Здесь будет закрепление материала за 1 спринт

1. Глубоко скопируйте объект superUser и измените в копии lat на ‘test coordinate’, при этом исходный объект должен оставаться неизменным. Реализуйте глубокое копирование 2-мя способами: через spread оператор и через structuredClone()

Исходный объект superUser:

const superUser = {

id: 10,

name: 'Clementina DuBuque',

username: 'Moriah.Stanton',

email: 'Rey.Padberg@karina.biz',

address: {

street: 'Kattie Turnpike',

suite: 'Suite 198',

city: 'Lebsackbury',

zipcode: '31428-2261',

geo: {

lat: '-38.2386',

lng: '57.2232',

},

},

phone: '024-648-3804',

website: 'ambrose.net',

company: {

name: 'Hoeger LLC',

catchPhrase: 'Centralized empowering task-force',

bs: 'target end-to-end models',

},

};

1. Напишите функцию sumAndDifference, которая принимает два числа и возвращает объект с их суммой и разностью. Используйте деструктуризацию.

Expected result:

console.log('Сумма:', sum); // Вывод: 15

console.log('Разность:', difference); // Вывод: 5

1. Напишите функцию getPositiveNumbers, которая принимает массив чисел и возвращает новый массив, содержащий только положительные числа(use filter)

Expected result:

const numbers = [-1, 2, -3, 4, -5, 6];

const positiveNumbers = getPositiveNumbers(numbers);

console.log(positiveNumbers); // Вывод: [2, 4, 6]

1. Напишите функцию mergeArrays, которая принимает несколько массивов и объединяет их в один массив (reduce, concat)

Expected result:

const arr1 = [1, 2, 3];

const arr2 = [4, 5, 6];

const mergedArray = mergeArrays(arr1, arr2);

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

1. Напишите функцию findString, которая принимает массив строк и строку для поиска, и возвращает true, если строка найдена в массиве, и false в противном случае(includes)

Expected result:

const strings = ['apple', 'banana', 'orange'];

console.log(findString(strings, 'banana')); // Вывод: true

console.log(findString(strings, 'grape')); // Вывод: false

1. Напишите функцию sortByPropertyValue, которая принимает массив объектов и строку, представляющую название свойства объекта. Функция должна вернуть массив объектов, отсортированных по возрастанию значений указанного свойства (sort)

Expected result:

const people = [

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

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

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

];

sortByProperty(people, 'age');

console.log(people); // Вывод: [{ name: "Bob", age: 25 }, { name: "Alice", age: 30 }, { name: "Charlie", age: 35 }]

1. Дан массив целых чисел. Напишите функцию extractSubarray, которая принимает этот массив и два индекса start и end. Функция должна вернуть новый массив, содержащий элементы исходного массива с индексами от start до end (не включая элемент с индексом end).После этого, исходный массив должен быть изменен, так что он больше не содержит извлеченные элементы(slice, splice)

Expected result:

let array = [1, 2, 3, 4, 5, 6, 7, 8, 9];

let extracted = extractSubarray(array, 2, 6);

console.log(extracted); // Вывод: [3, 4, 5, 6]

console.log(array); // Вывод: [1, 2, 7, 8, 9]

1. Напишите функцию addPropertyToObject, которая добавляет новое свойство к объекту с указанным именем и значением(use dynamic property for that)

Expected result:

let obj = { name: 'Alice' };

addPropertyToObject(obj, 'age', 30);

console.log(obj); // Вывод: { name: "Alice", age: 30 }

1. У вас есть объект, представляющий информацию о студентах и их оценках. Напишите функцию, которая принимает этот объект и возвращает массив строк, каждая из которых представляет собой имя студента, за которым следует строка с его оценками в формате "Имя: Оценки"(Object.entries() + map)

Expected result:

const students = {

Alice: [4, 5, 3],

Bob: [3, 2, 4],

Charlie: [5, 5, 5]

};

console.log(formatStudentGrades(students));

// Вывод:

// [

// "Alice: 4, 5, 3",

// "Bob: 3, 2, 4",

// "Charlie: 5, 5, 5"

// ]

1. У вас есть объект, представляющий информацию о студентах и их оценках. Напишите функцию, которая принимает этот объект и возвращает массив имен студентов (Object,keys())

Expected result:

const students = {

Alice: [4, 5, 3],

Bob: [3, 2, 4],

Charlie: [5, 5, 5]

};

console.log(getStudentNames(students));

// Вывод: ["Alice", "Bob", "Charlie"]

1. У вас есть объект, представляющий информацию о книгах в библиотеке. Каждая книга имеет свойство genre, представляющее жанр книги. Напишите функцию, которая принимает этот объект и возвращает массив уникальных жанров книг (new Set, Object.values(), map)

Expected result:

const library = {

'The Great Gatsby': { genre: 'Fiction' },

'To Kill a Mockingbird': { genre: 'Fiction' },

1984: { genre: 'Science Fiction' },

'Pride and Prejudice': { genre: 'Romance' },

'The Catcher in the Rye': { genre: 'Fiction' },

};

console.log(getUniqueGenres(library));

// Вывод: ["Fiction", "Science Fiction", "Romance"]

1. У вас есть массив пар ключ-значение. Напишите функцию, которая принимает этот массив и возвращает новый объект Map с этими парами.

