

Основи на JavaScript

I-ва част

29.10.2018

Филип Сидеров

filip.siderov@sap.com

Public



Съдържание

- ✓ Въведение
- ✓ Синтаксис
- ✓ Типове данни
- ✓ Оператори
- ✓ Цикли



Да припомним

<HTML />

.css { }

Да припомним (1/2)

КАКВО Е HTML?

Hypertext Markup Language

Език за репрезентиране на уеб
страници

Дефинира съдържанието на
страницата, но не
функционалността ѝ

КАКВО Е CSS?

Cascading Style Sheets

Описва презентацията на HTML или
XML документи

Описва как елементите трябва да се
изобразят (рендират) на страницата

Да припомним (2/2)

КАКВА Е РАЗЛИКАТА МЕЖДУ

a, #a, .a, *

color, background-color, border-color

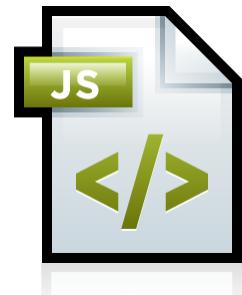
display: block, inline, flex, сещате ли се за други?

Какво е !important?



Въведение

```
i > 3) {  
} else {  
} alert("The universe is in order.");  
  
var firstName = {miller: "Henry",  
thompson: "Hunter",  
nietzsche: "Friedrich"};  
"5 onions and 6 potatoes".replace(/\d/g, function(x){  
});  
return parseInt(x) + 1;  
alert(23);  
if(a == 3) {  
} alert('Hello');  
  
function copyArray(array) {  
    var newArr = new Array(array.length);  
    for (var i = 0; i != array.length; i++) {  
        newArr[i] = array[i];  
    }  
    return newArr;  
}
```



Въведение

Възможности на JavaScript

Какво може да направи реалността на
JavaScript?

Разпознаване на браузер **Валидиране на форми**
Бисквитки Контролиране на браузери

Сървърни приложения

Обработване на HTTP заявки
Работа с бази от данни Работа с файлове

Други

Windows Gadgets, **Windows Tiles**

Въведение

Възможности на JavaScript

Какво НЕ може да направи

Записване на файлове на сървър, без помощта сървърен скрипт

Да работи с Бази от данни локално (изкл. браузер) нито на сървър без
помощен скрипт/услуга

Да пише и чете от локалната файлова система

Да затваря прозорци, които не са отворени от него

Да достъпва уеб страници, хоствани на друг домейн

Да защити използваните картинки или сурс-код

Въведение

История, версии и имена

In 10 days in **May 1995**, **Brendan Eich**, Netscape

Mocha -> LiveScript - > JavaScript

JavaScript or Java

Vanilla JS

EcmaScript

Въведение

EcmaScript стандарт и неговите версии

Edition	Published	Changes	Editor
1	06, 1997	First edition	Guy L. Steele, Jr.
2	06, 1998	Editorial changes to keep the specification fully aligned with ISO/IEC 16262 international standard	Mike Cowlishaw
3	12, 2009	Try/catch blocks, Regular Expressions, better string handling	Mike Cowlishaw
4	Abandoned	Due to political differences related to the language complexity.	
5	12, 2009	“Strict mode”, improvement error checking, getters/setters, JSON support & reflection	Pratap Lakshman, Allen Wirfs-Brock
5.1	12, 2011	This edition 5.1 of the ECMAScript Standard is fully aligned with third edition of the international standard ISO/IEC 16262:2011.	Pratap Lakshman, Allen Wirfs-Brock
6	06, 2015	New syntax, classes, modules, iterators, collections	Allen Wirfs-Brock
7	06 2017	Promises/concurrency, number and math enhancements, operator overloading,...	

Въведение

Основни характеристики



Мултиплатформен



Скриптов

Нетипизиран

Обектно-ориентиран

Еднонишков

При клиента / На сървъра

“C & Java like“



Типове данни

Примитивни и
референтни типове



Типове данни

Примитивни типове

Представят малки/прости парчета от данни*:

- ✓ **boolean** – true или false
- ✓ **number** – всяко цялочислено или число с плаваща запетая
[-9007199254740992 до 9007199254740992], **floating-point 64bit**
- ✓ **string** – всеки символ или поредица от символи
- ✓ **null** - притежава само една възможна стойност – **null**
- ✓ **undefined** – притежава само една възможна стойност – **undefined**
- ✓ **symbol** – уникални и непроменими (immutable) стойности

* Референтните типове реферират по-сложни структури от данни (обекти).

