**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ**

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**

**КЕМЕРОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ**

**Институт Цифры**

**ОТЧЕТ**

**О ВЫПОЛНЕНИИ ЛАБОРАТОРНОЙ РАБОТЫ №6**

«Spread, деструктуризация, модули, динамический импорт»

по дисциплине «Языки программирования»

студента 2 курса

**Лободы Максима Алексеевича**

Направление 02.03.02– «Фундаментальная информатика и информационные технологии»

Преподаватель:

ст. пр. Зимин А. И.

Работа защищена

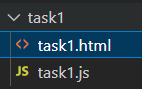
«\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_»

“\_\_\_\_” \_\_\_\_\_\_\_\_\_\_\_\_\_2023 г.

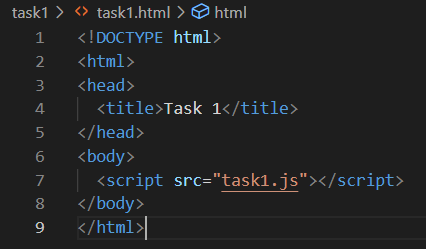
Кемерово 2023

**Основная часть**

Структура файлов каждого задания (на примере первого задания)



В .html-документе содержится такой код:



Данный код позволяет встраивать JavaScript-код непосредственно в веб-страницу.

**Spread оператор**

1. Дан массив с числами. Используя Math.min и spread, выведите на экран минимальное значение массива.

Код программы

numbers = [5, 2, 8, 1, 4];

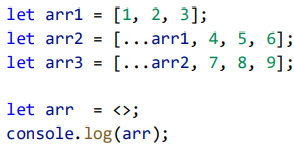
minNumber = Math.min(...numbers);

document.write(minNumber);

Результат выполнения



2. Имеется следующий код:



Что нужно подставить вместо <> (используя arr3 и spread), чтобы в arr появился массив чисел от 0 до 11 включительно.

Код программы

let arr1 = [1, 2, 3];

let arr2 = [...arr1, 4, 5, 6];

let arr3 = [...arr2, 7, 8, 9];

let arr = [...arr3, 10, 11];

document.write(arr);

Результат выполнения



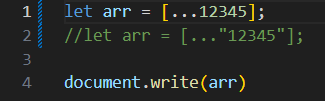
3. Даны два варианта инициализации:

a. let arr = [...12345];

b. let arr = [..."12345"];

Что произойдет при запуске кода каждого варианта и почему?

Код программы а



Результат выполнения



Код программы b



Так происходит из-за того, что число 12345 не является итерируемым, то есть его нельзя разбить на множество элементов.

4. Напишите функцию mean, которая будет принимать параметрами произвольное количество чисел и возвращать их среднее арифметическое.

Код программы

function mean(...numbers)

{

    let sum = 0;

    if (numbers.length === 0)

    {

      return 0;

    }

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

    {

        sum += numbers[i];

    }

    return sum / numbers.length;

}

document.write(mean(1, 2, 3, 4))

Результат выполнения



5. Напишите функцию unite, которая параметрами будет принимать произвольное количество массивов и сливать их в один двухмерный.



Код программы

function unite(...arrs)

{

    return arrs

}

let result = unite([1, 2, 3], [4, 5, 6], [7, 8, 9]);

document.write(result)

Результат выполнения



6. Напишите функцию merge, параметрами принимающую произвольное количество массивов и сливающую их элементы в один массив.



Код программы

function merge(...arrays)

{

    return [].concat(...arrays);

}

let result = merge([1, 2, 3], [4, 5, 6], [7, 8, 9]);

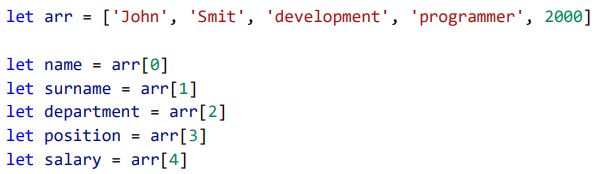
document.write(result);

Результат выполнения



**Деструктуризация массивов**

7. В следующем коде части массива записываются в соответствующие переменные:



Переделайте этот код через деструктуризацию.

