ДЗ на 30.01

1. В сортировке пузырьком добавить следующую оптимизацию (кто не успел): если за проход (внутренний цикл) не было произведено ни одной замены, то закончить сортировку (выйти из внешнего цикла).
2. \* Модифицировать функцию swap так, чтобы она не использовала дополнительную переменную (задание со звездочкой).
3. Написать функцию, которая делает все буквы в каждом элементе массива заглавными.
4. Написать функцию, которая делает первую букву в каждом элементе массива заглавной, а не строчные элементы (числа, булеаны, объекты и т.д.) удаляет.
5. Написать функцию, которая округляет все числовые значения в массиве в большую сторону, а не числовые элементы удаляет.

В следующих задачах нужно применить знания по циклу for, те функции работы с массивами, которые я вам дал на последнем занятии. Предпочтительнее решать используя функции filter, map, reduce, reduceRight, includes, find, если в задаче не указано иного. Если получается только с циклами for и функциями, изменяющими массив (pop, push, shift, unshift, splice), не страшно. Также иногда надо гуглить на предмет того, существует ли такая функция, которая решает задачу в 1 строку. Для лучшего эффекта предлагаю сначала решить задачу самому, а потом попробовать найти решение со встроенной функцией. Например:

|  |
| --- |
| /\*\*  \* Описание задачи: Напишите функцию, которая заполняет новый массив предоставленным значением.  \* Ожидаемый результат: (3, 'a') => ['a', 'a', 'a']  \* @param {number} arraySize - размер массива  \* @param {?} value - значение для массива  \* @returns {Array} \*/  const fill = (arraySize, value) => {  throw new Error('Напишите здесь свое решение'); }  const data = 3; const valueToFill = 'a'; console.log(fill(data, valueToFill)) // ['a', 'a', 'a'] |

Можно написать цикл на arraySize итераций, где на каждом шаге мы будем добавлять в массив value. А также можно просто вызвать встроенный метод fill и решить задачу так

|  |
| --- |
| const fill = (arraySize, value) => new Array(arraySize).fill(value); |

Массив создается через ключевое слово new и конструктор затем, чтобы сразу выделить память под нужное кол-во элементов, иначе, при каждом push массив будет расширяться и занимать новую память (этот процесс может быть времязатратным, если свободного места впереди мало и нужно переносить весь массив в другое место в памяти, где этого места достаточно).

Свои решения можете опубликовывать на <https://commie.io/> и скидывать ссылки мне, желательно сразу все задачи, пронумеруйте их и подпишите комментариями (в примере выше уже есть такой комментарий, только добавьте номер). Так я смогу оставлять вам комментарии, которые вы сможете поправить. Скидывать нужно только js код, без html.

Для удобствы выберите себе 1 метод вывода результата и придерживайтесь его: console.log, alert, document.write (или document.writeln).

|  |
| --- |
| // 1. Напишите функцию, которая заполняет новый массив предоставленным значением.  Ваше решение   // 2. Условие задачи 2  Ваше решение |

6 Напишите не используя reverse

|  |
| --- |
| /\*\*  \* Описание задачи: Напишите функцию, которая разворачивает массив в обратном порядке.  \* Ожидаемый результат: [1, 2, 3] => [3, 2, 1]  \* @param {Array} array - Массив любых элементов  \* @returns {Array} \*/ const reverse = (array) => {  throw new Error('Напишите здесь свое решение'); }  const data = [1, 2, 3]; console.log(reverse(data)); // [3, 2, 1] |

7

|  |
| --- |
| /\*\*  \* Описание задачи: Напишите функцию, которая очищает массив от нежелательных значений, таких как false, undefined, пустые строки, ноль, null.  \* Ожидаемый результат: [0, 1, false, 2, undefined, '', 3, null] => [1, 2, 3]  \* Сложность задачи: 1 of 5  \* @param {Array} array - Массив любых элементов  \* @returns {Array} \*/ const compact = (array) => {  throw new Error('Напишите здесь свое решение'); }  const data = [0, 1, false, 2, undefined, '', 3, null]; console.log(compact(data)) // [1, 2, 3] |