Expected result:

const pairs = [

['a', 1],

['b', 2],

['c', 3],

];

console.log(createMapFromArray(pairs));

// Вывод: Map { 'a' => 1, 'b' => 2, 'c' => 3 }

**Решение задач**

//1)

const superUser = {

id: 10,

name: 'Clementina DuBuque',

username: 'Moriah.Stanton',

email: 'Rey.Padberg@karina.biz',

address: {

street: 'Kattie Turnpike',

suite: 'Suite 198',

city: 'Lebsackbury',

zipcode: '31428-2261',

geo: {

lat: '-38.2386',

lng: '57.2232',

},

},

phone: '024-648-3804',

website: 'ambrose.net',

company: {

name: 'Hoeger LLC',

catchPhrase: 'Centralized empowering task-force',

bs: 'target end-to-end models',

},

};

const newSuperGeo = {

...superUser,

address: { ...superUser.address, geo: { ...superUser.address.geo, lat: 'test coordinate' } },

};

const structureCloneGeo = structuredClone(superUser);

structureCloneGeo.address.geo.lat = 'test coordinate';

console.log('newSuperGeo: ', newSuperGeo.address.geo); //newSuperGeo: { lat: 'test coordinate', lng: '57.2232' }

console.log('structureCloneGeo: ', structureCloneGeo.address.geo); //structureCloneGeo: { lat: 'test coordinate', lng: '57.2232' }

//2)

function sumAndDifference(a, b) {

return {

sum: a + b,

difference: a - b,

};

}

const { sum, difference } = sumAndDifference(10, 5);

console.log('Сумма:', sum); // Вывод: 15

console.log('Разность:', difference); // Вывод: 5

//3)

function getPositiveNumbers(numbers) {

return numbers.filter((num) => num > 0);

}

const numbers = [-1, 2, -3, 4, -5, 6];

const positiveNumbers = getPositiveNumbers(numbers);

console.log(positiveNumbers); // Вывод: [2, 4, 6]

//4)

function mergeArrays(...arrays) {

return arrays.reduce((acc, arr) => acc.concat(arr), []);

}

const arr1 = [1, 2, 3];

const arr2 = [4, 5, 6];

const mergedArray = mergeArrays(arr1, arr2);

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

//5

function findString(array, str) {

return array.includes(str);

}

const strings = ['apple', 'banana', 'orange'];

console.log(findString(strings, 'banana')); // Вывод: true

console.log(findString(strings, 'grape')); // Вывод: false

//6

function sortByProperty(objects, property) {

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

}

const people = [

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

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

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

];

sortByProperty(people, 'age');

console.log(people); // Вывод: [{ name: "Bob", age: 25 }, { name: "Alice", age: 30 }, { name: "Charlie", age: 35 }]

//7

function extractSubarray(arr, start, end) {

const extracted = arr.slice(start, end);

arr.splice(start, extracted.length);

return extracted;

}

let array = [1, 2, 3, 4, 5, 6, 7, 8, 9];

let extracted = extractSubarray(array, 2, 6);

console.log(extracted); // Вывод: [3, 4, 5, 6]

console.log(array); // Вывод: [1, 2, 7, 8, 9]

//8

function addPropertyToObject(obj, propName, propValue) {

obj[propName] = propValue;

}

let obj = { name: 'Alice' };

addPropertyToObject(obj, 'age', 30);

console.log(obj); // Вывод: { name: "Alice", age: 30 }

//9

function formatStudentGrades(students) {

return Object.entries(students).map(([name, grades]) => `${name}: ${grades.join(', ')}`);

}

const students = {

Alice: [4, 5, 3],

Bob: [3, 2, 4],

Charlie: [5, 5, 5],

};

console.log(formatStudentGrades(students));

// Вывод:

// [

// "Alice: 4, 5, 3",

// "Bob: 3, 2, 4",

// "Charlie: 5, 5, 5"

// ]

//10

function getStudentNames(students) {

return Object.keys(students);

}

const students = {

Alice: [4, 5, 3],

Bob: [3, 2, 4],

Charlie: [5, 5, 5],

};

console.log(getStudentNames(students));

// Вывод: ["Alice", "Bob", "Charlie"]

//11

function getUniqueGenres(library) {

const genres = new Set(Object.values(library).map((book) => book.genre));

return [...genres];

}

const library = {

'The Great Gatsby': { genre: 'Fiction' },

'To Kill a Mockingbird': { genre: 'Fiction' },

1984: { genre: 'Science Fiction' },

'Pride and Prejudice': { genre: 'Romance' },

'The Catcher in the Rye': { genre: 'Fiction' },

};

console.log(getUniqueGenres(library));

// Вывод: ["Fiction", "Science Fiction", "Romance"]

//12

function createMapFromArray(pairs) {

return new Map(pairs);

}

const pairs = [

['a', 1],

['b', 2],

['c', 3],

];

console.log(createMapFromArray(pairs));

// Вывод: Map { 'a' => 1, 'b' => 2, 'c' => 3 }