



УЕБ РАЗРАБОТКА

JS Масиви

Съдържание

- Деклариране и създаване на масив
- Достъпване елементите на масива
- Работа с елементи на масива
- Добавяне/премахване елементи на масива
- Операции с масивите:
 - Конкатениране
 - Разрязване
 - Манипулация
- Обекти в JS

Массивы в JS

Масиви в JS

- Какво представляват масивите?
- Деклариране на масив
 - Масив от числа
 - `var numbers = [1, 2, 3, 4, 5];`
 - Масив с елементи – стрингове
 - `var names = ['Pesho', 'Gosho', 'Ivan']`
 - Масив с елементи от смесен тип
 - `var mixedArr = [1, new Date(), 'hello'];`

Масиви в JS

- Матрица – масив от масиви/многомерен масив

```
var matrix = [ ['0,0', '0,1', '0,2'],
               ['1,0', '1,1', '1,2'],
               ['2,0', '2,1', '2,2']];
```

Деклариране и инициализиране на масиви

- Препоръчителен вариант – []

- var arr = [1, 2, 3, 4, 5];
- var myArr = [];



var arr = new Array(1, 2, 3, 4, 5); //повече от едно число – елементи на масива

var arr = new Array(10); //едно число – брой на елементите на масива, със стойност - undefined

Достъпване елементите на масива

Достъпване елементите на масива

Достъпваме и работим с елементите на масива

arr[index]

Задаваме/променяме стойност на елемент на масива

arr[index] = some value

index = 0 до arr.length-1

Добавяме елемент на масива

arr[index] = some new element in array

Достъпване на елементите на масива - 2

arr.push()

arr.unshift()

Задача: Обърнете елементите на масив с помощта на

- push()
- unshift()

първия елемент да стане с най-голям индекс, втория с предпоследен индекс и т.н.

Достъпване елементите на масива - 3

Итериране на масива с for/for-in

For – когато индексите са последователни и известни

For-in – когато не знаем индексите на масива и/или не са последователни

Добавяне/премахване на елементи на масива

Добавяне/премахване на елементи на масива

Масивите са динамични – в процеса на изпълнение на скрипта може да добавяме/премахваме елементи.

- `Array.push()` – добавя елемент в края на масива
- `Array.pop()` – премахва последния елемент в масива, връща премахнатия елемент.
- `Delete array[index]` – изтрива стойността на съответния елемент, но не и самия елемент
- Премахваме всички елементи от масив
 - `Arr.length = 0;`

Добавяне/премахване на елементи на масива - 2

Масивите са динамични – в процеса на изпълнение на скрипта може да добавяме/премахваме елементи.

- `Array.unshift(element)` **добавя елемент в началото на масива**
- `Array.shift()` – премахва първия елемент в масива, връща премахнатия елемент

JS функции за манипулация на масиви

JS функции за манипулация на масиви

- **Array.reverse()**
 - обръща реда/индексите/ на елементите;
 - Връща нов масив;
- **Array.join(separator)**
 - Конкатенира елементите на масив /с разделител между тях/
 - Връща стринг
- **arr1.concat(arr2)**
 - добавя елементите на arr2 в края на arr1
 - Връща нов масив. arr1 и arr2 не се променят!

JS функции за манипулация на масиви - 2

- Добавяне на елементите на един масив към друг
 - var arr1 = [1, 2, 3];
 - var arr2 = ["one", "two", "three"];
 - **[].push.apply(arr1, arr2);**
- Array.slice(fromIndex [, toIndex])
 - Връща нов масив
 - Копие на част от масива, съдържащо елементите с индекси от fromIndex до (без да включва toIndex).
 - Може да използвате за клониране на масив

JS функции за манипулация на масиви - 3

- `Array.splice(index, count, elements)`
 - Премахва елемент/и, от който и да е index на масива, като посочваме колко елемента да премахне

Премахва един елемент на позиция index

`items.splice(index, 1);`

Премахва посочения брой елементи на позиция index

`items.splice(index, count);`

Пробвайте – какво се случва с оригиналния масив и какво връща splice?

JS функции за манипулация на масиви - 4

- **Array.splice**

Добавяне на елемент на позиция index в масива.

`items.splice(index, 0, element)`

Добавяне на елементи, започвайки от позиция index.

`items.splice(index, 0, el1, el2)`

JS функции за манипулация на масиви

- [Array.splice](#)

Добавяне на елемент на позиция index в масива.

```
var myFish = ['angel', 'clown', 'mandarin', 'sturgeon'];
myFish.splice(2, 0, 'drum');
```

```
// insert 'drum' at 2-index position
// myFish is ["angel", "clown", "drum", "mandarin", "sturgeon"]
```

JS функции за манипулация на масиви

- **Array.indexOf(element [, rightOf])**
 - Връща индекса на първия елемент, равен на търсения//
 - **rightOf – откъде да започне търсенето**
 - Връща -1, ако няма такъв елемент в масива
- **Array.lastIndexOf(element, [leftOf])**
 - Връща индекса, на първия елемент, отговарящ на търсенето
 - **leftOf – откъде да започне търсенето**
 - Връща -1, ако няма такъв елемент в масива
- **Array.indexOf() и Array.lastIndexOf()** не работят на всички браузъри!

JS функции за манипулация на масиви

- Официалната документация за масиви:
 - https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Array
- Как да потвърдим, че променливата е масив - **array**
 - `typeof([1, 2, 3]) // резултат - object`
 - `Array.isArray([1, 2, 3]) // резултат – true`
 - `Array.isArray('string') // резултат – false`

Обекти в JS

Обекти в JS

Object



Properties

car.name = Fiat

car.model = 500

car.weight = 850kg

car.color = white

Methods

car.start()

car.drive()

car.brake()

car.stop()

Обекти в JS

```
myObject = {} //      декларираме променлива, тип обект
```

```
myObject = { property1: 10,  
            property2: 'anotherValue',  
            property3: true } //декларираме свойствата на обекта
```

```
console.log(myObject.property1)//10
```

```
myObject.newProperty = 123654
```

```
console.log(myObject.newProperty)//123654
```

Обекти в JS

```
var object1 = {
    a: 'somestring',
    b: 42,
    c: false
};

for (var property in object1) {
    if (object1.hasOwnProperty(property)) {
        console.log(object1[property])
    }
}

'something', 42, false
```

Обекти в JS

```
var object1 = {  
    a: 'somestring',  
    b: 42,  
    c: false  
};  
  
console.log(Object.keys(object1));  
// expected output: Array ["a", "b", "c"]
```

Обекти в JS

```
var fruits = {  
    apple: 28,  
    orange: 17,  
    pear: 54,  
}
```

```
var values = Object.values(fruits)  
console.log(values) // [28, 17, 54]
```

Обекти в JS

```
const fruits = {  
    apple: 28,  
    orange: 17,  
    pear: 54,  
}  
  
const entries = Object.entries(fruits)  
console.log(entries)  
// [ [apple, 28],  
//   [orange, 17],  
//   [pear, 54] ]
```