Министерство науки и высшего образования Российской Федерации

**Федеральное государственное бюджетное образовательное учреждение высшего образования**

«Владимирский государственный университет

имени Александра Григорьевича и Николая Григорьевича Столетовых»

**(ВлГУ)**

**Кафедра информационных систем и программной инженерии**

Лабораторная работа №2

по дисциплине «Базовые информационные технологии»

Тема: «Язык JavaScript: Работа со строками. Объекты и массивы. Функции.»

Выполнил:

студент гр. ПРИ-123

А.Ц. Нямаа

Принял:

Ст. пр. кафедры ИСПИ

О.Н. Шамышев

Владимир, 2025 г.

ЦЕЛЬ РАБОТЫ:

Научиться вычислять длину строки Юникода, совершать поиск подстроки в строке с использованием регулярных выражений, проводить валидацию введенных пользователем данных.

Решить задачи с использованием встроенных объектов JavaScript (Map, Set, Date), с помощью методов для обработки списков (.map, .filter, .reduce), научиться созданию сложных объектов (дескрипторы свойств, Proxy, Reflect).

Решить задачи на следующие темы: функции и рекурсию, создание набора функций, замкнутых на общую переменную.

ЗАДАНИЕ 1 (вариант 12).

Создать одностраничный сайт на тему «Сайт часов». На сайте должны осуществляться следующие действия: вычисления длины строки Юникода, поиск подстроки в строке с использованием регулярных выражений, валидация формы введенных пользователем данных.

ВЫПОЛНЕНИЕ РАБОТЫ:

Я выполнил данное задание, написав вычисление длины строки Юникода, поиск подстроки в строке с использованием регулярных выражений и валидацию введенных пользователем данных в форму.

Листинг index.html:

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <link rel="preconnect" href="https://fonts.googleapis.com">

    <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>

    <link href="https://fonts.googleapis.com/css2?family=Overpass:ital,wght@0,100..900;1,100..900&display=swap" rel="stylesheet">

    <title>Clock Shop</title>

    <link rel="stylesheet" href="style.css">

</head>

<body>

    <header>

        <div class="leftText">

            <h1> ДИЗАЙНЕРСКИЙ ЧАСЫ </h1>

            <p> ПОЛНОСТЬЮ РУЧНОЕ ИЗГОТОВЛЕНИЕ </p>

            <button class="buyHeader buttonToChange openModel" data-price="900000"> КУПИТЬ </button>

            <dialog class="dialogMain">

                <form class="orderForm">

                    <h2>Оформление заказа</h2>

                    <p class="hidden" id="validSuccess"></p>

                    <p class="hidden" id="validError"></p>

                    <p><label for="name">Имя:</label>

                    <input type="text" id="name" name="name" placeholder="Имя" required>

                    <p class="hidden text-danger" id="nameP"></p></p>

                    <p><label for="secondName">Фамилия:</label>

                    <input type="text" id="secondName" name="secondName" placeholder="Фамилия" required>

                    <p class="hidden text-danger" id="secondNameP"></p></p>

                    <p><label for="thirdName">Отчество:</label>

                    <input type="text" id="thirdName" name="thirdName" placeholder="Отчество">

                    <p class="hidden text-danger" id="thirdNameP"></p></p>

                    <p><label for="address">Адрес доставки:</label>

                    <input type="text" id="address" name="address" placeholder="Ул. Пушкино, д. 2, кв. 1" required></p>

                    <p><label for="quantity">Количество:</label>

                    <input type="number" id="quantity" name="quantity" placeholder="1" min="1" max="10" required></p>

                    <p><label for="promo">Промокод:</label>

                    <input type="text" id="promo" placeholder="\*Промокод\*" name="promo"></p>

                    <p class="priceHeader"></p>

                    <input type="submit" value="Заказать!">

                </form>

                <button id="closeModel" class="closeModel">&#x2715 </button>

            </dialog>

        </div>

        <div class="rightText">

            <p> JACOB & CO <br>EPIC X TOURBILON BAGUETTE <br>900.000$</p>

            <img src="imgs/mainPhotoWatches.png">

        </div>

    </header>

    <main>

        <div class="textInMain">

            <p>ХИТЫ</p>

            <p>ПОСЛЕДНИЕ ПОСТУПЛЕНИЯ</p>

        </div>

        <div class="productsList">

            <div class="productUnit">

                <img src="imgs/appleulta.png">

                <p>Apple watch ulta 2</p>

                <p>700$</p>

                <button class="buttonUnit buttonToChange openModel" data-price="700"> Купить </button>

            </div>

            <div class="productUnit">

                <img src="imgs/rolexDayDate36.png">

                <p>Rolex Day-Date 36</p>

                <p>117.250$</p>

                <button class="buttonUnit buttonToChange openModel" data-price="117250"> Купить </button>

            </div>

            <div class="productUnit">

                <img src="imgs/KingSeiko.png">

                <p>King Seiko Automatic</p>

                <p>2.000$</p>

                <button class="buttonUnit buttonToChange openModel" data-price="2000"> Купить </button>

            </div>

            <div class="productUnit">

                <img src="imgs/PatekPhilippe.png">

                <p>Paket Philippe 5270/1R</p>

                <p>230.000$</p>

                <button class="buttonUnit buttonToChange openModel" data-price="230000"> Купить </button>

            </div>

            <div class="productUnit">

                <img src="imgs/rolexDateJust31.png">

                <p>Rolex DateJust 31</p>

                <p>51.900$</p>

                <button class="buttonUnit buttonToChange openModel" data-price="51900"> Купить </button>

            </div>

            <div class="productUnit">

                <img src="imgs/PatekPhilippeBlackWhite.png">

                <p>Patek Philippe 6300/400G</p>

                <p>9.000.000$</p>

                <button class="buttonUnit buttonToChange openModel" data-price="9000000"> Купить </button>

            </div>

            <div class="productUnit">

                <img src="imgs/appleseries10.png">

                <p>Apple watch series 10</p>

                <p>450$</p>

                <button class="buttonUnit buttonToChange openModel" data-price="450"> Купить </button>

            </div>

            <div class="productUnit">

                <img src="imgs/Jacob&Co.png">

                <p>Jacob&Co Palatial Classic</p>

                <p>16.500$</p>

                <button class="buttonUnit buttonToChange openModel" data-price="16500"> Купить </button>

            </div>

        </div>

        <div class="addModule">

            <div class="leftSide">

                <div class="textInMain addText">

                    <p>ВЫЧИСЛЕНИЕ</p>

                    <p>ДЛИНЫ СТРОКИ ЮНИКОДА</p>

                </div>

                <form class="unicodeForm">

                    <p><label for="unicodeCalc">Введите Юникод</label>

                    <input type="text" required for="unicodeCalc" placeholder="Юникод..." id="unicodeInput"></p>

                    <input type="submit" value="Посчитать длину">

                </form>

                <p class="unicodeResult"></p>

            </div>

            <div class="rightSide">

                <div class="textInMain addText">

                    <p>ПОИСК ПОДСТРОКИ</p>

                </div>

                <form class="strFind">

                    <p><label for="toSearchFor">Строка для сравнения</label>

                    <input type="text" id="toSearch" for="toSearchFor" placeholder="Введите строку"></p>

                    <p><label for="toFindFor">Строка для поиска</label>

                    <input type="text" id="toFind" for="toFindFor" placeholder="Введите регулярное выражение"></p>

                    <input type="submit" value="Найти!">

                </form>

                <p class="checkResult"></p>

            </div>

        </div>

    </main>

    <script src="script.js"></script>

</body>

</html>

Листинг script.js:

// Смена цвета кнопки при наведении

const button = document.body.querySelectorAll(".buttonToChange");

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

    button[i].addEventListener("mouseover", function () {

        button[i].classList.add("changebackground");

    });

    button[i].addEventListener("mouseout", function () {

        button[i].classList.remove("changebackground");

    });

}

// Модальное окно

const dialog = document.querySelector(".dialogMain");

const html = document.querySelector("html");

const openModels = document.querySelectorAll(".openModel");

const priceHeader = document.querySelector(".priceHeader");

let price;

