1. Деструктуризация объекта: Дан объект с информацией о пользователе. Извлеките с помощью деструктуризации имя, возраст и email в переменные из объекта пользователя.

const user = { name: 'Иван', age: 25, email: 'ivan@example.com' };

1. Деструктуризация вложенных объектов: имеется объект с информацией о книге, включая автора, который также является объектом. Используйте деструктуризацию, чтобы извлечь название книги и имя автора в переменные.

const book = { title: 'Война и мир', author: { firstName: 'Лев', lastName: 'Толстой' } };

1. Деструктуризация массивов: Дан массив с именами. Используйте деструктуризацию, чтобы извлечь первое и второе имя в отдельные переменные.

const names = ['Иван', 'Алексей', 'Светлана'];

1. Пропуск значений в массиве: имеется массив чисел. Используйте деструктуризацию, чтобы извлечь первое и третье число, пропустив второе.

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

1. Использование остаточных параметров: Дан массив с несколькими значениями. Используйте деструктуризацию для извлечения первых двух элементов в переменные и оставшиеся элементы поместите в новый массив.

const fruits = ['яблоко', 'банан', 'апельсин', 'киви', 'манго'];

1. Деструктуризация параметров функции (почитайте про это в интернете): Создайте функцию, которая принимает объект с параметрами и использует деструктуризацию для извлечения этих параметров в переменные внутри функции и выводит результат в консоль.

function displayUserInfo() {

console.log()

}

const user = { name: 'Петр', age: 30, email: 'petr@example.com' };

displayUserInfo(user);

1. Значения по умолчанию в деструктуризации: Дан объект с информацией о пользователе, но некоторые поля (например возраст - age) могут отсутствовать. Используйте деструктуризацию со значениями по умолчанию, чтобы избежать ошибок при отсутствии данных.

const user = { name: 'Иван', email: 'ivan@example.com' };