



Преподаватель:

Коляда

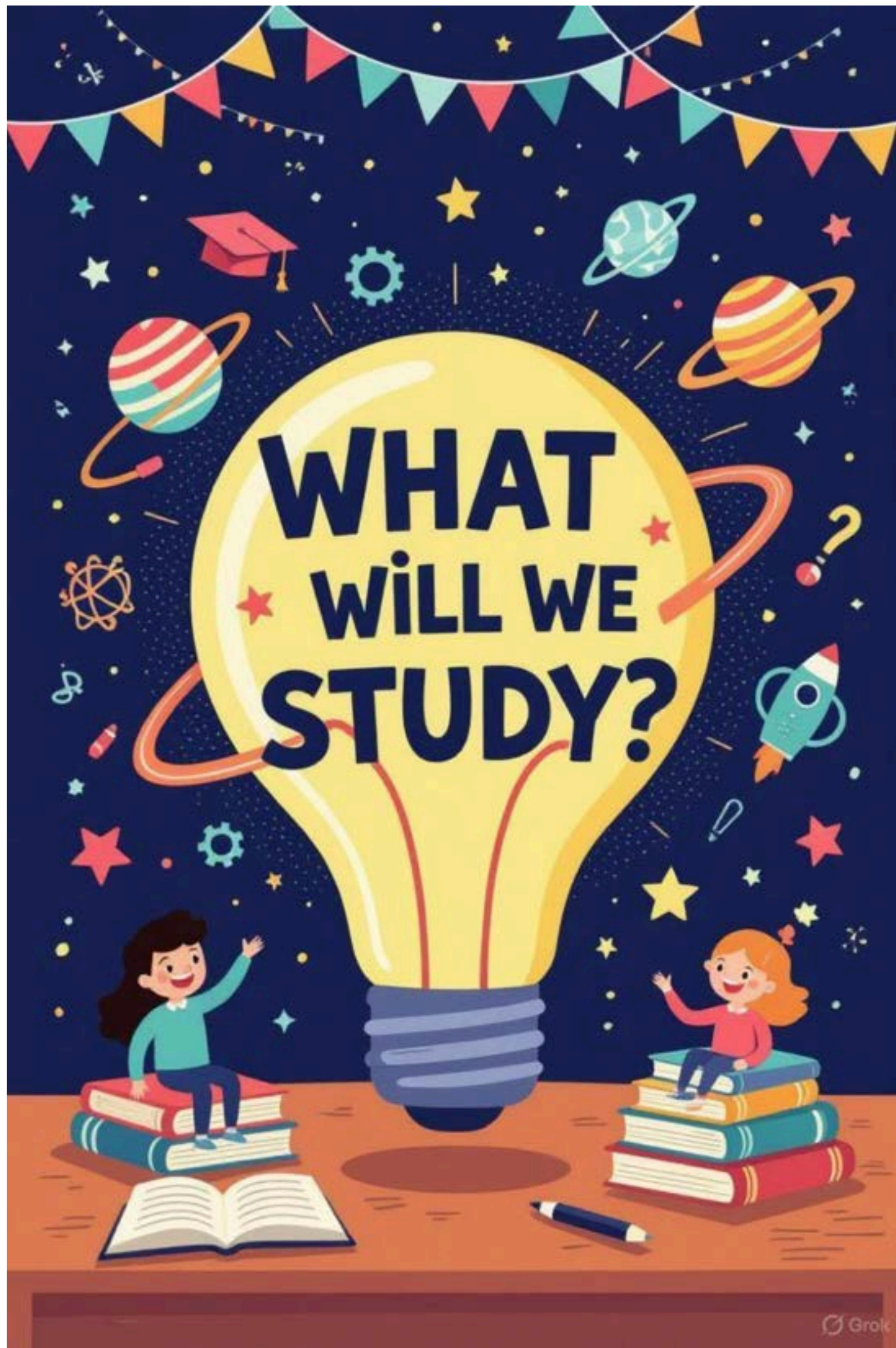
Никита Владимирович

Обратная связь:

- сообщения на inStudy

# РАБОТА С DOM В JAVASCRIPT

ПРОЕКТИРОВАНИЕ И РАЗРАБОТКА  
ИНТЕРФЕЙСОВ ПОЛЬЗОВАТЕЛЯ

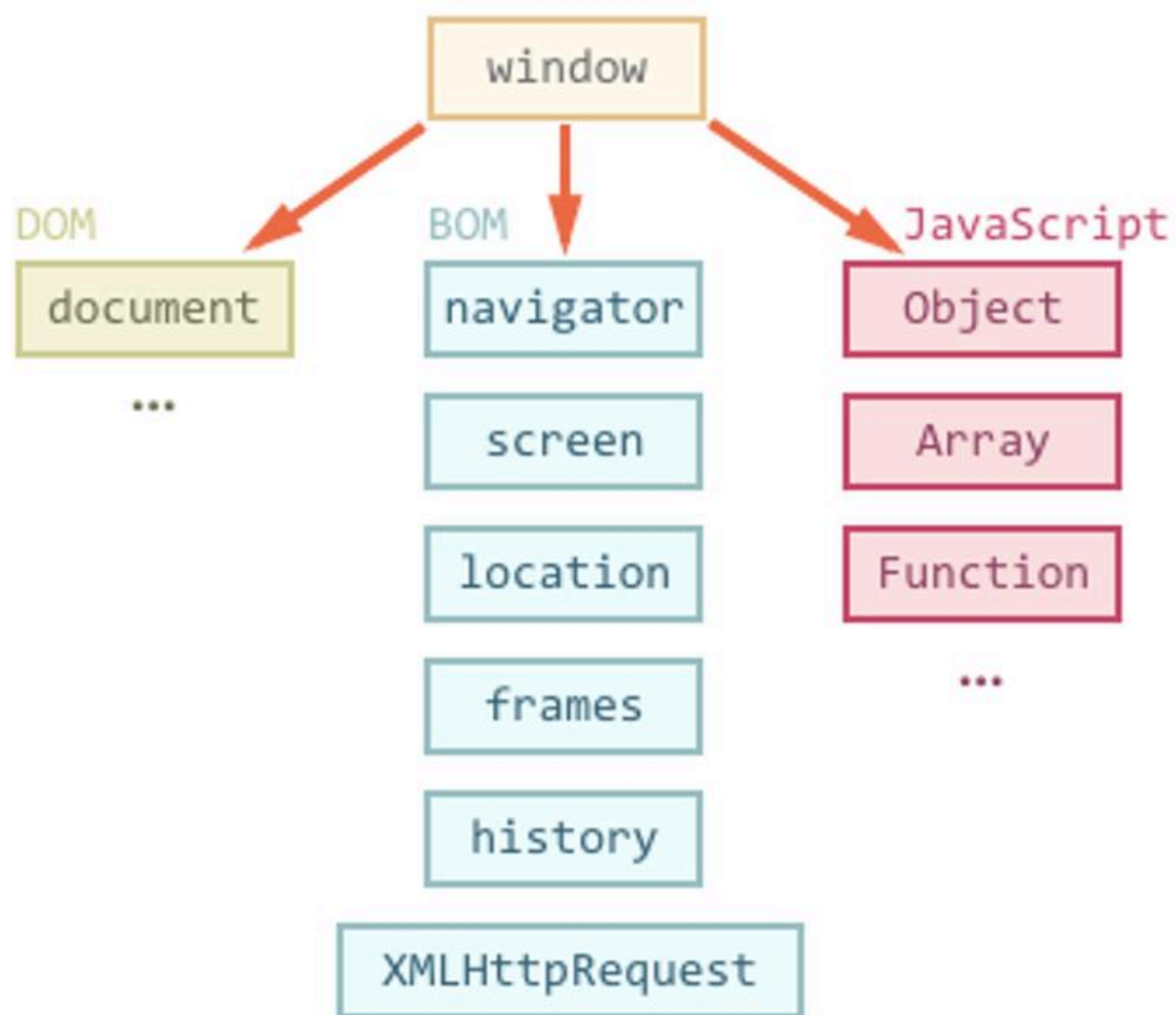


## ЧТО ИЗУЧИМ СЕГОДНЯ?

1. Что такое DOM?
2. Основные возможности работы с DOM.
3. Манипуляция элементами, атрибутами и событиями.
4. Навигация по DOM-дереву.
5. Создание и удаление узлов.