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

    openModels[i].addEventListener("click", function () {

        dialog.showModal();

        html.classList.add("changehtml");

        if (openModels[i].getAttribute("data-price") > 0) {

            price = parseInt(openModels[i].getAttribute("data-price"));

        }

        priceHeader.textContent = `Цена: ${price.toLocaleString()}$`;

    });

}

document.querySelector("#closeModel").onclick = function () {

    dialog.close();

    html.classList.remove("changehtml");

};

//Изменения цены

const formElement = document.querySelector(".orderForm");

const quantity = document.querySelector("#quantity");

const promo = document.querySelector("#promo");

const promoList = {

    "10procentSaleWOW": 0.9,

    "15yearShop": 0.85,

};

function updatePrice() {

    let quantityVal = 1;

    if (quantity.value > 1) {

        quantityVal = quantity.value;

    }

    let discount = promo.value in promoList ? promoList[promo.value] : 1;

    priceHeader.textContent = `Цена: ${createPrice(price, quantityVal, discount).toLocaleString()}$ (${quantityVal} шт, скидка ${Math.round((1 - discount) \* 100)}%)`;

}

function createPrice(priceFunc, quantityFunc, discountFunc) {

    let finalPrice = priceFunc \* quantityFunc \* discountFunc;

    return finalPrice;

}

quantity.addEventListener("input", updatePrice);

promo.addEventListener("input", updatePrice);

// Отправка формы

formElement.addEventListener("submit", (event) => {

    event.preventDefault();

    const formData = new FormData(formElement);

    const name = formData.get("name");

    const surname = formData.get("secondName");

    const thirdName = formData.get("thirdName");

    const address = formData.get("address");

    if (!checker(name, "nameP") || !checker(surname, "secondNameP") || !checker(thirdName, "thirdNameP")) {

        document.getElementById("validError").textContent = "Ошибка! Некоторый данный были введены неправильно";

        document.getElementById("validError").classList.remove("hidden");

        document.getElementById("validSuccess").classList.add("hidden");

    } else {

        document.getElementById("validSuccess").textContent = `Спасибо, ${name} ${surname} ${thirdName}!\nВаш заказ на адрес - ${address} в количестве ${quantity.value}шт будет исполнен в ближайшее время!`;

        document.getElementById("validSuccess").classList.remove("hidden");

        document.getElementById("validError").classList.add("hidden");

    }

});

function checker(str, idEl) {

    const regexStr = /^[A-Za-zА-Яа-яЁё]+$/;

    if (!regexStr.test(str)) {

        document.getElementById(idEl).textContent = "Не должно быть цифр!";

        document.getElementById(idEl).classList.remove("hidden");

        return false;

    }

    if (str.length > 50 || str.length < 3) {

        document.getElementById(idEl).textContent = "От 3 до 50 символов!";

        document.getElementById(idEl).classList.remove("hidden");

        return false;

    }

    document.getElementById(idEl).textContent = "";

    return true;

}

// Посчитать длину строки Unicode

const formUnicode = document.querySelector(".unicodeForm");

function calcUnicode(event) {

    event.preventDefault();

    const checker = /[\u0300-\u036F\u1AB0-\u1AFF\u1DC0-\u1DFF\u20D0-\u20FF\uFE20-\uFE2F]/;

    const unicodeString = formUnicode.querySelector("#unicodeInput").value;

    let count = 0;

    for (let item of unicodeString.normalize()) {

        if (!checker.test(item)) {

            count++;

        }

    }

    document.querySelector(".unicodeResult").textContent = `Длина строки Unicode - ${count}`;

}

formUnicode.addEventListener("submit", calcUnicode);

//Поиск подстроки

const formStrFind = document.querySelector(".strFind");

function findStr(event) {

    event.preventDefault();

    const strToCheck = document.getElementById("toSearch").value;

    const strToFind = document.getElementById("toFind").value;

    const regex = new RegExp(strToFind, "g");

    const matches = strToCheck.match(regex);

    if (!matches) {

        document.querySelector(".checkResult").textContent = "Совпадения нет";

    } else {

        document.querySelector(".checkResult").textContent = `Найдено ${matches.length} совпадение(ий) - ${matches}`;

    }

}

formStrFind.addEventListener("submit", findStr);

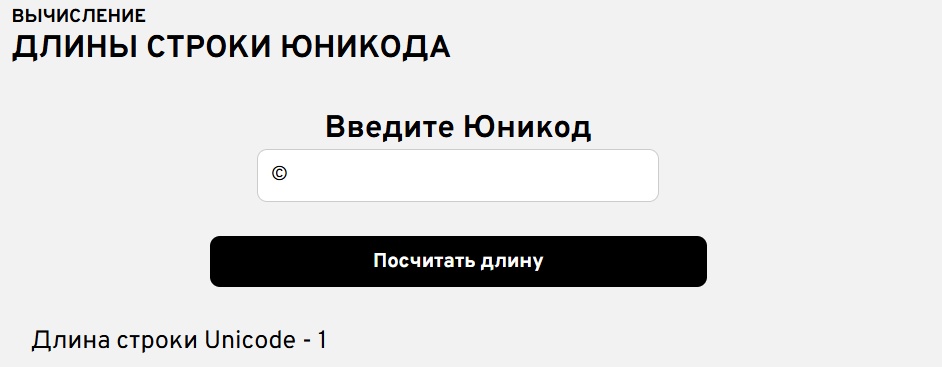


Рисунок 1. Вычисление строки Unicode.

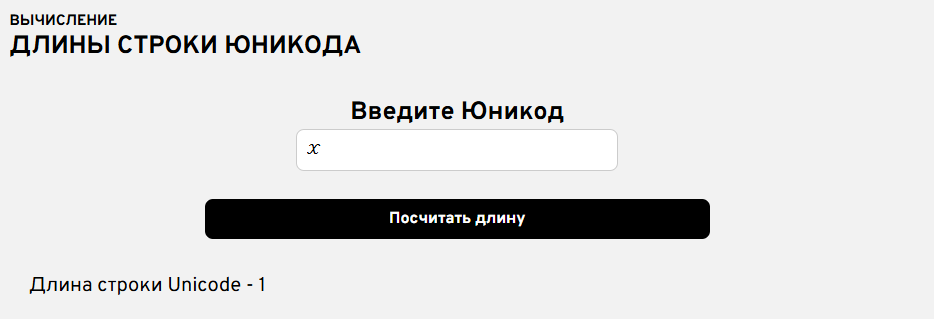


Рисунок 2. Вычисление строки Unicode.



Рисунок 3. Поиск подстроки.



Рисунок 4. Ошибка валидации.

