Lodash - это библиотека JavaScript, которая помогает программистам писать более компактный и простой в обслуживании JavaScript код. Lodash делает JavaScript легче с математическими формулами, а именно:

* Повторение массивов, объектов и строк
* Манипулирование и проверка значений
* Создание составных функций

Определение математических формул в библиотеке Lodash:

Utilities — для упрощения общих задач программирования, таких как определение типа, а также упрощение математических операций;

String — функции преобразования для выполнения основных операций с строками, такие как обрезка, преобразование в верхний регистр, [случай верблюда](https://en.wikipedia.org/wiki/Camel_case) и т.д.;

Array — создание, разбиение, объединение, изменение и сжатие;

Collection — итерация, сортировка, фильтрация, расщепление и строительство;

Object — доступ, расширение, слияние, дефолты и преобразование;

Seq — caching, упаковка, фильтрация и тестирование.

Function — упрощение связывания, декорирование, сдерживание, дросселирование, debouncing, currying и изменение указателя

Примеры использования Lodash:

1. Удаление выбранного значения из массива

\_.pull(array, [values])

Удаляет выбранные значения из всего массива

var array = ['a', 'b', 'c', 'a', 'b', 'c'];

\_.pull(array, 'a', 'c');

console.log(array);

// => ['b', 'b']

2. Получение рандомного числа

\_.random([lower=0], [upper=1], [floating])

Производит случайное число между включительно нижней и верхней границами. Если указан только один аргумент, возвращается число от 0 до данного числа. Если значение с плавающей запятой истинно или нижний или верхний предел являются числами с плавающей запятой, вместо целого числа возвращается число с плавающей запятой.

\_.random(0, 5);

// => an integer between 0 and 5

\_.random(5);

// => also an integer between 0 and 5

\_.random(5, true);

// => a floating-point number between 0 and 5

\_.random(1.2, 5.2);

// => a floating-point number between 1.2 and 5.2

3. Получение уникальных значений

\_.union([arrays])

Создает массив уникальных значений из всех переданных массивов

\_.union([2], [1, 2]);

// => [2, 1]

4. Сглаживание массива

\_.flatten(array)

Сглаживание массива на один уровень вложенности

\_.flatten([1, [2, [3, [4]], 5]]);

// => [1, 2, [3, [4]], 5]

5. Удаление свойств из объекта

\_.omit(object, [paths])

Этот метод создает объект, состоящий из собственных и унаследованных путей перечисляемых свойств объекта, которые не пропущены.

// Стандартный метод: Удаление свойств объекта по ключу

Object.prototype.remove = function(arr) {

var that = this;

arr.forEach(function(key){

delete(that[key]);

});

};

var objA = {"name": "colin", "car": "suzuki", "age": 17};

objA.remove(['car', 'age']);

objA; // {"name": "colin"}

// Lodash

objA = \_.omit(objA, ['car', 'age']); // {"name": "colin"}

6. Цикл, повторяющийся n раз

\_.times(n, [iteratee=\_.identity])

Вызывает итерацию n раз, возвращая массив результатов каждого вызова. Итерация вызывается с одним аргументом; (индекс).

// 1. Базовый цикл.

for(var i = 0; i < 5; i++) {

// ....

}

// 2. Использование методов объединения и разделения массива

Array.apply(null, Array(5)).forEach(function(){

// ...

});

// Lodash

\_.times(5, function(){

// ...

});

7. Создания массива с n размера с уникальными значениями

\_.uniqueId([prefix=''])

Создает уникальный id. Если указан префикс, к нему добавляется идентификатор.  
Мы уже знаем, насколько полезно \_.times из предыдущего примера. Используя его в сочетании с методом \_.uniqueId, мы можем предложить более лаконичное решение. Если вы не хотите повторно указывать контекст, в Lodash есть способ и для этого.

// Создайте массив длиной 6 и заполните его уникальными значениями. Значение должно быть префиксом с "ball\_".

// например: [ball\_0, ball\_1, ball\_2, ball\_3, ball\_4, ball\_5]

// Метод map.

Array.apply(null, Array(6)).map(function(item, index){

return "ball\_" + index;

});

// Lodash

\_.times(6, \_.uniqueId.bind(null, 'ball\_'));

Установка библиотеки в браузере:



Ссылка на GitHub:

[GitHub - lodash/lodash: A modern JavaScript utility library delivering modularity, performance, & extras.](https://github.com/lodash/lodash?ysclid=lenu7898gl444825270)

Пример:

