**В конце файла будут ответы на задачи !!!!! Сначала будет описание всех задач, а в конце будет решение**push():

Добавьте один или несколько элементов в конец массива

Исходный массив:

let fruits = ['apple', 'banana', 'orange'];

Ожидаемый результат:

console.log(fruits); // Output: ['apple', 'banana', 'orange', твои добавленные элементы в конец]

pop():

Удалите последний элемент в массиве и верните этот элемент

Исходный массив:

const users = [

{

id: 1,

name: 'Bob',

isStudent: true,

},

{

id: 2,

name: 'Alex',

isStudent: true,

},

{

id: 3,

name: 'Ann',

isStudent: true,

},

{

id: 4,

name: 'Donald',

isStudent: false,

},

];

Ожидаемый результат:

console.log(lastUser); // { id: 4, name: 'Donald', isStudent: false }

console.log(users); // [ { id: 1, name: 'Bob', isStudent: true }, { id: 2, name: 'Alex', isStudent: true },{ id: 3, name: 'Ann', isStudent: true }]

shift():

Удалите первого юзера и верните его

Исходный массив:

const users = [

{

id: 1,

name: 'Bob',

isStudent: true,

},

{

id: 2,

name: 'Alex',

isStudent: true,

},

{

id: 3,

name: 'Ann',

isStudent: true,

},

{

id: 4,

name: 'Donald',

isStudent: false,

},

];

Ожидаемый результат:

console.log(firstUser); // { id: 1, name: 'Bob', isStudent: true }

console.log(users); // [{ id: 2, name: 'Alex', isStudent: true }, { id: 3, name: 'Ann', isStudent: true }, { id: 4, name: 'Donald', isStudent: false }]

unshift():

Добавьте один и более элементов в начало массива

Исходный массив:

const users = [

{

id: 1,

name: 'Bob',

isStudent: true,

},

{

id: 2,

name: 'Alex',

isStudent: true,

},

{

id: 3,

name: 'Ann',

isStudent: true,

},

{

id: 4,

name: 'Donald',

isStudent: false,

},

];

Ожидаемый результат:

console.log(users); //[

// { id: 0, name: 'TestName', isStudent: false },

// { id: 1, name: 'Bob', isStudent: true },

// { id: 2, name: 'Alex', isStudent: true },

// { id: 3, name: 'Ann', isStudent: true },

// { id: 4, name: 'Donald', isStudent: false }

// ]

reverse(), join(), split():

Сделать reverse слова

Исходная строка:

const str = 'JavaScript is awesome';

Ожидаемый результат:

console.log(res); // awesome is JavaScript

concat():

Напишите функцию mergeArrays, которая принимает на вход два массива и возвращает новый массив, содержащий все элементы из обоих массивов.

Ожидаемый результат:

console.log(mergeArrays([1, 2, 3], [4, 5, 6])); // Output: [1, 2, 3, 4, 5, 6]

console.log(mergeArrays(['apple', 'banana'], ['orange', 'kiwi'])); // Output: ["apple", "banana", "orange", "kiwi"]

flat():

Напишите функцию JavaScript, которая сглаживает вложенный массив любой глубины

Ожидаемый результат:

console.log(flattenArray([1, [2, 3], [[4], [5, 6]]])); // Output: [1, 2, 3, 4, 5, 6]

console.log(flattenArray([1, [2, [3, [4, [5]]]]])); // Output: [1, 2, 3, 4, 5]

find():

Напишите функцию JavaScript, которая находит первый элемент в массиве, удовлетворяющий заданному условию.

Пример:

Входной массив: [10, 20, 30, 40, 50]

Условие: Найти первый элемент больше 25

Выходные данные: 30

Входной массив: ["яблоко", "банан", "апельсин", "киви"].

Условие: Найти первый элемент, начинающийся на "o".

Выходные данные: "апельсин"

Необходимо написать функцию findFirstElement, которая принимает на вход массив и функцию условия и возвращает первый элемент в массиве, удовлетворяющий условию.

Ожидаемый результат:

let firstNumberGreaterThan25 = findFirstElement(numbers, (num) => num > 25);

console.log(firstNumberGreaterThan25); // Output: 30

let firstFruitStartingWithO = findFirstElement(fruits, (fruit) => fruit.startsWith('o'));

console.log(firstFruitStartingWithO); // Output: "orange"

includes():

Напишите функцию JavaScript, которая проверяет, присутствует ли заданный элемент в массиве.

