SWIFTACADEMY

НИЕ ВЯРВАМЕ ВЪВ ВАШЕТО БЪДЕЩЕ


BRACE YOURSELF
A LOT OF JAVASCRIPT IS COMING
memegenerator.net

Изрази и оператори - II част

- Миналият път споменахме логическите и аритметичните оператори:
  - те се използват за да извършваме операции като сравнение и събиране върху **прости типове данни**
  - те могат да бъдат унарни (напр: !) или бинарни (напр: &&)
- също така говорихме за групиращия оператор **()** и как с негова помощ да правим композитни изрази като:

```
(wallet.amount >= (item.price * 3)) || (creditCard.balance > 100)
```

Изрази и оператори - II част

- В JavaScript обаче има още цял куп оператори, които можем да ползваме, като:
  - typeof
  - ++ и --
  - += и -=
- https://developer.mozilla.org/en/docs/Web/JavaScript/Reference/Operators

Тринарен оператор и switch

- if и else са запазени думи от езика и сформират конструкция
- подобен ефект има и конструкцията switch - case (https://developer.mozilla.org/en/docs/Web/JavaScript/Reference/Statements/switch)
- от зората на програмирането е останала и една short-hand конструкция или т.нар. тринарен оператор, който е кратък запис за if - else израза
- Синтаксис:

```
condition ? true_actions : false_actions
```

Цикли (loops)

- Когато искаме да повторим една и съща операция (изчисление) много пъти, използваме специална конструкция за повторение, която се нарича "цикъл"
- JS има три основни оператора за цикъл: for, while и do
  - for() {} цикъл
  - while() {} цикъл
  - do {} while() цикъл
- Array.prototype.forEach() - функция за обхождане на списъци от висок ред

Цикли (loops) - Синтаксис

- `for (`*задаване на начална стойност*`;` *условие за прекратяване*`;` *стъпка*`) {`
  `// действия`
  `}`

- `// задаване на начална стойност`
  `while (`*условие за прекратяване*`) {`
  `// действия`
  `// стъпка`
  `}`

- `// задаване на начална стойност`
  `do {`
  `// действия (ще се изпълнят минимум 1 път)`
  `// стъпка`
  `} while (`*условие за прекратяване*`);`

Примери

- ```
  for (var i=0; i < 10; i++) {
    console.log(i) // 0, 1, 2, 3, 4, 5, 6, 7, 8, 9
  }
  ```

- ```
  var i=0;
  while (i < 10) {
    console.log(i) // 0, 1, 2, 3, 4, 5, 6, 7, 8, 9
    i += 1;
  }
  ```

- ```
  var i=0;
  do {
    console.log(i) // 0, 1, 2, 3, 4, 5, 6, 7, 8, 9
    i += 1;
  } while (i < 10);
  ```
  ```

Пример forEach

```
function log(x) {
  console.log(x);
}

function double(x) {
  console.log(x*2);
}

var array = [1, 2, 3, 4, 5, 6, 7, 8];

array.forEach(log);    // 1, 2, 3, 4, 5, 6, 7, 8
array.forEach(double); // 2, 4, 6, 8, 10, 12, 14, 16
```

Упражнение

- Напишете програма, която проверява дали в един списък от числа, всички числа са четни
- Използвайте всеки един от операторите за обхождане, като за всяко от решенията използвайте различна функция (forCheckEven(), doCheckEven(), whileCheckEven())
- Пример:

```
array = [2, 14, 6, 24, 8]
forCheckEven(array) // => true
array = [32, 44, 26, 9, 4]
whileCheckEven(array) // => false
```

Упражнение 2

- Напишете функция, която получава като аргумент списък от числа и връща резултат сумата от числата
- Напишете функция, която сумира само четните числа и такава, която сумира само нечетните

- Пример:

```
array = [2, 1, 16, 3, 9]
sum(array) // => 31
sumEven(array) // => 18
sumOdd(array) // => 13
```