido!");
            }
        }

        // Funzione per rimuovere l'ultimo elemento dell'array
        function rimuoviUltimoElemento() {
            lista.pop(); // Rimuove l'ultimo elemento dell'array
            aggiornaLista();
        }

        // Funzione per rimuovere il primo elemento dell'array
        function rimuoviPrimoElemento() {
            lista.shift(); // Rimuove il primo elemento dell'array
            aggiornaLista();
        }

        // Funzione per aggiornare l'output della lista

```

```

function aggiornaLista() {
    let listaUl = document.getElementById("listaElementi");
    listaUl.innerHTML = ""; // Svuota la lista esistente
    lista.forEach(function(elemento) {
        let li = document.createElement("li");
        li.textContent = elemento;
        listaUl.appendChild(li);
    });

    // Visualizza l'array come stringa
    document.getElementById("arrayOutput").textContent = JSON.stringify(lista);
}
</script>

</body>
</html>

```

## Spiegazione del Codice:

### 1. HTML:

- La pagina contiene un modulo con un campo di input (<input>) e tre pulsanti per interagire con l'array: "Aggiungi", "Rimuovi Ultimo" e "Rimuovi Primo".
- Un'area <ul> viene utilizzata per visualizzare l'elenco degli elementi attuali nell'array.
- Un paragrafo (<p>) visualizza l'array in formato JSON, che cambia dinamicamente quando l'array viene modificato.

### 2. CSS:

- È stato aggiunto uno stile di base per la pagina, incluse le liste e i pulsanti, per migliorare l'aspetto.

### 3. JavaScript:

- `lista`: è un array che contiene gli elementi inseriti dall'utente.
- La funzione `aggiungiElemento()` aggiunge un nuovo elemento alla fine dell'array usando `push()` e poi aggiorna la visualizzazione dell'array.
- La funzione `rimuoviUltimoElemento()` rimuove l'ultimo elemento dall'array usando `pop()`.
- La funzione `rimuoviPrimoElemento()` rimuove il primo elemento dell'array usando `shift()`.
- La funzione `aggiornaLista()` aggiorna la visualizzazione dell'array nella lista HTML (<ul>) e nella sezione di output JSON.

## Come funziona:

1. L'utente inserisce un valore nel campo di input e clicca su "Aggiungi" per aggiungere l'elemento all'array.
2. Gli elementi vengono mostrati in una lista HTML sotto la sezione "Array Attuale".
3. Puoi rimuovere l'ultimo elemento con il pulsante "Rimuovi Ultimo" o il primo elemento con "Rimuovi Primo".

4. L'array viene visualizzato anche come una stringa JSON sotto "Array dopo l'operazione", così da poter vedere come cambia dinamicamente.