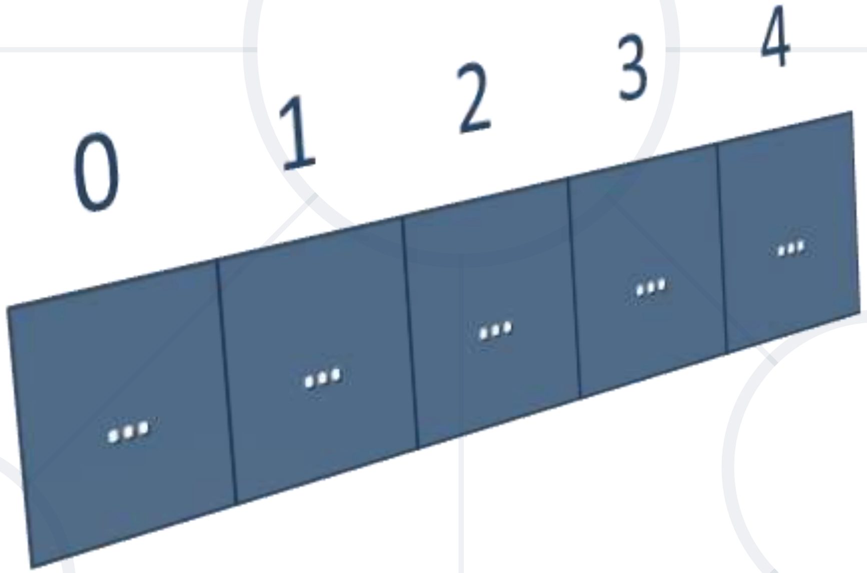
Arrays Advanced

Additional Array Operations



SoftUni Team

Technical Trainers

Software University

SoftUni Foundation

Software University

http://softuni.bg

Table of Contents

SoftUni Foundation

Have a Question?