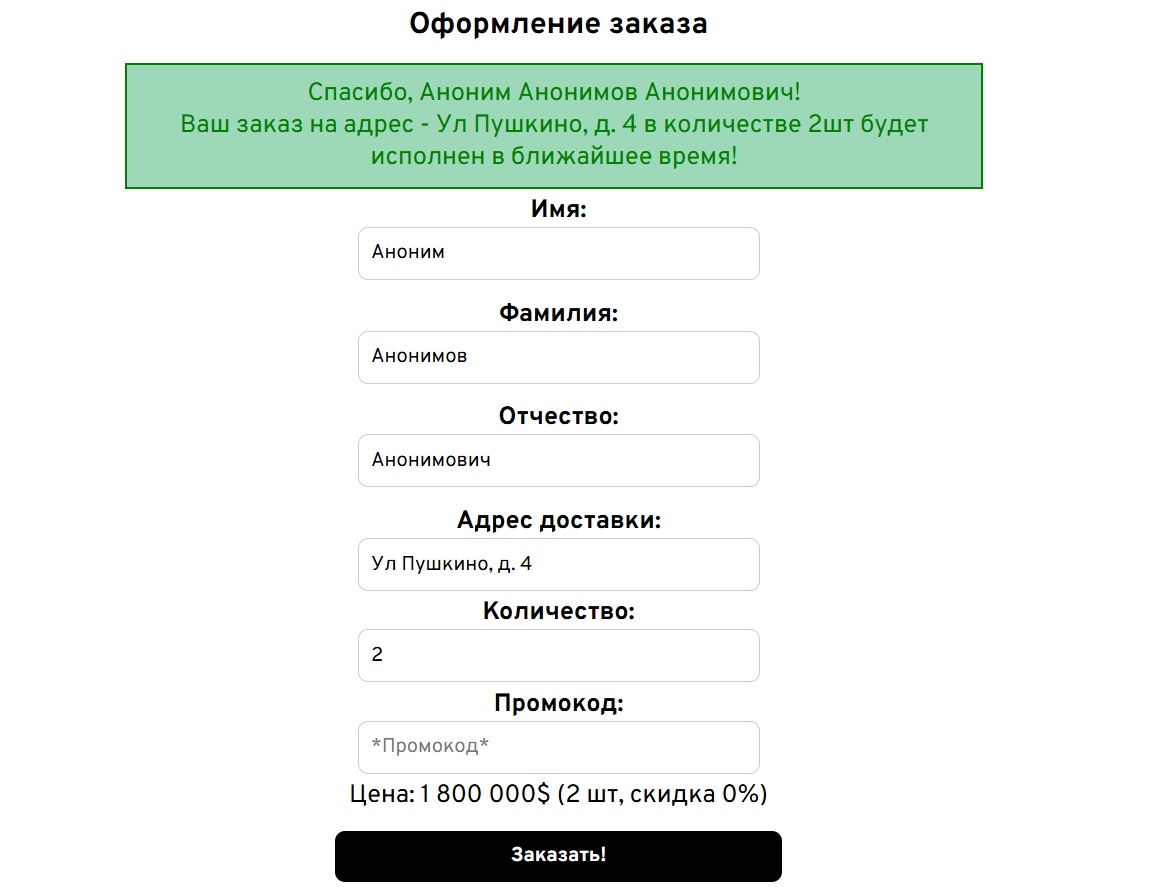


Рисунок 5. Валидация пройдена.

ЗАДАНИЕ 2 (вариант 12).

Создать одностраничный сайт на тему «Сайт часов». На сайте должны осуществляться следующие действия: использование объектов Map, Set, Date, обработка списка с помощью методов .map, .filter, .reduce, использование сложных объектов (дескрипторы свойств, Proxy, Reflect)

ВЫПОЛНЕНИЕ РАБОТЫ:

Я выполнил данное задание, использовав объекты Map, Set, Date, обработав список с помощью методов .map, .filter, .reduce и использовав сложные объекты.

Листинг index.html:

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <link rel="preconnect" href="https://fonts.googleapis.com">

    <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>

    <link href="https://fonts.googleapis.com/css2?family=Overpass:ital,wght@0,100..900;1,100..900&display=swap" rel="stylesheet">

    <title>Clock Shop</title>

    <link rel="stylesheet" href="style.css">

</head>

<body>

    <header>

        <span class="time" id="time"></span>

        <div class="leftText">

            <h1> ДИЗАЙНЕРСКИЙ ЧАСЫ </h1>

            <p> ПОЛНОСТЬЮ РУЧНОЕ ИЗГОТОВЛЕНИЕ </p>

            <button class="buyHeader buttonToChange openModel" data-price="900000"> КУПИТЬ </button>

            <dialog class="dialogMain">

                <form class="orderForm">

                    <h2>Оформление заказа</h2>

                    <p class="hidden" id="validSuccess"></p>

                    <p class="hidden" id="validError"></p>

                    <p><label for="name">Имя:</label>

                    <input type="text" id="name" name="name" placeholder="Имя" required>

                    <p class="hidden text-danger" id="nameP"></p></p>

                    <p><label for="secondName">Фамилия:</label>

                    <input type="text" id="secondName" name="secondName" placeholder="Фамилия" required>

                    <p class="hidden text-danger" id="secondNameP"></p></p>

                    <p><label for="thirdName">Отчество:</label>

                    <input type="text" id="thirdName" name="thirdName" placeholder="Отчество">

                    <p class="hidden text-danger" id="thirdNameP"></p></p>

                    <p><label for="address">Адрес доставки:</label>

                    <input type="text" id="address" name="address" placeholder="Ул. Пушкино, д. 2, кв. 1" required></p>

                    <p><label for="quantity">Количество:</label>

                    <input type="number" id="quantity" name="quantity" placeholder="1" min="1" max="10" required></p>

                    <p><label for="promo">Промокод:</label>

                    <input type="text" id="promo" placeholder="\*Промокод\*" name="promo"></p>

                    <p class="priceHeader"></p>

                    <input type="submit" value="Заказать!">

                </form>

                <button id="closeModel" class="closeModel">&#x2715 </button>

            </dialog>

        </div>

        <div class="rightText">

            <p> JACOB & CO <br>EPIC X TOURBILON BAGUETTE <br>900.000$</p>

            <img src="imgs/mainPhotoWatches.png">

        </div>

    </header>

    <main>

        <div class="textInMain">

            <p>ХИТЫ</p>

            <p>ПОСЛЕДНИЕ ПОСТУПЛЕНИЯ</p>

        </div>

        <div class="productsList">

            <div class="productUnit">

                <img src="imgs/appleulta.png">

                <p>Apple watch ulta 2</p>

                <p>700$</p>

                <button class="buttonUnit buttonToChange openModel" data-price="700"> Купить </button>

            </div>

            <div class="productUnit">

                <img src="imgs/rolexDayDate36.png">

                <p>Rolex Day-Date 36</p>

                <p>117.250$</p>

                <button class="buttonUnit buttonToChange openModel" data-price="117250"> Купить </button>

            </div>

            <div class="productUnit">

                <img src="imgs/KingSeiko.png">

                <p>King Seiko Automatic</p>

                <p>2.000$</p>

                <button class="buttonUnit buttonToChange openModel" data-price="2000"> Купить </button>

            </div>

            <div class="productUnit">

                <img src="imgs/PatekPhilippe.png">

                <p>Paket Philippe 5270/1R</p>

                <p>230.000$</p>

                <button class="buttonUnit buttonToChange openModel" data-price="230000"> Купить </button>

            </div>

            <div class="productUnit">

                <img src="imgs/rolexDateJust31.png">

                <p>Rolex DateJust 31</p>

                <p>51.900$</p>

                <button class="buttonUnit buttonToChange openModel" data-price="51900"> Купить </button>

            </div>

            <div class="productUnit">

                <img src="imgs/PatekPhilippeBlackWhite.png">

                <p>Patek Philippe 6300/400G</p>

                <p>9.000.000$</p>

                <button class="buttonUnit buttonToChange openModel" data-price="9000000"> Купить </button>

            </div>

            <div class="productUnit">

                <img src="imgs/appleseries10.png">

                <p>Apple watch series 10</p>

                <p>450$</p>

                <button class="buttonUnit buttonToChange openModel" data-price="450"> Купить </button>

            </div>

            <div class="productUnit">

                <img src="imgs/Jacob&Co.png">

                <p>Jacob&Co Palatial Classic</p>

                <p>16.500$</p>

                <button class="buttonUnit buttonToChange openModel" data-price="16500"> Купить </button>

            </div>

        </div>

        <div class="addModule">

            <div class="leftSide">

                <div class="textInMain addText">

                    <p>ОБЪЕКТ MAP</p>

                </div>

                <form class="decorateBlock">

                    <p><label for="MAPkeyAdd">Ключ</label>

                    <input type="text" required for="MAPkeyAdd" placeholder="1.." id="keyInput"></p>

                    <p><label for="MAPvalueAdd">Значение</label>

                    <input type="text" required for="MAPvalueAdd" placeholder="Значение..." id="valueInput"></p>

