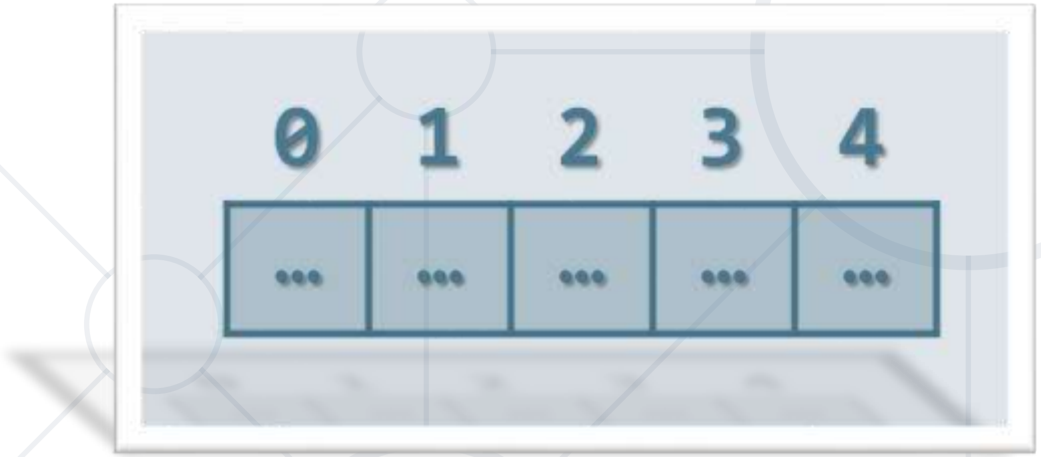
Arrays and Matrices

Arrays, Array Operations, Matrices, Multi-Dimensional Arrays



SoftUni Team

Technical Trainers

Software University
SoftUni Foundation

Software University

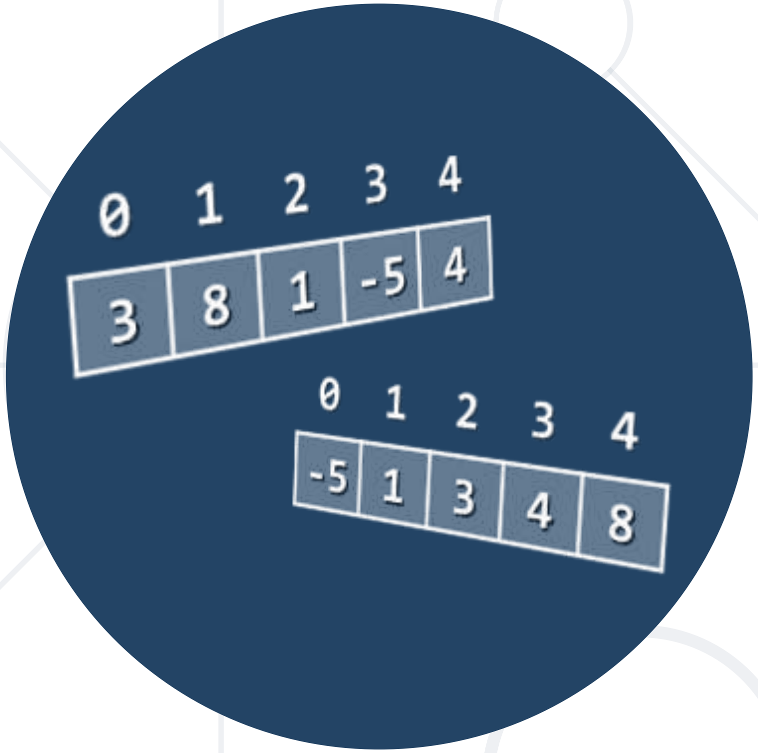
http://softuni.bg

Table of Content

Have a Question?

