

ОСНОВНИ ТИПОВЕ ДАННИ



Как ще се пресметне в JavaScript:

$$2 + 2 = 4$$

$$4 + "4" = 44$$

$$"44" - 4 = 40$$

$$44 + "2" - 2 = 440$$

$$44 - "2" + 2 = 44$$

$$44 - 2 + "2" = 422$$

Динамично типизиране

□ JavaScript е **динамично типизиран език**. При него променливите нямат предварително указан тип – той се определя според стойността, която е въведена в тях.

□ **пример:**

```
var x;  
alert(typeof x); // undefined  
x = 5;  
alert(typeof x); // number  
x = "Текст";  
alert(typeof x); // string  
x = true;  
alert(typeof x); // boolean
```

Числа

□ В JavaScript няма отделни категории за цели и дробни числа:

□ **пример:**

// цяло число в различни бройни системи

var n1 = 34; // 34, в десетична бройна система

var n2 = 0b10100001; // 161, в двоична бройна система

// дробно число в десетичен формат

var x1 = 34.00;

// дробно число в експонециален формат

var x3 = 123e5; // 12300000

var x3 = 123e-5; // 0.00123

Аритметични операции с числа

■ За числовите стойности са приложими следните операции:

Действие	Аритметичен оператор	Пример
Събиране	+	$a = 4 + 5.0;$
Изваждане	-	$b = 4 - 50;$
Умножение	*	$c = 4 * 5;$
Деление	/	$d = 10 / 2;$
Остатък	%	$e = 10 \% 4;$

Булев тип данни

□ Булевият тип данни има само две стойности: true и false.

□ пример:

```
// проверка намерено ли е нещо - отначало обявяваме, че  
    го няма
```

```
var nameren = false;
```

```
...
```

```
// ако го намерим, отбелязваме го в променливата
```

```
nameren = true;
```

Операции за булев тип данни

□ Приложими са следните основни операции:

Логически оператор	Действие	Пример
!	НЕ, логическо отрицание – връща вярно, когато операндът има стойност невярно	! true
&&	И, логическо умножение – връща вярно, когато и двата операнда имат стойност вярно	true && false
	ИЛИ, логическо събиране – връща вярно, когато поне единият операнд има стойност вярно	true false

Низове (текст)

□ Низовият тип данни съдържа текст. При низовите константи този текст трябва да бъде ограден с кавички или апострофи и може да съдържа всеки друг знак.

□ За низовете е дефинирана само операцията + (слепване)

□ пример:

```
// използване на кавички
```

```
var t1 = "Това е текст";
```

```
// използване на апострофи
```

```
var t2 = 'И това е текст';
```

```
// комбиниране на двете
```

```
var answer = "It's alright";
```

```
var answer = "His name is 'John' " + 'He is called "Johnny" ';
```


Операции за сравнение

□ Без проверка на типа:

Сравнителен оператор	Действие	Пример
==	Равенство, равни стойности	4 == "4"
!=	Неравенство, различни стойности	4 != 4.0
<	По-малко	-4 < 4
>	По-голямо	-10 > 4
<=	По-малко или равно	4 <= "4"
>=	По-голямо или равно	"10" >= 4

Операции за сравнение

□ С проверка на типа:

Сравнителен оператор	Действие	Пример
===	Равни стойности и от един и същ тип	4 === "4"
!==	Различна стойност или различен тип	4 !== "4"

Преобразуване между типовете

- Стойностите се преобразуват, така че да се извърши операцията:

```
alert(16 + "текст"); // 16текст
```

```
alert("текст" + 16); // текст16
```

```
alert(37 + "7"); // "377"
```

- Ако има операции, неподдържани от типа на първия операнд, се преминава към типа на втория операнд:

```
alert("37" - 7); // 30, за низовете няма операция изваждане
```

```
alert("37" + 7); // "377", но има операция събиране (слеждане)
```

- Изразите се пресмятат отляво надясно. Така че редът на стойностите има голямо значение:

```
document.write(16 + 4 + "текст"); // 20текст
```

```
document.write("текст" + 16 + 4); // текст164
```

Край

