

Язык JavaScript (основы: массивы)

A large, bold, black 'JS' logo is centered on the page. The 'J' and 'S' are stylized with thick strokes and rounded terminals, typical of the JavaScript logo.

МАССИВЫ С ЧИСЛОВЫМИ ИНДЕКСАМИ

Массив – разновидность объекта, которая предназначена для хранения пронумерованных значений и предлагает дополнительные методы для удобного манипулирования такой коллекцией.

Обычно используются для хранения упорядоченных коллекций данных, например – картинок галереи, статей и т.п.

В массиве может храниться любое число элементов любого типа.

Объявление пустого массива

```
let arr = [ ];
```

Объявление массива с элементами

```
let pictures = [ 'summer.png', 'winter.png', 'autumn.png',  
'spring.png' ];
```

Элементы массива нумеруются, начиная с нуля

МАССИВЫ С ЧИСЛОВЫМИ ИНДЕКСАМИ

Элементы нумеруются, начиная с нуля.

Чтобы получить нужный элемент из массива необходимо указать его номер в квадратных скобках:

```
let pictures = [ 'summer.png', 'winter.png', 'autumn.png',  
'spring.png' ];
```

```
console.log(pictures[0]); //'summer.png'
```

```
console.log(pictures[1]); //'winter.png'
```

```
console.log(pictures[2]); //'autumn.png'
```

```
console.log(pictures[3]); //'spring.png'
```

МАССИВЫ С ЧИСЛОВЫМИ ИНДЕКСАМИ

Изменить элемент массива

```
pictures[1] = 'другая картинка';
```

Добавить новый элемент в массив

```
pictures[4] = 'новая картинка';
```

Получить количество элементов массива

```
let picturesLength = pictures.length; // свойство length
```

Длина `length` – не количество элементов массива, а последний индекс + 1.

МЕТОДЫ ДЛЯ РАБОТЫ С МАССИВОМ

Метод - это функция, принадлежащая какому-то объекту.

Обращаться к методам можно через оператор току . ,

например **имяОбъекта.имяМетода()**

Методы **pop()** / **push()** работают с концом массива (работают быстрее):

- **pop()** - удаляет последний элемент из массива и возвращает его
- **push(добавляемый элемент)** - добавляет элемент в конец массива

Методы **shift()** / **unshift()** работают с началом массива (работают медленнее)

- **shift()** - удаляет из массива первый элемент и возвращает его
- **unshift(добавляемый элемент)** - добавляет элемент в начало массива

МЕТОДЫ ДЛЯ РАБОТЫ С МАССИВОМ

Из массива в строку / из строки в массив

1. **split**(разделитель, [ограничение на кол-во элементов]) - превращает строку в массив, разбив ее по разделителю (пробел и пустая строка "" - тоже символы). Аргумент ограничение на кол-во элементов позволяет создать массив заданной длины.
2. **join**([разделитель]) превращает массив в строку, элементы массива склеиваются в строку через разделитель.

МЕТОДЫ ДЛЯ РАБОТЫ С МАССИВОМ

splice(**start**, **delCount**[, **item1**[, **item2**[, ...]]]) позволяет удалять элементы, вставлять элементы, заменять элементы – по очереди и одновременно.

Возвращает массив, содержащий удаленные элементы.

- **start** - индекс, по которому начинается изменяться массив. Если он больше длины массива, реальный индекс будет установлен на длину массива. Если отрицателен, указывает индекс элемента с конца.

- **delCount** количество удаляемых из массива элементов. Если **delCount** равен 0, элементы не удаляются. В этом случае необходимо указать как минимум один новый элемент. Если **delCount** больше количества элементов, оставшихся в массиве, начиная с индекса **start**, то будут удалены все элементы до конца массива.

- **itemN** - элементы, добавляемые в массив.

МЕТОДЫ ДЛЯ РАБОТЫ С МАССИВОМ

slice([**begin**[, **end**]]) копирует участок массива от **begin** до **end**, не включая **end** и возвращает его в виде массива. Исходный массив при этом не меняется.

- **begin** - индекс элемента массива, по которому начинается извлечение.

Если begin отрицательный, **begin** указывает смещение от конца последовательности.

Если begin опущен, **slice()** начинает работать с индекса 0.

- **end** - Индекс элемента массива, по которому, по которому заканчивается извлечение.

Если end отрицательный, **end** указывает смещение от конца последовательности.

Если end опущен, **slice()** извлекает все элементы до конца последовательности.

МЕТОДЫ ДЛЯ РАБОТЫ С МАССИВОМ

reverse() - меняет порядок элементов в массиве на обратный.

concat(value1[, value2[, ...[, valueN]]) - возвращает новый массив, состоящий из массива, на котором он был вызван, соединённого с другими массивами и/или значениями.

- **valueN** - массивы и/или значения, соединяемые в новый массив.

indexOf(searchElement[, fromIndex]) возвращает номер элемента searchElement в массиве или -1, если его нет. Поиск начинается с номера fromIndex, если он указан. Если нет – с начала массива.

lastIndexOf(searchElement[, fromIndex]) ищет справа-налево: с конца массива или с номера fromIndex, если он указан.

КАК ПЕРЕБРАТЬ МАССИВ

Для перебора элементов массива используются циклы
Например,

```
let pictures = [ 'summer.png', 'winter.png', 'autumn.png',  
'spring.png' ];  
for (var i = 0; i < pictures.length; i++) {  
    console.log('Элемент массива №' + i + ' = ' +  
        pictures[i]);  
}
```

Цикл позволяет взять каждый (или необходимый, в зависимости от условия) элемент массива и совершить с ним какое-либо действие.

!!! Ненадо использовать цикл `for..in` для массивов, где важен порядок индексов, цикл `for` безопаснее и работает быстрее

ЧТО ПОЧИТАТЬ ДОПОЛНИТЕЛЬНО

<https://learn.javascript.ru/array> - Массивы

<https://learn.javascript.ru/array-methods> - Методы массивов