

Массивы

Обработка последовательностей элементов

Что такое массив?

- **массив** это последовательность элементов
 - Порядок следования элементов фиксирован
 - Не имеют фиксированного размера
 - Можно получить текущий размер (`Array.length`)
- Массив из 5 элементов
- Индекс элемента - 3 `array[3]`
- Элемент массива передан в переменную `let a = array[3]`

Объявление массива

```
// Массив хранящий цифры
var numbers = [1, 2, 3, 4, 5];

// Массив хранящий строки
var weekdays = ['Понедельник', 'Вторник', 'Среда',
  'Четверг', 'Пятница', 'Суббота', 'Воскресенье']

// Массив хранящий различные типы
var mixedArr = [1, new Date(), 'привет'];

// Массив массивов (матрица)
var matrix = [
  ['0,0', '0,1', '0,2'],
  ['1,0', '1,1', '1,2'],
  ['2,0', '2,1', '2,2']];
```

- Проинициализировать массив в JavaScript можно тремя способами:
 - Используя `new Array(elements)` :

```
var arr = new Array(1, 2, 3, 4, 5);
```

- Используя `new Array(initialLength)` :

```
var arr = new Array(10);
```

- Используя литерал массив (рекомендуется):

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

Как получить доступ к элементам массива?

- Получить доступ к элементу массива можно с помощью **индексного оператора**: `[]` (квадратные скобки)
 - Индекс массива может быть в диапазоне `[0; length-1]`
 - Первый элемент массива `0`
 - Последний элемент массива `length-1`
- Массив элементов может быть сброшен или обновлен так - `[]` (индексный оператор)

```
//всегда используйте var переменные вверху области видимости!  
var array,  
    len,  
    reversed,  
    i,  
    j;  
  
array = [1, 2, 3, 4, 5];  
reversed = [];  
  
for (i = 0, len = array.length; i < len; i += 1) {  
    j = len - i - 1;  
    reversed.push(array[j]);  
}
```

Проход по массиву с помощью конструкции `for`

- Используйте `for` для прохода по массиву, если Вам надо контролировать доступ к членам массива
- В теле цикла используйте индекс по которому осуществляется доступ к элементу (`array[index]`):

```
var i, len;  
for (i = 0, len = array.length; i < len; i += 1) {  
    squares[i] = array[i] * array[i];  
}
```

- Распечатать элементы массива в обратном порядке:

```
var arr, i, len;
arr = [1, 2, 3, 4, 5];
for (len = arr.length, i = len - 1; i >= 0; i -= 1) {
    console.log(arr[i]);
}
// Результат: 5 4 3 2 1
```

- Проинициализировать все элементы массива их индексом:

```
var i, len
for (i = 0, len = array.length; i < len; i += 1) {
    array[i] = i;
}
```

Проход по массиву с помощью цикла `for-in`

- Как работает цикл `for-in` ?
 - `index` интерация происходит по индексам всего массива
- Используйте, когда индексы неизвестны
 - Доступ ко всем элементам осуществляется последовательно
 - Порядок не гарантируется
 - Подходит для работы с объектами

```
var index;
for (index in array) {
    // ваш код
}
```

- Вывести построчно все элементы массива:

```
var capitals, i;
capitals = [
    'Sofia',
    'Washington',
    'London',
    'Paris'
];

for (i in capitals) {
    console.log(capitals[i]);
}
```

Добавление и удаление элементов из массива

- Все массивы в JavaScript являются **динамическими**

- Их размер может меняться в процессе выполнения программы
 - В массив могут быть добавлены новые элементы
 - Из массива могут быть удалены любые элементы
- `Array#push(element)`
 - **Добавляет** новый элемент в **конец** массива
- `Array#pop()`
 - **Удаляет** элемент из **конца** массива
 - **Возвращает** удаленный элемент
- `Array#unshift(element)`
 - **Добавляет** новый элемент в **начало** массива
- `Array#shift()`
 - **Удаляет** элемент из **начала** массива
 - **Возвращает** удаленный элемент

Методы массива

- `Array#reverse()`
 - **Переворачивает** порядок элементов в массива
 - **Возвращает** новый массив

```
var items = [1, 2, 3, 4, 5, 6];
var reversed = items.reverse();
//reversed = [6, 5, 4, 3, 2, 1]
```

- `Array#join(separator)`
 - **Позлементная конкатенация** всех элементов через разделитель
 - **Возвращает** строку

```
var names = ["John", "Jane", "George", "Helen"];
var namesString = names.join(", ");
//namesString = "John, Jane, George, Helen"
```

Конкатенация массивов

- `arr1.concat(arr2)`
 - **Добавляет** элементы массива `arr2` в **конец** массива `arr1`
 - **Возвращает** новый массив
 - `arr1` и `arr2` остаются неизмененными!

```
var arr1 = [1, 2, 3];
var arr2 = ["one", "two", "three"];
var result = arr1.concat(arr2);
//result = [1, 2, 3, "one", "two", "three"]
```

- Добавление элементов одного массива к другому массиву

```
var arr1 = [1, 2, 3];
var arr2 = ["one", "two", "three"];
[].push.apply(arr1, arr2);
//arr1 = [1, 2, 3, "one", "two", "three"]
```

Получение части массива

- `Array#slice(fromIndex [, toIndex])`
 - **Возвращает новый массив**
 - Неполную копию части массива
 - Новый массив содержит элементы с `fromIndex` до `to` (исключая `toIndex`)
 - Может быть применимо для клонирования массива

```
var items = [1, 2, 3, 4, 5];
var part = items.slice(1, 3);
//part = [2, 3]
var clonedItems = items.slice();
```

Вставка в массив

- `Array#splice(index, count, elements)`
 - Удаляет `count` элементов, начиная с `index` позиции
 - Добавляет `elements` с позиции `index`
 - Возвращает новый массив содержащий удаленные индексы

```
var numbers = [1, 2, 3, 4, 5, 6, 7];
var result = numbers.splice(3, 2, "four", "five", "five.five");
//result = [4, 5]
//numbers = [1, 2, 3, "four", "five", "five.five", 6, 7];
```

- Удаление элементов по индексу из массива:

```
//удалить один элемент с позиции `index`
items.splice(index, 1);
//удаляет count элементов начиная с позиции index
items.splice(index, count);
```

- Добавить элемент по индексу в массив:

```
//Добавляет один элемент в позицию index
items.splice(index, 0, element);
//добавляет несколько элементов начиная с позиции index
items.splice(index, 0, item1, item2, item3);
```

Поиск в массиве

- `Array#indexOf(element [, rightOf])`
 - Возвращает индекс первого вхождения искомого значения в массиве
 - Возвращает `-1` если элемент не найден
- `Array#lastIndexOf(element, [leftOf])`
 - Возвращает индекс последнего вхождения искомого значения в массиве
 - Возвращает `-1` если элемент не найден
- `Array#indexOf()` и `Array#lastIndexOf()` работают в большинстве браузеров

Другие функции массивов

- Официальная документация по массивам:
 - https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Array
- Проверка массива
 - `typeof([1, 2, 3]) -> object`
 - Не работает
 - `Array.isArray([1, 2, 3]) -> true`
 - Поддерживается в большинстве современных браузеров