Лабораторная работа №2

«Функции. Массивы. Объекты»

**Задание 1. Функции.** Реализовать функцию своего варианта.

**Вариант 1**

Функция должна принимать один аргумент и возвращать его

*function returnFirstArgument(arg) {*

*}*

**Вариант 2:**

Функция должна принимать два аргумента и возвращать сумму переданных значений. Значение по умолчанию второго аргумента должно быть 100

*function defaultParameterValue(a, b) {*

*}*

**Вариант 3:**

Функция должна возвращать все переданные в нее аргументы в виде массива. Количество переданных аргументов заранее неизвестно

*function returnArgumentsArray() {*

*}*

**Вариант 4:**

Функция должна принимать другую функцию и возвращать результат вызова переданной функции

*function returnFnResult(fn) {*

*}*

**Дополнительное задание:**

Функция должна принимать другую функцию (F) и некоторое количество дополнительных аргументов. Функция должна привязать переданные аргументы к функции F и вернуть получившуюся функцию

*function bindFunction(fn) {*

*}*

**Задание 2. Массивы.**

**Вариант 1:**

Напишите аналог встроенного метода forEach для работы с массивами

*function forEach(array, fn) {*

*}*

**Вариант 2:**

Напишите аналог встроенного метода map для работы с массивами

*function map(array, fn) {*

*}*

**Вариант 3:**

Напишите аналог встроенного метода reduce для работы с массивами

*function reduce(array, fn, initial) {*

*}*

**Вариант 4:**

Напишите аналог встроенного метода [slice](http://learn.javascript.ru/array-methods" \l "%D0%BC%D0%B5%D1%82%D0%BE%D0%B4-slice) для работы с массивами

*function* [*slice*](http://learn.javascript.ru/array-methods#%D0%BC%D0%B5%D1%82%D0%BE%D0%B4-slice) *(array, fn, initial) {*

*}*

**Задание 3. Объекты.**

**Вариант 1:**

Функция принимает объект и имя свойства, которое необходимо удалить из объекта. Функция должна удалить указанное свойство из указанного объекта

*function deleteProperty(obj, prop) {*

*}*

**Вариант 2:**

Функция принимает объект и имя свойства и возвращает true или false. Функция должна проверить существует ли указанное свойство в указанном объекте

*function hasProperty(obj, prop) {*

*}*

**Вариант 3:**

Функция должна получить все перечисляемые свойства объекта и вернуть их в виде массива

*function getEnumProps(obj) {*

*}*

**Вариант 4:**

Функция должна перебрать все свойства объекта, преобразовать их имена в верхний регистра и вернуть в виде массива

*function upperProps(obj) {*

*}*

**Дополнительное задание\*:**

Функция принимает объект и должна вернуть Proxy для этого объекта. Proxy должен перехватывать все попытки записи значений свойств и возводить это значение в квадрат

*function createProxy(obj) {*

*}*