

# Основы Программирования

## Занятие №4 Массив



**IT Education  
Academy**

[WWW.ITEA.UA](http://WWW.ITEA.UA)

# План занятия

- Массив как структура данных
- Операции с массивами
- Задачи с массивами

# Повторение материала

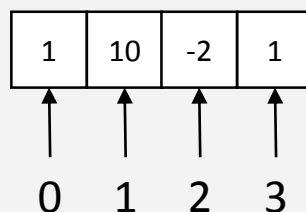
---

# Массив

---

Массив - это структура данных, которая позволяет хранить несколько значений в одной переменной.

Каждому элементу в массиве присвоен его порядковый номер (индекс) для того, чтобы можно было однозначно обратиться к этому элементу:



В большинстве языков программирования, индексация начинается с 0. Т.е. первый элемент (1) имеет индекс 0, второй элемент (10) имеет индекс 1... Также массив характеризуется его размером (длиной), на рисунке - размер равен 4.

В памяти элементы массива расположены последовательно и непосредственно друг за другом.

# Массив в JavaScript

---

Для создания массива в JavaScript используется синтаксис: [набор, значений, через, запятую]

Например:

```
var array; // создание переменной
```

```
array = [1, 2, 3]; // создание массива и сохранение его в переменной
```

```
var array2;
```

```
array2 = []; // создание пустого массива и сохранение его в переменной
```

Т.к. JavaScript динамически типизированный язык, в массиве могут содержаться значения разных типов:

```
var array3 = [1, 'a', true, undefined];
```

# Массив в JavaScript

---

Создать переменную с именем `digits`, и записать в нее массив из 10-ти элементов от 0 до 9.

# Обращение к элементам

---

После создания массива, для того чтобы обратиться к его элементу используется синтаксис: `имя_переменной[индекс]`;

Например:

```
var anotherArray = [10, 20, 30];  
console.log(anotherArray[0]); //выведет на экран 10  
console.log(anotherArray[2]); //выведет на экран 30
```

```
anotherArray[1] = 'Hello world!';  
console.log(anotherArray[1]); //выведет на экран Hello world!
```

# Обращение к элементам

---

Создать переменную с именем `letters`, и присвоить ей массив `['h', 'l', 'w', 'r', 'd', 'e', 'o', ' ']`.

Вывести на экран `"hello world"`, используя только операции конкатенации и доступа к элементу массива.



# Динамический массив

---

Массивы в JavaScript - динамические, это означает что они могут менять свой размер после того как были созданы.

```
var array = [1, 2, 3]
console.log(array[3]); //этого элемента не существует поэтому
выведет undefined
```

```
array[3] = 4;
array[4] = 5;
console.log(array[4]); //выведет 5
```

В этом коде мы изменили его размер с 3 до 5.

# Динамический массив

---

Создать переменную с пустым массивом и заполнить ее числами от 0 до 9.

# Свойство length

---

Для того, чтобы узнать текущий размер массива, можно обратиться к его свойству length:

```
var array = [1, 2, 3];  
console.log(array.length); //3
```

```
array[3] = 4; // array == [1, 2, 3, 4]  
console.log(array.length); //4
```

# Свойство length

---

Вывести все элементы массива на экран, используя цикл.  
Например, если есть массив [1, 2, 3], вывести на экран

1  
2  
3

# Массив

---

Задача 1. Задать массив чисел любой длины и найти максимальный элемент в массиве.

Задача 2. Создать массив размера 2 с двумя произвольными элементами. Поменять местами элемент 0 и 1.

Задача 3. Найти максимальный элемент в массиве, начиная поиск не с 0 элемента, а с указанного вами в коде. Т.е. если есть массив [1, 4, -1, -10, 3], и поиск начинать с элемента под индексом 2, то максимальный элемент будет 3.

Задача 4. Найти индекс максимального элемента в массиве.

# Сортировка выбором

Словесное описание.

Начиная с нулевого элемента, последовательно находить минимальный элемент массива и обменивать его с текущим. На каждом шаге переходить на следующий элемент.

