



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

Масиви 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
 - Demo 1

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`

Demo 2

Достъпване елементите на масива - 2

`arr.push()`

Задача: Обърнете елементите на масив с помощта на `push()`/първия елемент да стане с най-голям индекс, втория с предпоследен индекс и т.н.

Достъпване елементите на масива - 3

/виж презентация – Loops in JS/

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

Demo

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;`

demo

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

demo

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"];`
 - `arr1.push.apply(arr1, arr2); /demo`
- `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 [/demo](#)

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)
```

[demo](#)

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! [demo](#)

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
- [demo](#)