Код программы

let arr = ['John ', 'Smit ', 'development ', 'programmer ', 2000]

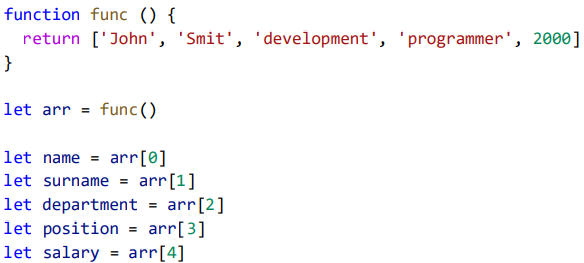
let [name, surname, department, position, salary] = arr

document.write(name, surname, department, position, salary)

Результат выполнения



8. В следующем коде части массива записываются в соответствующие переменные:



Переделайте этот код через деструктуризацию.

Код программы

function func()

{

    return ['John ', 'Smit ', 'development ', 'programmer ', 2000];

}

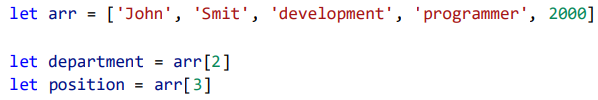
let [name, surname, department, position, salary] = func();

document.write(name, surname, department, position, salary)

Результат выполнения



9. В следующем коде части массива записываются в соответствующие переменные:



Переделайте этот код через деструктуризацию, используя пропуски значений.

Код программы

let arr = ['John', 'Smit', 'development ', 'programmer ', 2000];

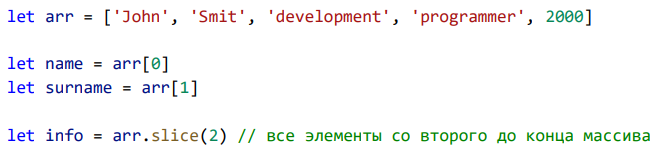
let [, , department, position] = arr;

document.write(department, position)

Результат выполнения



10. В следующем коде части массива записываются в соответствующие переменные:



Переделайте этот код через деструктуризацию, используя запись части значений в массив.

Код программы

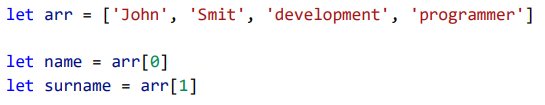
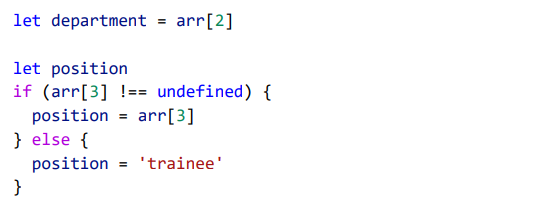
let arr = ['John ', 'Smit ', 'development ', 'programmer ', 2000]

let [name, surname, ...info] = arr;

document.write(name, surname, info)

Результат выполнения



11. В следующем коде части массива записываются в соответствующие переменные:  

Переделайте этот код через деструктуризацию, используя значения по умолчанию.

Код программы

let arr = ['John ', 'Smit ', 'development ', 'programmer '];

//let arr = ['John ', 'Smit ', 'development '];

let [name, surname, department, position = 'trainee'] = arr;

document.write(name, surname, department, position);

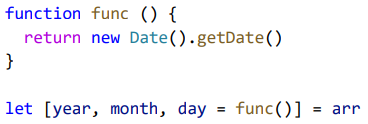
Результат выполнения без использования значения по умолчанию



Результат выполнения со значением по умолчанию



12. В следующем коде значение по умолчанию указывается в виде результата работы функции:



Модифицируйте код так, чтобы при отсутствии в массиве значения для месяца по умолчанию брался текущий месяц, а при отсутствии значения для года - соответственно текущий год.

