1. Створити функцію validateFormFields(), яка перевіряє, чи всі обов’язкові поля форми заповнені (наприклад, ім’я, email). Якщо ні — вивести повідомлення та не дозволити надсилання форми (preventDefault()).

2. Створити модуль sortModule.js, який експортує функцію sortByKey(array, key) для сортування масиву об’єктів. В іншому файлі імпортувати її, викликати з тестовим масивом і вивести результат на сторінку.

3. Створити функцію setupClickDelegation(listId), яка додає один обробник подій click на список ul з заданим id. При натисканні на будь-який li, у консоль виводиться його текстовий вміст.

4. Створити функцію toggleElementStyle(element), яка додає елементу нові стилі при наведенні миші (mouseenter) і повертає початкові стилі при відведенні (mouseleave). Наприклад, змінити фон на синій, а текст — на білий.

5. Створити функцію handleLocalStorageActions(), яка додає слухачі подій на дві кнопки з класами .save-btn та .remove-btn. При натисканні .save-btn зберігає об’єкт {name: 'Боб', age: 25} у localStorage під ключем 'user'; При натисканні .remove-btn видаляє цей ключ із localStorage.

6. Створити функцію checkForSpam(message), яка приймає рядок (параметр message), перевіряє його на вміст заборонених слів spam і sale, і повертає результат перевірки.

7. Створити функцію mockServerRequest(), яка повертає Promise, що виконується через 2 секунди з результатом 'Дані успішно отримані'. Вивести результат у консоль після завершення запиту.

8. Створити модуль geometry.js, який експортує функцію getRectangleArea(width, height) для обчислення площі прямокутника. В іншому файлі імпортувати функцію та вивести результат у консоль.

9. Створити функцію changeElementContent(), яка при натисканні кнопки #changeBtn змінює текст і колір елемента #target.

10. Створити функцію processArray(arr, callback), яка приймає масив чисел та функцію-колбек. Використати метод map, передаючи в нього цей колбек. Вивести оброблений масив у консоль.

11. Створити функцію filterByCategory(items, category), яка приймає масив об’єктів з полем category і повертає новий масив, що містить лише об’єкти відповідної категорії.

12. Створити функцію toggleDarkMode(), яка перемикає тему сторінки між світлою та темною. При натисканні кнопки з id="themeBtn" функція має додавати або видаляти клас dark-mode до body.

13. Створити функцію displayCurrentDateTime(), яка отримує поточну дату і час за допомогою об’єкта Date та виводить їх у форматі: 'Сьогодні: 10.04.2025, Час: 14:30'.

14. Створити функцію saveThemePreference(theme), яка зберігає тему інтерфейсу у localStorage. При завантаженні сторінки інша функція applySavedTheme() має застосовувати збережену тему до сторінки.

15. Створити функцію fetchDataFromAPI(url), яка отримує дані з вказаного API за допомогою fetch, обробляє JSON-відповідь та виводить дані у вигляді списку на сторінці.

16. Створити функцію generateProductCard(product), яка приймає об’єкт {name, price, description} і повертає HTML-рядок з розміткою картки товару. Результат виводиться на сторінку в елемент #product-list.

17. Написати функцію checkVariableType(value), яка перевіряє тип переданої змінної та виводить повідомлення у форматі: Тип змінної — string (або number, boolean тощо).

18. Написати функцію createList, яка приймає масив рядків (наприклад: ['яблуко', 'банан', 'груша']) і створює список <ul> з елементами <li>, що містять значення з масиву. Додати цей список до тіла сторінки.

19. Створити функцію liveCharacterCount(inputId, outputId), яка відслідковує подію input на текстовому полі і виводить кількість введених символів у відповідному елементі.

20. Створити функцію processUserInput(value), яка намагається перетворити введене значення в число і обчислити його квадрат. Якщо значення не є числом, вивести повідомлення про помилку через try...catch.

21. Створити функцію saveAndReadJSON(obj), яка зберігає переданий об’єкт у localStorage у форматі JSON під ключем "userData", а потім зчитує цей об’єкт і відображає його на сторінці у вигляді списку.

22. Створити функцію createListFromArray(array), яка приймає масив рядків, наприклад ["HTML", "CSS", "JavaScript"], та створює список <ul> з відповідними <li>, додає його до DOM-дерева сторінки.

23. Створити функцію changeButtonTextOnInput(inputId, buttonId), яка змінює текст кнопки залежно від поточного значення в полі введення.

24. Створити функцію changeBackgroundByKey(), яка змінює фон сторінки при натисканні клавіш: 1 → червоний, 2 → зелений, 3 → синій.

25. Створити клас User, який має властивості name та email, що ініціалізуються через конструктор. Додати метод getInfo().

26. Створити функцію calculateSum(...numbers), яка приймає довільну кількість чисел за допомогою rest параметрів та повертає їх суму.

27. Написати функцію showMessageByInput(value), яка виводить різні повідомлення в залежності від значення введеного в поле input.

28. Написати функцію generateProductCard(product), яка приймає об’єкт товару (наприклад {name: "Смартфон", price: 12000}) та повертає HTML-рядок картки товару.

29. Створити асинхронну функцію loadDataWithAsync(url), яка виконує запит до API за допомогою fetch і виводить отримані дані. У разі помилки — повідомлення про помилку.

30. Написати функцію handleLocalStorage, яка додає або видаляє ключ "data" у localStorage в залежності від події кліку на кнопку з класом .save-btn або .remove-btn.