**DOM: доступ к элементам, изменение, события**

**8.1. Что такое DOM?**

**DOM (Document Object Model)** — это объектное представление HTML-документа.  
Каждый элемент страницы (тег, текст, атрибут) становится **объектом**, с которым можно работать в JS.

Пример:

html

КопироватьРедактировать

<h1 id="title">Привет</h1>

→ В JS:

js

КопироватьРедактировать

document.getElementById("title") // доступ к заголовку

**8.2. Методы доступа к элементам**

| **Метод** | **Что делает** |
| --- | --- |
| getElementById(id) | Поиск по id |
| getElementsByClassName() | Поиск по классу (HTMLCollection) |
| getElementsByTagName() | Поиск по тегу |
| querySelector() | Один элемент по CSS-селектору |
| querySelectorAll() | Все элементы по CSS-селектору |

js

КопироватьРедактировать

const el = document.querySelector(".btn");

const title = document.getElementById("title");

**8.3. Изменение содержимого**

js

КопироватьРедактировать

title.textContent = "Новый заголовок"; // только текст

title.innerHTML = "<em>Курс</em>"; // с HTML-разметкой

**8.4. Работа со стилями**

js

КопироватьРедактировать

title.style.color = "red";

title.style.fontSize = "28px";

**8.5. Атрибуты**

js

КопироватьРедактировать

let img = document.querySelector("img");

img.getAttribute("src"); // получить

img.setAttribute("alt", "Фото"); // задать

img.removeAttribute("width"); // удалить

**8.6. Классы: classList**

js

КопироватьРедактировать

title.classList.add("active");

title.classList.remove("hidden");

title.classList.toggle("dark-theme");

title.classList.contains("error"); // true/false

**8.7. Создание и удаление элементов**

js

КопироватьРедактировать

let p = document.createElement("p");

p.textContent = "Новый параграф";

document.body.append(p); // вставка

p.remove(); // удалить

**8.8. Обработчики событий**

js

КопироватьРедактировать

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

alert("Клик!");

});

Можно добавить несколько событий на один элемент.

События: "click", "input", "submit", "keydown", "mouseover", "change" и др.

**8.9. this в обработчике**

js

КопироватьРедактировать

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

this.classList.toggle("active"); // this = элемент

});

⚠️ Не работает со стрелочными функциями: this теряется.

**8.10. Event Object**

js

КопироватьРедактировать

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

console.log(e.target); // элемент, по которому кликнули

});

**8.11. Всплытие событий**

js

КопироватьРедактировать

div.addEventListener("click", () => console.log("div"));

btn.addEventListener("click", () => console.log("button"));

Клик по btn сначала вызывает btn, затем div — это **всплытие**.

**8.12. Частые ошибки**

❌ querySelectorAll() не возвращает массив → нужно forEach  
❌ Прямое изменение innerHTML — может быть небезопасно  
❌ Стрелочная функция в addEventListener теряет this  
❌ Работа с элементом до загрузки DOM → null

**8.13. Глоссарий**

| **Термин** | **Значение** |
| --- | --- |
| DOM | Объектная модель документа (структура HTML) |
| Селектор | Способ найти элемент |
| Обработчик | Функция, вызываемая при событии |
| Всплытие | Механизм передачи событий вверх по DOM |
| Атрибут | Свойство тега (src, alt, href и т.д.) |

**8.14. Вопросы для самопроверки**

1. Чем отличается textContent от innerHTML?
2. Что делает classList.toggle()?
3. Как создать новый элемент с JS?
4. В чём разница между querySelector и getElementById?
5. Почему this не работает в стрелочной функции в addEventListener?