

Массивы

Основные понятия. Работа с массивами

Определение

Массив – совокупность переменных, которая позволяет хранить несколько однотипных значений под одним именем. Каждая переменная в массиве называется *элементом*. Каждый элемент имеет свой порядковый номер, называемый *индексом*. Нумерация элементов начинается с 0. Количество элементов в массиве называют *длиной массива*.

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

Пожалуйста, обратите внимание на то, что сами по себе массивы (или их аналоги) существуют практически во всех языках программирования. Однако, особенность заключается в том, что **в языке JavaScript массив является объектом**. Это означает, что у любого массива в JS есть определенный набор свойств и методов для работы с ним.

Синтаксис

// Способ №1

```
var firstNumbers = [4, 5, 3, 6, 23];
```

// Способ №2

```
var secondNumbers = new Array(4, 5, 3, 6, 23);
```

ОСНОВНЫЕ СВОЙСТВА И МЕТОДЫ

`length`

Длина массива.

`arr.push(...items)`

Добавляет элементы в конец.

`arr.pop()`

Извлекает элемент из конца.

`arr.unshift(...items)`

Добавляет элементы в начало.

`arr.shift()`

Извлекает элемент из начала.

`arr.splice(index[, deleteCount, elem1, ..., elemN])`

Изменяет содержимое массива, удаляя существующие элементы и/или добавляя новые.

`arr.slice(start, end)`

Возвращает новый массив, содержащий копию (части) исходного массива.

`arr.concat(arg1, arg2...)`

Возвращает новый массив, состоящий из массива, на котором он был вызван, соединённого с другими массивами и/или значениями, переданными в качестве аргументов.

`arr.forEach(function(item, index, array) {...});`

Выполняет указанную функцию один раз для каждого элемента в массиве.

`arr.indexOf(item, from)`

Ищет item, начиная с индекса from, и возвращает индекс, на котором был найден искомый элемент, в противном случае -1.

`arr.lastIndexOf(item, from)`

То же самое, но ищет справа налево.

`arr.includes(item, from)`

Ищет item, начиная с индекса from, и возвращает true, если поиск успешен.

`arr.map(function callback(item, index, array) {...})`

Создает новый массив с результатом вызова указанной функции для каждого элемента массива.

Задачи

Задание 1

Создать массив из 10 случайных чисел и написать несколько функций для работы с ним.

1. Функция принимает массив и выводит его на экран.
2. Функция принимает массив и выводит только четные элементы.
3. Функция принимает массив и возвращает сумму всех элементов массива.
4. Функция принимает массив и возвращает его максимальный элемент.
5. Функция добавления нового элемента в массив по указанному индексу.
6. Функция удаления элемента из массива по указанному индексу.

Задание 2

Создать массив из 5 случайных чисел и написать следующие функции.

1. Функция принимает 2 массива и возвращает новый массив, в котором собраны все элементы из двух массивов без повторений.
2. Функция принимает 2 массива и возвращает новый массив, в котором собраны общие элементы (то есть элементы, которые встречаются и в первом и во втором массивах) без повторений.
3. Функция принимает 2 массива и возвращает новый массив, в котором собраны все элементы из первого массива, которых нет во втором массиве.

Задание 3

Создать массив фруктов и отсортировать его по алфавиту. Написать следующие функции.

1. Вывод на экран с помощью `document.write()` в виде списка (с помощью тегов `` и ``).
2. Поиск фрукта в массиве. Функция принимает название фрукта и возвращает индекс найденного элемента или `-1`, если не найден. Поиск должен быть не регистрозависимым.

Запрещается использование стандартных методов массива!