

WEB. Работа с NodeJS.

Урок 3. Практика. Установка сторонних модулей.

Задача. Самостоятельно разобраться с библиотекой **Lodash**.

Краткое описание методов **Lodash**.

- `_.chunk(array, [size=1])` - Разбивает массив на группы элементов по указанному размеру.
- `_.compact(array)` - Удаляет ложные значения (`false`, `null`, `0`, `""`, `undefined`, `NaN`) из массива.
- `_.concat(array, [values])` - Объединяет массив с любыми дополнительными значениями или массивами.
- `_.difference(array, [values])` - Создает массив, содержащий элементы из первого массива, которых нет в последующих массивах.
- `_.drop(array, [n=1])` - Создает срез массива, убирая `n` элементов с начала.
- `_.fill(array, value, [start=0], [end=array.length])` - Заполняет все элементы массива значением, начиная с `start` до `end`.
- `_.find(collection, [predicate=.identity], [fromIndex=0])` - Возвращает первый элемент коллекции, который удовлетворяет предикату.
- `_.flatten(array)` - Создает новый массив с "плоской" структурой, выравнивая вложенные массивы.
- `_.merge(object, [sources])` - Рекурсивно объединяет свойства источников в объект.
- `_.orderBy(collection, [iteratees=[.identity]], [orders])` - Сортирует коллекцию по указанным полям и порядкам.
- `_.random([lower=0], [upper=1], [floating])` - Возвращает случайное число в диапазоне от `lower` до `upper`.
- `_.uniq(array)` - Создает дубликат массива с уникальными значениями.
- `_.debounce(func, [wait=0], [options={}])` - Создает функцию, которая будет вызвана с задержкой.
- `_.throttle(func, [wait=0], [options={}])` - Создает функцию, которая будет вызываться не чаще чем раз в указанный интервал времени.

Задача.

Используя библиотеку **Lodash**, выполните следующие шаги:

1. Простая задача.

- У вас есть массив чисел: `[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]`.
- Разбейте этот массив на подмассивы размером 3.
- Заполните все значения второго подмассива числом 0.

2. Задача средней сложности.

Найдите уникальные значения в объединенном массиве [2, 3, 3, 4, 5, 5, 6].

3. Создайте объект, объединив два других объекта: {a: 1, b: 2} и {b: 3, c: 4}.

4. Сортируйте массив объектов по полю age в порядке убывания: [{name: 'John', age: 25}, {name: 'Jane', age: 30}, {name: 'Jim', age: 20}].