Пример:

Входной массив: [1, 2, 3, 4, 5]

Элемент для проверки: 3

Выход: true

Входной массив: ["яблоко", "банан", "апельсин", "киви"].

Элемент для проверки: "виноград"

Выход: false

Необходимо написать функцию checkElement, которая принимает на вход массив и элемент и возвращает true, если элемент присутствует в массиве, и false в противном случае.

Ожидаемый результат:

let numbers = [1, 2, 3, 4, 5];

let fruits = ['apple', 'banana', 'orange', 'kiwi'];

console.log(checkElement(numbers, 3)); // Output: true

console.log(checkElement(fruits, 'grape')); // Output: false

filter():

Напишите функцию JavaScript, которая фильтрует массив чисел и возвращает только те элементы, которые больше заданного значения.

Пример:

Входной массив: [10, 20, 30, 40, 50]

Значение для фильтрации: 25

Выходной массив: [30, 40, 50]

Входной массив: [5, 15, 25, 35, 45]

Значение для фильтрации: 20

Выходные данные: [25, 35, 45]

Необходимо написать функцию filterGreaterThan, которая принимает на вход массив чисел и значение и возвращает новый массив, содержащий только элементы, превышающие указанное значение.

Ожидаемый результат:

let numbers1 = [10, 20, 30, 40, 50];

let numbers2 = [5, 15, 25, 35, 45];

console.log(filterGreaterThan(numbers1, 25)); // Output: [30, 40, 50]

console.log(filterGreaterThan(numbers2, 20)); // Output: [25, 35, 45]

sort():

Напишите функцию JavaScript, которая сортирует массив объектов в алфавитном порядке на основе заданного свойства.

Ожидаемый результат:

let people = [

{ name: 'John', age: 30 },

{ name: 'Alice', age: 25 },

{ name: 'Bob', age: 35 },

];

console.log(sortAlphabeticallyByProperty(people, 'name')); // [ { name: 'Alice', age: 25 },{ name: 'Bob', age: 35 },{ name: 'John', age: 30 }]

Напишите функцию JavaScript, которая сортирует массив чисел в порядке убывания

Ожидаемый результат:

let numbers = [5, 2, 8, 1, 4];

console.log(sortDescending(numbers)); // Output: [8, 5, 4, 2, 1]

map():

1. Отформатируйте data в массив объектов с такими вот св-ми

[

{ value: 1, label: 'Option 1' },

{ value: 2, label: 'Option 2' },

{ value: 3, label: 'Option 3' }

]

Исходный массив:

let data = [

{ value: 1, name: 'Option 1' },

{ value: 2, name: 'Option 2' },

{ value: 3, name: 'Option 3' },

];

Ожидаемый результат:

console.log(formattedData); // [

// { value: 1, label: 'Option 1' },

// { value: 2, label: 'Option 2' },

// { value: 3, label: 'Option 3' }

// ]

1. У вас есть массив объектов, и вы хотите добавить к каждому объекту новое свойство на основе существующих данных. Этим свойством будет discountPrice, где значение будет вычисляться таким образом price \* 0.9

Исходный массив:

let products = [

{ id: 1, name: 'Apple', price: 1.99 },

{ id: 2, name: 'Banana', price: 0.99 },

{ id: 3, name: 'Orange', price: 2.49 },

];

Ожидаемый результат:

console.log(productsWithDiscount); // [

// { id: 1, name: 'Apple', price: 1.99, с: 1.791 },

// { id: 2, name: 'Banana', price: 0.99, discountedPrice: 0.891 },

// { id: 3, name: 'Orange', price: 2.49, discountedPrice: 2.241 }

// ]

Дополнительные задачки:

1. Напишите функцию, которая определяет, является ли переданная строка палиндромом. Палиндромом считается строка, которая читается одинаково как слева направо, так и справа налево(split, reverse, join)

// Ожидаемый результат:

console.log(isPalindrome('level')); // Вернет true

console.log(isPalindrome('racecar')); // Вернет true

console.log(isPalindrome('hello')); // Вернет false

1. Выведите все дублирующиеся элементы массива(filter, indexOf)

