**Рамазанов Линар, ПДо-35**

**Основы JS.**  
**Задание 1:**

function getType(input) {

return typeof input;

}

console.log(getType(25));

**Задание 2:**

a

undefined

Это происходит потому, что сначала выполняется выражение a || b, которое возвращает функцию a, так как она определена и имеет значение истины. Затем выполняется вызов функции a() и выводится в консоль строка "a". Однако, так как функция a не возвращает никакого значения, переменная c получает значение undefined, которое выводится на следующей строке.

**Функции.**

**Задание 1:**

function sumNumbers(a, b, c, d, e) {

let sum = 0;

if (typeof a === "number") {

sum += a;

}

if (typeof b === "number") {

sum += b;

}

if (typeof c === "number") {

sum += c;

}

if (typeof d === "number") {

sum += d;

}

if (typeof e === "number") {

sum += e;

}

return sum;

}

console.log(sumNumbers(3, 2, 1, 5, 7)) // 18

console.log(sumNumbers(3, null, 1, 'asad'))

**Объекты и массивы.**

**Задание 1:**

function convertArrToObj(arr) {

let obj = {};

for (let i = 0; i < arr.length; i++) {

obj[i] = arr[i];

}

return obj;

}

const arr = [1, null, 'test', undefined];

const obj = convertArrToObj(arr);

console.log(obj);

**Задание 2.**

function countFromArr(arr) {

let obj = {};

for (let i = 0; i < arr.length; i++) {

if (obj[arr[i]]) {

obj[arr[i]]++;

} else {

obj[arr[i]] = 1;

}

}

return obj;

}

const arr = [1, 1, 1, 'test', 'test'];

const res = countFromArr(arr);

console.log(res);

**Задание 3:**

function groupByField(arr, field) {

let obj = {};

for (let i = 0; i < arr.length; i++) {

const value = arr[i][field];

if (obj[value]) {

obj[value].push(arr[i]);

} else {

obj[value] = [arr[i]];

}

}

return obj;

}

const arr = [{test: 1},{test: 2},{test: 3},{test: 1},{test: 1}];

const res = groupByField(arr, 'test');

console.log(res);

**Прототипы, наследование.**

**Задание 1:**

Number.prototype.plus = function(num) {

return this + num;

};

console.log((1).plus(2));

**Асинхронная работа JS.**

**Задание 1:**

start

promise constructor

final

p2

p4

p5

timeout