

Основи на JavaScript

I-ва част

26.10.2015

Владислав Илиев

vladislav.iliev@sap.com

Public



Съдържание

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



Въведение

```
1 if (a > 3) {  
2   alert("Impossible!");  
3 } else {  
4   alert("The universe is in order.");  
5 }  
  
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 интерактивна е страницата на

JavaScript?
Разпознаване на браузър 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 достъпва бази от данни

Да пише и чете от локалната файлова система

Да затваря прозорци, които не са отворени от него

Да пише и чете от локалната файлова система

Да достъпва уеб страници, хоствани на друг домейн

Да защити използваните картинки или сорс-код

Въведение

История и версии

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

Mocha -> LiveScript - > JavaScript

JavaScript or Java

Въведение

История и 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 Cowlshaw
3	12, 2009	Try/catch blocks, Regula Expressions, better string handling	Mike Cowlshaw
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, modues, iterators, collections	Allen Wirfs-Brock
7	Work in Progress	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



Типове данни

Примитивни типове

Представят малки парчета от данни:

- ✓ Boolean – true или false
- ✓ Number – всяко целочислено 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

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

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 2015);             // "number"
3 console.log(typeof true);             // "boolean"
4 console.log(typeof 5.5);              // "number"
5 console.log(typeof undefined);        // "undefined"
```

```
console.log(typeof null);              // "object"
```

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

Типове данни

Примитивни типове - методи

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

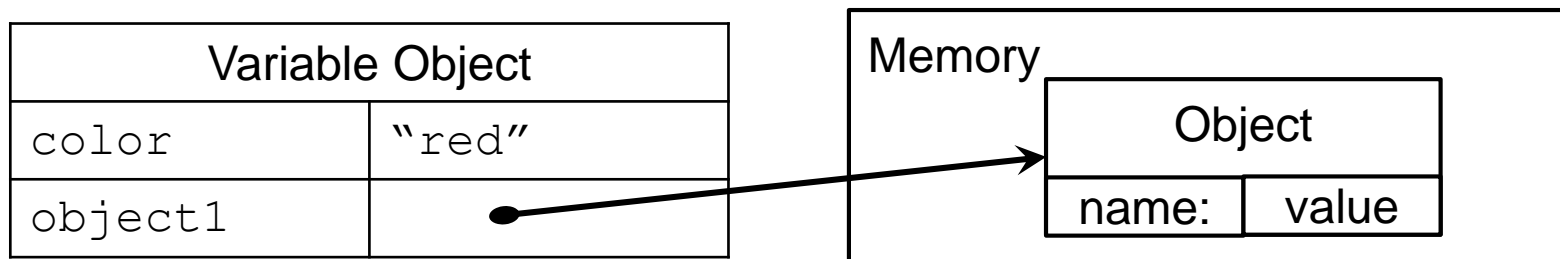
autoboxing

.. въпреки това те не са обекти!

Типове данни

Референтни типове

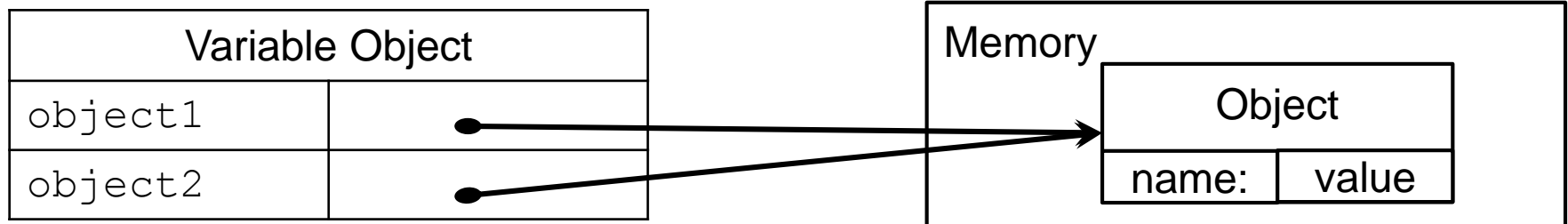
- ✓ Представяват обекти в езика
- ✓ Всеки референтен тип може да има референтни стойности (инстанции/обекти)
- ✓ Непореден списък със свойства(properties), всяко от които има име и стойност
- ✓ Когато стойността на дадено свойство е функция, тя се нарича метод
- ✓ Присвояването на стойност отговаря на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



