**Вопросы 💎**

1. Что такое DOM? - это внутреннее представление веб-страницы в виде дерева объектов, которое браузер использует для отображения страницы и управления ее содержимым. Независимый от платформы и языка программирования формат, который позволяет программам получать доступ к содержимому HTML-документов и изменять их структуру, содержимое и дизайн.

Примеры действий, которые можно выполнить с помощью DOM JS: добавление и удаление элементов, изменение содержимого элементов, изменение стилей и атрибутов элементов, добавление и удаление обработчиков событий и т. д.

1. Что такое BOM?

BOM (Browser Object Model) — – это объектная модель браузера, которая предоставляет доступ к компонентам браузера и дает возможность управлять ими программно.

Примеры действий, которые можно выполнить с помощью BOM JS: оперировать объектами BOM и работать с различными свойствами окна браузера, включая URL, заголовок, историю браузера, размеры окна, расположение мыши и т.д.

1. Каким методом можно получить элемент по его идентификатору в DOM?

HTML-элементу можно присвоить уникальное имя, или идентификатор, с помощью атрибута id. Обозначив элемент уникальным id можем воспользоваться DOM-методом

document.getElementById(). В качестве аргумента методу передается строка с идентификатором элемента (например, "myElement"). Метод возвращает найденный элемент или null, если элемент с указанным идентификатором не был найден.

1. Каким методом можно создать новый элемент в DOM?

Существует пять методов, которые позволяют добавлять элементы на веб-страницу относительно других элементов в DOM-дереве. Эти методы позволяют размещать элементы перед или после определенных узлов, внутри них или заменять существующие элементы.

1. **node.append(...nodes or strings)** - добавляет узлы или строки в конец элемента
2. **node.prepend(...nodes or strings)** - добавляет узлы или строки в начало элемента
3. **node.before(...nodes or strings)** - добавляет узлы или строки перед элементом
4. **node.after(...nodes or strings)** - добавляет узлы или строки после элемента
5. **node.replaceWith(...nodes or strings)** - заменяет элемент заданными узлами или строками
6. Каким методом можно добавить класс к элементу в DOM?

Метод add добавляет элементу класс. В качестве аргументов нужно передавать строку с именем класса (Можно так же передать несколько строк-аргументов, тогда добавится несколько классов) cat.classList.add('bright'); (cat.classList.add('black', 'white', 'fluffy');)

1. Каким методом можно изменить текстовое содержимое элемента в DOM?

1. Свойство innerHTML: это свойство позволяет установить или получить HTML содержимое элемента. Для изменения текста потребуется передать новое значение в качестве строки, например, element.innerHTML = "Новый текст";

2. Метод textContent или innerText: оба img.src = image.jpg; метода позволяют установить или получить текстовое содержимое элемента. Для изменения текста потребуется передать новое значение в качестве строки, например, element.innerText = "Новый текст";

3. Метод setAttribute: можно установить атрибут "value" для элементов формы, который является текстовым содержимым. Для изменения текста потребуется передать новое значение в качестве строки, например, element.setAttribute("value", "Новый текст");

1. Что такое объект события и какие у него могут быть свойства?

В JavaScript объект события представляет собой структуру данных, содержащую информацию о событии, которое произошло в браузере. Объект события создается автоматически при возникновении события и передается в обработчик события.

У объекта события в JavaScript могут быть различные свойства, которые предоставляют информацию о событии. Вот некоторые из наиболее распространенных свойств объекта события в JavaScript:

- Свойство innerHTML: это свойство позволяет установить или получить HTML содержимое элемента. Для изменения текста потребуется передать новое значение в качестве строки, например, element.innerHTML = "Новый текст";

Метод textContent или innerText: оба img.src = image.jpg; метода позволяют установить или получить текстовое содержимое элемента. Для изменения текста потребуется передать новое значение в качестве строки, например, element.innerText = "Новый текст";