                    <button id="MAPadd"> Добавить </button>

                </form>

                <form class="decorateBlock">

                    <p><label for="MAPkeyKey">Ключ</label>

                    <input type="text" required for="MAPkeyKey" placeholder="1.." id="keyWork"></p>

                    <button id="MAPcheck"> Проверить </button>

                    <button id="MAPdelete"> Удалить </button>

                    <button id="MAPget"> Вернуть </button>

                </form>

                <div class="decorateBlock">

                    <button id="MAPclear"> Очистить </button>

                    <button id="MAPgetAll"> Вернуть все </button>

                    <button id="CreateMass"> Массив </button>

                    <button id="MAPLength"> Длина </button>

                </div>

                <p id="MapResult" class="messageText"></p>

            </div>

            <div class="rightSide">

                <div class="textInMain addText">

                    <p>ОБЪЕКТ SET</p>

                </div>

                <form class="decorateBlock">

                    <p><label for="SETvalue">Значение</label>

                    <input type="text" required for="SETvalue" placeholder="1.." id="valueWork"></p>

                    <button id="SETadd"> Добавить </button>

                    <button id="SETcheck"> Проверить </button>

                    <button id="SETdelete"> Удалить </button>

                </form>

                <div class="decorateBlock">

                    <button id="SETclear"> Очистить </button>

                    <button id="SETgetAll"> Вернуть все </button>

                    <button id="SETmulty"> Все числа </button>

                    <button id="SETSumm"> Сумма чисел </button>

                </div>

                <p id="SETResult" class="messageText"></p>

            </div>

        </div>

        <div class="addModule">

            <div class="leftSide">

                <div class="textInMain addText">

                    <p>ДЕСКРИПТОРЫ</p>

                </div>

                <form class="decorateBlock">

                    <p><label for="DescriptorProperty">Свойство</label>

                    <input type="text" required for="DescriptorProperty" placeholder="Свойство..." id="DescriptorPropertyInput"></p>

                    <p><label for="DescriptorValue">Значение</label>

                    <input type="text" required for="DescriptorValue" placeholder="Значение..." id="DescriptorValueInput"></p>

                    <button id="DescriptorAdd"> Записать </button>

                    <button id="DescriptorChange"> Перезначить </button>

                </form>

                <div class="decorateBlock">

                    <p><label for="DescriptorPropertyOtherAction">Свойство</label>

                    <input type="text" required for="DescriptorPropertyOtherAction" placeholder="Свойство..." id="DescriptorPropertyOtherActionInput"></p>

                    <button id="DescriptorWritable"> Перезапись </button>

                    <button id="DescriptorEnumerable"> Видимость </button>

                    <button id="DescriptorConfigurate"> Переконфиг. </button>

                </div>

                <div class="decorateBlock">

                    <button id="DescriptorShowAll"> Показать все </button>

                </div>

                <p id="DescriptorResult" class="messageText"></p>

            </div>

            <div class="rightSide">

                <div class="textInMain addText">

                    <p>ПРОКСИ</p>

                </div>

                <form class="decorateBlock">

                    <button id="ProxyGetButton"> Название </button>

                </form>

                <div class="decorateBlock">

                    <p><label for="ProxyActionValue">Цена</label>

                    <input type="number" required for="ProxyActionValue" placeholder="Свойство..." id="ProxyActionValueInput"></p>

                    <button id="ProxyActionValueButton"> Перезаписать </button>

                </div>

                <p id="ProxyResult" class="messageText"></p>

            </div>

        </div>

    </main>

    <script src="script.js"></script>

</body>

</html>

Листинг script.js:

// Смена цвета кнопки при наведении

const button = document.body.querySelectorAll(".buttonToChange");

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

    button[i].addEventListener("mouseover", function () {

        button[i].classList.add("changebackground");

    });

    button[i].addEventListener("mouseout", function () {

        button[i].classList.remove("changebackground");

    });

}

// Модальное окно

const dialog = document.querySelector(".dialogMain");

const html = document.querySelector("html");

const openModels = document.querySelectorAll(".openModel");

const priceHeader = document.querySelector(".priceHeader");

let price;

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

    openModels[i].addEventListener("click", function () {

        dialog.showModal();

        html.classList.add("changehtml");

        if (openModels[i].getAttribute("data-price") > 0) {

            price = parseInt(openModels[i].getAttribute("data-price"));

        }

        priceHeader.textContent = `Цена: ${price.toLocaleString()}$`;

    });

}

document.querySelector("#closeModel").onclick = function () {

    dialog.close();

    html.classList.remove("changehtml");

};

//Изменения цены

const formElement = document.querySelector(".orderForm");

const quantity = document.querySelector("#quantity");

const promo = document.querySelector("#promo");

const promoList = {

    "10procentSaleWOW": 0.9,

    "15yearShop": 0.85,

};

function updatePrice() {

    let quantityVal = 1;

    if (quantity.value > 1) {

        quantityVal = quantity.value;

    }

    let discount = promo.value in promoList ? promoList[promo.value] : 1;

    priceHeader.textContent = `Цена: ${createPrice(price, quantityVal, discount).toLocaleString()}$ (${quantityVal} шт, скидка ${Math.round((1 - discount) \* 100)}%)`;

}

function createPrice(priceFunc, quantityFunc, discountFunc) {

    let finalPrice = priceFunc \* quantityFunc \* discountFunc;

    return finalPrice;

}

quantity.addEventListener("input", updatePrice);

promo.addEventListener("input", updatePrice);

// Отправка формы

formElement.addEventListener("submit", (event) => {

    event.preventDefault();

    const formData = new FormData(formElement);

    const name = formData.get("name");

    const surname = formData.get("secondName");

    const thirdName = formData.get("thirdName");

    const address = formData.get("address");

    if (!checker(name, "nameP") || !checker(surname, "secondNameP") || !checker(thirdName, "thirdNameP")) {

        document.getElementById("validError").textContent = "Ошибка! Некоторый данный были введены неправильно";

        document.getElementById("validError").classList.remove("hidden");

        document.getElementById("validSuccess").classList.add("hidden");

    } else {

        document.getElementById("validSuccess").textContent = `Спасибо, ${name} ${surname} ${thirdName}!\nВаш заказ на адрес - ${address} в количестве ${quantity.value}шт будет исполнен в ближайшее время!`;

        document.getElementById("validSuccess").classList.remove("hidden");

        document.getElementById("validError").classList.add("hidden");

    }

});

function checker(str, idEl) {

    const regexStr = /^[A-Za-zА-Яа-яЁё]+$/;

    if (!regexStr.test(str)) {

        document.getElementById(idEl).textContent = "Не должно быть цифр!";

        document.getElementById(idEl).classList.remove("hidden");

        return false;

    }

    if (str.length > 50 || str.length < 3) {

        document.getElementById(idEl).textContent = "От 3 до 50 символов!";

        document.getElementById(idEl).classList.remove("hidden");

        return false;

    }

    document.getElementById(idEl).textContent = "";

    return true;

}

//Объект MAP

let mapBlock = new Map();

document.getElementById("MAPadd").addEventListener("click", function (event) {

    event.preventDefault();

    mapBlock.set(document.getElementById("keyInput").value, document.getElementById("valueInput").value);

    document.getElementById("MapResult").textContent = "Значение добавилось успешно!";

});

document.getElementById("MAPcheck").addEventListener("click", function (event) {

    event.preventDefault();

    if (!mapBlock.has(document.getElementById("keyWork").value)) {

        document.getElementById("MapResult").textContent = "Значения под таким ключом нет((";

    } else {

        document.getElementById("MapResult").textContent = "Значение под таким ключом существует!";

    }

});

document.getElementById("MAPdelete").addEventListener("click", function (event) {

    event.preventDefault();

    if (mapBlock.has(document.getElementById("keyWork").value)) {

        mapBlock.delete(document.getElementById("keyWork").value);

        document.getElementById("MapResult").textContent = "Значение удалено успешно!";

    } else {

        document.getElementById("MapResult").textContent = "Такого значения и не было!";

    }

});

document.getElementById("MAPget").addEventListener("click", function (event) {

    event.preventDefault();

    if (mapBlock.has(document.getElementById("keyWork").value)) {

        keyToGet = document.getElementById("keyWork").value;

        valueToGet = mapBlock.get(keyToGet);

        document.getElementById("MapResult").textContent = `Значение под ключом ${keyToGet}: ${valueToGet}`;

    } else {

        document.getElementById("MapResult").textContent = "Такого значения и не было!";

    }

});

document.getElementById("MAPclear").addEventListener("click", function (event) {

    event.preventDefault();

    mapBlock.clear();

    document.getElementById("MapResult").textContent = "Список полностью очищен!";

});

document.getElementById("MAPgetAll").addEventListener("click", function (event) {

    event.preventDefault();

    let result = "";

    for (let [k, v] of mapBlock) {

        result += `${k}: ${v},  `;

