Практические задания

Давайте студентам задания по одному, чтобы они могли сосредоточиться.

Задание 1: Поиск и выбор элементов

**Цель:** Научиться находить нужные элементы на странице с помощью различных методов.

**Задачи:**

1. Найдите параграф с id="first-paragraph" и выведите его в консоль.
2. Найдите все параграфы с классом content-text и выведите их в консоль. Посмотрите, что получилось (это будет NodeList или HTMLCollection).
3. Найдите элемент ссылки (<a>) по её id.
4. Выведите в консоль текст (textContent) найденной ссылки.

**Подсказки:**

* Используйте document.getElementById('id').
* Используйте document.querySelectorAll('.class').
* Чтобы получить текст элемента, используйте свойство .textContent.

Задание 2: Изменение содержимого и атрибутов

**Цель:** Научиться менять текст, HTML-содержимое и атрибуты элементов.

**Задачи:**

1. Найдите заголовок с id="change-me". Измените его текст на "Текст изменен с помощью JavaScript!".
2. Найдите картинку с id="image". Измените её атрибут src на новую картинку. Например: https://picsum.photos/200.
3. Найдите ссылку из Задания 1 (id="link") и измените её атрибут href, чтобы она вела на https://developer.mozilla.org/ru/docs/Web/API/Document\_Object\_Model.

**Подсказки:**

* Для изменения текста: element.textContent = 'Новый текст'.
* Для изменения атрибутов: element.src = 'новый\_путь' или element.href = 'новая\_ссылка'.

Задание 3: Работа со стилями

**Цель:** Научиться изменять CSS-стили элементов напрямую из JS.

**Задачи:**

1. Найдите блок с id="style-box".
2. Найдите кнопку с id="style-button".
3. Добавьте обработчик события click на кнопку.
4. При клике на кнопку, измените у блока style-box цвет фона на green, добавьте ему тень (box-shadow: 2px 2px 5px grey;) и увеличьте ширину до 200px.
5. **Бонусное задание:** Сделайте так, чтобы при повторном клике стили возвращались в исходное состояние (используйте classList.toggle с заранее подготовленным классом в CSS).

**Подсказки:**

* Стили меняются через свойство style: element.style.backgroundColor = 'green'.
* Обратите внимание, что CSS-свойства с дефисом (like box-shadow) в JS пишутся в camelCase (boxShadow).
* Для событий используйте button.addEventListener('click', function() { ... });.

Задание 4: Обработка событий

**Цель:** Научиться реагировать на действия пользователя.

**Задачи:**

1. Найдите кнопку id="event-button". При клике на неё должен появляться alert с текстом "Кнопка была нажата!".
2. Найдите поле ввода id="text-input" и элемент span с id="output-text".
3. Добавьте на поле ввода обработчик события input.
4. Когда пользователь печатает что-то в поле ввода, текст из поля должен в реальном времени отображаться внутри span#output-text.

**Подсказки:**

* Событие для отслеживания ввода в реальном времени — 'input'.
* Значение поля ввода можно получить через inputElement.value.

Задание 5: Создание и удаление элементов

**Цель:** Научиться динамически добавлять и удалять элементы со страницы.

**Задачи:**

1. Найдите список ul с id="item-list" и кнопку id="add-item-button".
2. Добавьте обработчик события click на кнопку.
3. При каждом клике должен создаваться новый элемент <li>.
4. Устанавливайте для нового <li> текст, например, "Новый элемент".
5. Добавляйте созданный <li> в конец списка ul.
6. **Бонусное задание:** Сделайте так, чтобы при клике на любой элемент списка (<li>) он удалялся со страницы. (Подсказка: можно повесить один обработчик на <ul> и использовать event.target).

**Подсказки:**

* Для создания элемента: document.createElement('li').
* Для добавления элемента в конец родителя: parentElement.appendChild(newElement).
* Для удаления элемента: element.remove().