**3. Работа с элементами DOM средствами JavaScript**

**Цель:** Ознакомление с принципами работы с DOM-деревом средствами JavaScript. Научиться создавать и управлять элементами DOM-дерева. Изучить организацию работы с событиями в JavaScript. Научиться обрабатывать пользовательские сценарии с помощью обработчиков событий в JavaScript.

Для продолжения работы над интерактивными элементами на странице создан файл JavaScript под названием main.js. После создания файл был подключен к разрабатываемой HTML-странице. Теперь отдельный файл main.js используется для реализации дополнительной функциональности и логики для веб-страницы. Это помогает структурировать и организовать код, делая его более модульным и удобным для поддержки.

Листинг кода:

document.addEventListener("DOMContentLoaded", () => {

    const modal = document.querySelector(".subscribe");

    const modalWindow = document.querySelector("#modal");

    const openButton = modal.querySelector("#openModal");

    const closeButton = modalWindow.querySelector(".close-button");

    const inputEmail = modal.querySelector(".subscribe\_\_input");

    const modelOverlay = modalWindow.querySelector('.modal\_\_overlay');

    const modalForm = modalWindow.querySelector("#modalForm");

    openButton.addEventListener("click", onOpenButtonClick);

    modalForm.addEventListener("submit", onModalFormSubmit);

    function onOpenButtonClick() {

        const email = inputEmail.value;

        if (email === "") {

            showErrorEmptyString();

        } else if (isValidEmail(email)) {

            openModalWindow();

        } else {

            showErrorIsNotValidEmail();

        }

    }

    function onCloseButtonClick() {

        closeModalWindow();

    }

    function onWindowOverlayClick(evt) {

        if (evt.target == modelOverlay) {

            closeModalWindow();

        }

    }

**Контрольные вопросы:**

1. Что такое DOM?

DOM (Document Object Model) — это объектная модель, представляющая HTML или XML-документ в виде дерева.

1. Что такое цель события или целевой элемент (event.target)?

Цель события или целевой элемент (event.target) — элемент, на котором изначально вызвано событие.

1. Что такое текущая цель события (event.currentTarget)?

Текущая цель события (event.currentTarget) – элемент, к которому привязан текущий обработчик события.

1. В чем разница между ключевыми словами «var», «let» и «const»?

var: функциональная область видимости, можно переобъявлять. let: блочная область видимости, нельзя переобъявлять. const: блочная область видимости, нельзя переопределять и изменять.

1. Что такое «стрелочные функции» (Arrow Functions)?

Стрелочные функции (Arrow Functions) — компактный синтаксис функций (`() => {}`), не имеют своего `this`.

1. Для чего необходимо удалять неиспользуемые обработчики событий?

Удаление неиспользуемых обработчиков событий нужно для предотвращения утечек памяти и повышения производительности.

1. Что такое область видимости?

Область видимости — контекст, в котором можно обращаться к переменным и функциям. Бывает глобальная, функциональная и блочная.

1. Что такое анонимная функция?

Анонимная функция — функция без имени, часто используется как аргумент.

1. Что такое callback-функция? Примеры использования callback-функции.

Callback-функция — функция, передаваемая в другую функцию как аргумент и вызываемая позже.

Пример:

function greet(name, callback) {

console.log('Hello ' + name);

callback();

}

greet('Alice', function() {

console.log('This is a callback');

});

1. Зачем нужен метод preventDefault?

preventDefault() — метод, предотвращающий стандартное поведение события.

Пример:

event.preventDefault();

1. Каким образом можно добавить/удалить css-класс у DOM-элемента?

Добавить/удалить CSS-класс у DOM-элемента:

element.classList.add('my-class');

element.classList.remove('my-class');

1. Как выполняется поиск элементов по DOM-дереву?

Поиск элементов по DOM-дереву:

document.getElementById('id');

document.getElementsByClassName('class');

document.getElementsByTagName('tag');

document.querySelector('.class');

document.querySelectorAll('.class');