8 Напишите функцию copyArr(arr), которая копирует массив не изменяя оригинал.

|  |
| --- |
| const vegetables = ['Капуста', 'Репа', 'Редиска', 'Морковка']; function arrayClone(arr) {  // Ваш код  }  const arr1 = arrayClone(vegetables);  document.writeln(arr1); // ['Капуста', 'Репа', 'Редиска', 'Морковка'] |

9 Напишите код, который преобразовывает и объединяет все элементы массива в одно строковое значение. Элементы массива будут разделены запятой.

|  |
| --- |
| const vegetables = ['Капуста', 'Репа', 'Редиска', 'Морковка'];  // Ваш код  document.writeln(str1); // "Капуста, Репа, Редиска, Морковка" |

10 Пользователь вводит многозначное число через promt. Напишите функцию colonOdd(num), которая принимает число **num** в качестве аргумента и вставляет двоеточие (:) между двумя нечетными числами. Например, если вводится число **55639217**, то на выход должно быть **5:563:921:7**. Подсказка: по строкам можно итерироваться циклом for так же как и по массиву. Второй аргумент в функции prompt предзаполняет поле для ввода. Четные числа делятся на 2 без остатка, нечетные - нет (вспоминайте оператор деления по модулю). Перед проверкой строку нужно привести к числу.

|  |
| --- |
| const num = prompt('Введите число', 55);  function colonOdd(num) {  // Ваш код  }  document.writeln(colonOdd(num)); // 5:5 |

11

|  |
| --- |
| /\*\*  \* Описание задачи: Напишите функцию, которая убирает повторяющиеся значения.  \* Ожидаемый результат: [1, 2, 3, 1, 2] => [1, 2, 3]  \* @param {Array<string | number>} array - Массив с примитивными значениями  \* @returns {Array} \*/ const unique = (array) => {  throw new Error('Напишите здесь свое решение'); }  const data = [1, 2, 1, 2, 3]; console.log(unique(data)); // [1, 2, 3] |

12

|  |
| --- |
| /\*\*  \* Описание задачи: Напишите функцию, которая сравнивает два массива и возвращает true, если они идентичны.  \* Ожидаемый результат: ([1, 2, 3], [1, 2, 3]) => true  \* @param {Array} firstArray - Массив с примитивными значениями  \* @param {Array} secondArray - Массив с примитивными значениями  \* @returns {boolean} \*/ const isEqual = (firstArray, secondArray) => {  throw new Error('Напишите здесь свое решение'); }  const arr1 = [1, 2, 3, 4]; const arr2 = [1, 2, 3, 4]; const arr3 = [1, 2, 3, 5]; const arr4 = [1, 2, 3, 4, 5]; console.log(isEqual(arr1, arr2)); // true console.log(isEqual(arr1, arr3)); // false console.log(isEqual(arr1, arr4)); // false |

13 Високосным годом является каждый четвертый год. Напишите функцию chooseYears(start, end), которая принимает в качестве аргументов диапазон лет и возвращает массив високосных лет в заданном диапазоне включительно.

|  |
| --- |
| function chooseYears(start, end) {  // Ваш код  }  document.writeln(chooseYears(2000,2018)); // [2000,2004,2008,2012,2016] |

14 Напишите функцию getFirst(array, n), которая возвращает фрагмент массива, содержащий первые **n** элементов массива.

|  |
| --- |
| const array = [1,2,3,4,5,6,7,8,9];   function getFirst(array, n) {  ваш код };  document.writeln(getFirst(array, 1)); // 1 document.writeln(getFirst(array, 4)); // 1,2,3,4 document.writeln(getFirst(array,-3)); // 1,2,3,4,5,6 |