Типове данни

Примитивни типове - пример

```
let num = 5,  
  emptyObj = null,  
  notDefined,  
  myBool = true,  
  myText = "Hello World!",  
  myNewText = 'Today is "Monday"',  
  sym = Symbol("foo");
```

```
console.log(num); // 5  
console.log(emptyObj); // null  
console.log(notDefined); // undefined  
console.log(myBool); // true  
console.log(myText); // Hello World!  
console.log(myNewText); // Today is "Monday"  
console.log(sym); // Symbol(foo)
```

Типове данни

Примитивни типове

- ✓ В “heap” паметта
- ✓ “Variable Object”
- ✓ Присвояване по стойност

Variable Object	
age	20

```
1 var color1 = "red";  
2 var color2 = color1;  
3 color1 = "blue";
```



Variable Object	
color1	"blue"
color2	"red"

Типове данни

Примитивни типове - идентифициране

✓ **typeof**

```
1 console.log(typeof "FMI");      // "string"  
2 console.log(typeof 2018);       // "number"  
3 console.log(typeof true);       // "boolean"  
4 console.log(typeof 5.5);        // "number"  
5 console.log(typeof undefined);  // "undefined"  
6 console.log(typeof symbol());   // "symbol"
```

```
console.log(typeof null);        // "object"  
console.log(value === null);     // true or false
```

Типове данни

Примитивни типове - методи

```
1 var name = "JavaScript rocks";  
  
2 let price = 1.95583;  
    temp = new String("JavaScript rocks");  
  
3 name.charCodeAt(4);  
    temp.charCodeAt(4); temp = null;  
    temp = new Number(1.95583);  
  
4 price.toFixed(2);  
    temp.toFixed(2); temp = null;
```

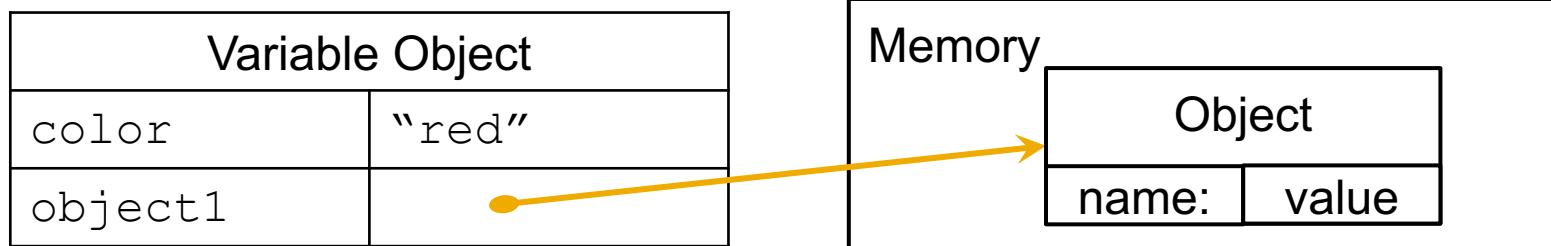
autoboxing

.. Въпреки това те не са обекти!

Типове данни

Референтни типове

- ✓ Репрезентират обекти
- ✓ Всеки референтен тип може да има референтни стойности (инстанции/обекти)
- ✓ Неподреден списък със свойства(**properties**), всяко от които има име и стойност
- ✓ Когато стойността на дадено свойство е **функция**, тя се нарича **метод**
- ✓ Присвояването на стойност отговаря на записване на **указател** към истинския обект