sli.do

#JS-CORE

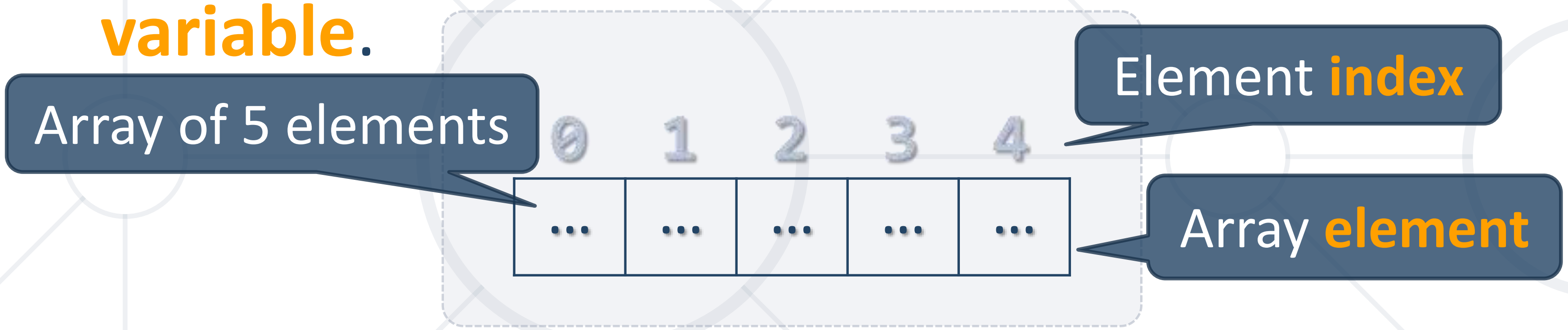


Arrays in JS

Working with Arrays of Elements

What is an Array?

- JS arrays are used to store **multiple values** in a **single variable**.



- Elements are **numbered** from **0** to **length-1**.
- Creating an array using **an array literal**.

The better way.

```
let numbers = [10, 20, 30, 40, 50];
```

- Creating an array using the keyword **new**.

May create some unexpected results.

```
let numbers = new Array(10, 20, 30, 40, 50);
```

Arrays of Different Types

```
// Array holding numbers
let numbers = [10, 20, 30, 40, 50];
```

```
// Array holding strings
let weekDays = ['Monday', 'Tuesday', 'Wednesday',
  'Thursday', 'Friday', 'Saturday', 'Sunday'];
```

```
// Array holding mixed data
var mixedArr =
  [20, new Date(), 'hello', {x:5, y:8}];
```

Accessing Elements

- Array elements are accessed using their **index number**.

```
let cars = ['BMW', 'Audi', 'Opel'];
let firstCar = cars[0];    // BMW
let lastCar = cars[arr.length - 1];  // Opel
```

- Accessing indexes that do not exist in the array returns **undefined**.

```
console.log(cars[3]);   // undefined
console.log(cars[-1]);  // undefined
```

Problem: Sum First Last

- You are given an array of string elements.
  - Multiplicate each element with the length of the array and print each index with its value.

```
function sumFirstLast(numArr) {
        let listInput = JSON.parse(document.getElementById("arr").value);
            let divResult = document.getElementById('result');
            function calculate(list) {
                for (let i = 0; i < list.length; i++) {
                    let p = document.createElement('p');
                     p.textContent = `${i} -> ${list[i]*list.length}`
                    divResult.appendChild(p) }
            }
            calculate(listInput);
}
```

Check your solution here: https://judge.softuni.bg/Contests/Practice/Index/1464#0

Changing elements

- Elements can be modified.

```
let fruits = ['Apple', 'Kiwi'];
fruits[0] = 'Peach';
console.log(fruits); // ['Peach', 'Kiwi']
```

- JS arrays are **resizable**.

```
fruits.push('Banana');
fruits[fruits.length] = 'Mango';
// both examples add a new element to the array
```

- Note that adding elements with **high indexes** can create **undefined "holes"** in an array!

Problem: Even Position Element

- You are given an array of numbers.
  - Find the elements at even position and print them.

```
function solve(numArr) {
  let listInput = JSON.parse(document.getElementById("arr").value);
  let result = [];
  function calculate(numArr) {
    numArr.forEach((element, index) => {
      if (index % 2 === 0) {
        result.push(element);
      }
    });
  }
  calculate(listInput);
  document.getElementById("result").innerHTML = result.join(' x ');
}
```

Check your solution here: https://judge.softuni.bg/Contests/Practice/Index/1464#1

Properties and Methods

- The **length property** returns the **number of elements**.

```
let fruits = ['Mango', ‘Kiwi', 'Orange'];
console.log(fruits.length); // returns 3
```

- The **sort()** method sorts the item of an array. By default, it sorts the values as **strings** in **alphabetical** and **ascending** order.

```
console.log(fruits.sort());
// ['Kiwi', 'Mango', 'Orange']
```

- However, if numbers are sorted as strings, **"25"** is **bigger than "100"**, because **"2"** is bigger than **"1"**.