SoftUni Foundation

sli.do

#tech-fund

Additional Array Functionality

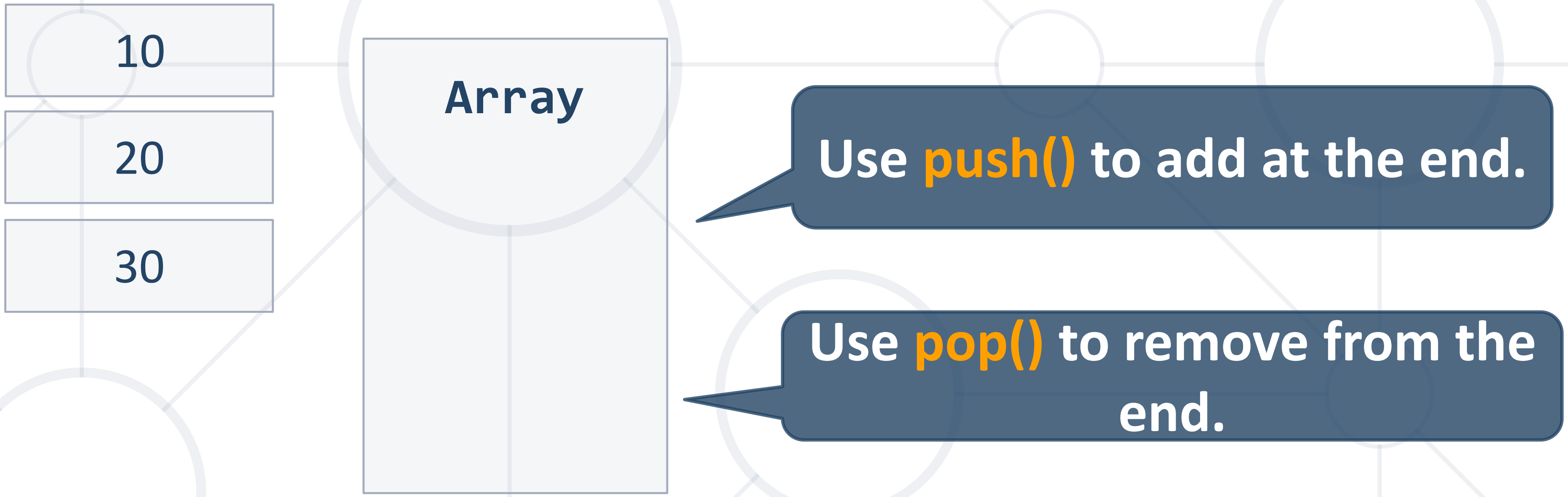
Inserting at Start, Removing at End

array() – Advanced Overview

- array() - Advanced functionality of the array consists of the following functions in JS:
  - push() – add to the end
  - pop() – remove from the end
  - unshift() – add to the beginning
  - shift() – remove from the beginning
  - slice() – remove a range of elements
  - splice() – insert at position/delete from position

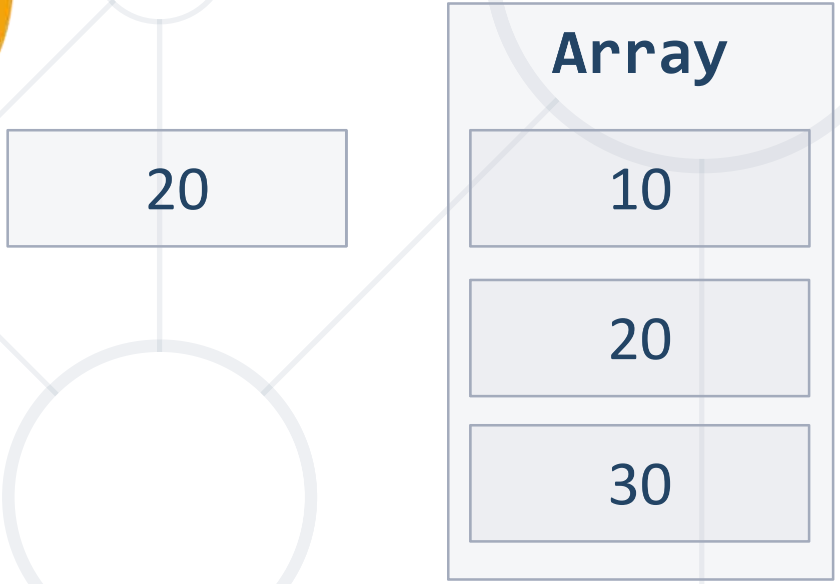
Add at the End, Remove from the End

- JavaScript arrays provide **push()** and **pop()**


10
20
30
Array
Use push() to add at the end.
Use pop() to remove from the end.

Add at the Start, Remove from the Start

- We can use **unshift()** to add at the start and **shift()** to remove from the start.



Use **shift()** to remove from the start.

Use **unshift(20)** to add at the start.

Array Operations

Push, Pop ,Shift, Unshift, Slice, ...

pop() – Removes the last element

- The **pop** method removes the last element from an array and returns that value to the caller.
- If you call **pop()** on an empty array, it returns undefined.

```
let myArray = ["one","two","three","four","five"];
let popped = myArray.pop();
console.log(myArray); //["one","two","three","four"]
console.log(popped); //"five"
```

Problem: Sum First Last

- Calculate and print the sum of the **first** and the **last** elements in an array.
- The input comes as **array of string** elements holding numbers.
- The output is the return value of your function.