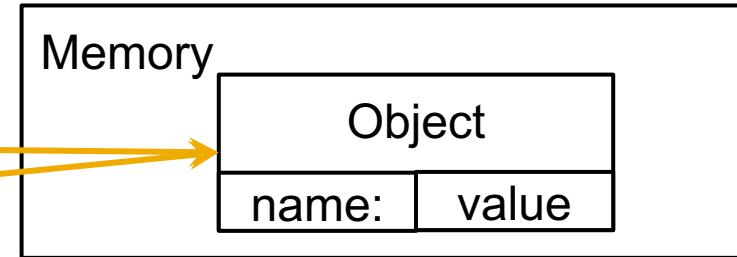
Типове данни

Референтни типове – присвояване

```
1 let object1 = new Object();  
2 let object2 = object1;
```



Variable Object	
object1	
object2	



Типове данни

Ключови думи за дефиниране

```
1 const outer = 42;
2 if (true) {
3     var foo = 5;
4     let bar = 10;
5     const inner = 45;
6 }
7 console.log(foo);           // 5
8 console.log(bar);          ? // Uncaught ReferenceError: bar is not defined
9 console.log(inner);        ? // Uncaught ReferenceError: inner is not defined
10 console.log(outer);       // 42
11 outer += 10;              // Uncaught TypeError: Assignment to constant variable
```

const - константи

var – видими в цялата функция

*let – видими в блока, в който са
дефинирани*



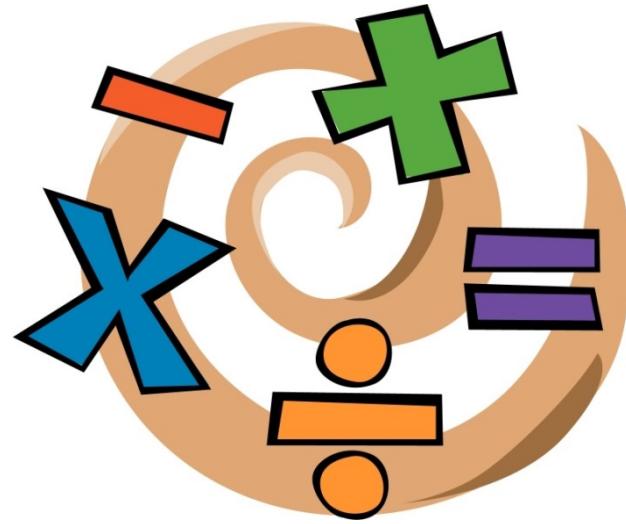
Demo

Public





Оператори



Оператори

Преглед

Аритметични оператори

+ , - , * , / , % , ++ , --

```
let a = 5;
let b = 4;

console.log(a + b); // 9
console.log(a - b); // 1
console.log(a * b); // 20
console.log(a / b); // 1.25
console.log(a % b); // 1

console.log(a++); // ?
console.log(b--); // ?

console.log(a); // ?
console.log(b); // ?
```

Оператори

Преглед

Логически оператори

&&, ||, !

```
const a = true;  
const b = false;
```

```
console.log(a && b); // false  
console.log(a || b); // true  
console.log(!a); // false
```

Бележка: Short circuit evaluation

```
let a = false && (5 + 3);  
  
let c;  
let b = c && (5 + 3);
```

Оператори

Преглед

Побитови оператори

&, |, ^, ~, <<, >>

```
let a = 5;  
let b = 1;
```

`console.log(a & b); // 1` 0101 & 0001 = 0001

`console.log(a | b); // 5` 0101 | 0001 = 0101

`console.log(a ^ b); // 4` 0101 ^ 0001 = 0100

`console.log(~ a); // 10` ~0101 = 1010

`console.log(a >> 1); // 2` 0101 >> 1 = 0010

`console.log(a << 1); // 10` 0101 << 1 = 1010

Оператори

Преглед

Оператори за сравнение

`==`, `!=`, `<`, `>`, `<=`, `>=`, `====`, `!==`

```
let a = 5;
let b = 4;

console.log(a == b); // false
console.log(a != b); // true
console.log(a === b); // false
console.log(a !== b); // true
console.log(a > b); // true
console.log(a < b); // false
```

Оператори

Преглед

Оператори за присвояване

=, +=, -=, *=, /=, %=

```
let a = 0;

console.log(a += 5); // 5
console.log(a); // 5

console.log(a -= 3); // 2
console.log(a *= 6); // 12
console.log(a /= 3); // 4
console.log(a % = 2); // 0
```