# БРАУЗЕРНОЕ ОКРУЖЕНИЕ



## Окружение: DOM, BOM и JS

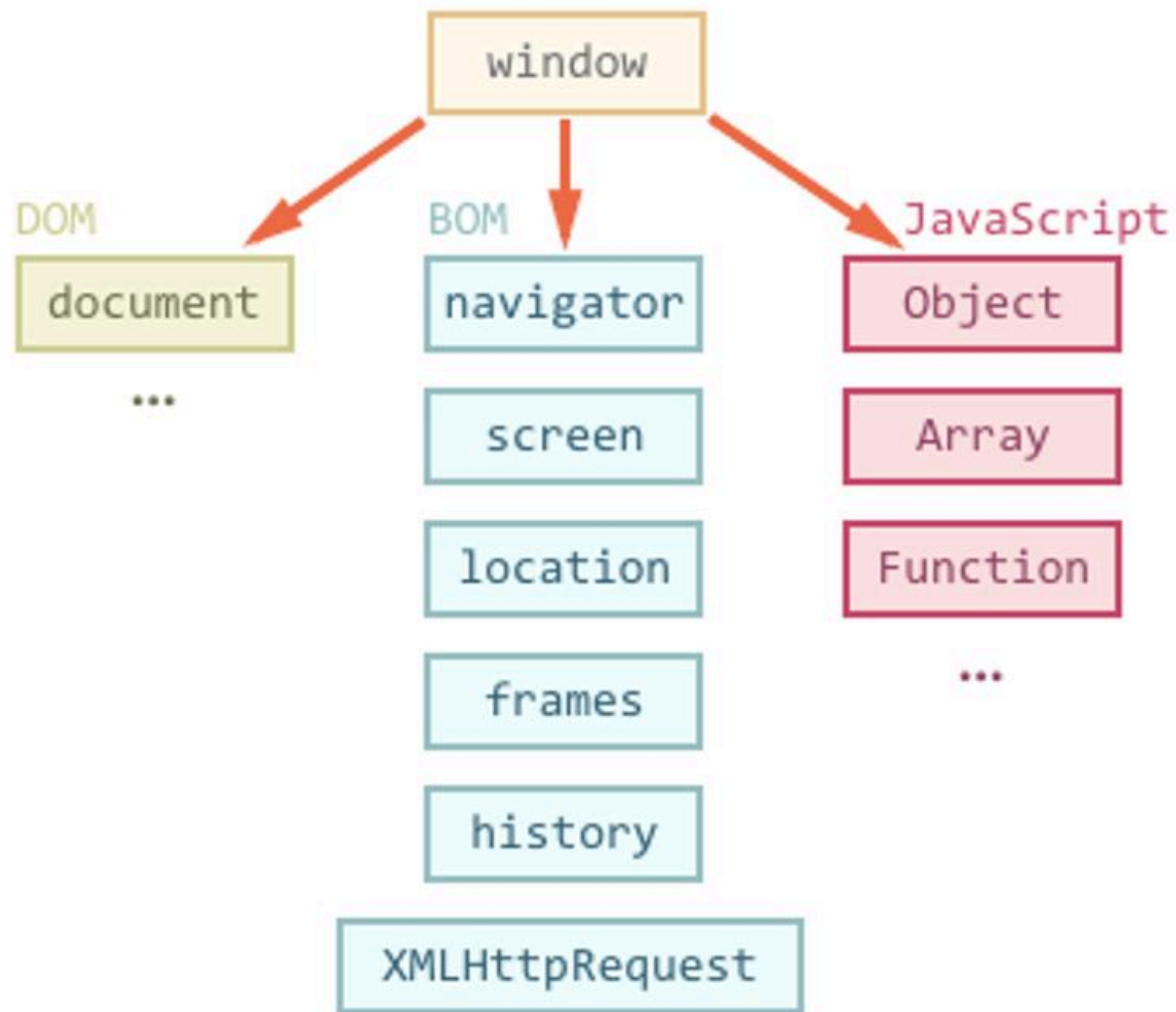
Современный JavaScript используется не только в браузерах. Среда, в которой он запускается, будь то браузер, сервер или что-то ещё, называется окружением.

У **разных окружений** разные возможности и функциональность. Окружение предоставляет языку дополнительные возможности и функции. Браузерное окружение, например, даёт возможность работать со страницами сайтов.

Представление этих дополнительных возможностей и функциональности **в виде объектов**, к которым у языка есть доступ — это объектная модель.

В случае с браузерным окружением — это объектная модель браузера (**Browser Object Model, BOM**). Она предоставляет доступ к **navigator**, **location**, **fetch** и другим объектам .

