

Государственное бюджетное профессиональное образовательное  
учреждение Самарской области  
«Самарский техникум авиационного и промышленного машиностроения  
имени Д.И.Козлова»

# Методы массивов

Группа ИС-4

Выполнили работу студенты:

Галеева Эльвира, Долгашев Игорь, Вандюков Дмитрий.

# Массив

Массивы – это упорядоченная коллекция данных в массивах могут храниться любые типы данных.

```
1 <script>
2   var arr = [1, 3, "Elena", true];
3
4   console.log(arr);
5   console.log(arr.length);
6   //Свойство указывающее длину массива.
7   console.log(typeof(arr));
8
9   console.log(arr[0], typeof(arr[0]));
10  console.log(arr[2], typeof(arr[2]));
11  console.log(arr[3], typeof(arr[3]));
12
13  arr[0] = "Ivan";
14  console.log(arr[0], typeof(arr[0]));
15
16  arr.push(777);
17  //Добавление элемента в конец массива
18  console.log(arr, arr.length);
19
20  var x = arr.pop();
21  //Удаление последнего элемента из массива
22  console.log(arr, arr.length);
23  console.log(x);
24
25  arr.unshift("Julia");
26  //Добавление элемента в начало массива
27  console.log(arr, arr.length);
28
29  arr.shift();
30  //Удаление первого элемента массива
31  console.log(arr, arr.length);
32
33 </script>
```



[1, 3, "Elena", true]	example.html:4
4	example.html:5
object	example.html:7
1 "number"	example.html:9
Elena string	example.html:10
true "boolean"	example.html:11
Ivan string	example.html:14
["Ivan", 3, "Elena", true, 777] 5	example.html:18
["Ivan", 3, "Elena", true] 4	example.html:22
777	example.html:23
["Julia", "Ivan", 3, "Elena", true] 5	example.html:27
["Ivan", 3, "Elena", true] 4	example.html:31

# Метод массива splice.

Его синтаксис:

```
array.splice(start, deleteCount[, item1[, item2[, ...]]])
```

Метод `splice()` изменяет содержимое массива, удаляя существующие элементы и/или добавляя новые.

Пример:

```
1 | var myFish = ['angel', 'clown', 'drum', 'mandarin', 'sturgeon'];  
2 | var removed = myFish.splice(3, 1);  
3 |  
4 | // removed равен ["mandarin"]  
5 | // myFish равен ["angel", "clown", "drum", "sturgeon"]
```

# Метод массива concat.

Метод `arr.concat` создаёт новый массив, в который копирует данные из других массивов и дополнительные значения.

Его синтаксис:

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

Пример:

```
1 let arr = [1, 2];
2
3 // создать массив из: arr и [3,4]
4 alert( arr.concat([3, 4])); // 1,2,3,4
5
6 // создать массив из: arr и [3,4] и [5,6]
7 alert( arr.concat([3, 4], [5, 6])); // 1,2,3,4,5,6
8
9 // создать массив из: arr и [3,4], потом добавить значения 5 и 6
10 alert( arr.concat([3, 4], 5, 6)); // 1,2,3,4,5,6
```

# Метод массива forEach

Метод `arr.forEach` позволяет запускать функцию для каждого элемента массива.

```
1 arr.forEach(function(item, index, array) {  
2     // ... делать что-то с item  
3 });
```

Например, этот код выведет на экран каждый элемент массива:

```
1 // Вызов alert для каждого элемента  
2 ["Bilbo", "Gandalf", "Nazgul"].forEach(alert);
```

# Метод массива map

Метод `arr.map` является одним из наиболее полезных и часто используемых.

Он вызывает функцию для каждого элемента массива и возвращает массив результатов выполнения этой функции.

Синтаксис:

```
1 let result = arr.map(function(item, index, array) {  
2   // возвращается новое значение вместо элемента  
3 });
```

Пример, здесь мы преобразуем каждый элемент в его длину:

```
1 let lengths = ["Bilbo", "Gandalf", "Nazgul"].map(item => item.length);  
2 alert(lengths); // 5,7,6
```

# Метод массива `find` и `findIndex`

Метод `arr.findIndex` – по сути, то же самое, но возвращает индекс, на котором был найден элемент, а не сам элемент, и `-1`, если ничего не найдено

```
1 let result = arr.find(function(item, index, array) {
2   // если true - возвращается текущий элемент и перебор прерывается
3   // если все итерации оказались ложными возвращается undefined
4 });
```

Пример:

```
1 let users = [
2   {id: 1, name: "Вася"},
3   {id: 2, name: "Петя"},
4   {id: 3, name: "Маша"}
5 ];
6
7 let user = users.find(item => item.id == 1);
8
9 alert(user.name); // Вася
```

# Метод массива filter

Метод `find` ищет один (первый попавшийся) элемент, на котором функция-колбэк (или функция обратного вызова) вернёт `true`.

На тот случай, если найденных элементов может быть много, предусмотрен метод `arr.filter(fn)`.

Синтаксис этого метода схож с `find`, но `filter` возвращает массив из всех подходящих элементов:

```
1 let results = arr.filter(function(item, index, array) {
2   // если true - элемент добавляется к результату, и перебор продолжается
3   // возвращается пустой массив в случае, если ничего не найдено
4 });
```

Пример:

```
1 let users = [
2   {id: 1, name: "Вася"},
3   {id: 2, name: "Петя"},
4   {id: 3, name: "Маша"}
5 ];
6
7 // возвращает массив, состоящий из двух первых пользователей
8 let someUsers = users.filter(item => item.id < 3);
9
10 alert(someUsers.length); // 2
```