Код программы

function func ()

{

    return [new Date().getFullYear(), new Date().getMonth(), new Date().getDate()]

}

let arr = [undefined, undefined, undefined]

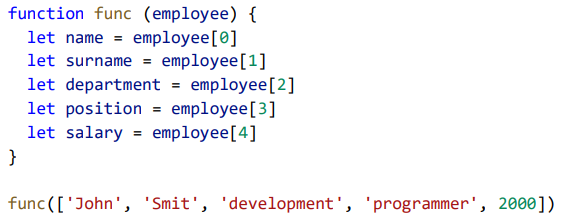
let [year = func()[0], month = func()[1], day = func()[2]] = arr

document.write(`${year} ${month} ${day}`);

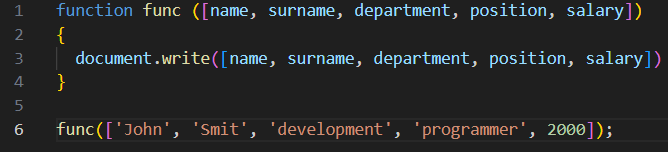
Результат выполнения



13. Переделайте следующий код через деструктуризацию параметров функции:



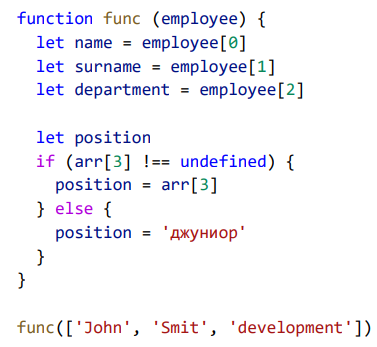
Код программы



Результат выполнения



14. Переделайте следующий код через деструктуризацию параметров функции:



Код программы

function func ([name, surname, department, position = 'джуниор'])

{

  document.write([name, surname, department, position])

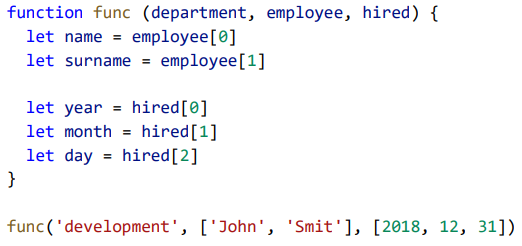
}

func(['John', 'Smit', 'development'])

Результат выполнения



15. Переделайте следующий код через деструктуризацию параметров функции:



Код программы

function func (department, [name, surname], [year, month, day])

{

  document.write(department, [name, surname], [year, month, day])

}

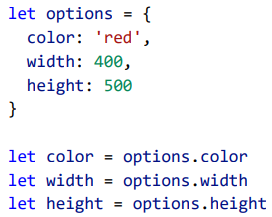
func('development', ['John', 'Smit'], [2018, 12, 31])

Результат выполнения



**Деструктуризация объектов**

16. В следующем коде части объекта записываются в соответствующие переменные:



Переделайте этот код через деструктуризацию.

Код программы

let options = {

    color: 'red',

    width: 400,

    height: 500

}

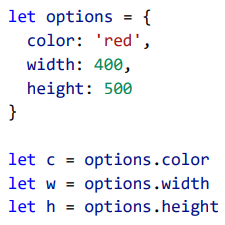
let {color, width, height} = options;

document.write(color + ' ' + width + ' ' + height);

Результат выполнения



17. В следующем коде части объекта записываются в соответствующие переменные:



Переделайте этот код через деструктуризацию, чтобы имена переменных не совпадали с именами ключей объекта.

