# Лабораторная работа № 4 Работа с массивами в JavaScript

**Цель работы:** изучить возможности JavaScript для создания и обработки массивов.

**Краткие теоретические сведения:**

Для работы с наборами данных предназначены массивы. Для создания массива применяется литерал массива или конструкция new Array():

let array\_name1 = [item1, item2, ...];

let array\_name2 = new Array([item1, item2, ...]);

Для повышения производительности и читабельности программного кода рекомендуется использовать литерал массива.

Для получения доступа к элементам массива используется индекс. Индексация элементов начинается с нуля:

let cars = ["Saab", "Volvo", "BMW"];

console.log(cars[0]); // Saab

Индех используется как для чтения, так и для записи элемента массива. Причем в отличие от других языков, таких как C# или Java, можно установить элемент, который изначально не установлен:

let cars = ["Saab", "Volvo", "BMW"];

cars[10] = "Toyota"

console.log(cars[10]); // Toyota

console.log(cars[3]); // undefined

Также стоит отметить, что в отличие от ряда языков программирования в JavaScript массивы не являются строго типизированными, один массив может хранить данные разных типов:

var objects = ["Tom", 12, true, 3.14, false];

console.log(objects.toString());

Массивы могут быть одномерными и многомерными. Каждый элемент в многомерном массиве может представлять собой отдельный массив.

const students = [

["Иванов", 20, 5.5],

["Петров", 18, 8.2],

["Сидоров", 21, 7.8]

];

students[0][1] = 19; // присваиваем отдельное значение

console.log(students[0][1]); // 56

В языке JavaScript все свойства и методы обработки массивов собраны в глобальном объекте Array.prototype, от которого автоматически наследуются все создаваемые массивы.

Все массивы обладают свойством length, которой устанавливает или возвращает количество элементов в массиве:

let cars = ["Saab", "Volvo", "BMW"];

console.log(cars.length); // 3

cars.length = 5;

console.log(cars[4]); // undefined

**Методы массивов**

Для добавления и удаления элементов массива используются следующие методы:

* **push(...items)** – добавляет элементы в конец,
* **pop()** – извлекает элемент из конца,
* **shift()** – извлекает элемент из начала,
* **unshift(...items)** – добавляет элементы в начало.

Для удаления эедемента массива можно использовать оператор **delete**. Однако этот оператор удаляет только значение элеменета с заданным ключом без переиндексации:

let cars = ["Saab", "Volvo", "BMW"];

delete cars[1];

console.log(cars.length); // 3

console.log(cars[1]); // undefined

Универсальный метод **splice()** используется для добавления, удаления и замены элементов массива:

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

Он начинает с позиции index удалять deleteCount элементов и вставлять elem1, ..., elemN на их место. Возвращает массив из удалённых элементов.

Метод **slice()** возвращает новый массив, в который копирует элементы, начиная с индекса start и до end (не включая end). Оба индекса start и end могут быть отрицательными. В таком случае отсчёт будет осуществляться с конца массива:

slice([start], [end])

Метод **forEach()** позволяет запускать функцию для каждого элемента массива. Его синтаксис:

forEach(function(item, index, array) {

// ... делать что-то с item

});

Функция обратного вызова (callback) вызывается по очереди для каждого элемента массива и принимает следующие парметра:

* **item** – очередной элемент;
* **index** – его индекс;
* **array** – сам массив.

Для поиска элементов в массиве используются следующие методы:

* **indexOf(item, from)** ищет item, начиная с индекса from, и возвращает индекс, на котором был найден искомый элемент, в противном случае -1.
* **lastIndexOf(item, from)** – то же самое, но ищет справа налево.
* **includes(item, from)** – ищет item, начиная с индекса from, и возвращает true, если поиск успешен.

Методы **find**, **findIndex** и **filter** в качестве условия поиска используют функцию-предикат:

let result = arr.find(function(item, index, array) {

// если true - возвращается текущий элемент и перебор прерывается

// если все итерации оказались ложными, возвращается undefined

});

let result = arr.findIndex(function(item, index, array) {

// если true - возвращается индекс, на котором был найден элемент, и перебор прерывается

// если все итерации оказались ложными, возвращается -1

});

let results = arr.filter(function(item, index, array) {

// если true - элемент добавляется к результату, и перебор продолжается

// возвращается пустой массив в случае, если ничего не найдено

});

Метод **map()** является одним из наиболее полезных и часто используемых. Он вызывает функцию для каждого элемента массива и возвращает массив результатов выполнения этой функции:

let result = arr.map(function(item, index, array) {

// возвращается новое значение вместо элемента

});

Метод **sort(fn)** сортирует массив на месте, меняя в нём порядок элементов. Он возвращает отсортированный массив, но обычно возвращаемое значение игнорируется, так как изменяется сам массив.

Полный список методов есть в [справочнике MDN](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Array/).

**Задание к лабораторной работе:**

**Задание 1.**

В файле со скриптом lab4-1.html:

1. Создайте массив treug «треугольных» чисел, т.е. чисел вида n(n+1)/2 (где n=1,2,… 10) и выведите значения этого массива на экран в строку (через 2 пробела).

2. Создайте массив kvd квадратов натуральных чисел от 1 до 10, выведите значения этого массива на экран в строку.

3. Объедините эти 2 массива в массив rez, выведите результат на экран.

4. Отсортируйте массив rez в обратном порядке, выведите результат на экран.

