

Массивы данных. Одномерные и многомерные массивы

За урок мы научимся работать с массивами данных в языке JavaScript. При помощи массивов можно хранить большие объемы информации в одном единственном месте. Мы изучим одномерные и многомерные массивы данных.

Массивы позволяют хранить большой объем информации в одном месте. В языке **JavaScript** можно найти несколько основных типов массивов.

Одномерный массив

Одномерный массив состоит из нескольких элементов, объединенных под одним именем. Чтобы создать массив необходимо создать переменную и присвоить ей значение `new Array()`. Внутри вы можете через запятую записывать различные значения, которые и будут элементами массива.

В массивах отсчет начинается с 0, поэтому первый элемент по индексу будет равен 0, второй - 1 и так далее.

Примеры создания массива:

```
var some = new Array(); // Создание пустого массива
some[0] = '1'; // Добавление 1 элемента
some[1] = 2; // Добавление 2 элемента
console.log(some[0]); // Вывод первого элемента
var array = new Array(1, 5, 2); // Создание массива со значениями сразу же
```

Работать с элементами массива можно точно как с переменными. Мы можем их выводить или же устанавливать для них новые значения.

Для массивов существует несколько дополнительных функций, которые ещё будут изучены в ходе курса. Сейчас был изучен лишь метод `length`, который позволяет получить длину всего массива.

Пример метода:

```
var elements = new Array(23, 6, 0, true, "Первый");
```

```
// Выведет значение 5, так как в массиве 5 элементов  
console.log(elements.length);
```

Сперва массивы могут показаться бесполезными, но на самом деле это не так. В дальнейшем вы сможете понять всю важность массивов.

Многомерные массивы

Многомерный массив - это массив, в котором каждый элемент является другим массивом. На практике очень редко используются массивы с более чем третьим уровнем вложенности. То есть массивы, в которых все элементы являются другими массивами и в котором все элементы также другие массивы встречаются очень и очень редко на практике.

Мы не будем изучать подобные массивы, так как принцип их построения точно такой же как при работе с двумерными массивами.

Для создания двумерных массивов необходимо использовать двойные квадратные скобки после типа данных. Выглядит такой массив как матрица, а записывается следующим образом:

```
var x = new Array(new Array(0, 34, 2), new Array(3, 4, 5));  
console.log(x[0][1]); // Выведет 34
```

```
// Можно их сразу не присваивать  
var symbols = new Array(new Array(), new Array());  
symbols [0][1] = 'A';
```

Мы видим, что элементы первого массива являются другими массивами. Чтобы выбрать какой-либо объект используйте ту же структуру что и для одномерных массивов, вот только теперь указывайте индекс как первого массива, так и второго:

```
x[0][1] = 1; // Вместо 34 теперь будет 1
```

Создание массивов

```
var arr = [5, true, "stroka", 5.7, 0, -100];  
arr[3] = "word";
```

```
console.log(arr.length);
```

```
var matrix = [  
  [4, 6, 8], ["stroka", 5.7], [0, -100]  
];
```

```
matrix[1][0] = 90;  
console.log(matrix);
```