Sorting an array of numbers

SoftUni Foundation

```
let nums = [20, 40, 10, 30, 100, 5];
console.log(nums.join(' ')); // 20 40 10 30 100 5
```

```
nums.sort(); // Works incorrectly on arrays of numbers
console.log(nums.join('|')); // 10 100 20 30 40 5
```

- Sorting numbers in **ascending order**

```
nums.sort((a, b) => a - b); // Compare elements as numbers
console.log(nums.join(' ')); // 5 10 20 30 40 100
```

- Sorting numbers **in descending order**

```
nums.sort((a, b) => b - a); // Compare elements as numbers
console.log(nums.join(' ')); // 100 40 30 20 10 5
```

Add / Remove Elements at Both Ends

```
let nums = [10, 20, 30];
console.log(nums.join('|')); // 10|20|30
```

```
nums.push(40);
console.log(nums.join('|')); // 10|20|30|40
```

```
let tail = nums.pop();       // tail = 40
console.log(nums.join('|')); // 10|20|30
```

```
nums.unshift(0);
console.log(nums.join('|')); // 0|10|20|30
```

```
let head = nums.shift();     // head = 0
console.log(nums.join('|')); // 10|20|30
```

Problem: Replace and Reverse

- You are given **an array of strings**.
- Make the **first letter** of each element **upper** and **reverse** the array.

```
function solve(numArr) {
  let listInput = JSON.parse(document.getElementById("arr").value);
  function calculate(numArr) {
    numArr.forEach((element, index) => {
      numArr[index] = element.split('').reverse().join('');
    });
    numArr.forEach((element, index) => {
      numArr[index] = element.charAt(0).toUpperCase().concat(element.slice(1));
    });
    return numArr.join(' ');
  }
  let result = calculate(listInput);
  document.getElementById("result").innerHTML = result;
}
```

Check your solution here: https://judge.softuni.bg/Contests/Practice/Index/1464#2

Slicing Arrays

```
let nums = ['one', 'two', 'three', 'four'];
console.log(nums.join('|')); // one|two|three|four
```

```
let firstNums = nums.slice(0, 2); // start, end+1
console.log(firstNums.join('|')); // one|two
```

```
let lastNums = nums.slice(2, 4); // start, end+1
console.log(lastNums.join('|')); // three|four
```

```
let midNums = nums.slice(1, 3); // start, end+1
console.log(midNums.join('|')); // two|three
```

Splice: Cut and Insert Array Elements

```
let nums = [5, 10, 15, 20, 25, 30];
console.log(nums.join('|')); // 5|10|15|20|25|30
```

```
let mid = nums.splice(2, 3); // start, delete-count
console.log(mid.join('|')); // 15|20|25
console.log(nums.join('|')); // 5|10|30
```

```
nums = [5, 10, 15, 20, 25, 30];
nums.splice(3, 2, "twenty", "twenty-five");
console.log(nums.join('|'));
// 5|10|15|twenty|twenty-five|30
```

Looping through an array

```
let nums = [5, 10, 15, 20, 25, 30];
```

- **for** … **of** works like **foreach**

```
for (let num of nums)
  console.log(num);
```

- **for** … **in** goes through array indices

```
for (let i in nums)
  console.log(i + " " + nums[i]);
```

- Traditional **for**-loop

```
for (let i = 0; i < nums.length; i++)
  console.log(nums[i]);
```

Problem: Find Element

- You are given a string and **an array of strings**.
  - Search if the string is **present** in **each element** of the array.

```
function solve() {
  let number = parseInt(document.getElementById("num").value);
  let listInput = JSON.parse(document.getElementById("arr").value);
  let answer = [];
  function calc(searched, input) {
    for (let element of input) {
      let result = element.includes(searched);
      let index = element.indexOf(searched);
      answer.push(result + ' -> ' + index);
    }
    return answer;
  }
  let result = calc(number, listInput);
  document.getElementById("result").innerHTML = result;
}
```

Check your solution here: https://judge.softuni.bg/Contests/Practice/Index/1464#3

More Array Methods

SoftUni Foundation

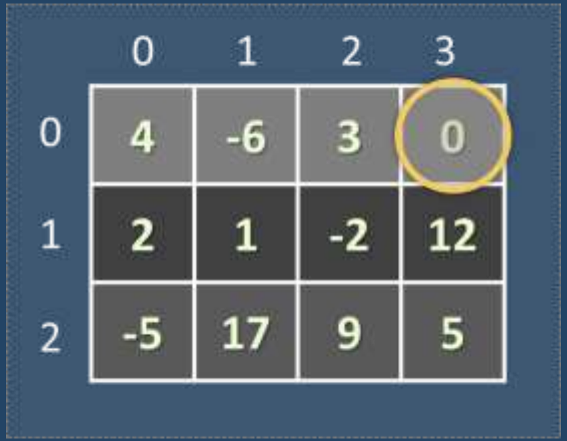
- **includes()** – check if an array contains a specific element
- **indexOf()** - returns the position of an element in an array
- **isArray()** – checks if an object is an array
- **reverse()** – reverses the order of elements in an array
- **toString()** – converts an array to string
- **reduce()** – reduces the values of an array (from left to right)
- Note that you **CANNOT reduce** an **empty array**!

Problem: Multiple Sort

- You are given **an array of strings**. Sort the elements by **ascending** order and then **alphabetically**.

```
function solve() {
  let listInput = JSON.parse(document.getElementById("arr").value);
  let sortAscending = [];
  let sortAlphabetically = [];
  function calc(input) {
    let resultContainer = document.getElementById("result");
    let div1 = document.createElement('div');
    let div2 = document.createElement('div');
    sortAscending = input.sort((a, b) => a - b);
    div1.textContent = sortAscending.join(', ')
    resultContainer.appendChild(div1);
    sortAlphabetically = input.sort((a, b) => a.localeCompare(b));
    div2.textContent = sortAscending.join(', ')
    resultContainer.appendChild(div2);
  }
  calc(listInput);
}
```

Check your solution here: https://judge.softuni.bg/Contests/Practice/Index/1464#4



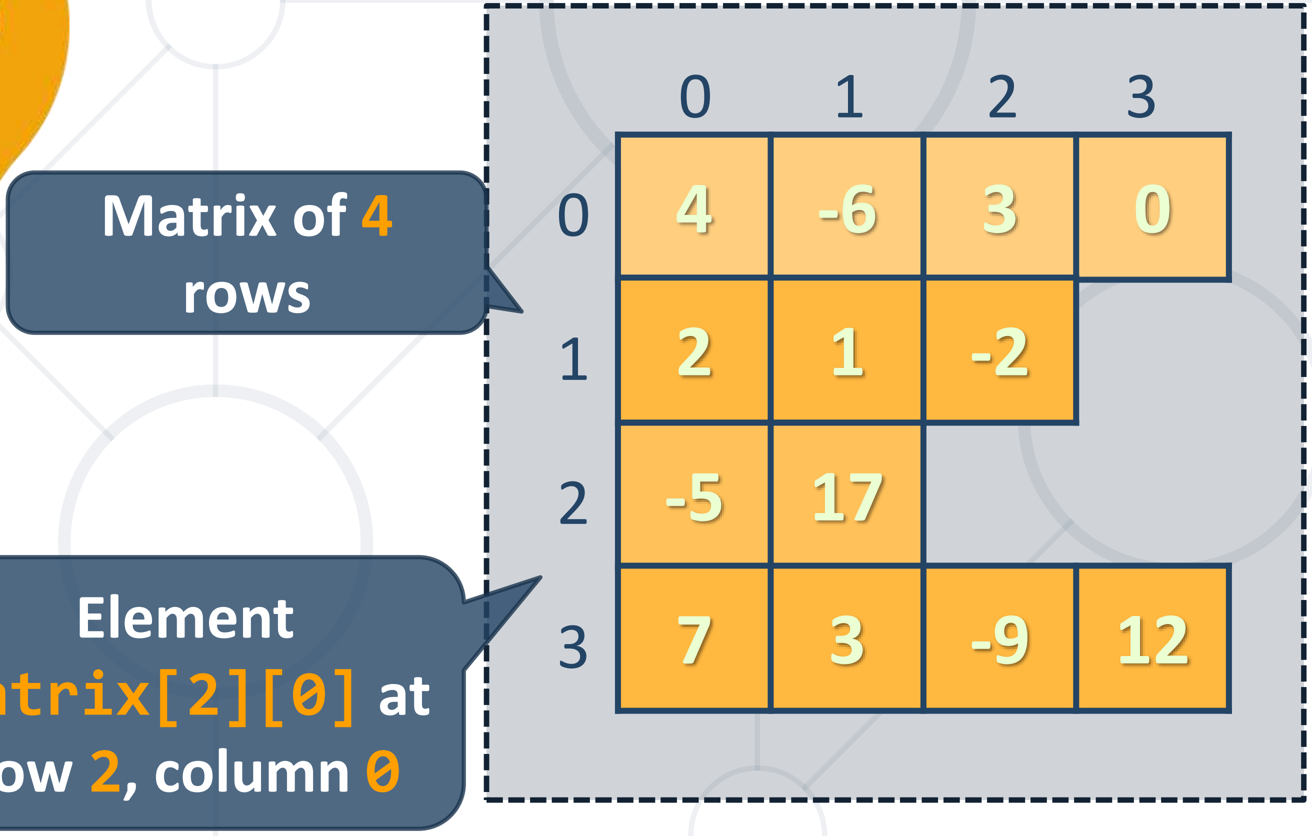
	0	1	2	3
0	4	-6	3	0
1	2	1	-2	12
2	-5	17	9	5