**Задание 2.**

В файле со скриптом lab4-2.html:

1. Сформируйте одномерный массив (не менее 15 элементов, для генерации значений массива можно воспользоваться функцией random объекта Math), выведите значения этого массива на экран в строку.

2. Выполните задание в соответствии с вариантом, не используя встроенные функции.

| Вариант | Задание |
| --- | --- |
| 1 | Найти максимальный элемент и поменять его местами с последним элементом массива |
| 2 | Найти минимальный элемент и поменять его местами с предыдущим элементом массива |
| 3 | Найти минимальный элемент и поменять его местами с последующим элементом массива |
| 4 | Найти максимальный элемент и поменять его местами с шестым элементом массива |
| 5 | Найти максимальный элемент, присвоить его значение последнему элементу массива, а вместо максимального числа записать - 1 |
| 6 | Найти минимальный элемент, присвоить его значение первому элементу массива, а вместо минимального элемента записать число 9999 |
| 7 | Найти минимальный элемент и поменять его местами с третьим элементом массива |
| 8 | Найти минимальный элемент и заменить его на сумму первого и последнего элементов |
| 9 | Найти максимальный элемент и поменять его местами с предпоследним элементом массива |
| 10 | Найти минимальный элемент, присвоить его значение последнему элементу массива, а вместо минимального элемента записать значение 3N |
| 11 | Найти минимальный элемент и поменять его местами с элементом массива, номер которого задан |
| 12 | Найти максимальный элемент и поменять его местами со вторым элементом массива |
| 13 | Найти минимальный элемент и поменять его местами с последним элементом массива |
| 14 | Найти минимальный элемент и поменять его местами с предпоследним элементом массива |
| 15 | Найти минимальный элемент и поменять его местами с третьим элементом массива |

Задание 3.

В файле со скриптом lab4-3.html:

1. Сформируйте одномерный массив (не менее 15 элементов, для генерации значений массива можно воспользоваться функцией random объекта Math), выведите значения этого массива на экран в строку.

2. Выполните задание в соответствии с вариантом, используя встроенные методы массивов.

| **Вариант** | **Задание** |
| --- | --- |
| 1 | Найти количество чисел, принадлежащих промежутку [a,b], и сумму чисел, стоящих на местах, кратных 3. |
| 2 | Найти сумму чисел, меньших заданного D, и количество чисел, стоящих на четных местах и больших заданного С. |
| 3 | Найти произведение всех чисел, стоящих на местах, кратных 4, и количество чисел, небольших заданного A. |
| 4 | Найти количество чисел, меньших заданного X, и произведение всех отрицательных чисел, стоящих на нечетных местах. |
| 5 | Найти количество чисел, непринадлежащих промежутку (X,Y], и сумму отрицательных чисел, стоящих на четных местах. |
| 6 | Найти количество неотрицательных чисел и определить сумму чисел, стоящих на местах, кратных 3, и неравных заданному F. |
| 7 | Найти среднее арифметическое отрицательных чисел и определить количество чисел, по величине больших А и стоящих на четных местах. |
| 8 | Найти среднее арифметическое положительных чисел, стоящих на нечетных местах, и количество чисел, меньших заданного B. |
| 9 | Найти среднее арифметическое чисел, принадлежащих промежутку [A,B), и количество положительных чисел, стоящих на местах, кратных 4. |
| 10 | Найти среднее арифметической чисел, неравных заданному С, и произведение неположительных чисел, стоящих на четных местах. |
| 11 | Найти среднее арифметическое чисел, больших заданного D и стоящих на нечетных местах, и определить количество чисел, небольших заданного F. |
| 12 | Найти среднее арифметическое чисел, непопадающих в промежуток [A,B], и количество положительных чисел, стоящих на местах, кратных 3. |
| 13 | Найти среднее арифметическое ненулевых чисел и количество чисел, по величине небольших A и стоящих на четных местах. |
| 14 | Вычислить произведение чисел, принадлежащих промежутку (A,B], и количество отрицательных чисел, стоящих на местах, кратных 3. |
| 15 | Найти среднее арифметическое положительных чисел, стоящих на нечетных местах, и произведение чисел, меньших заданного С. |

**Задание 4.**

В файле со скриптом lab4-4.html:

1. Создайте массивы fruits=[‘apple’,’pineapple’,’mango’,’melon’,’grape’]; citrus=[‘orange’,’lemon’,’lime’].
2. Последовательно, в каждой отдельной строчке, выведите:

а) массив fruits;

б) массив citrus;

в) объединенный массив fruits;

г) массив fruits, в котором после mango вставлены: pear, cherry, plum, raspberry, strawberry;

д) массив fruits, в котором удалены последние 3 элемента;

е) выведите отсортированный в алфавитном порядке массив fruits;

ж) выведите массив fruits в обратном алфавитном порядке;

з) выведите массив fruits, отсортированный по длине слов.

**Требование по содержанию отчета:**

В отчете должны быть отображены следующие пункты:

1. Титульный лист.
2. Цель работы.
3. Задания.

5. Листинг кода сценария с комментариями.

6. Результат выполнения скрипта.

8. Выводы.

**Контрольные вопросы для зашиты:**

1. Что такое массив?
2. Какие массивы используются в JavaScript?
3. Как индексируются элементы массив?
4. Как создается многомерный массив?
5. Как просмотреть структуру и значения элементов массива?
6. Как вывести значения элементов массива в окно браузера?
7. Как отсортировать массив по возрастанию и убыванию?
8. Какие методы объекта Array вы знаете?
9. Как добавить элементы в массив?
10. Как объединить несколько массивов?
11. Как удалить элементы массива?