Функции

• Напишите функцию **max**, которая сравнивает два числа и *возвращает* большее:

console.log( max(0, -1) ); // 0

• Напишите функцию-аналог **Math.min()**. Функция принимает любое количество чисел и *возвращает* меньшее из них:

console.log( min(0, -1, 100, 500, 100500) ); // -1

• Изучите *перебирающие методы массивов*: **forEach**, **filter**, **map**. Создайте массив объектов users (~10 объектов), каждый объект имеет поля firstname, lastname, age с разными значениями. Используя встроенные функции массивов:

• Отфильтруйте пользователей младше 18 лет

• Добавьте каждому объекту поле fullName, которое является конкатенацией имени и фамилии

• Сформируйте новый массив, который содержит только полное имя пользователей

⁃ Напишите функцию аналог метода массива **shift**. Функция удаляет из переданного в параметре массива первый элемент.

• Напишите функцию аналог метода массива **push**. Функция добавляет в конец переданного в параметре массив произвольное количество элементов.

• Напишите функцию аналог метода Object.assign(). Первый параметр функции - целевой объект, поля которого будут изменены или расширены. Остальные параметры - объекты-источники, полями которых будет расширяться целевой объект.

var source = {firstname: 'Tom', age: 10}

var s = extend(source, {firstname: 'John'}, {lastname: 'Doe'});

console.log(source); // {firstname: 'John', age: 10, lastname: 'Doe'}

console.log(s); // {firstname: 'John', age: 10, lastname: 'Doe'}

• Напишите функцию **setComment** с параметрами: date, message, author. Дата и текст сообщения - обязательные параметры, если какой-то из них или оба отсутствуют, то выполнение функции должно обрываться, а пользователю выдаваться предупреждение (alert) о том, что данные переданы некорректно. Параметр author - опциональный, но должна происходить проверка: если параметр не передан, то вместо него подставляется значение ‘Anonymous’. Функция распечатывает в консоле текст в формате:   
 <имя\_автора>, <дата>  
 <текст\_сообщения>

setComment('2016-11-02', 'Everything is ok', 'John');

**John**, **2016-12-22**  
Everything is ok

setComment('2016-11-02', 'You could do it better!');

**Anonymous**, **2016-12-22**  
You could do it better!

Замыкания

• Используя замыкание, напишите функцию createTimer, которая использует метод **performance.now()** для получения текущей временн*о*й метки и служит для замера времени выполнения другого кода:

var timer = createTimer();

alert('!') // код, время выполнения которого нужно измерить

alert( timer() ); // время в мкс от начала выполнения createTimer() до момента вызова timer()

• Используя замыкания, создайте функцию createAdder(), которая принимает на вход любой примитивный параметр и возвращает функцию, которая добавляет к первому параметру второй. Функция работает по следующему принципу:

var hello = createAdder('Hello, ');

alert( hello('John') ); // Hello, John

alert( hello('Harry') ); // Hello, Harry

var plus = createAdder(5);

alert( plus(1) ); // 6

alert( plus(5) ); // 10