**[Независимы ли счётчики?](https://learn.javascript.ru/closure" \l "nezavisimy-li-schyotchiki)**

важность: 5

Здесь мы делаем два счётчика: counter и counter2, используя одну и ту же функцию makeCounter.

Они независимы? Что покажет второй счётчик? 0,1 или 2,3 или что-то ещё?

function makeCounter() {

let count = 0;

return function() {

return count++;

};

}

let counter = makeCounter();

let counter2 = makeCounter();

alert( counter() ); // 0

alert( counter() ); // 1

alert( counter2() ); // ?

alert( counter2() ); // ?

Ответ: Два счетчика независимы. Второй счетчик покажет 0,1

**[Объект счётчика](https://learn.javascript.ru/closure" \l "obekt-schyotchika)**

важность: 5

Здесь объект счётчика создан с помощью функции-конструктора.

Будет ли он работать? Что покажет?

function Counter() {

let count = 0;

this.up = function() {

return ++count;

};

this.down = function() {

return --count;

};

}

let counter = new Counter();

alert( counter.up() ); // ?

alert( counter.up() ); // ?

alert( counter.down() ); // ?

Ответ: Будет работать и покажет 1, 2, 1

### [Функция в if](https://learn.javascript.ru/closure" \l "funktsiya-v-if)

Посмотрите на код. Какой будет результат у вызова на последней строке?

let phrase = "Hello";

if (true) {

let user = "John";

function sayHi() {

alert(`${phrase}, ${user}`);

}

}

sayHi();

Ответ: ошибка

**[Сумма с помощью замыканий](https://learn.javascript.ru/closure" \l "summa-s-pomoschyu-zamykaniy)**

важность: 4

Напишите функцию sum, которая работает таким образом: sum(a)(b) = a+b.

Да, именно таким образом, используя двойные круглые скобки (не опечатка).

Например:

sum(1)(2) = 3

sum(5)(-1) = 4

function sum(a) {

            return function (b) {

                return a + b;

            };

        }

        alert(sum(1)(2));

        alert(sum(5)(-1));

### [Фильтрация с помощью функции](https://learn.javascript.ru/closure" \l "filtratsiya-s-pomoschyu-funktsii)

важность: 5

У нас есть встроенный метод arr.filter(f) для массивов. Он фильтрует все элементы с помощью функции f. Если она возвращает true, то элемент добавится в возвращаемый массив.

Сделайте набор «готовых к употреблению» фильтров:

* inBetween(a, b) – между a и b (включительно).
* inArray([...]) – находится в данном массиве.

Они должны использоваться таким образом:

* arr.filter(inBetween(3,6)) – выбирает только значения между 3 и 6 (включительно).
* arr.filter(inArray([1,2,3])) – выбирает только элементы, совпадающие с одним из элементов массива

Например:

/\* .. ваш код для inBetween и inArray \*/

let arr = [1, 2, 3, 4, 5, 6, 7];

alert( arr.filter(inBetween(3, 6)) ); // 3,4,5,6

alert( arr.filter(inArray([1, 2, 10])) ); // 1,2

function inBetween(a, b) {

            return function (x) {

                return x >= a && x <= b;

            };

        }

        function inArray(arr) {

            return function (x) {

                return arr.includes(x);

            };

        }

        let arr = [1, 2, 3, 4, 5, 6, 7];

        alert(arr.filter(inBetween(3, 6)));

        alert( arr.filter(inArray([1, 2, 10])) );

### [Сортировать по полю](https://learn.javascript.ru/closure" \l "sortirovat-po-polyu)

важность: 5

У нас есть массив объектов, который нужно отсортировать:

let users = [

{ name: "John", age: 20, surname: "Johnson" },

{ name: "Pete", age: 18, surname: "Peterson" },

{ name: "Ann", age: 19, surname: "Hathaway" }

];

Обычный способ был бы таким:

// по имени (Ann, John, Pete)

users.sort((a, b) => a.name > b.name ? 1 : -1);

// по возрасту (Pete, Ann, John)

users.sort((a, b) => a.age > b.age ? 1 : -1);

Можем ли мы сделать его короче, например вот таким?

users.sort(byField('name'));

users.sort(byField('age'));

То есть чтобы вместо функции мы просто писали byField(fieldName).

Напишите функцию byField, которая может быть использована для этого.

let users = [

            { name: "John", age: 20, surname: "Johnson" },

            { name: "Pete", age: 18, surname: "Peterson" },

            { name: "Ann", age: 19, surname: "Hathaway" }

        ];

        function byField(field) {

            return (a, b) => a[field] > b[field] ? 1 : -1;

        }

        users.sort(byField('name'));

        users.forEach(user => alert(user.name));

        users.sort(byField('age'));

        users.forEach(user => alert(user.name));

### [Армия функций](https://learn.javascript.ru/closure" \l "armiya-funktsiy)

важность: 5

Следующий код создаёт массив из стрелков (shooters).

Каждая функция предназначена выводить их порядковые номера. Но что-то пошло не так…

function makeArmy() {

let shooters = [];

let i = 0;

while (i < 10) {

let shooter = function() { // функция shooter

alert( i ); // должна выводить порядковый номер

};

shooters.push(shooter);

i++;

}

return shooters;

}

let army = makeArmy();

army[0](); // у 0-го стрелка будет номер 10

army[5](); // и у 5-го стрелка тоже будет номер 10

// ... у всех стрелков будет номер 10, вместо 0, 1, 2, 3...

Почему у всех стрелков одинаковые номера? Почините код, чтобы он работал как задумано.

Всё потому, что внутри функций shooter нет локальной переменной i. Когда вызывается такая функция, она берёт i из своего внешнего лексического окружения.