Matrices

Array of Arrays

Matrices in JS

- A **matrix** is a **table of values**.
- Matrices are represented as **nested arrays** in JavaScript.



$$\begin{array}{c|cccc} & 0 & 1 & 2 & 3 \\ \hline 0 & 4 & -6 & 3 & 0 \\ 1 & 2 & 1 & -2 & \\ 2 & -5 & 17 & & \\ 3 & 7 & 3 & -9 & 12 \end{array}$$

```
let matrix = [
   [4, -6, 3, 0],
   [2, 1, -2],
   [-5, 17],
   [7, 3, -9, 12]
];
```

Looping through a matrix

```
let matrix = [[4, 5, 6],
              [6, 5, 4],
              [5, 5, 5]];
```

```
for (let row = 0; row < matrix.length; row++) {
     console.log(matrix[row]);  // [4, 5, 6]
                                // [6, 5, 4]
                                // [5, 5, 5]
       for (let col = 0; col < matrix[row].length; col++) {
     console.log(matrix[row][col]);
    // prints each element of the matrix on a separate line
    }
}
```

Live Exercises

Summary

- Arrays are used to store **multiple values** in a **single variable**.
- Elements are **accessed** using their **index number**.
- **Sorting** and **modifying** elements using methods.
- Looping through arrays with **for … of**, **for … in** and traditional **for-loop**.
- A matrix is a **table of values**.

Questions?

SoftUni

Software University

SoftUni Svetlina

SoftUni Creative

SoftUni Digital

SoftUni Foundation

SoftUni Kids

https://softuni.bg/courses/javascript-fundamentals

SoftUni Diamond Partners

SoftUni Foundation

XSsoftware

SBTech we know sports

telenor

SoftwareGroup doing it right

NETPEAK

SmartIT

æternity

SUPER HOSTING .BG

INDEAVR Serving the high achievers

INFRAGISTICS

LIEBHERR

Postbank Решения за твоето утре

SoftUni Organizational Partners

SoftUni Foundation

ИНФОРМАЦИОННО ОБСЛУЖВАНЕ

OneBit SOFTWARE

Lukanet.com

WORLD OF MYTHS

Trainings @ Software University (SoftUni)

- Software University – High-Quality Education and Employment Opportunities
  - softuni.bg
- Software University Foundation
  - http://softuni.foundation/
- Software University @ Facebook
  - facebook.com/SoftwareUniversity
- Software University Forums
  - forum.softuni.bg

License