Оператори

Предимство

Тежест	Тип	Примери
20	Групиране	(..)
19	Достъп до член променливи	object.key
...		
16	Логическо NOT	!myVar
14	Умножение, деление, остатък	3 * 5, 5 / 2, 5 % 3 (left to right)
13	Събиране, изваждане	3 + 5, 5 - 2
11	Сравнение	x > y
10	Равенство	x == y, x != y
6	Логическо И	x && y
5	Логическо ИЛИ	x y
1	Запетая, последователност	x, y

(5 + 8 + 2) * 2

5 + 8 + 2 * 2

!true && false || true

!true && true || false

Пълна информация - [тук](#)

Оператори

Условни оператори

- ✓ **if конструкция - >**

```
1 if (условие){  
2     код за изпълнение  
3 }
```

- ✓ **if-else конструкция - >**

```
1 if (условие) {  
2     код за изпълнение  
3 } else {  
4     // ако условието не е вярно се изпънява този блок  
5     код за изпълнение  
6 }
```

- ✓ **Вложени условия - >**

```
1 if (условие_1) {  
2     код за изпълнение  
3 } else if (условие_2) {  
4     // ако условие_1 е грено, а условие_2 вярно  
5     код за изпълнение  
6 } else {  
7     // ако всички по-горни условия са грешни  
8     if (условие_3) {  
9         код за изпълнение  
10    } else {  
11        // ако условие_3 е грешно  
12    }  
13 }
```

Оператори

Условни оператори

✓ switch-case конструкция

```
1  switch (day) {  
2      case 1:  
3          console.log('Monday');  
4          break;  
5      case 2:  
6          console.log('Tuesday');  
7          break;  
8      case 3:  
9          console.log('Wednesday');  
10         break;  
11     case 4:  
12         console.log('Thursday');  
13         break;  
14     case 5:  
15         console.log('Friday');  
16         break;  
17     case 6:  
18         console.log('Saturday');  
19         break;  
20     case 7:  
21         console.log('Sunday');  
22         break;  
23     default:  
24         console.log('Error!');  
25         break;  
26 }
```

Оператори

Условни оператори – false стойности (1/3)

```
1 let logged = false;  
2 if (logged) {  
3     alert("Отключи сейфа");  
4 } else {  
5     alert("Грешна парола");  
6 }  
7
```



An embedded page at www.google.de says

Грешна парола

OK

Оператори

Условни оператори – false стойности (2/3)

```
1 let logged = new Boolean(false); //Wrapper for prim. boolean
2 if (logged) {
3     alert("Отключи сейфа");
4 } else {
5     alert("Грешна парола");
6 }
```



An embedded page at www.google.de says
Отключи сейфа

OK

- ✓ **false**
- ✓ **0 (нула)**
- ✓ **null**
- ✓ **"" (празен низ)**
- ✓ **undefined**
- ✓ **NaN**

Всички други стойности в JavaScript се считат за true!

Оператори

Условни оператори – false стойности (3/3)

```
1 if (false) { ...not true... }
2 if (0) { ...not true... }
3 if ("") { ...not true... }
4 if (null) { ...not true... }
5 if (undefined) { ...not true... }
6
7 if (1) { ...it is true... }
8 if ("0") { ...it is true... }
9 if (new Object()) { ...it is true... }
```



Demo

Public



SAP®

> `typeof(NaN)`
< "number"





