

Урок 4.3 – 4.4. Методы объектов, массивы

Массивы

Объявление массива

```
const arr = []; // пустой массив
const arr2 = ['яблоко', 'банан', 'персик'];
// массив строк// Массив объектов
const objectsList = [
  {
    name: 'Andrew',
    lastName: 'Gulin'
  },
  {
    name: 'Boris',
    lastName: 'Ivanov'
  },
  {
    name: 'Alexey',
    lastName: 'Petrov'
  }
]
```

Доступ по индексу

```
// Получить третий элемент (индексы с 0)
const element = arr2[2];

// Последний элемент (индекс - длина массива - 1)
const lastElement = arr2[arr2.length - 1];
```

forEach - цикл по массиву

```
// Функция, объявленная внутри forEach
// вызывается для каждого элемента списка.
// forEach нельзя прервать.// Параметры:
```

```
// item - очередной элемент списка
// index - индекс элемента списка
// array - это ссылка на сам массив, по которому бежим
arr2.forEach((item, index, array) => {
  console.log(item);
});
```

push - добавить в конец цикла

```
arr2.push('Ананас');
```

pop - удалить с конца списка

```
const lastElem = arr2.pop();
```

unshift - добавить в начало списка

```
arr2.unshift('Папайя');
```

shift - удалить элемент с начала списка

```
const firstElement = arr2.shift();
```

indexOf - найти индекс элемента массива

```
const index = arr2.indexOf('Ананас');
```

findIndex - найти индекс (параметр - callback-функция)

```
const index = arr2.findIndex((item) => {
  // Если длина элемента равна 4.
  // Ищем первый такой и возвращаем его index
  return item.length === 4;
});
```

find - найти элемент в массиве

```
const findElement = arr2.find((item, index, array) => {
  // Начинается с буквы Б
  return item[0] === 'Б';
});
```

splice - удалить часть из массива (меняет массив)

```
// Параметры
// 2 - индекс, с которого начинать
// 3 - количество удаляемых элементов
arr2.splice(2, 3);
```

slice - вернуть часть массива (не меняет массив, возвращает копию)

```
const arr2copy = arr2.slice();
// полная копия
// Частичная копия. Параметры:
// 0 - индекс начального элемента (включительно)
// 3 - индекс конечного элемента (не включительно)
const arr2part = arr2.slice(0, 3);
```

filter - фильтрация массива

Возвращает новый массив, удовлетворяющий условиям.

```
const filteredArray = arr2.filter((item, index, array) => {
  // Функция выполняется для каждого элемента
  // В filteredArray попадут те элементы, где функция вернула true.
});
```

```
    return item[2] === 'a'; // третья буква a
  });

// Короткая запись
const filteredArrayShort = arr2.filter(item => item[2] === 'a');
```

map - возвращает преформированный массив

```
const objectsList = [
  {
    name: 'Andrew',
    lastName: 'Gulin'
  },
  {
    name: 'Boris',
    lastName: 'Ivanov'
  },
  {
    name: 'Alexey',
    lastName: 'Petrov'
  }
];

const objectsRemapped = objectsList.map((item, index, array) => {
  // Функция выполняется для каждого элемента.
  // Возвращаемое значение будет использоваться вместо текущего значения элемента
  return {
    ...item, // копируем изначальный объект с name и lastName
    id: index // добавляем новое свойство
  }
});
```

reduce - обработка каждого элемента с сохранением результата

```
const arrNumbers = [
  {
    x: 5,
    y: 10
  },
  {
    x: 15,
    y: 20
  },
  {
```

```

        x: 39,
        y: 27
    }
];

// Считаем (x+y)*(x+y)*(x+y)
const result = arr2.reduce((accumulator, item, index, array) => {
    accumulator *= (item.x + item.y);
    // Необходимо вернуть новое значение аккумулятора
    return accumulator;
}, 1); // Начальное значение аккумулятора после функции

```

Методы объектов

Object.assign

Копирует один или несколько объектов в заданный объект. Возвращает изменённый объект.

```

// Полное копирование объекта.
const newObj = Object.assign({}, oldObj);
const newObj2 = Object.assign(targetObject, sourceObject1, sourceObject2);

```

Object.entries

Возвращает массив с парами [key, value]

```

const entries = Object.entries(obj);
// Пробежаться по свойствам объекта
Object.entries(obj).forEach(([key, value]) => {
    console.log('Ключ: ', key, 'Значение: ', value);
});

```

Object.values

Возвращает массив значений объекта (без ключей);

```

const values = Object.values(obj);
// Пробежаться по свойствам объекта
Object.values(obj).forEach((value) => {
    console.log('Значение: ', value);
});

```

Set

Set - множество значений. По сути, очень сильно перекликается с массивом.

Главное отличие - не может содержать повторяющихся значений.

```
const set = new Set();
set.add('яблоко');
set.add('яблоко');
set.add('банан');
// в результате в set будет только два значения - яблоко и банан

set.clear(); // очистить множество
set.delete('банан'); // удалить элемент по значению
set.size; // количество элементов в множестве

// Пробежаться по значениям множества
for (let entry of set) {
}
```

Map

Раньше для создания объекта с парами ключ-значение использовали объект, но у Map есть несколько преимуществ (именно для пар ключ-значение).[Сравнение объектов и мап на MDN](#)

```
const map = new Map();
map.set('Ключ', 'Значение');
map.set('Ключ2', 'Значение');
map.size; // Количество пар ключ-значение

map.get('Ключ'); // Возвращает значение с ключом "Ключ"

map.has('Ключ'); // true, если map содержит значение с ключом "Ключ"
map.delete('Ключ'); // Удаляет пару ключ-значение с ключом "Ключ"

// Пробежаться по элементам map
for (let [key, value] of map) {
}

// Другой вариант
for (let entry of map) {
  const key = entry[0];
}
```

```
    const value = entry[1];  
  }
```