

# JAVASCRIPT

**ВЪВЕДЕНИЕ. СИНТАКСИС. ТИПОВЕ  
ДАНИИ**

# СЪДЪРЖАНИЕ

- МАЛКО ИСТОРИЯ ...
- ЗА КАКВО СЕ ИЗПОЛЗВА JAVASCRIPT?
- ТИПОВЕ ДАННИ

**ДА ПРИПОМНИМ**



**<html>**

**.css {}**



## КАКВО Е HTML?

HyperText Markup Language

Език за репрезентиране на уеб страници

Дефинира съдържанието на страницата, но не  
функционалостта ѝ

# КАКВО Е CSS?

Cascading Style Sheets

Описва презентацията на HTML или XML документи

Описва как елементите трябва да се рендират на страницата

## КАКВО Е РАЗЛИКАТА МЕЖДУ

`a, #a, .a, *`

`color, background-color, border-color`

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

Какво е `!important`?

# JAVASCRIPT





## **МАЛКО ИСТОРИЯ**

- Създаден за 10 дни през май 1995
- Mocha - > LiveScript - > JavaScript
- Brendan Eich (Netscape)
- 6 големи версии и 7ма в прогрес

## JAVASCRIPT МОЖЕ ДА

- Добавя функционалност в уеб сайтове
- Валидация на форми
- Връзка client-server
- Кеширане при клиента
- Контрол на бисквитки

## JAVASCRIPT НЕ МОЖЕ ДА

- Записва файлове на сървър
- Пише и чете от файловата система
- Контролира прозорци, които не са отворени от него
- Защити използваните картинки и сорс код в уеб страница

# NODEJS - СЪРВЪРЕН JAVASCRIPT





## КАТО ЦЯЛО

- Мултиплатформен
- Скриптов
- Нетипизиран
- Обектно ориентиран
- Може да работи при клиента и на сървъра

# JAVASCRIPT & ТИПОВЕ ДАННИ



# ДОБАВЯНЕ НА JAVASCRIPT

script таг в HTML страницата

```
<script>  
    // my JavaScript here  
</script>
```

чрез външен файл

```
<script src="path/script.js" type="text/javascript"></script>
```

async & defer

## VAR VS. \*LET VS. \*CONST

- var - function scope
- \*let - block scope
- \*const - constants



## ПРИМИТИВНИ ТИПОВЕ

- булеви
- null
- undefined
- Number
- String
- \*Symbol

# ПРИМЕР

```
var a = 5;  
var b = a;
```

```
b = 10;
```

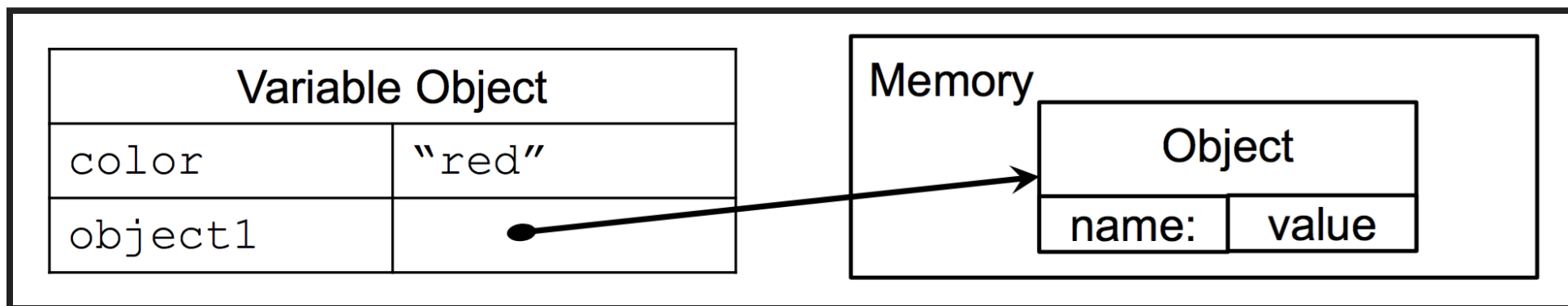
```
console.log(a); // 5  
console.log(b); // 10
```

# typeof

```
console.log(typeof "FMI"); // string
console.log(typeof 2016); // number
console.log(typeof true); // boolean
console.log(typeof 5.5) // number
console.log(typeof undefined); // undefined
console.log(typeof null); //object
```

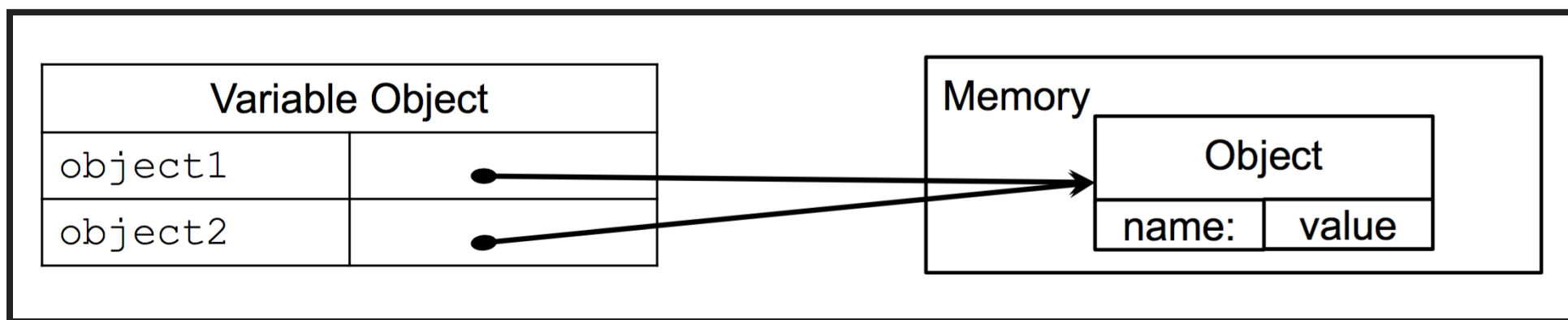
# РЕФЕРЕНТНИ ТИПОВЕ - ОБЕКТИ

- Неподреден списък от свойства, всяко от които има име и стойност
- В JavaScript почти всичко е обекти (освен примитивните типове)
- Присвояването на стойност отговаря на записването на указател към истинския обект в паметта



# РЕФЕРЕНТНИ ТИПОВЕ - ПРИСВОЯВАНЕ

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



**VAR THANK = "YOU!";**

```
var tutors = [{  
  name: "Мартин Христов",  
  email: "martin.r.hristov@sap.com"  
}];
```