    }

    document.getElementById("MapResult").textContent = result;

});

document.getElementById("CreateMass").addEventListener("click", function (event) {

    event.preventDefault();

    const mass = Array.from(mapBlock);

    const result = mass.map(([key, value]) => `${key} = ${value} `);

    document.getElementById("MapResult").textContent = result;

});

document.getElementById("MAPLength").addEventListener("click", function (event) {

    event.preventDefault();

    const mass = Array.from(mapBlock);

    const result = mass.map(([key, value]) => `${value.length} `);

    document.getElementById("MapResult").textContent = result;

});

//Объект SET

let setBlock = new Set();

document.getElementById("SETadd").addEventListener("click", function (event) {

    event.preventDefault();

    setBlock.add(document.getElementById("valueWork").value);

    document.getElementById("SETResult").textContent = "Значение добавилось успешно!";

});

document.getElementById("SETcheck").addEventListener("click", function (event) {

    event.preventDefault();

    if (!setBlock.has(document.getElementById("valueWork").value)) {

        document.getElementById("SETResult").textContent = "Такого значения в множестве нет((";

    } else {

        document.getElementById("SETResult").textContent = "Такое значение в множестве есть!";

    }

});

document.getElementById("SETdelete").addEventListener("click", function (event) {

    event.preventDefault();

    if (setBlock.has(document.getElementById("valueWork").value)) {

        setBlock.delete(document.getElementById("valueWork").value);

        document.getElementById("SETResult").textContent = "Значение удалено успешно!";

    } else {

        document.getElementById("SETResult").textContent = "Такого значения и не было!";

    }

});

document.getElementById("SETclear").addEventListener("click", function (event) {

    event.preventDefault();

    setBlock.clear();

    document.getElementById("SETResult").textContent = "Список полностью очищен!";

});

document.getElementById("SETgetAll").addEventListener("click", function (event) {

    event.preventDefault();

    let result = "";

    setBlock.forEach((value, valueAgain, set) => {

        result += `${value}\n`;

    });

    document.getElementById("SETResult").textContent = result;

});

document.getElementById("SETmulty").addEventListener("click", function (event) {

    event.preventDefault();

    const massSet = Array.from(setBlock);

    const massSetNumbers = massSet.filter((item) => Number.isInteger(parseInt(item)));

    document.getElementById("SETResult").textContent = massSetNumbers;

});

document.getElementById("SETSumm").addEventListener("click", function (event) {

    event.preventDefault();

    const massSet = Array.from(setBlock)

        .filter((item) => Number.isInteger(parseInt(item)))

        .map((item) => parseInt(item));

    const resultSumm = massSet.reduce((sum, current) => sum + current, 0);

    document.getElementById("SETResult").textContent = resultSumm;

});

//Объект Date

const date = new Date();

document.getElementById("time").textContent = date.toLocaleString("ru-RU", {

    day: "2-digit",

    month: "2-digit",

    year: "2-digit",

    hour: "2-digit",

    minute: "2-digit",

});

// Дескрипторы

let descripterTask = {};

document.getElementById("DescriptorAdd").addEventListener("click", function (event) {

    event.preventDefault();

    const propertyToUse = document.getElementById("DescriptorPropertyInput").value;

    const valueToUse = document.getElementById("DescriptorValueInput").value;

    Object.defineProperty(descripterTask, propertyToUse, {

        value: valueToUse,

        writable: true,

        enumerable: true,

        configurable: true,

    });

    document.getElementById("DescriptorResult").textContent = "Успешно! Свойство записано/перезаписано!";

});

document.getElementById("DescriptorChange").addEventListener("click", function (event) {

    event.preventDefault();

    const propertyToUse = document.getElementById("DescriptorPropertyInput").value;

    const valueToUse = document.getElementById("DescriptorValueInput").value;

    descripterTask[propertyToUse] = valueToUse;

    document.getElementById("DescriptorResult").textContent = "Успешно! Свойство переназначено!";

});

document.getElementById("DescriptorWritable").addEventListener("click", function (event) {

    event.preventDefault();

    const propertyToUse = document.getElementById("DescriptorPropertyOtherActionInput").value;

    Object.defineProperty(descripterTask, propertyToUse, {

        writable: false,

    });

    document.getElementById("DescriptorResult").textContent = "Успешно! Теперь нельзя перезаписать значение!";

});

document.getElementById("DescriptorEnumerable").addEventListener("click", function (event) {

    event.preventDefault();

    const propertyToUse = document.getElementById("DescriptorPropertyOtherActionInput").value;

    Object.defineProperty(descripterTask, propertyToUse, {

        enumerable: false,

    });

    document.getElementById("DescriptorResult").textContent = "Успешно! Теперь свойство не видно при перечислении!";

});

document.getElementById("DescriptorConfigurate").addEventListener("click", function (event) {

    event.preventDefault();

    const propertyToUse = document.getElementById("DescriptorPropertyOtherActionInput").value;

    Object.defineProperty(descripterTask, propertyToUse, {

        configurable: false,

    });

    document.getElementById("DescriptorResult").textContent = "Успешно! Теперь свойство нельзя перезаписать даже в конфигураторе!";

});

document.getElementById("DescriptorShowAll").addEventListener("click", function (event) {

    event.preventDefault();

    let resultDescriptor = "";

    for (let key in descripterTask) {

        resultDescriptor += `${key}: ${descripterTask[key]}\n`;

    }

    document.getElementById("DescriptorResult").textContent = resultDescriptor;

});

//Прокси

const target = {

    watch: "Apple watch series 10",

    cost: 600,

};

const handler = {

    get: function (target, prop) {

        document.getElementById("ProxyResult").textContent = `Название: ${prop}. Позиция: ${target[prop]}`;

    },

    set: function (target, prop, value) {

        document.getElementById("ProxyResult").textContent = `Перезаписано свойство ${prop} со значение ${value}`;

        target[prop] = value;

    },

};

const proxy = new Proxy(target, handler);

document.getElementById("ProxyGetButton").addEventListener("click", function (event) {

    event.preventDefault();

    proxy.watch;

});

document.getElementById("ProxyActionValueButton").addEventListener("click", function (event) {

    event.preventDefault();

    proxy.cost = document.getElementById("ProxyActionValueInput").value;

});

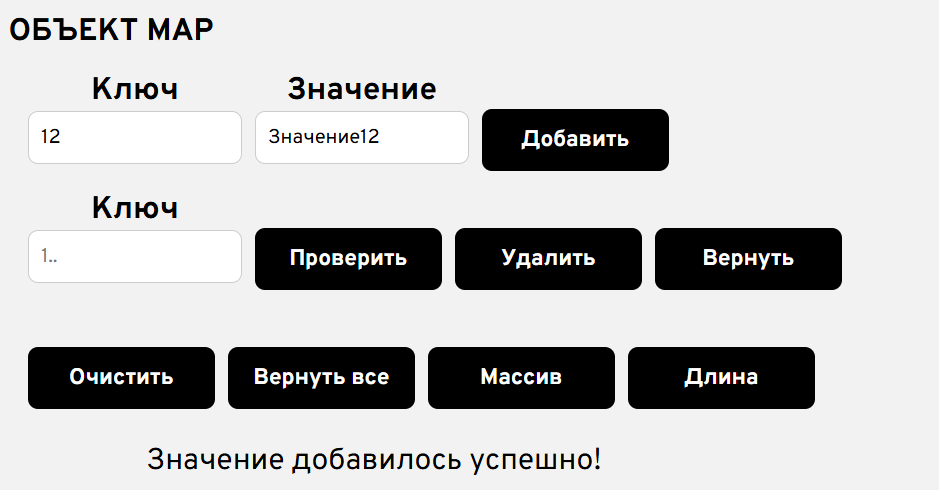


Рисунок 6. Добавление нового значения в объект Map.



Рисунок 7. Вывод всех значений объекта Map.

