**Задание №1. Деструктуризация**

Есть следующий код:

let arr = ["Tom", "Sam", "Ray", "Bob"];

let [/\* Ваш код \*/] = arr;

console.log(x); // "Tom"

console.log(y); // "Sam"

console.log(z); // [Bob]

Допишите код используя деструктуризацию, чтобы в консоли браузера появились строки, которые написаны в комментариях.

**Задание №2. Деструктуризация**

Есть следующий код:

let names = {

first: "Tom",

second: "Sam",

third: "Ray",

fourth: "Bob"

};

let {/\* Ваш код \*/} = names;

console.log(f); // "Tom"

console.log(x); // "Ray"

console.log(fifth); // "Name №5"

Допишите код используя деструктуризацию, чтобы в консоли браузера появились строки, которые написаны в комментариях.

**Задание №3. Деструктуризация**

Есть следующий код:

let data = {

names: ["Sam", "Tom", "Ray", "Bob"],

ages: [20, 24, 22, 26],

};

let /\* Ваш код \*/ = data;

console.log(name2); // "Tom"

console.log(age2); // 24

console.log(name4); // "Bob"

console.log(age4); // 26

Допишите код используя деструктуризацию, чтобы в консоли браузера появились строки, которые написаны в комментариях.

**Задание №4. Функции**

Напишите функцию mul(), которая принимает любое количество параметров разного типа и возвращает произведение параметров типа Number.

Если параметры типа Number отсутствуют, возвращает число 0.

function mul() {

// Ваш код

}

console.log(mul(1, "str", 2, 3, true)); // 6

console.log(mul(null, "str", false, true)); // 0

Нельзя использовать свойство arguments, но в функцию mul() можно добавить один параметр.

**Задание №5. Функции**

Есть следующий код:

let obj = {

name: "obj",

print: function () {

// Ваш код

}

};

function multiCaller(func, count) {

for (let i = 0; i < count; i++) {

func();

}

}

multiCaller(obj.print(), 3); // "obj", 3 раза

Допишите код используя стрелочные функции, чтобы в консоли браузера появились строки, которые написаны в комментариях.

**Задание №6. Функции**

Есть следующий код:

let server = {

data: 0,

convertToString: function (callback) {

callback((function () {

return this.data + "";

}).bind(this));

}

};

let client = {

server: server,

result: "",

calc: function (data) {

this.server.data = data;

this.server.convertToString(this.notification());

},

notification: function () {

return (function (callback) {

this.result = callback();

}).bind(this);

}

};

client.calc(123);

console.log(client.result); // "123"

console.log(typeof client.result); // "string"

Измените код используя стрелочные функции, чтобы в коде не использовались методы bind().

**Задание №7. Строки**

Есть следующий код:

let names = {

first: "Tom",

second: "Sam",

third: "Ray"

};

console.log(/\* Ваш код \*/); // "Первый: -Tom-, Второй - "Sam", Третий => 'Ray'"

Допишите код используя интерполяцию, чтобы в консоли браузера появилась строка, которая написана в комментариях.

**Задание №8. Строки**

Напишите функцию шаблонизации format(), которая выводит строку в обратном порядке.

Примеры использования функции format():

function format(s, ...v) {

// Ваш код

}

let res1 = format`${1} + ${2} = ${3}`; // "3 = 2 + 1"

console.log(res1);

let res2 = format`Привет ${"Мир "}`; // "Мир Привет "

console.log(res2);

let res3 = format`left${"<->"}right`; // "right<->left"

console.log(res3);