Метод setAttribute: можно установить атрибут "value" для элементов формы, который является текстовым содержимым. Для изменения текста потребуется передать новое значение в качестве строки, например, element.setAttribute("value", "Новый текст");

type: Содержит строковое значение, указывающее тип события. Например, " Свойство innerHTML: это свойство позволяет установить или получить HTML содержимое элемента. Для изменения текста потребуется передать новое значение в качестве строки, например, element.innerHTML = "Новый текст";

1. Метод textContent или innerText: оба img.src = image.jpg; метода позволяют установить или получить текстовое содержимое элемента. Для изменения текста потребуется передать новое значение в качестве строки, например, element.innerText = "Новый текст";
2. Метод setAttribute: можно установить атрибут "value" для элементов формы, который является текстовым содержимым. Для изменения текста потребуется передать новое значение в качестве строки, например, element.setAttribute("value", "Новый текст");
3. click" для события клика мыши или "keydown" для события нажатия клавиши.
4. **target**: Содержит ссылку на элемент DOM, на котором произошло событие. Это может быть, например, ссылка на кнопку, которую пользователь кликнул.
5. **currentTarget**: Содержит ссылку на элемент DOM, который в данный момент обрабатывает событие. Это может отличаться от **target**, если событие всплывает или делегируется.
6. **eventPhase**: Указывает на текущую фазу обработки события. Может иметь одно из трех значений: **Event.CAPTURING\_PHASE** (фаза перехвата), **Event.AT\_TARGET** (фаза цели) или **Event.BUBBLING\_PHASE** (фаза всплытия).
7. timeStamp: Временная метка, показывающая время возникновения события в миллисекундах с момента загрузки документа.
8. keyCode или key: В случае событий клавиатуры, содержит информацию о нажатой клавише. keyCode предоставляет числовой код клавиши, а key предоставляет строковое значение.
9. Есть элемент: <input id= "input" value = "Привет">. Какой вызов поменяет значение в нём?

document.getElementById("input").value = "Новое значение";

1. Для чего используется запись: setAttribute(name, value)?

В JavaScript, метод setAttribute(name, value) используется для установки значения атрибута элемента. Этот метод позволяет программно изменять или добавлять атрибуты элемента HTML.

1. Как сделать переход на другую страницу при клике на кнопку (без <a href=...>, только средствами JavaScript)?

можно воспользоваться методом **window.location.href:**

var button = document.getElementById("myButton");

button.addEventListener("click", function() {

window.location.href = "новая\_страница.html";

});

1. Найдите ошибку в коде, которая мешает изменить цвет текста при нажатии на кнопку.

<body>

<h1 id="myElement">Hello, World!</h1>

<button onclick="changeColor()">Change Color</button>

<script>

function changeColor() {

let element = document.getElementByid('myElement'); // document.getElementById

element.style.color = 'red';

}

</script>

</body>

1. Найдите ошибку в коде, которая мешает добавить новый элемент списка при нажатии на кнопку.

<body>

<ul id="myList">

<li>Item 1</li>

<li>Item 2</li>

</ul>

<button onclick="addListItem()">Add Item</button>

<script>

function addListItem() {

let list = document.getElementById('myList');

let item = document.createElement('li');

item.innerHTML = 'New Item';

item.append(item); // list.append(item);

}

</script>

</body>

1. Найдите ошибку в коде, которая мешает удалить параграф при нажатии на кнопку.

<body>

<p class="myParagraph ">This is a paragraph.</p>

<button onclick="removeParagraph">Remove Paragraph</button> // <button onclick="removeParagraph()">Remove Paragraph</button>

<script>

function removeParagraph() {

let paragraph = document.getElementsByClassName('myParagraph'); // let paragraph = document.getElementsByClassName('myParagraph') [0];

paragraph.remove();

}

</script>

</body>

</html>

1. Найдите ошибку в коде, которая мешает добавить изображение при нажатии на кнопку.

<body>

<button onclick="showImage()">Показать изображение</button>

</body>

<script>

function showImage() {

const img = document.createElement('img');

img.src = image.jpg; // img.src = “image.jpg”;

document.body.append(img);

}

</script>

// </body>