

Массивы в JavaScript

Рассматриваемые темы

- создание массивов
- доступ к элементам и индексы
- методы массивов
- объекты, подобные массивами

Задача 1

Напишите функцию `range()`, принимающую два аргумента: начало и конец диапазона, и возвращающую массив, который содержит все числа из диапазона, включая начальное и конечное. Третий необязательный аргумент функции `range()` – шаг для построения массива. Убедитесь, что функция `range()` работает с отрицательным шагом: например, `range(5, 2, -1)` возвращает `[5, 4, 3, 2]`.

Задача 2

Самостоятельно реализуйте любой алгоритм сортировки массива. Желательно, чтобы ваша функция сортировки могла принимать компаратор для сравнения элементов (как это делает стандартный метод `sort()`).

Задача 2*

Реализуйте функцию `sort()`, чтобы она сохраняла индексы разреженного массива, а не сдвигала индексы в начало.

Задача 3

Создать функцию `createMatrix()`, принимающую количество строк и количество столбцов матрицы и возвращающее матрицу (массив массивов), заполненную случайными числами в диапазоне от 0 до 100. Написать функцию, выполняющую суммирование двух таких «матриц».

Задача 4

Реализуйте функцию `union()`, которая объединит уникальные элементы всех массивов, переданных ей.

Пример: `union([1, 2, 2, 3], [101, 2, 1, 10], [2, 1])` вернёт `[1, 2, 3, 101, 10]`.

Задача 5

Реализуйте функцию `flatten()`, которая в случае, если массив обладает уровнями вложенности, приведёт его к элементарному виду (вложенность может быть любой глубины).

Пример: `flatten([1, [2], [3, [4]]])` вернёт `[1, 2, 3, 4]`.

Задача 6

Функция принимает на вход массив или объект, подобный массиву. Функция возвращает массив, полученный из аргумента удалением триплов. Триплом будем считать элемент, которые повторяется 3 и более раз.

Пример:

```
var a = f([1, 2, 1, 3, 1, 3, 1, 4]); // a = [2, 3, 4]
var b = f("mama ramu");           // b = [" ", "r", "u"];
```