Цикли

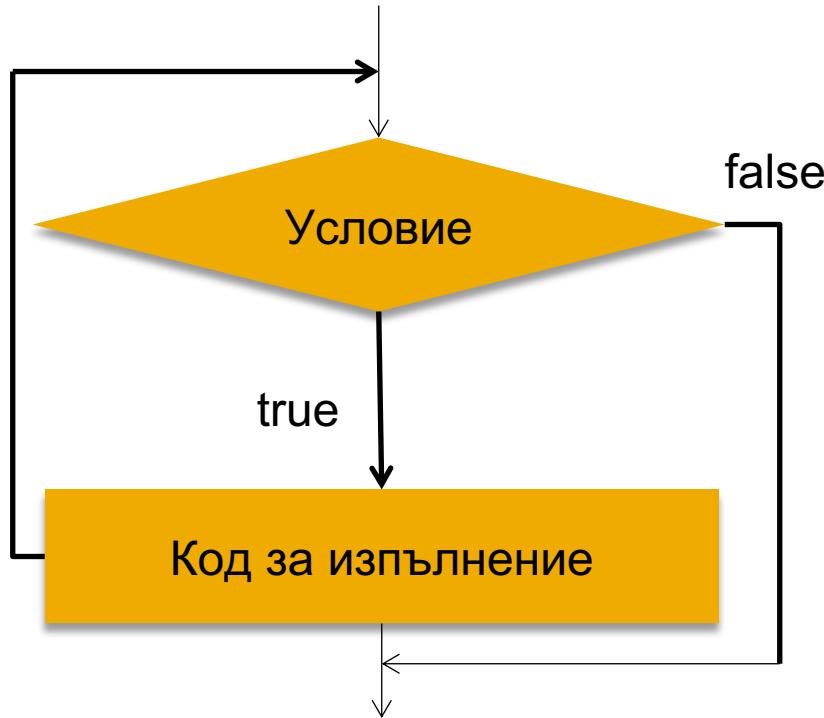


LOOPS REPEAT
ACTIONS...
SO YOU DON'T HAVE TO

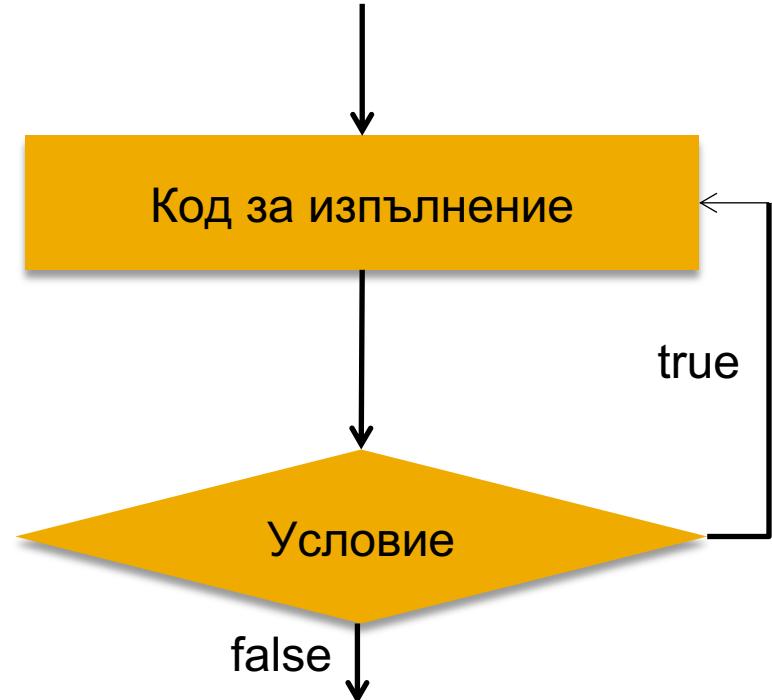
Цикли

while и for цикли

while



do-while



Цикли

while и for цикли

- ✓ **while - >**

```
1 while (условие) {  
2     |     код за изпълнение  
3 }
```

- ✓ **do while - >**

```
1 var counter = 0;  
2 do {  
3     |     counter++;  
4     |     console.log(counter);  
5 } while (counter < 10);
```

- ✓ **for - >**

```
1 for (var i = 1; i <= 10; i++) {  
2     |     alert(i);  
3 }
```

- ✓ **for in - >**

```
1 var array = [9, 8, 7, 6, 5, 4, 3, 2, 1];  
2  
3 for (var index in array) {  
4     |     console.log('index: ' + index + ' Number: ' + array[index]);  
5 }
```

Цикли

Приложение

- ✓ **Търсене**
- ✓ **Сортиране**
- ✓ **Обхождане на масиви, матрици, кубове и т.н.**

- ✓ **Повтаряне на една и съща операция няколко пъти**



Demo

Public





Благодаря Ви !

Контакти:

Филип Сидеров
filiп.siderov@sap.com

SAP Labs Bulgaria