



DOM

В JavaScript **DOM (Document Object Model)** — это объектная модель HTML-документа. Когда браузер загружает веб-страницу, он превращает HTML в древовидную структуру объектов — **DOM-дерево**, где каждый HTML-элемент становится узлом (node), а тег — объектом, доступным через JavaScript.

Что такое DOM?

DOM — это интерфейс, предоставляемый браузером, позволяющий JavaScript взаимодействовать с HTML-документом.

С его помощью можно:

- читать и изменять содержимое страницы,
- добавлять или удалять элементы,
- обрабатывать события (например, клики),
- менять стили и классы элементов.

Пример DOM-дерева:

Для кода:

```
html

<html>
  <body>
    <h1>Привет, мир!</h1>
```

```
<p>Это параграф.</p>
</body>
</html>
```

DOM будет выглядеть примерно так:

css

```
document
├── html
│   └── body
│       ├── h1
│       └── p
```



Часто используемые методы DOM в JavaScript

1. Поиск элементов на странице

Метод	Описание
<code>document.getElementById(id)</code>	Ищет элемент по <code>id</code>
<code>document.getElementsByClassName(className)</code>	Возвращает коллекцию элементов с указанным классом
<code>document.getElementsByTagName(tagName)</code>	Возвращает коллекцию элементов по тегу
<code>document.querySelector(selector)</code>	Возвращает первый элемент по CSS-селектору
<code>document.querySelectorAll(selector)</code>	Возвращает все элементы по CSS-селектору

Примеры:

js

```
const title = document.getElementById("main-title");
const paragraphs = document.getElementsByTagName("p");
const firstButton = document.querySelector(".btn");
const allButtons = document.querySelectorAll(".btn");
```

2. Изменение содержимого элементов

Метод/свойство	Описание
<code>element.textContent</code>	Получить или задать текст без HTML
<code>element.innerHTML</code>	Получить или задать HTML-содержимое
<code>element.innerText</code>	Как <code>textContent</code> , но учитывает стили и переносы строк

Пример:

```
js

const p = document.querySelector("p");
p.textContent = "Новый текст!";
```

3. Изменение атрибутов

Метод/свойство	Описание
<code>element.getAttribute(attrName)</code>	Получает значение атрибута
<code>element.setAttribute(attrName, value)</code>	Устанавливает значение атрибута
<code>element.removeAttribute(attrName)</code>	Удаляет атрибут
<code>element.id</code> , <code>element.className</code> , <code>element.href</code> , и др.	Быстрый доступ к атрибутам

Пример:

```
js

const link = document.querySelector("a");
link.setAttribute("href", "https://example.com");
link.setAttribute("target", "_blank");
```

4. Работа с классами

Метод	Описание
<code>element.classList.add("class")</code>	Добавляет класс
<code>element.classList.remove("class")</code>	Удаляет класс
<code>element.classList.toggle("class")</code>	Включает/выключает класс
<code>element.classList.contains("class")</code>	Проверяет наличие класса

Пример:

```
js
```

```
const box = document.querySelector(".box");  
box.classList.add("active");  
box.classList.toggle("hidden");
```

5. Создание и добавление элементов

Метод	Описание
<code>document.createElement(tagName)</code>	Создает новый элемент
<code>element.appendChild(child)</code>	Добавляет дочерний элемент
<code>element.append(...)</code>	Добавляет один или несколько элементов/строк
<code>element.prepend(...)</code>	Добавляет в начало
<code>element.remove()</code>	Удаляет элемент
<code>element.replaceWith(...)</code>	Заменяет текущий элемент

Пример:

```
js
```

```
const newDiv = document.createElement("div");  
newDiv.textContent = "Привет!";  
document.body.appendChild(newDiv);
```

6. Обработка событий

Метод	Описание
<code>element.addEventListener(event, handler)</code>	Добавляет обработчик события
<code>element.removeEventListener(...)</code>	Удаляет обработчик

Пример:

```
js
```

```
const button = document.querySelector("button");  
  
button.addEventListener("click", () ⇒ {
```

```
    alert("Кнопка нажата!");  
  });
```

Пример на практике

html

```
<button id="myBtn">Нажми меня</button>  
<p id="output"></p>  
  
<script>  
  const btn = document.getElementById("myBtn");  
  const output = document.getElementById("output");  
  
  btn.addEventListener("click", () => {  
    output.textContent = "Привет от JavaScript!";  
    btn.classList.toggle("clicked");  
  });  
</script>
```





Итого



DOM — это способ взаимодействия JavaScript с HTML-страницей.

Он позволяет:

- находить элементы,
- изменять их содержимое и стиль,
- добавлять и удалять узлы,
- реагировать на действия пользователя.

Таблица-шпаргалка по DOM в JavaScript

Категория	Метод / Свойство	Описание	Пример
 Поиск элементов	<code>getElementById(id)</code>	Находит элемент по <code>id</code>	<code>document.getElementById("header")</code>
	<code>getElementsByClassName(class)</code>	Возвращает коллекцию по классу	<code>document.getElementsByClassName("box")</code>
	<code>getElementsByTagName(tag)</code>	Возвращает коллекцию по тегу	<code>document.getElementsByTagName("p")</code>
	<code>querySelector(sel)</code>	Первый элемент по CSS-селектору	<code>document.querySelector(".item")</code>
	<code>querySelectorAll(sel)</code>	Все элементы по CSS-селектору	<code>document.querySelectorAll("li.active")</code>
 Работа с содержимым	<code>textContent</code>	Устанавливает / получает только текст	<code>el.textContent = "Текст"</code>
	<code>innerHTML</code>	Устанавливает / получает HTML	<code>el.innerHTML = "Жирный"</code>
	<code>innerText</code>	Как <code>textContent</code> , но с учётом CSS	<code>el.innerText = "Видимый текст"</code>
 Атрибуты	<code>getAttribute(name)</code>	Получает значение атрибута	<code>el.getAttribute("href")</code>
	<code>setAttribute(name, value)</code>	Устанавливает значение атрибута	<code>el.setAttribute("id", "main")</code>
	<code>removeAttribute(name)</code>	Удаляет атрибут	<code>el.removeAttribute("disabled")</code>
 Классы	<code>classList.add(class)</code>	Добавляет класс	<code>el.classList.add("active")</code>
	<code>classList.remove(class)</code>	Удаляет класс	<code>el.classList.remove("hidden")</code>
	<code>classList.toggle(class)</code>	Вкл/выкл класс	<code>el.classList.toggle("open")</code>
	<code>classList.contains(class)</code>	Проверяет наличие класса	<code>el.classList.contains("btn")</code>

Категория	Метод / Свойство	Описание	Пример
 Элементы	<code>createElement(tag)</code>	Создает новый элемент	<code>document.createElement("div")</code>
	<code>appendChild(node)</code>	Добавляет в конец	<code>el.appendChild(newDiv)</code>
	<code>append(...nodes)</code>	Добавляет строки/элементы	<code>el.append("Привет", newEl)</code>
	<code>prepend(...)</code>	Добавляет в начало	<code>el.prepend(newEl)</code>
	<code>remove()</code>	Удаляет элемент	<code>el.remove()</code>
	<code>replaceWith(...)</code>	Заменяет элемент	<code>el.replaceWith(newNode)</code>
 События	<code>addEventListener(event, fn)</code>	Назначает обработчик события	<code>btn.addEventListener("click", fn)</code>
	<code>removeEventListener(...)</code>	Удаляет обработчик	<code>el.removeEventListener("click", fn)</code>