['5' , '10']	60
['20', '30', '40']	60

```
function solve(input) {
  input = input.map(Number);
  console.log(input[0]
              + input.pop());
}
```

Check your solution here: https://judge.softuni.bg/Contests/1254/Arrays-Advanced-Lab

Pushing into an Array

- The **push** method adds **one** or **more** elements to the end of an array and returns the new length of the array

```
let fruits = ["apple","banana","kiwi"];
fruits.push("pineapple");
console.log(fruits);
 // ["apple","banana","kiwi","pineapple"]
```

Element is added at the end

Shifting and Unshifting

SoftUni Foundation

shift() - Removes the first element of an array

```
let myArray = ["one","two","three","four","five"];
myArray.shift(); // ["two","three","four","five"]
```

unshift() - Adds elements to the beginning

```
let myArray = ["red","green","blue"];
myArray.unshift("purple");
// ["purple","red","green","blue"]
```

New element added

Problem: Negative / Positive Numbers

- You are given an array of numbers **`arr`**
  - Process them one by one and produce a new array **`result`**
    - Prepend each negative element at the front of result
    - Append each positive (or 0) element at the end of result
  - Print the **`result`** array, each element at separate line

$$\boxed{\begin{array}{c}7\\-2\\8\\9\end{array}} \rightarrow \boxed{\begin{array}{c}-2\\7\\8\\9\end{array}} \qquad \boxed{\begin{array}{c}3\\-2\\0\\-1\end{array}} \rightarrow \boxed{\begin{array}{c}-1\\-2\\3\\0\end{array}} \qquad \boxed{\begin{array}{c}-3\\2\\-1\\0\end{array}} \rightarrow \boxed{\begin{array}{c}-1\\-3\\2\\0\end{array}}$$

Solution: Negative / Positive Numbers

```
function negativePositiveNumbers(arr) {
  let result = [];
  for (num of arr)
    if (num < 0)
      result.unshift(num); // Insert at the start
    else
      result.push(num);    // Append at the end
  console.log(result.join('\n'));
}
```

Check your solution here: https://judge.softuni.bg/Contests/1254/Arrays-Advanced-Lab

Deleting Elements

- Elements can be removed by using **delete**

```
let myArray = ["one","two","three","four"];
delete myArray[0];
// Changes the first element to undefined
```

- Using delete may leave **undefined spots** in the array.
- Use **pop()** or **shift()** instead.

Slicing Arrays

- The **slice()** function returns a newly created array
- Can **remove** a range of elements from selected **start** to **end**
- Note that the original array will **not be modified**

```
let myArray = ["one","two","three","four","five"];
let sliced = myArray.slice(2);
console.log(myArray);
//["one","two","three","four","five"]
console.log(sliced); // ["three","four","five"]
console.log(myArray.slice(2,4)); // ["three","four"]
```

Splice: Cut and Insert Array Elements

- The **splice()** function adds/removes items to/from an array, and returns the removed item(s).
- This function **changes the original array**.

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

Problem: First and Last K Numbers

- You are given an array of numbers
  - The first element holds an integer **k**
  - Print the first **k** and the last **k** from the other elements in the array (space separated)

```
function firstLastKElements(arr) {
  let k = arr.shift();
  console.log(arr.slice(0, k).join(' '));
  console.log(arr.slice(arr.length-k,
    arr.length).join(' '));
}
```

Input	Output
2 7 8 9	7 8 8 9
3 6 7 8 9	6 7 8 7 8 9

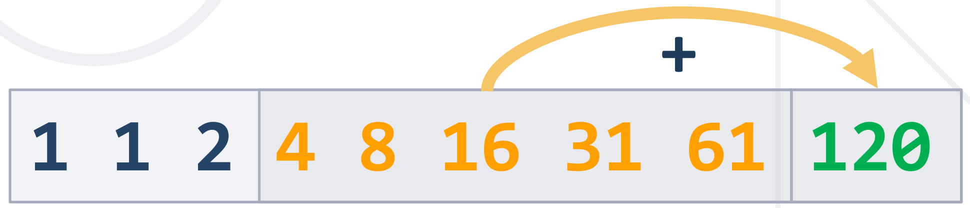
Check your solution here: https://judge.softuni.bg/Contests/1254/Arrays-Advanced-Lab