Типове данни

Референтни типове – присвояване

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



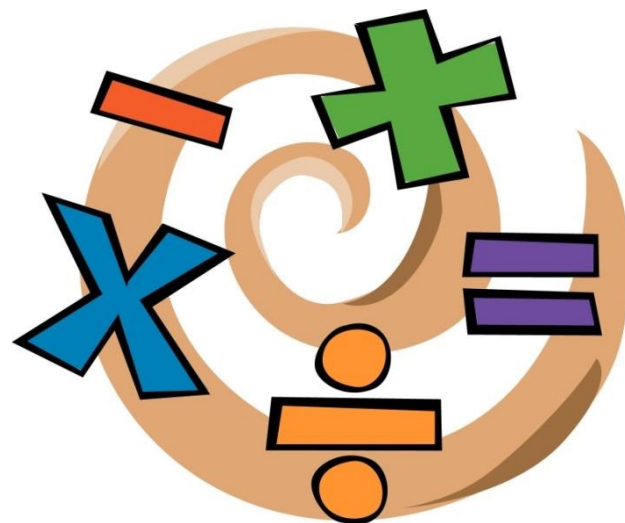


Demo

Public



Оператори



Оператори

Преглед

Аритметични оператори

✓ + - * / % ++ --

Логически оператори

✓ && || !

Побитови оператори

✓ & | ^ ~ << >>

Оператори за сравнение

✓ == != < > <= >= === !==

Оператори за присвояване

= += -= *= /= %= (пример $x/=4 \rightarrow x = x / 4$)

Оператори

Предимство

Тип	Примери
member	. []
call / create instance	() new
negation/ increment	! ~ - + ++ -- typeof void delete
multiply/divide	* / %
addition/ subtraction	+ -
bitwise shift	<< >> >>>
relational	< <= > >= in instanceof
equality	== != === !==
bitwise shift	<< >> >>>

Тип	Примери
relational	< <= > >= in instanceof
equality	== != === !==
bitwise-and	&
bitwise-xor	^
bitwise-or	
logical-and	&&
logical-or	
conditional	?:
assignment	= += -= *= /= %= <<= >>= >>>= &= ^= =
comma	,

Detailed list:

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 изпълнява 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 грешно, 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

✓ swich-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 var logged = new Boolean(false);  
2 if (logged) {  
3     alert("Отключи сейфа");  
4 } else {  
5     alert("Грешна парола");  
6 }
```



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

Всички други стойности в JavaScript се считат за true!



Demo

Public