Код программы

let options = {

  color: 'red',

  width: 400,

  height: 500

}

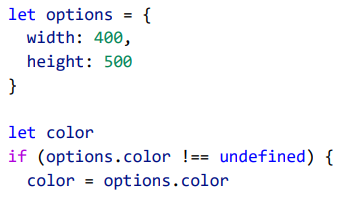
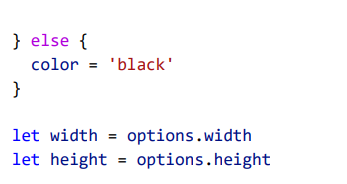
let {color: c, width: w, height: h} = options;

document.write(c + ' ' + w + ' ' + h);

Результат выполнения



18. В следующем коде части объекта записываются в соответствующие переменные:

Переделайте этот код через деструктуризацию, используя значения по умолчанию.

Код программы

let options = {

    width: 400,

    height: 500

  }

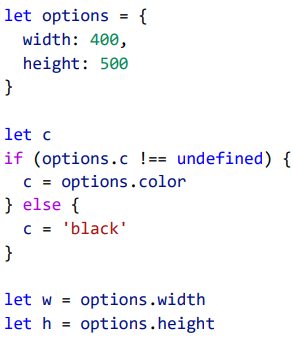
  let {color = 'black', width, height} = options;

document.write(color + ' ' + width + ' ' + height);

Результат выполнения



19. В следующем коде части объекта записываются в соответствующие переменные:



Переделайте этот код через деструктуризацию, используя значения по умолчанию и названия переменных, не совпадающих с ключами объекта.

Код программы

let options = {

    width: 400,

    height: 500

}

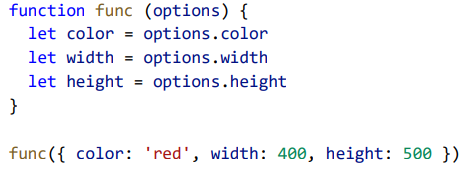
let { color: c = 'black', width: w, height: h } = options;

document.write(c + ' ' + w + ' ' + h);

Результат выполнения



20. Переделайте следующий код через деструктуризацию параметров функции:



Код программы

function func ({color, width, height}) {

    document.write(color + ' ' + width + ' ' + height);

}

func({ color: 'red', width: 400, height: 500 })

Результат выполнения



21. Переделайте следующий код через деструктуризацию параметров функции:



Код программы

function func ({color='black', width, height}) {

    document.write(color + ' ' + width + ' ' + height);

}

func({ color: 'red', width: 400, height: 500 })

Результат выполнения



**Модули, динамический импорт**

22. Возьмите версию задания №2 из лабораторной №4, которая реализована в лабораторной №5 через async/await. Поместите каждую функцию в отдельный модуль. В главном файле выполните динамический импорт всех функций из соответствующих модулей и запустите их в нужном порядке.

Код программы

async function complete()

{

    await(await import('./Modules/readConfig.mjs')).default('myConfig')

    await(await import('./Modules/doQuery.mjs')).default('select \* from cities');

    await(await import('./Modules/httpGet.mjs')).default('http://google.com');

    await(await import('./Modules/readFile.mjs')).default('README.md');

}

complete();

Код каждого из модулей

export default function readConfig (name) {

    return new Promise((resolve, reject) => {

        setTimeout(() => {

        console.log('(1) config from ' + name + ' loaded')

        resolve()

    }, Math.floor(Math.random() \* 1000))

    })

}

export default function doQuery (statement) {

    return new Promise((resolve, reject) => {

        setTimeout(() => {

        console.log('(2) SQL query executed: ' + statement)

        resolve()

    }, Math.floor(Math.random() \* 1000))})

}

export default function httpGet (url) {

    return new Promise((resolve, reject) => {setTimeout(() => {

        console.log('(3) Page retrieved: ' + url)

        resolve()

    }, Math.floor(Math.random() \* 1000))})

}

export default function readFile (path) {

    return new Promise((resolve, reject) => {setTimeout(() => {

        console.log('(4) Readme file from ' + path + ' loaded')

        resolve()

    }, Math.floor(Math.random() \* 1000))})

}

Результат выполнения