3	2	4	1	5
1	2	4	3	5
1	2	4	3	5
1	2	4	3	5
1	2	3	4	5
1	2	3	4	5
1	2	3	4	5
1	2	3	4	5

# Действия над массивами

---

Поскольку массив является популярной структурой данных, существуют готовые наборы доступных операций над ними.

Это такие операции, как:

- добавление элемента в конец массива;
- удаление элемента с конца;
- удаление первого элемента;
- добавление элемента в начало;
- проверка наличия элемента в массиве;
- поиск позиции элемента в массиве;
- [другие](#).

# Добавление/удаление в конец

---

push - добавление в конец

```
var array = [1,2,3];  
console.log(array); // [1, 2, 3]  
array.push(4);  
console.log(array); // [1, 2, 3, 4]  
array.push(5, 6, 7);  
console.log(array); // [1, 2, 3, 4, 5, 6, 7]
```

pop - удаление с конца, результат операции - удаленный элемент

```
var array = ['hi', 'dear', 'friend'];  
var lastElement = array.pop();  
console.log(lastElement); // friend  
console.log(array); // ['hi', 'dear']  
var anotherLast = array.pop();  
console.log(anotherLast); // что будет выведено?  
console.log(array); // что будет выведено?
```



# Добавление/удаление в конец

---

Опустошить массив, используя: цикл, push и length. На каждой итерации выводить удаленный элемент.

# Добавление/удаление в начало

---

unshift - добавление в начало

```
var array = [1,2,3];
console.log(array); // [1, 2, 3]
array.unshift(4);
console.log(array); // [4, 1, 2, 3]
array.unshift(5, 6, 7);
console.log(array); // [5, 6, 7, 4, 1, 2, 3]
```

shift - удаление с начала, результат операции - удаленный элемент

```
var array = ['hi', 'dear', 'friend'];
var firstElement = array.shift();
console.log(firstElement); //hi
console.log(array); // ['dear', 'friend']
var anotherFirst = array.shift();
console.log(anotherFirst); // что будет выведено?
console.log(array); // что будет выведено?
```

# Добавление/удаление в конец

---

Опустошить массив, используя: цикл, unshift и length. На каждой итерации выводить удаленный элемент.

# Поиск

---

includes - проверка наличия элемента

```
var array = [1,2,3];
console.log(array); // [1, 2, 3]
var i1 = array.includes(1);
console.log(i1); // true
var i4 = array.includes(4);
console.log(i4); // false
```

indexOf - поиск позиции значения

```
var array = ['hi', 'dear', 'friend'];
var index = array.indexOf('dear');
console.log(index); //1
var index2 = array.indexOf('JavaScript');
console.log(index2); // -1
```

# Поиск

---

Задать массив лексики JavaScript (до 10 слов). Запрашивать ввод пользователя, до тех пор, пока не введет 'exit'. На каждый ввод говорить пользователю является ли его слово зарезервированным, или нет.

\* Если пользователь введет слово 'all' - вывести весь список зарезервированных слов.

# Массив в массиве

---

Массив может содержать другой массив(или массивы).

```
var matrix = [ [0, 1, 2], [3, 4, 5], [6, 7, 8] ];
```

```
console.log(matrix[1][1]); //4
```

```
console.log(matrix[2]); // [6, 7, 8]
```

# Summary

---

1. С помощью массива храним набор данных в одной переменной.
2. Размер массива в JavaScript меняется динамически.
3. Индекс - позиция элемента в массиве.
4. “.length” - размер массива.
5. Дополнительные операции: push, pop, shift, unshift, includes и тд.
6. При работе с массивами удобно использовать циклы.
7. Массив в массиве.

# Домашнее задание

---

1. Просить пользователя ввести слово, пока не введет 'стоп'. После этого просить пользователя ввести число от 0 до (N-1) (где N - это кол-во введенных слов) и на каждый ввод выводить сохраненное слово, которому соответствует число.
2. Создать массив чисел. Задать два числа a и b ( $a < b$ ). Найти в нем все элементы, которые больше a и меньше b (находятся в диапазоне от a до b). Вывести их на экран.
3. Задать массив чисел. Найти сумму всех положительных чисел и сумму всех отрицательных чисел. Вывести на экран.
4. \*\*Используя решения задач 1-4, реализовать алгоритм сортировки выбором.



**Спасибо за внимание!**