

## **ПМ3 Разработка модулей ПО.**

**РО 3.2 Разрабатывать модули с применением DOM API,  
Regexp, HTTP**

**Тема 1. JS: DOM API.**

**Лекция 2. JavaScript в браузере.**

# Цели занятия:

- Познакомиться с ролью JavaScript в браузере.
- Рассмотреть глобальный объект window и основные BOM-объекты (navigator, location, history).
- Изучить базовые методы взаимодействия с пользователем (alert, confirm, setTimeout).
- Понять разницу между DOM-элементами и текстовыми узлами (children и childNodes).

# **Учебные вопросы:**

**1. DOM-элементы и текстовые узлы.**

**2. Методы взаимодействия с пользователем.**

# 1. DOM-элементы и текстовые узлы.

## Что такое DOM?

DOM (Document Object Model) — это объектное представление HTML-документа.

Браузер при загрузке страницы строит DOM-дерево:

- каждый HTML-тег → DOM-элемент (объект типа Element),
- текст между тегами → текстовый узел (Text),
- комментарии → узлы-комментарии (Comment).

👉 Таким образом, HTML превращается в структуру данных (объект), с которой можно работать из JavaScript.

## Узлы и элементы.

Всё в DOM — это узлы.

Когда браузер превращает HTML в DOM, каждый фрагмент документа становится узлом (node).

Узлы бывают разных типов:

- Элемент (Element) → соответствует HTML-тегу (<div>, <p>, <span>).
- Текст (Text) → кусок текста внутри элемента.
- Комментарий (Comment) → содержимое <!-- ... -->.
- Документ (Document) → сам объект document, корень всего дерева.
- Есть и другие (doctype, атрибуты и т.д.), но чаще всего нужны именно эти три.

## Не все узлы — элементы.

👉 Элемент — это частный случай узла.

То есть:

- каждый элемент — узел,
- но не каждый узел — элемент.

ВСЁ в DOM — это узлы. Но только теги = элементы.

## Свойства children и childNodes.

### `element.children`

Возвращает коллекцию только дочерних элементов.

Игнорирует текстовые узлы и комментарии.

Пример:

```
let div = document.querySelector("div");  
console.log(div.children);  
// HTMLCollection(3) [p, p, p]
```



## **element.childNodes**

Возвращает коллекцию всех дочерних узлов, включая текст и комментарии.

Пример:

```
let div = document.querySelector("div");  
console.log(div.childNodes);  
//NodeList(7) [text, p, text, p, text, p, text]
```

## Выводы:

- DOM — это объектное представление HTML.
- Всё в DOM — узлы (Node), но не все узлы являются элементами.
- children → только элементы.
- childNodes → все узлы, включая текст и комментарии.

## 2. Методы взаимодействия с пользователем.

### Методы окна (window)

Браузер предоставляет встроенные функции для диалога с пользователем.

Все они принадлежат глобальному объекту window, но обычно пишутся без window.:

```
window.alert("Привет!");  
alert("Привет!"); // то же самое
```

## **alert() — уведомление**

- Показывает модальное окно с сообщением и кнопкой ОК.
- Используется для простых уведомлений или отладки.

**Минус:** блокирует выполнение кода и работу с вкладкой, пока пользователь не нажмёт ОК.

## **confirm()** — запрос подтверждения.

- Показывает модальное окно с текстом и кнопками OK / Cancel.
- Возвращает true, если нажали OK, и false — если Cancel.

Пример:

```
let isSure = confirm("Удалить запись?");  
if (isSure) {  
    console.log("Запись удалена");  
} else {  
    console.log("Удаление отменено");  
}
```

## **prompt() — запрос ввода (дополнительно)**

- Показывает окно с полем ввода и кнопками OK / Cancel.
- Возвращает введённую строку или null, если нажали Cancel.

Пример:

```
let name = prompt("Введите ваше имя:", "Гость");  
alert("Привет, " + name + "!");
```

## setTimeout() — отложенный запуск

- Запускает функцию один раз через заданное время (в миллисекундах).
- Возвращает id таймера, который можно отменить через clearTimeout(id).

Синтаксис:

`setTimeout(func, delay, arg1, arg2, ...);`

Пример:

```
setTimeout(() => {  
  alert("Прошло 3 секунды");  
}, 3000);
```

## Выводы:

- Методы `alert`, `confirm`, `prompt` — простые модальные диалоги браузера.
- Они блокируют страницу, пока пользователь не ответит.
- В реальных проектах часто заменяют на кастомные модальные окна.
- `setTimeout` позволяет запускать код через заданное время, не блокируя выполнение.



## Выводы по лекции:

- JS в браузере делает страницы интерактивными, но работает в «песочнице».
- Скрипты подключаются через `<script>` (inline или файл, в head или перед `</body>`, с `defer/async`).
- Глобальный объект `window` хранит BOM, DOM и встроенные функции.
- DOM = дерево узлов: элементы, текст, комментарии.
- Методы `alert`, `confirm`, `prompt` блокируют работу страницы → в проектах заменяют кастомными окнами.
- `setTimeout` позволяет запускать код с задержкой, не блокируя выполнение.

# Контрольные вопросы:

- Какие BOM-объекты входят в window и что они делают?
- В чём разница между children и childNodes?
- Чем отличаются alert и confirm?
- Для чего используется setTimeout?

# Домашнее задание:

1. <https://ru.hexlet.io/courses/js-dom>