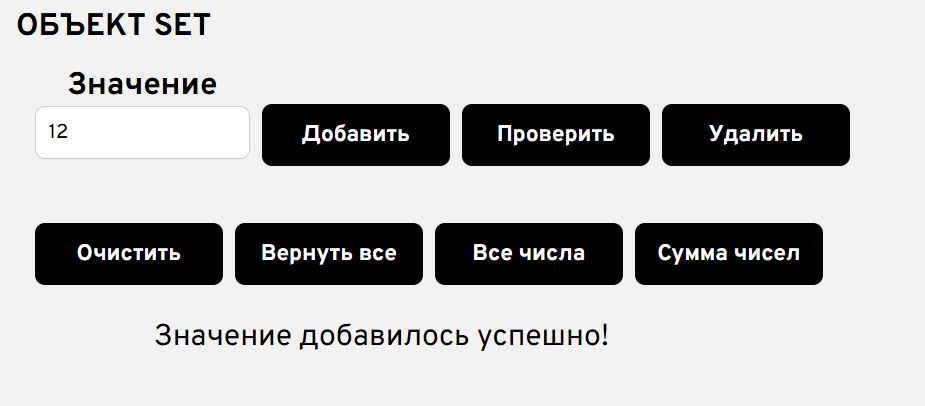


Рисунок 8. Добавление нового значения в объект Set.

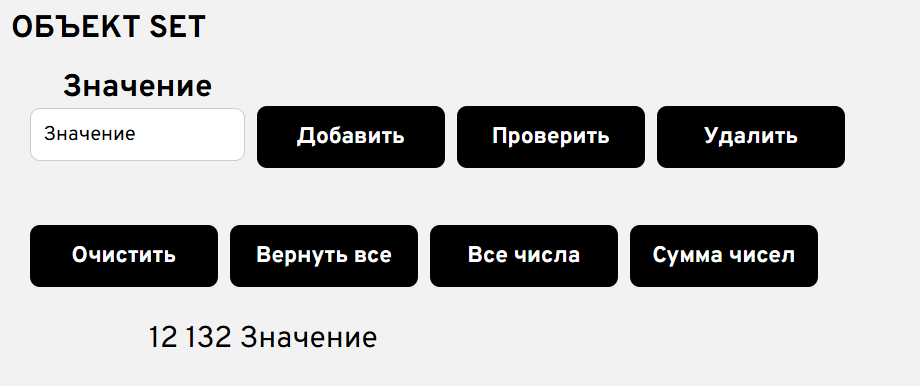


Рисунок 9. Вывод всех значений объекта Set.

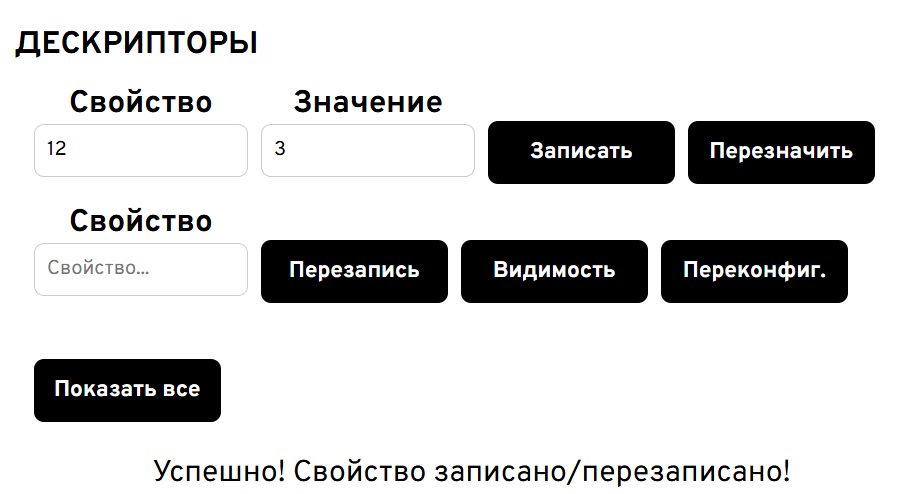


Рисунок 10. Добавление нового значения в объект.

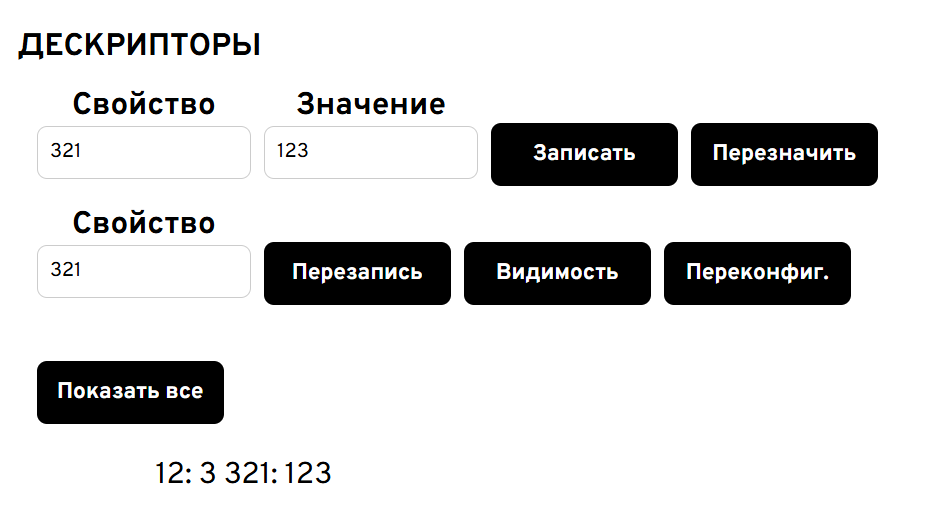


Рисунок 11. Вывод всех значений объекта.

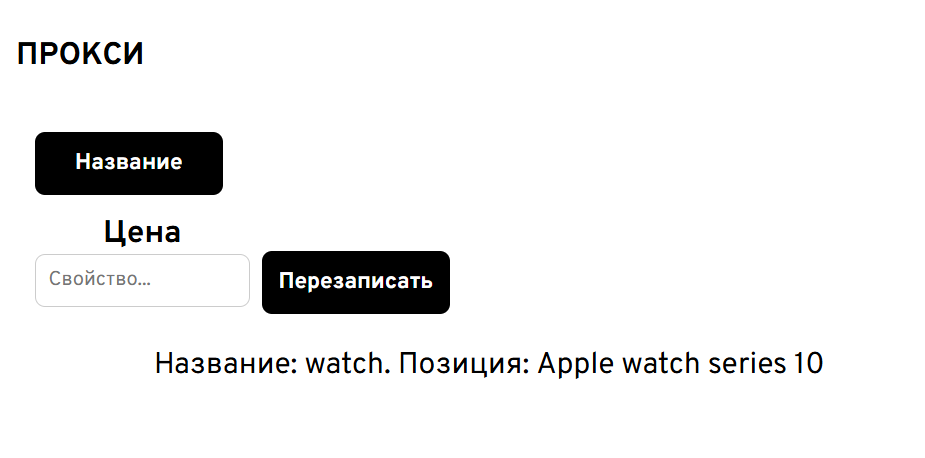


Рисунок 12. Вывод методом get.

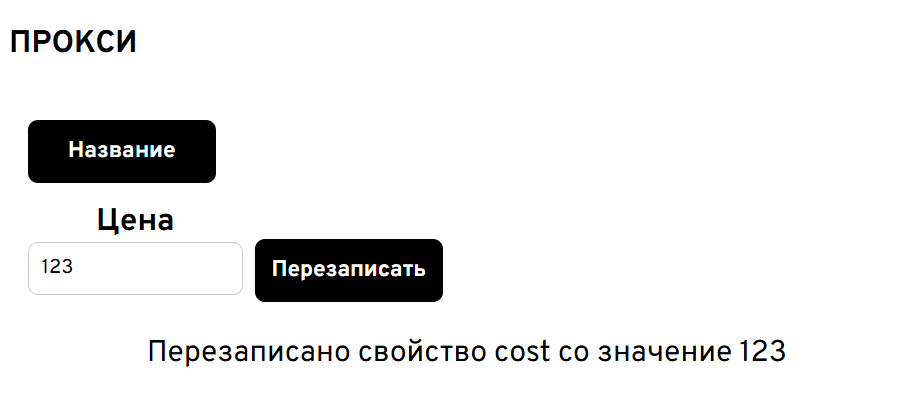


Рисунок 13. Изменение значения методом set.