например исходный массив

let arr = [1, 1, 5, 6, 7, 8, 9, 9, 3, 4, 0, 19];

ожидаемый результат

console.log(removeDublicates(arr));//[ 1, 9 ]

1. Напишите функцию JavaScript для сортировки массива объектов по определенному свойству в порядке возрастания, используя метод Array.prototype.sort()(тут кастомная сортировка придется чуть-чуть поискать и подумать)

Пример

const persons = [

{ name: 'Alice', age: 30 },

{ name: 'Bob', age: 25 },

{ name: 'Charlie', age: 35 },

];

console.log(sortByProperty(persons, 'age'));

// [{ name: 'Bob', age: 25 }, { name: 'Alice', age: 30 }, { name: 'Charlie', age: 35 }]

**ОТВЕТЫ НА ЗАДАЧИ**

//1 PUSH

let fruits = ['apple', 'banana', 'orange'];

fruits.push('grape', 'kiwi');

console.log(fruits); // Output: ['apple', 'banana', 'orange', 'grape', 'kiwi']

//2 POP

const users = [

{

id: 1,

name: 'Bob',

isStudent: true,

},

{

id: 2,

name: 'Alex',

isStudent: true,

},

{

id: 3,

name: 'Ann',

isStudent: true,

},

{

id: 4,

name: 'Donald',

isStudent: false,

},

];

let lastUser = users.pop();

console.log(lastUser); // { id: 4, name: 'Donald', isStudent: false }

console.log(users); // [ { id: 1, name: 'Bob', isStudent: true }, { id: 2, name: 'Alex', isStudent: true },{ id: 3, name: 'Ann', isStudent: true }]

//3 SHIFT

const users = [

{

id: 1,

name: 'Bob',

isStudent: true,

},

{

id: 2,

name: 'Alex',

isStudent: true,

},

{

id: 3,

name: 'Ann',

isStudent: true,

},

{

id: 4,

name: 'Donald',

isStudent: false,

},

];

const firstUser = users.shift();

console.log(firstUser); // { id: 1, name: 'Bob', isStudent: true }

console.log(users); // [{ id: 2, name: 'Alex', isStudent: true }, { id: 3, name: 'Ann', isStudent: true }, { id: 4, name: 'Donald', isStudent: false }]

//4 UNSHIFT

const users = [

{

id: 1,

name: 'Bob',

isStudent: true,

},

{

id: 2,

name: 'Alex',

isStudent: true,

},

{

id: 3,

name: 'Ann',

isStudent: true,

},

{

id: 4,

name: 'Donald',

isStudent: false,

},

];

users.unshift({ id: 0, name: 'TestName', isStudent: false });

console.log(users); //[

// { id: 0, name: 'TestName', isStudent: false },

// { id: 1, name: 'Bob', isStudent: true },

// { id: 2, name: 'Alex', isStudent: true },

// { id: 3, name: 'Ann', isStudent: true },

// { id: 4, name: 'Donald', isStudent: false }

// ]

//5 reverse

const str = 'JavaScript is awesome';

const res = str.split(' ').reverse().join(' ');

console.log(res); // awesome is JavaScript

//6 concat

function mergeArrays(arr1, arr2) {

let mergedArray = arr1.concat(arr2);

return mergedArray;

}

console.log(mergeArrays([1, 2, 3], [4, 5, 6])); // Output: [1, 2, 3, 4, 5, 6]

console.log(mergeArrays(['apple', 'banana'], ['orange', 'kiwi'])); // Output: ["apple", "banana", "orange", "kiwi"]

//7 flat

function flattenArray(arr) {

let flattenedArray = arr.flat(Infinity);

return flattenedArray;

}

console.log(flattenArray([1, [2, 3], [[4], [5, 6]]])); // Output: [1, 2, 3, 4, 5, 6]

console.log(flattenArray([1, [2, [3, [4, [5]]]]])); // Output: [1, 2, 3, 4, 5]

//8 forEach()

function calculateSquare(arr) {

arr.forEach(function (num) {

let square = num \* num;

console.log(`${num} squared is ${square}`);

});

}

calculateSquare([1, 2, 3, 4, 5]);

