**[Вычислить сумму чисел до данного](https://learn.javascript.ru/recursion" \l "vychislit-summu-chisel-do-dannogo)**

важность: 5

Напишите функцию sumTo(n), которая вычисляет сумму чисел 1 + 2 + ... + n.

Например:

sumTo(1) = 1

sumTo(2) = 2 + 1 = 3

sumTo(3) = 3 + 2 + 1 = 6

sumTo(4) = 4 + 3 + 2 + 1 = 10

...

sumTo(100) = 100 + 99 + ... + 2 + 1 = 5050

Сделайте три варианта решения:

1. С использованием цикла.
2. Через рекурсию, т.к. sumTo(n) = n + sumTo(n-1) for n > 1.
3. С использованием формулы [арифметической прогрессии](https://ru.wikipedia.org/wiki/%D0%90%D1%80%D0%B8%D1%84%D0%BC%D0%B5%D1%82%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%B0%D1%8F_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B5%D1%81%D1%81%D0%B8%D1%8F).

Пример работы вашей функции:

function sumTo(n) { /\*... ваш код ... \*/ }

alert( sumTo(100) ); // 5050

P.S. Какой вариант решения самый быстрый? Самый медленный? Почему?

P.P.S. Можно ли при помощи рекурсии посчитать sumTo(100000)?

1. С использованием цикла.

function sumTo(n) {

            let sum = 0;

            for (let i = 1; i <= n; i++) {

                sum += i;

            }

            return sum;

        }

        alert(sumTo(100));

1. Через рекурсию, т.к. sumTo(n) = n + sumTo(n-1) for n > 1.
2. function sumTo(n) {
3. if (n == 1) return 1;
4. return n + sumTo(n - 1);
5. }
6. alert(sumTo(100));
7. С использованием формулы [арифметической прогрессии](https://ru.wikipedia.org/wiki/%D0%90%D1%80%D0%B8%D1%84%D0%BC%D0%B5%D1%82%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%B0%D1%8F_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B5%D1%81%D1%81%D0%B8%D1%8F).
8. function sumTo(n) {
9. return n \* (n + 1) / 2;
10. }
11. alert(sumTo(100));

### [Вычислить факториал](https://learn.javascript.ru/recursion" \l "vychislit-faktorial)

важность: 4

[Факториал](https://ru.wikipedia.org/wiki/%D0%A4%D0%B0%D0%BA%D1%82%D0%BE%D1%80%D0%B8%D0%B0%D0%BB) натурального числа – это число, умноженное на "себя минус один", затем на "себя минус два", и так далее до 1. Факториал n обозначается как n!

Определение факториала можно записать как:

n! = n \* (n - 1) \* (n - 2) \* ...\*1

Примеры значений для разных n:

1! = 1

2! = 2 \* 1 = 2

3! = 3 \* 2 \* 1 = 6

4! = 4 \* 3 \* 2 \* 1 = 24

5! = 5 \* 4 \* 3 \* 2 \* 1 = 120

Задача – написать функцию factorial(n), которая возвращает n!, используя рекурсию.

alert( factorial(5) ); // 120

P.S. Подсказка: n! можно записать как n \* (n-1)! Например: 3! = 3\*2! = 3\*2\*1! = 6

 function factorial(n) {

            return n ? n \* factorial(n - 1) : 1;

        }

        alert(factorial(5));

### [Числа Фибоначчи](https://learn.javascript.ru/recursion" \l "chisla-fibonachchi)

важность: 5

Последовательность [чисел Фибоначчи](https://ru.wikipedia.org/wiki/%D0%A7%D0%B8%D1%81%D0%BB%D0%B0_%D0%A4%D0%B8%D0%B1%D0%BE%D0%BD%D0%B0%D1%87%D1%87%D0%B8) определяется формулой Fn = Fn-1 + Fn-2. То есть, следующее число получается как сумма двух предыдущих.

Первые два числа равны 1, затем 2(1+1), затем 3(1+2), 5(2+3) и так далее: 1, 1, 2, 3, 5, 8, 13, 21....

Числа Фибоначчи тесно связаны с [золотым сечением](https://ru.wikipedia.org/wiki/%D0%97%D0%BE%D0%BB%D0%BE%D1%82%D0%BE%D0%B5_%D1%81%D0%B5%D1%87%D0%B5%D0%BD%D0%B8%D0%B5) и множеством природных явлений вокруг нас.

Напишите функцию fib(n) которая возвращает n-е число Фибоначчи.

Пример работы:

function fib(n) { /\* ваш код \*/ }

alert(fib(3)); // 2

alert(fib(7)); // 13

alert(fib(77)); // 5527939700884757

P.S. Все запуски функций из примера выше должны работать быстро. Вызов fib(77) должен занимать не более доли секунды.

 function fib(n) {

            let a = 1;

            let b = 1;

            for (let i = 3; i <= n; i++) {

                let c = a + b;

                a = b;

                b = c;

            }

            return b;

        }

        alert(fib(3));

alert( fib(7) );

        alert( fib(77) );

### [Вывод односвязного списка](https://learn.javascript.ru/recursion" \l "vyvod-odnosvyaznogo-spiska)

важность: 5

Допустим, у нас есть односвязный список (как описано в главе [Рекурсия и стек](https://learn.javascript.ru/recursion)):

let list = {

value: 1,

next: {

value: 2,

next: {

value: 3,

next: {

value: 4,

next: null

}

}

}

};

Напишите функцию printList(list), которая выводит элементы списка по одному.

Сделайте два варианта решения: используя цикл и через рекурсию.

Как лучше: с рекурсией или без?

Решение с использованием цикла:

   let list = {

            value: 1,

            next: {

                value: 2,

                next: {

                    value: 3,

                    next: {

                        value: 4,

                        next: null

                    }

                }

            }

        };

        function printList(list) {

            let tmp = list;

            while (tmp) {

                alert(tmp.value);

                tmp = tmp.next;

            }

        }

        printList(list);

через рекурсию:

let list = {

            value: 1,

            next: {

                value: 2,

                next: {

                    value: 3,

                    next: {

                        value: 4,

                        next: null

                    }

                }

            }

        };

        function printList(list) {

            alert(list.value);

            if (list.next) {

                printList(list.next);

            }

        }

        printList(list);

### [Вывод односвязного списка в обратном порядке](https://learn.javascript.ru/recursion" \l "vyvod-odnosvyaznogo-spiska-v-obratnom-poryadke)

важность: 5

Выведите односвязный список из предыдущего задания [Вывод односвязного списка](https://learn.javascript.ru/task/output-single-linked-list) в обратном порядке.

Сделайте два решения: с использованием цикла и через рекурсию.

с использованием цикла:

let list = {

            value: 1,

            next: {

                value: 2,

                next: {

                    value: 3,

                    next: {

                        value: 4,

                        next: null

                    }

                }

            }

        };

        function printReverseList(list) {

            let arr = [];

            let tmp = list;

            while (tmp) {

                arr.push(tmp.value);

                tmp = tmp.next;

            }

            for (let i = arr.length - 1; i >= 0; i--) {

                alert(arr[i]);

            }

        }

        printReverseList(list);

через рекурсию:

let list = {

            value: 1,

            next: {

                value: 2,

                next: {

                    value: 3,

                    next: {

                        value: 4,

                        next: null

                    }

                }

            }

        };

        function printReverseList(list) {

            if (list.next) {

                printReverseList(list.next);

            }

            alert(list.value);

        }

        printReverseList(list);