JavaScript Syntax

Functions, Objects, Arrays, Strings

Au Mau Duong
Technical Trainers

Table of Contents

Functions

- Functions in JS hold a piece of code (script)
  - Can take parameters and return result
  - Similar to functions in C and PHP and methods in C++ / C# / Java

```
function multiply(a, b) {
  return a * b;
}

console.log(multiply(2, 3)); // 6 == 2 * 3
console.log(multiply(2)); // NaN == 2 * undefined
console.log(multiply(5, 6, 7)); // 30 = 5 * 6
```

Functions practice - Exercise 1

Write a JS program to input any number from user and find cube of the given number using function

- *Input*
  - *Input any number: 5*
- *Output*
  - *Cube of 5 = 125*

Functions practice - Exercise 2

Write a JS program to input two or more numbers from user and find maximum and minimum of the given numbers using functions

- *Input*
  - *Input two numbers: 10, 20*
- *Output*
  - *Maximum = 20*
  - *Minimum = 10*

Functions practice - Exercise 3

Write a function in JS programming to print all natural numbers between 1 to n

- *Input*
  - *Input lower limit: 1*
  - *Input upper limit: 10*
- *Output*
  - *Natural numbers between 1 to 10: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10,*

Objects

- Objects in JavaScript hold key-value pairs:

```
let obj = { name : "SoftUni", age : 2 }
console.log(obj); // Object {name: "SoftUni", age: 2}
obj['site'] = "http://www.softuni.bg"
obj.age = 10
obj['name'] = "Software University"
console.log(obj) // Object {name: "Software University",
age: 10, site: "http://www.softuni.bg"}
delete obj.name
delete obj.site
console.log(obj) // Object {age: 10}
```

Objects and JSON

- JavaScript objects can be stored as text in JSON format


```
let obj = { name : "SoftUni", age : 2 }
let str = JSON.stringify(obj)
console.log(str) // {"name":"SoftUni","age":2}
```


```
let str = "{\"name\":\"Nakov\",\"age\":24}"
let obj = JSON.parse(str)
console.log(obj) // Object {name: "Nakov", age: 24}
```

Problem: Sums by Town

- You are given a sequence of JSON strings holding town + income

```
{"town":"Sofia","income":200}
{"town":"Varna","income":120}
{"town":"Pleven","income":60}
{"town":"Varna","income":70}
```

Towns can appear multiple times

- Write a JS function to sum and print the incomes for each town

```
Pleven -> 60
Sofia -> 200
Varna -> 190
```

Order the towns by name

Solution: Sums by Town

```
function calcSumsByTown(arr) {
  let objects = JSON.parse(arr);
  let sums = {};
  for (let obj of objects)
    if (obj.town in sums)
      sums[obj.town] += obj.income;
    else
      sums[obj.town] = obj.income
  let towns = Object.keys(sums).sort();
  for (let town of towns)
    console.log(town + " -> " + sums[town]);
}
```

Arrays in JavaScript

```
// Array holding numbers
let numbers = [1, 2, 3, 4, 5];

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

// Array of mixed data
var mixedArr = [1, new Date(), 'hello'];

// Array of arrays (matrix)
var matrix = [
  ['0,0', '0,1', '0,2'],
  ['1,0', '1,1', '1,2'],
  ['2,0', '2,1', '2,2']];
```

Processing Arrays Elements

- Print all elements of an array of strings:

```
let capitals = ['Sofia', 'Washington', 'London', 'Paris'];

for (let capital of capitals)
  console.log(capital);

for (let i in capitals)
  console.log(capitals[i]);

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

Works like **foreach**

This is not **foreach**! It goes through the array indices.

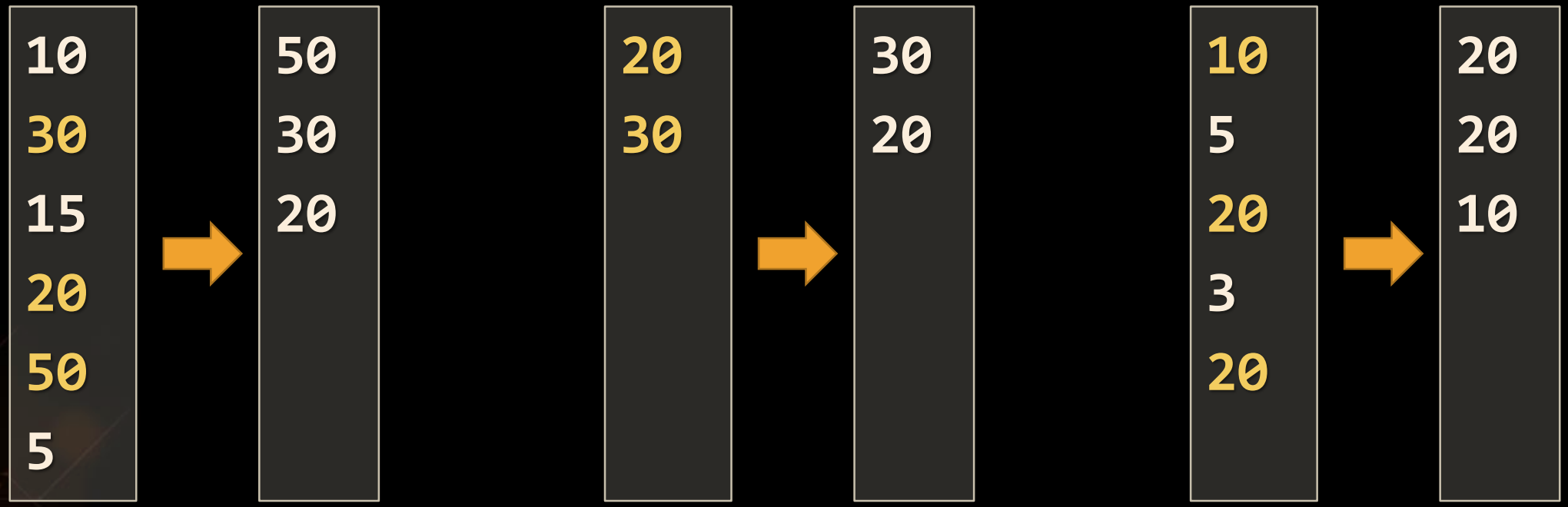
Traditional **for**-loop

Array Operations

```
let numbers = [1, 2, 3, 4];
console.log(numbers.join('|')); // result: 1|2|3|4|5

numbers.push(5);
console.log(numbers.join('|')); // result: 1|2|3|4|5

let tail = numbers.pop();       // tail = 5;
console.log(numbers.join('|')); // result: 1|2|3|4

numbers.unshift(0);
console.log(numbers.join('|')); // result: 0|1|2|3|4

let head = numbers.shift();     // head = 0;
console.log(numbers.join('|')); // result: 1|2|3|4
```

Problem: Largest 3 Numbers

- Write a program to read an array of numbers and find and print the largest 3 of them



Solution: Largest 3 Numbers

```
function largest3Numbers(arr) {
  let nums = arr.map(Number);
  let numsSorted = nums.sort((a, b) => b - a);
  let count = Math.min(3, arr.length);
  for (let i = 0; i < count; i++)
    console.log(numsSorted[i]);
}
```

```
largest3Numbers(['10', '30', '15', '20', '50', '5'])
```

Strings

- Strings in JavaScript hold a sequence of Unicode characters

```
let str1 = "Some text in a string variable"
let str2 = 'Text enclosed in single quotes'
for (let i = 0; i < str1.length; i++)
  console.log(str1[i] + ' ' + str2[i])
```

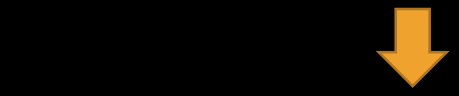
```
let tokens = 'C#, Java, PHP ,HTML'.split(',');
// tokens = ['C#', ' Java', ' PHP ', 'HTML']
tokens = tokens.map(s => s.trim());
console.log(tokens);
// ['C#', 'Java', 'PHP', 'HTML']
```

Filter by lambda function

Problem: Extract Capital-Case Words

- Write a JavaScript function to extract from array of strings all capital-case words. All non-letter chars are considered separators.

```
We start by HTML, CSS, JavaScript, JSON and REST.
Later we touch some PHP, MySQL and SQL.
Later we play with C#, EF, SQL Server and ASP.NET MVC.
Finally, we touch some Java, Hibernate and Spring.MVC.
```



```
HTML, CSS, JSON, REST, PHP, SQL, C, EF, SQL, ASP, NET,
MVC, MVC
```

Solution: Extract Capital-Case Words

```
function extractCapitalCaseWords(arr) {
  let text = arr.join(",");
  let words = text.split(/\W+/);
  let nonEmptyWords = words.filter(w => w.length > 0);
  let upWords = nonEmptyWords.filter(isUppercase);
  console.log(upWords.join(", "));

  function isUppercase(str) {
    return str == str.toUpperCase();
  }
}
```

```
extractCapitalCaseWords(['PHP, Java and HTML'])
```

Summary

- Arrays in JS combine traditional arrays, lists and dictionaries
- Strings in JS hold a sequence of Unicode characters
- Objects in JS hold key-value pairs
- JS is functional language: relies on functions, callbacks, lambdas, etc.

JavaScript Syntax

Questions?