// 1 squared is 1

// 2 squared is 4

// 3 squared is 9

// 4 squared is 16

// 5 squared is 25

//9 find()

function findFirstElement(arr, condition) {

return arr.find(condition);

}

let numbers = [10, 20, 30, 40, 50];

let fruits = ['apple', 'banana', 'orange', 'kiwi'];

let firstNumberGreaterThan25 = findFirstElement(numbers, (num) => num > 25);

console.log(firstNumberGreaterThan25); // Output: 30

let firstFruitStartingWithO = findFirstElement(fruits, (fruit) => fruit.startsWith('o'));

console.log(firstFruitStartingWithO); // Output: "orange"

//10 includes()

function checkElement(arr, element) {

return arr.includes(element);

}

let numbers = [1, 2, 3, 4, 5];

let fruits = ['apple', 'banana', 'orange', 'kiwi'];

console.log(checkElement(numbers, 3)); // Output: true

console.log(checkElement(fruits, 'grape')); // Output: false

//11 filter

function filterGreaterThan(arr, value) {

return arr.filter((num) => num > value);

}

let numbers1 = [10, 20, 30, 40, 50];

let numbers2 = [5, 15, 25, 35, 45];

console.log(filterGreaterThan(numbers1, 25)); // Output: [30, 40, 50]

console.log(filterGreaterThan(numbers2, 20)); // Output: [25, 35, 45]

//12 sort

function sortAlphabeticallyByProperty(arr, property) {

return arr.sort((a, b) => a[property].localeCompare(b[property]));

}

let people = [

{ name: 'John', age: 30 },

{ name: 'Alice', age: 25 },

{ name: 'Bob', age: 35 },

];

console.log(sortAlphabeticallyByProperty(people, 'name')); // [ { name: 'Alice', age: 25 },{ name: 'Bob', age: 35 },{ name: 'John', age: 30 }]

//SORT SECOND TASK

function sortDescending(arr) {

return arr.sort((a, b) => b - a);

}

let numbers = [5, 2, 8, 1, 4];

console.log(sortDescending(numbers)); // Output: [8, 5, 4, 2, 1]

//map

let data = [

{ value: 1, name: 'Option 1' },

{ value: 2, name: 'Option 2' },

{ value: 3, name: 'Option 3' },

];

let formattedData = data.map((item) => {

return {

value: item.value,

label: item.name,

};

});

console.log(formattedData); // [

// { value: 1, label: 'Option 1' },

// { value: 2, label: 'Option 2' },

// { value: 3, label: 'Option 3' }

// ]

//// MAP SECOND TASK

let products = [

{ id: 1, name: 'Apple', price: 1.99 },

{ id: 2, name: 'Banana', price: 0.99 },

{ id: 3, name: 'Orange', price: 2.49 },

];

let productsWithDiscount = products.map((product) => {

return {

...product,

discountedPrice: product.price \* 0.9,

};

});

console.log(productsWithDiscount); // [

// { id: 1, name: 'Apple', price: 1.99, с: 1.791 },

// { id: 2, name: 'Banana', price: 0.99, discountedPrice: 0.891 },

// { id: 3, name: 'Orange', price: 2.49, discountedPrice: 2.241 }

// ]

Дополнительные задачки ответы:

1)

function isPalindrome(str) {

const reversedStr = str.split('').reverse().join('');

return str === reversedStr;

}

// Ожидаемый результат:

console.log(isPalindrome('level')); // Вернет true

console.log(isPalindrome('racecar')); // Вернет true

console.log(isPalindrome('hello')); // Вернет false

2)

let arr = [1, 1, 5, 6, 7, 8, 9, 9, 3, 4, 0, 19];

function removeDublicates(arr) {

let result = arr.filter((item, index) => {

return arr.indexOf(item) !== index;

});

return result;

}

console.log(removeDublicates(arr));//[ 1, 9 ]

3)

function sortByProperty(arr, property) {

return arr.sort((a, b) => a[property] - b[property]);

}

const persons = [

{ name: 'Alice', age: 30 },

{ name: 'Bob', age: 25 },

{ name: 'Charlie', age: 35 },

];

console.log(sortByProperty(persons, 'age'));

// [{ name: 'Bob', age: 25 }, { name: 'Alice', age: 30 }, { name: 'Charlie', age: 35 }]