Problem: Sum Last K Numbers Sequence

SoftUni Foundation

- Take two integers **n** and **k**
- Generate and print the following sequence:
  - The first element is: **1**
  - All other elements = sum of the previous **k** elements
- Example: **n** = 9, **k** = 5
  - 120 = 4 + 8 + 16 + 31 + 61

Input	Output
6 3	Sequence: 1 1 2 4 7 13
8 2	Sequence: 1 1 2 3 5 8 13 21
9 5	Sequence: 1 1 2 4 8 16 31 61 120



Check your solution here: https://judge.softuni.bg/Contests/1254/Arrays-Advanced-Lab

Solution: Sum Last K Numbers Sequence

```
function sumLastKNumbersSequence(n, k) {
  let seq = [1];
  for (let current = 1; current < n; current++) {
    let start = Math.max(0, current - k);
    let end = current - 1;
    let sum = // TODO: sum the values of seq[start … end]
    seq[current] = sum;
  }
  console.log(seq.join(' '));
}
```

Check your solution here: https://judge.softuni.bg/Contests/1254/Arrays-Advanced-Lab

Filtering and Transforming Elements

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

```
let filteredNums =
 nums.filter(x => x.startsWith('t'));
console.log(filteredNums.join('|')); // two|three
```

```
let lengths = nums.map(x => x.length);
console.log(lengths.join('|')); // 3|3|5|4
```

```
let lengths = nums.map(x => [x.length, x[0]]);
console.log(lengths.join('|')); // 3,o|3,t|5,t|4,f
```

Problem: Process Odd Numbers

SoftUni Foundation

- You are given an **array of numbers**
  - Print the **odd** numbers, **doubled** and **reversed**

```
function firstLastKElements(arr) {
  let result = arr
    .filter((num, i) => i % 2 == 1)
    .map(x => 2*x)
    .reverse();
  return result.join(' ');
}
```

```
10
15
20
25
```

→

```
50 30
```

```
3
0
10
4
7
3
```

→

```
6 8 0
```

Check your solution here: https://judge.softuni.bg/Contests/1254/Arrays-Advanced-Lab

Sorting Arrays

Arranging Elements in Increasing Order

Sorting Arrays

- The **sort()** function sorts the items of an array.
- The sort order can be either **alphabetic** or **numeric**, and either **ascending (up)** or **descending (down).**
- By default, the **sort()** function sorts the values as strings in **alphabetical** and **ascending** order.
- The **sort()** function will produce an **incorrect** result when sorting numbers. You can fix this by providing a **compare function**.

Sorting Arrays

```
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
```

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

Problem: Smallest 2 Numbers

- You are given an **array of numbers**
  - Print the **smallest** two numbers

```
function smallestTwoNumbers(arr) {
  arr.sort((a, b) => a-b);
  let result = arr.slice(0, 2);
  return result.join(' ');
}
```

```
30
15
50
5
```

```
5 15
```

```
3
0
10
4
7
3
```

```
0 3
```

Check your solution here: https://judge.softuni.bg/Contests/1254/Arrays-Advanced-Lab

Live Exercises

Summary

- Arrays in JavaScript aren't **fixed**.
- Can **add** / **remove** / **insert** elements at runtime.
- Sorting arrays can be done with and without a **compare function**.

Questions?

SoftUni

Software University

SoftUni Svetlina

SoftUni Creative

SoftUni Digital

SoftUni Foundation

SoftUni Kids

SoftUni Diamond Partners

SoftUni Foundation

XSsoftware

SBTech
we know sports

telenor

SoftwareGroup
doing it right

NETPEAK

SmartIT

Postbank
Решения за твоето утре

SUPER HOSTING .BG

INDEAVR
Serving the high achievers

INFRAGISTICS

LIEBHERR

æternity

codexio

SoftUni Organizational Partners

SoftUni Foundation

ИНФОРМАЦИОННО ОБСЛУЖВАНЕ

OneBit SOFTWARE

WORLD OF MYTHS

Lukanet.com

codexio

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