```
> typeof(NaN)  
« "number"
```





Цикли

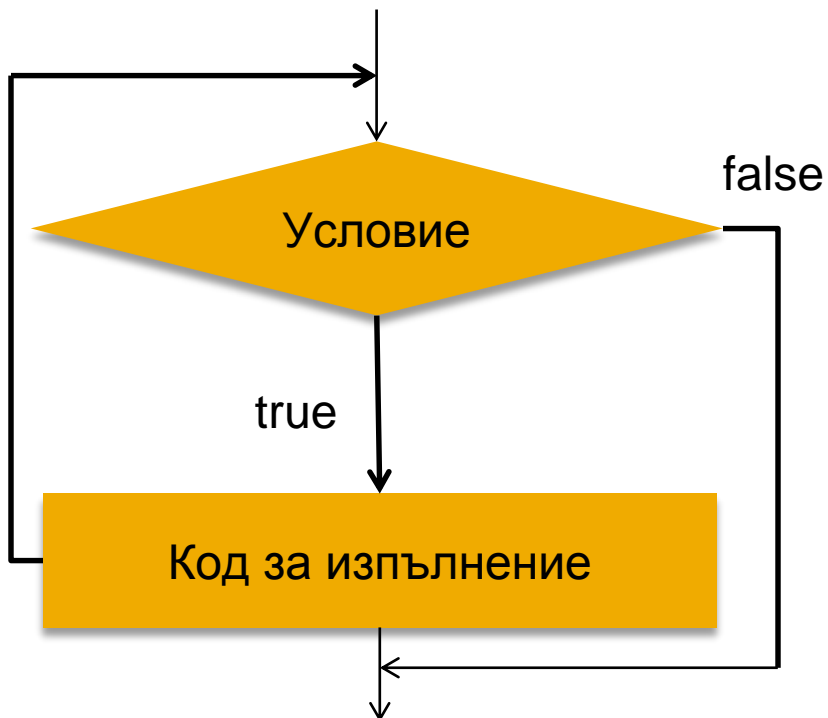


LOOPS REPEAT
ACTIONS...
SO YOU DON'T HAVE TO

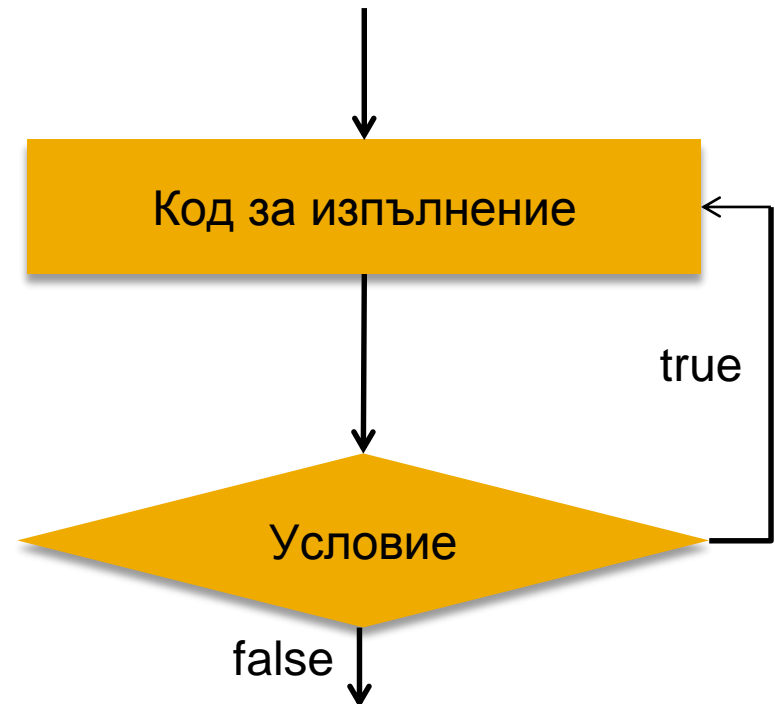
Цикли

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.**
- ✓ **Генериране на комбинации**
- ✓ **Повтаряне на една и съща операция няколко пъти**



Demo

Public



Благодаря Ви !

Контакти:

Владислав Илиев
vladislav.iliev@sap.com

SAP Labs Bulgaria
София, бул.Цар Борис III, 136А

Типове данни

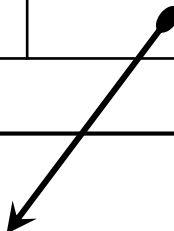
Примитивни и референтни типове

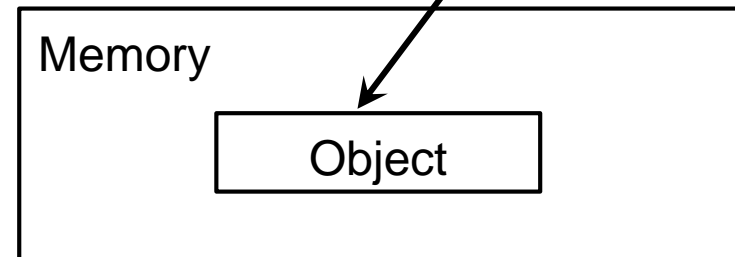
Примитивите типове данни се записват като прости типове данни, референтните – като обекти

В heap паметта

Variable Object

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

Variable Object	
color	"red"
object1	



Variable Object	
color1	"blue"
color2	"red"