**Задачи на закрепления slice, splice, reduce**

**Задачи на закрепления slice:**1) Напишите функцию, которая получает подстроку из строки, начиная с определенного индекса

Пример:

let str = 'Hello, world!';

let startIndex = 7;

console.log(getSubStr(str, startIndex)); // Выведет: "world!"

2) Напишите функцию, которая возвращает последние n элементов массива.

Expected:

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

3) Создайте функцию, которая извлекает часть пути URL-адреса, начиная с указанного индекса.

Expected:

console.log(extractPath('https://example.com/blog/article', 18)); // Выведет: "/blog/article"

**Задачи на splice:**

1) Напишите функцию, которая принимает массив и индекс элемента для удаления, а затем удаляет этот элемент из массива, модифицируя его

Expected:

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

console.log(removeElement(numbers, 2)); // Выведет: [1, 2, 4, 5]

2) Создайте функцию, которая принимает массив, индекс и новый элемент, а затем вставляет новый элемент в указанную позицию массива

Expected:

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

console.log(insertElement(numbers, 2, 3)); // Выведет: [1, 2, 3, 4, 5]

**Задачи на reduce:**

1. Напишите функцию, которая принимает массив чисел и использует метод reduce, чтобы вернуть сумму всех элементов в массиве

Expected:

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

console.log(sumArray(numbers)); // Выведет: 15

1. Создайте функцию, которая принимает массив строк и возвращает сумму их длин, используя метод reduce

Expected:

console.log(sumStringLengths(['apple', 'banana', 'cherry'])); // Выведет: 17

1. Напишите функцию, которая принимает массив, содержащий вложенные массивы чисел, и возвращает сумму всех чисел в этих массивах

Expected:

console.log(

sumNestedArrays([

[1, 2],

[3, 4, 5],

[6, 7, 8, 9],

])

); // Выведет: 45

1. Создайте функцию, которая принимает массив элементов и возвращает новый массив, содержащий только уникальные элементы, используя метод reduce.

Expected:

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

console.log(removeDuplicates(numbers)); // Выведет: [1, 2, 3, 4, 5, 6]

1. Найдите дубликаты в массиве с помощью метода reduce и верните массив дублированных элементов

Expected:

const array = [1, 2, 3, 2, 4, 5, 4, 5];

const duplicatesArray = findDuplicates(array);

console.log(duplicatesArray); // Output: [2, 4, 5]

1. Напишите функцию, которая принимает массив строк и возвращает объект, в котором ключи - это уникальные строки, а значения - количество их вхождений в массив

Expected:

const words = ['apple', 'banana', 'apple', 'cherry', 'banana', 'apple'];

console.log(wordCount); // Output: { apple: 3, banana: 2, cherry: 1 }

1. Создайте функцию, которая принимает массив объектов и возвращает массив значений определенного свойства объектов

Expected:

const people = [

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

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

{ name: 'Charlie', age: 22 },

];

const names = getPropertyValues(people, 'name');

console.log(names); // Output: ['Alice', 'Bob', 'Charlie']

**Решение задачек:**

**slice ответы:**

// slice 1-st

let str = 'Hello, world!';

let startIndex = 7;

function getSubStr(str, index) {

return str.slice(index);

}

console.log(getSubStr(str, startIndex)); // Выведет: "world!"

// slice 2-d

function getLastElements(arr, n) {

return arr.slice(-n);

}

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

// slice 3-d

function extractPath(url, startIndex) {

return url.slice(startIndex);

}

console.log(extractPath('https://example.com/blog/article', 18)); // Выведет: "/blog/article"

**splice ответы:**

// splice - 1st

function removeElement(arr, index) {

arr.splice(index, 1);

return arr;

}

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

console.log(removeElement(numbers, 2)); // Выведет: [1, 2, 4, 5]

// splice - 2d

function insertElement(arr, index, element) {

arr.splice(index, 0, element);

return arr;

}

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

console.log(insertElement(numbers, 2, 3)); // Выведет: [1, 2, 3, 4, 5]

**reduce ответы:**

// reduce - 1st

function sumArray(arr) {

return arr.reduce((accumulator, currentValue) => accumulator + currentValue, 0);

}

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

console.log(sumArray(numbers)); // Выведет: 15

// reduce - 2d

function sumStringLengths(arr) {

return arr.reduce((total, currentString) => total + currentString.length, 0);

}

console.log(sumStringLengths(['apple', 'banana', 'cherry'])); // Выведет: 17

// reduce - 3d

function sumNestedArrays(arr) {

return arr.reduce((total, nestedArray) => {

return total + nestedArray.reduce((acc, num) => acc + num, 0);

}, 0);

}

console.log(

sumNestedArrays([

[1, 2],

[3, 4, 5],

[6, 7, 8, 9],

])

); // Выведет: 45

// reduce - 4-th

function removeDuplicates(arr) {

return arr.reduce((unique, item) => {

if (!unique.includes(item)) {

unique.push(item);

}

return unique;

}, []);

}

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

console.log(removeDuplicates(numbers)); // Выведет: [1, 2, 3, 4, 5, 6]

// reduce - 5th

function findDuplicates(arr) {

return arr.reduce((duplicates, item, index) => {

if (arr.indexOf(item, index + 1) !== -1 && duplicates.indexOf(item) === -1) {

duplicates.push(item);

}

return duplicates;

}, []);

}

const array = [1, 2, 3, 2, 4, 5, 4, 5];

const duplicatesArray = findDuplicates(array);

console.log(duplicatesArray); // Output: [2, 4, 5]

// reduce -6th

const words = ['apple', 'banana', 'apple', 'cherry', 'banana', 'apple'];

const wordCount = words.reduce((acc, word) => {

acc[word] = (acc[word] || 0) + 1;

return acc;

}, {});

console.log(wordCount); // Output: { apple: 3, banana: 2, cherry: 1 }

//reduce - 7th

function getPropertyValues(arr, property) {

return arr.reduce((result, obj) => {

result.push(obj[property]);

return result;

}, []);

}

const people = [

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

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

{ name: 'Charlie', age: 22 },

];

const names = getPropertyValues(people, 'name');

console.log(names); // Output: ['Alice', 'Bob', 'Charlie']