# Теоретическая часть

* Массивы в JavaScript, синтаксис создания массивов – операция **[]**, операция создания объекта **new** Array()
* Идентичность массивов, созданных операциями **new** и **[]**, операция **instantof**
* Вывод массива
* Свойство length, создание и удаление элементов массива
* Операция доступа к элементам массива – операция **[]**
* Перебор элементов массива циклом **for … of** – синтаксис ES6+
* Ассоциативные массивы, создание объектов, доступ к свойствам объекта
* Методы массивов
* Двумерные массивы в JavaScript – создание и доступ к элементам
* Область видимости переменных в функциях, блоках кода
* Глобальные и локальные переменные
* Литеральные функции (функциональные выражения) – определение и вызов
* Анонимные функции – не именованный вариант функционального выражения
* Особенности рекурсивных функций для функционального выражения

# Практическая часть

Разработайте функции **JavaScript** для решения следующих задач из сборника Павловской Т.А., Абрамяна М.Э. Скрипты должны выполняться по мере загрузки страницы, не используйте формы ввода, слушатели событий страниц, анимацию интерфейса **JavaScript** – сконцентрируйтесь на изученных вопросах. Формируйте исходные данные при помощи генератора случайных чисел или задавайте присваиванием.

Выводите результаты работы скрипта в разметку при помощи метода **document.write()**, используйте стилизацию, навигацию по страницам. Каждая задача должна размещаться на собственной странице и иметь навигацию на все остальные страницы, на главной странице разместите задание на разработку и навигацию по страницам задач.

**Задача 1.** Введение в функции JavaScript

* **Proc36.** Описать функцию Fib(N), вычисляющую N-й элемент последовательности чисел Фибоначчи F~~K~~, которая описывается следующими формулами: F1 = 1, F2 = 1, FK = FK−2 + FK−1, K = 3, 4, ....

Используя функцию Fib, найти пять чисел Фибоначчи с данными номерами N1, N2, ..., N5. Номера – случайные числа, формировать их при каждой загрузке страницы.

* **Recur9**. Описать рекурсивную функцию NOD(A, B), находящую наибольший общий делитель (НОД) двух целых положительных чисел A и B, используя алгоритм Евклида:

НОД(A, B) = НОД(B, A mod B), если B ≠ 0; НОД(A,0) = A.

С помощью этой функции найти НОД(A, B), НОД(A, C), НОД(A, D), если даны числа A, B, C, D. Числа формировать при помощи генератора случайных чисел.

**Задача 2.** Формировать массив размера N вещественных чисел (и размер и значения) при каждой загрузке страницы:

* **Вычислить** сумму элементов массива, расположенных между первым и последним положительными элементами
* **Вычислить** произведение элементов массива, расположенных между первым и вторым нулевыми элементами
* **Преобразовать** массив таким образом, чтобы сначала располагались все элементы, модуль которых не превышает единицу, а потом — все остальные
* **Преобразовать** массив таким образом, чтобы элементы, равные нулю, располагались после всех остальных.

**Задача 3**. Формировать массив размера N целых чисел (и размер и значения) при каждой загрузке страницы.

* **Удалить** из массива все элементы, встречающиеся более двух раз, и вывести размер полученного массива и его содержимое.
* **Перед** каждым положительным элементом массива вставить элемент с нулевым значением.

**Задача 4.** Даны две непустые очереди, очереди содержат одинаковое количество элементов. Объединить очереди в одну, в которой элементы исходных очередей чередуются (начиная с первого элемента первой очереди). Очереди – массивы, заполненные случайными числами. Формировать массивы (и размер и элементы) при загрузке страницы.

# Дополнительно

Запись занятия можно скачать по [**этой ссылке**](https://cloud.mail.ru/public/C5K1/9uqd1QK6d). Материалы занятия в этом же архиве.