ЗАДАНИЕ 3 (вариант 12).

Создать одностраничный сайт на тему «Сайт часов». На сайте должны осуществляться следующие действия: вычисление факториала числа (написать рекурсивную функцию, которая вычисляет факториал числа), набор функций, замкнутых на 1 переменную.

ВЫПОЛНЕНИЕ РАБОТЫ:

Выполнил данное задание, вычислив факториал числа при помощи рекурсивной функции и написав функцию, замкнутую на одну переменную.

Листинг index.html:

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <link rel="preconnect" href="https://fonts.googleapis.com">

    <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>

    <link href="https://fonts.googleapis.com/css2?family=Overpass:ital,wght@0,100..900;1,100..900&display=swap" rel="stylesheet">

    <title>Clock Shop</title>

    <link rel="stylesheet" href="style.css">

</head>

<body>

    <header>

        <span class="time" id="time"></span>

        <div class="leftText">

            <h1> ДИЗАЙНЕРСКИЙ ЧАСЫ </h1>

            <p> ПОЛНОСТЬЮ РУЧНОЕ ИЗГОТОВЛЕНИЕ </p>

            <button class="buyHeader buttonToChange openModel" data-price="900000"> КУПИТЬ </button>

            <dialog class="dialogMain">

                <form class="orderForm">

                    <h2>Оформление заказа</h2>

                    <p class="hidden" id="validSuccess"></p>

                    <p class="hidden" id="validError"></p>

                    <p><label for="name">Имя:</label>

                    <input type="text" id="name" name="name" placeholder="Имя" required>

                    <p class="hidden text-danger" id="nameP"></p></p>

                    <p><label for="secondName">Фамилия:</label>

                    <input type="text" id="secondName" name="secondName" placeholder="Фамилия" required>

                    <p class="hidden text-danger" id="secondNameP"></p></p>

                    <p><label for="thirdName">Отчество:</label>

                    <input type="text" id="thirdName" name="thirdName" placeholder="Отчество">

                    <p class="hidden text-danger" id="thirdNameP"></p></p>

                    <p><label for="address">Адрес доставки:</label>

                    <input type="text" id="address" name="address" placeholder="Ул. Пушкино, д. 2, кв. 1" required></p>

                    <p><label for="quantity">Количество:</label>

                    <input type="number" id="quantity" name="quantity" placeholder="1" min="1" max="10" required></p>

                    <p><label for="promo">Промокод:</label>

                    <input type="text" id="promo" placeholder="\*Промокод\*" name="promo"></p>

                    <p class="priceHeader"></p>

                    <input type="submit" value="Заказать!">

                </form>

                <button id="closeModel" class="closeModel">&#x2715 </button>

            </dialog>

        </div>

        <div class="rightText">

            <p> JACOB & CO <br>EPIC X TOURBILON BAGUETTE <br>900.000$</p>

            <img src="imgs/mainPhotoWatches.png">

        </div>

    </header>

    <main>

        <div class="textInMain">

            <p>ХИТЫ</p>

            <p>ПОСЛЕДНИЕ ПОСТУПЛЕНИЯ</p>

        </div>

        <div class="productsList">

            <div class="productUnit">

                <img src="imgs/appleulta.png">

                <p>Apple watch ulta 2</p>

                <p>700$</p>

                <button class="buttonUnit buttonToChange openModel" data-price="700"> Купить </button>

            </div>

            <div class="productUnit">

                <img src="imgs/rolexDayDate36.png">

                <p>Rolex Day-Date 36</p>

                <p>117.250$</p>

                <button class="buttonUnit buttonToChange openModel" data-price="117250"> Купить </button>

            </div>

            <div class="productUnit">

                <img src="imgs/KingSeiko.png">

                <p>King Seiko Automatic</p>

                <p>2.000$</p>

                <button class="buttonUnit buttonToChange openModel" data-price="2000"> Купить </button>

            </div>

            <div class="productUnit">

                <img src="imgs/PatekPhilippe.png">

                <p>Paket Philippe 5270/1R</p>

                <p>230.000$</p>

                <button class="buttonUnit buttonToChange openModel" data-price="230000"> Купить </button>

            </div>

            <div class="productUnit">

                <img src="imgs/rolexDateJust31.png">

                <p>Rolex DateJust 31</p>

                <p>51.900$</p>

                <button class="buttonUnit buttonToChange openModel" data-price="51900"> Купить </button>

            </div>

            <div class="productUnit">

                <img src="imgs/PatekPhilippeBlackWhite.png">

                <p>Patek Philippe 6300/400G</p>

                <p>9.000.000$</p>

                <button class="buttonUnit buttonToChange openModel" data-price="9000000"> Купить </button>

            </div>

            <div class="productUnit">

                <img src="imgs/appleseries10.png">

                <p>Apple watch series 10</p>

                <p>450$</p>

                <button class="buttonUnit buttonToChange openModel" data-price="450"> Купить </button>

            </div>

            <div class="productUnit">

                <img src="imgs/Jacob&Co.png">

                <p>Jacob&Co Palatial Classic</p>

                <p>16.500$</p>

                <button class="buttonUnit buttonToChange openModel" data-price="16500"> Купить </button>

            </div>

        </div>

        <div class="addModule">

            <div class="leftSide">

                <div class="textInMain addText">

                    <p>ФУНКЦИИ И РЕКУРСИИ. ВАРИАНТ 12. ЗАДАНИЕ 1</p>

                </div>

                <form class="decorateBlock" id="recursiveFunction">

                    <p><label for="RecursiveValue">Число</label>

                    <input type="number" required for="RecursiveValue" placeholder="1.." min="1" id="RecursiveValueInput"></p>

                    <button id="RecursiveCalc"> Вычислить! </button>

                </form>

                <p id="recursiveResult" class="messageText"></p>

            </div>

            <div class="rightSide">

                <div class="textInMain addText">

                    <p>ЗАМКНУТЫЕ ФУНКЦИИ</p>

                </div>

                <form class="decorateBlock">

                    <p><label for="closedNum">Число</label>

                    <input type="text" required for="closedNum" placeholder="1.." id="closedNumInput"></p>

                    <button id="NumWrite"> Прочитать </button>

                </form>

                <div class="decorateBlock">

                    <p><label for="numToOperation">Число</label>

                    <input type="text" required for="numToOperation" placeholder="1.." id="numToOperationInput"></p>

                    <button id="multyNum"> Умножить </button>

                    <button id="summNum"> Сложить </button>

                    <button id="divideNum"> Разделить </button>

                    <button id="remainderNum"> Остаток </button>

                </div>

                <p id="closeResult" class="messageText"></p>

            </div>

        </div>

    </main>

    <script src="script.js"></script>

</body>

</html>

Листинг script.js:

// Смена цвета кнопки при наведении

const button = document.body.querySelectorAll(".buttonToChange");

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

    button[i].addEventListener("mouseover", function () {

        button[i].classList.add("changebackground");

    });

    button[i].addEventListener("mouseout", function () {

        button[i].classList.remove("changebackground");

    });

}

// Модальное окно

const dialog = document.querySelector(".dialogMain");

const html = document.querySelector("html");

const openModels = document.querySelectorAll(".openModel");

const priceHeader = document.querySelector(".priceHeader");

let price;

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

    openModels[i].addEventListener("click", function () {

        dialog.showModal();

        html.classList.add("changehtml");

        if (openModels[i].getAttribute("data-price") > 0) {

            price = parseInt(openModels[i].getAttribute("data-price"));

        }

        priceHeader.textContent = `Цена: ${price.toLocaleString()}$`;

    });

}

document.querySelector("#closeModel").onclick = function () {

    dialog.close();

    html.classList.remove("changehtml");

};

//Изменения цены

const formElement = document.querySelector(".orderForm");

const quantity = document.querySelector("#quantity");

const promo = document.querySelector("#promo");

const promoList = {

    "10procentSaleWOW": 0.9,

    "15yearShop": 0.85,

};

function updatePrice() {

    let quantityVal = 1;

    if (quantity.value > 1) {

        quantityVal = quantity.value;