15 Напишите функцию getLast(array, n), которая возвращает фрагмент массива, содержащий последние **n** элементов массива.

|  |
| --- |
| const array = [1,2,3,4,5,6,7,8,9];   function getLast(array, n) {  ваш код };  document.writeln(getLast(array)); // 9 document.writeln(getLast(array, 4)); // 6,7,8,9 document.writeln(getLast(array, 12)); // 1,2,3,4,5,6,7,8,9 |

16 Напишите функцию sumOfSquares(arr), которая возвращает сумму квадратов значений массива.

|  |
| --- |
| function sumOfSquares(arr) {    ...ваш код... }  document.writeln(sumOfSquares([1, 2, 3, 4, 5])); // 55 |

17 Напишите код, который определяет сумму и произведение значений массива.

|  |
| --- |
| const arr = [ 1, 2 , 3 , 4 , 5 , 6 , 7, 8 ,9];   ...ваш код...  document.writeln(" Сумма : "+sum + "Произведение : " + product); /\* Сумма : 45  Произведение : 362880 \*/ |

18 Напишите функцию removeItem(arr, num), которая удаляет определенный элемент из массива.

|  |
| --- |
| function removeItem(arr, num) {  // Ваш код  }  document.writeln(removeItem([3, 5, 7, 8, 5], 5)); // [3,7,8] |

19 Напишите функцию moveElement(arr,from,to), которая позволяет переместить элемент массива из позиции from в позицию to.

|  |
| --- |
| function moveElement(arr,from,to) {  // Ваш код  }  const arr = [ 'a', 'b', 'c', 'd', 'e']; document.writeln(moveElement(arr,3,1)); // ["a", "d", "b", "c", "e"] |

20 Напишите функцию generateRange(start, end), которая генерирует массив заданной длины len, заполненный целыми числами. Каждое последующее число в массиве должно быть больше предыдущего на единицу.

|  |
| --- |
| function generateRange(start, end) {  // Ваш код  }  document.writeln(generateNumbers(0, 5)); // 0,1,2,3,4 document.writeln(generateNumbers(-5, 4)); // -5,-4,-3,-2 |

21 Используя метод map() напишите код, который получает из массива строк новый массив, содержащий их длины.

|  |
| --- |
| const vegetables = ['Капуста', 'Репа', 'Редиска', 'Морковка'];  // Ваш код  alert( vegetableLength ); // 7,4,7,8 |

22 Имеется массив простых чисел: **numbers = [2, 3, 5, 7, 11, 13, 17, 19]**. Использую метод **reduce()** напишите функцию **currentSums(numbers)**, которая возвращает новый массив из такого же числа элементов, в котором на каждой позиции будет находиться сумма элементов массива **numbers** до этой позиции включительно.

|  |
| --- |
| const numbers = [2, 3, 5, 7, 11, 13, 17];  // ваш код  console.log(currentSums(numbers)); // [2, 2+3, 2+3+5, 2+3+5+7, 2+3+5+7+11, 2+3+5+7+11+13, 2+3+5+7+11+13+17] = [ 2,5,10,17,28,41,58] |

23 Перед вами переменная, содержащая строку. Напишите код, создащий массив, который будет состоять из первых букв слов строки **str**. При написании данного кода, используйте метод, который превращает строку в массив split. В него нужно передать символ (строку), по которой исходная строка будет разбита.

|  |
| --- |
| const str = "Каждый охотник желает знать, где сидит фазан.";   // Ваш код   document.write(array); // [К,о,ж,з,г,с,ф] |

24 Напишите код, объединяющий три массива цифр.

|  |
| --- |
| const a = [1,2,3]; const b = [4,5,6]; const c = [7,8,9]; function getArr(a, b, c) {  // Ваш код   }  console.log(getArr(a, b, c)); // [9 8 7 6 5 4 3 2 1] |