    }

    let discount = promo.value in promoList ? promoList[promo.value] : 1;

    priceHeader.textContent = `Цена: ${createPrice(price, quantityVal, discount).toLocaleString()}$ (${quantityVal} шт, скидка ${Math.round((1 - discount) \* 100)}%)`;

}

function createPrice(priceFunc, quantityFunc, discountFunc) {

    let finalPrice = priceFunc \* quantityFunc \* discountFunc;

    return finalPrice;

}

quantity.addEventListener("input", updatePrice);

promo.addEventListener("input", updatePrice);

// Отправка формы

formElement.addEventListener("submit", (event) => {

    event.preventDefault();

    const formData = new FormData(formElement);

    const name = formData.get("name");

    const surname = formData.get("secondName");

    const thirdName = formData.get("thirdName");

    const address = formData.get("address");

    if (!checker(name, "nameP") || !checker(surname, "secondNameP") || !checker(thirdName, "thirdNameP")) {

        document.getElementById("validError").textContent = "Ошибка! Некоторый данный были введены неправильно";

        document.getElementById("validError").classList.remove("hidden");

        document.getElementById("validSuccess").classList.add("hidden");

    } else {

        document.getElementById("validSuccess").textContent = `Спасибо, ${name} ${surname} ${thirdName}!\nВаш заказ на адрес - ${address} в количестве ${quantity.value}шт будет исполнен в ближайшее время!`;

        document.getElementById("validSuccess").classList.remove("hidden");

        document.getElementById("validError").classList.add("hidden");

    }

});

function checker(str, idEl) {

    const regexStr = /^[A-Za-zА-Яа-яЁё]+$/;

    if (!regexStr.test(str)) {

        document.getElementById(idEl).textContent = "Не должно быть цифр!";

        document.getElementById(idEl).classList.remove("hidden");

        return false;

    }

    if (str.length > 50 || str.length < 3) {

        document.getElementById(idEl).textContent = "От 3 до 50 символов!";

        document.getElementById(idEl).classList.remove("hidden");

        return false;

    }

    document.getElementById(idEl).textContent = "";

    return true;

}

// Рекурсия

function factorial(num) {

    if (num !== 0) {

        return num \* factorial(num - 1);

    }

    return 1;

}

document.getElementById("RecursiveCalc").addEventListener("click", function (event) {

    event.preventDefault();

    numberToCalc = parseInt(document.getElementById("RecursiveValueInput").value);

    document.getElementById("recursiveResult").textContent = `Вычисленное число - ${factorial(numberToCalc)}`;

});

// Замкнутые функции

let miniCalculator;

function operationCalc(numToOperate) {

    return {

        get value() {

            return numToOperate;

        },

        multipli: function (operator) {

            numToOperate \*= operator;

        },

        summ: function (operator) {

            numToOperate += operator;

        },

        divide: function (operator) {

            numToOperate /= operator;

        },

        remaind: function (operator) {

            numToOperate %= operator;

        },

    };

}

document.getElementById("NumWrite").addEventListener("click", function (event) {

    event.preventDefault();

    miniCalculator = operationCalc(parseInt(document.getElementById("closedNumInput").value));

    document.getElementById("closeResult").textContent = "Число успешно записано!";

});

document.getElementById("multyNum").addEventListener("click", function (event) {

    event.preventDefault();

    miniCalculator.multipli(parseInt(document.getElementById("numToOperationInput").value));

    document.getElementById("closeResult").textContent = `Удача! Вы умножили число! Полученное число - ${miniCalculator.value}`;

});

document.getElementById("summNum").addEventListener("click", function (event) {

    event.preventDefault();

    miniCalculator.summ(parseInt(document.getElementById("numToOperationInput").value));

    document.getElementById("closeResult").textContent = `Удача! Вы сложили число! Полученное число - ${miniCalculator.value}`;

});

document.getElementById("divideNum").addEventListener("click", function (event) {

    event.preventDefault();

    miniCalculator.divide(parseInt(document.getElementById("numToOperationInput").value));

    document.getElementById("closeResult").textContent = `Удача! Вы раздлелили число! Полученное число - ${miniCalculator.value}`;

});

document.getElementById("remainderNum").addEventListener("click", function (event) {

    event.preventDefault();

    miniCalculator.remaind(parseInt(document.getElementById("numToOperationInput").value));

    document.getElementById("closeResult").textContent = `Удача! Вы нашли остаток от числа! Полученное число - ${miniCalculator.value}`;

});

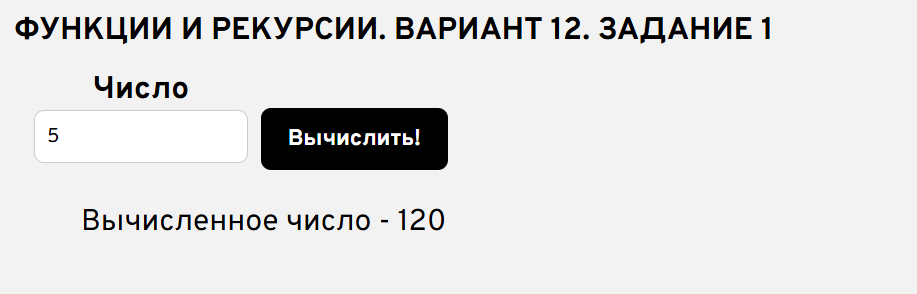


Рисунок 14. Вычисление факториала.

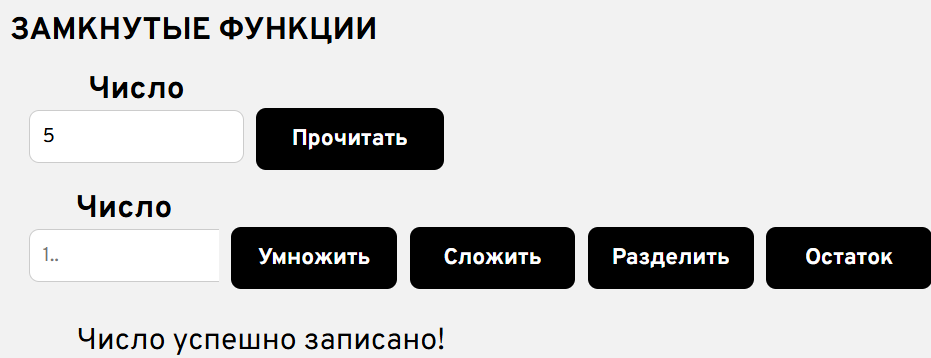


Рисунок 15. Прочитать число.

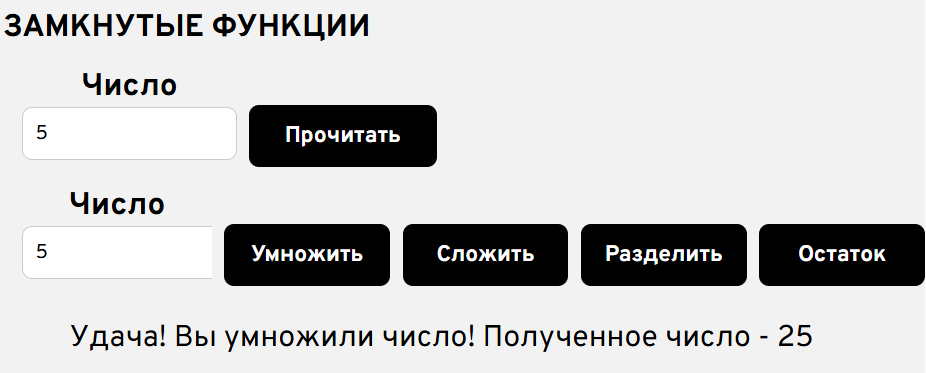


Рисунок 16. Умножение прочитанного числа.

ВЫВОД К РАБОТЕ:

Я научился вычислять длину строки Юникода, совершать поиск подстроки в строке с использованием регулярных выражений, проводить валидацию введенных пользователем данных.

Решил задачи с использованием встроенных объектов JavaScript (Map, Set, Date), с помощью методов для обработки списков (.map, .filter, .reduce), научился созданию сложных объектов (дескрипторы свойств, Proxy, Reflect).

Решил задачи на следующие темы: функции и рекурсию, создание набора функций, замкнутых на общую переменную.