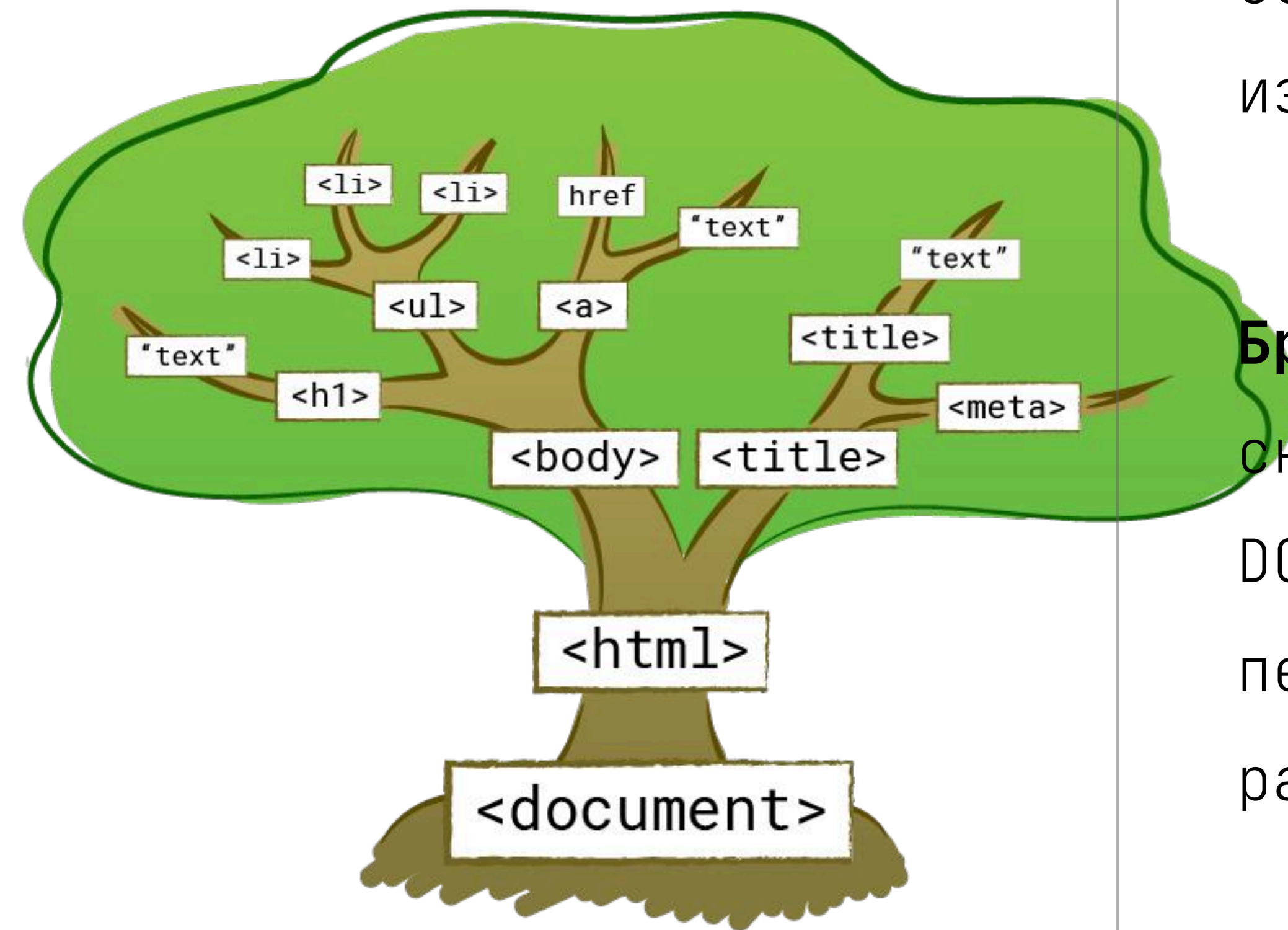
# БРАУЗЕРНОЕ ОКРУЖЕНИЕ



- Объект **navigator** содержит информацию о браузере: название, версия, платформа, доступные плагины, доступ к буферу обмена и прочее. Это один из самых больших объектов в окружении.
- Объект **screen** содержит информацию об экране браузера. **width** и **height** указывают ширину и высоту экрана в пикселях соответственно.
- Объект **location** даёт возможность узнать, на какой странице мы находимся (какой у неё URL) и перейти на другую страницу программно.
- **fetch** предоставляет возможность работы с сетью, с его помощью можно отправлять запросы на сервер.
- **history** даёт доступ к истории браузера, которая ограничена текущей вкладкой. То есть с её помощью можно перейти на страницу назад, только если мы пришли с неё.
- **localStorage**, **sessionStorage** Локальные хранилища используются, чтобы хранить какие-то данные в браузере пользователя. Они удобны, когда мы не хотим отправлять данные на сервер, потому что они, например, промежуточные, или нужны только на клиенте.



# ЧТО ТАКОЕ **DOM-ДЕРЕВО**?



**DOM** (Document Object Model) – это специальная древовидная структура, которая позволяет управлять HTML-разметкой из JavaScript-кода. Управление обычно состоит из добавления и удаления элементов, изменения их стилей и содержимого.

**Браузер** создаёт DOM при загрузке страницы, складывает его в переменную **document** и сообщает, что DOM создан, с помощью события **DOMContentLoaded**. С переменной **document** начинается любая работа с HTML-разметкой в JavaScript.

## ИЗ ЧЕГО СОСТОИТ DOM ?

```
1 <!DOCTYPE html>
2 <head>
3   <title>Личный кабинет</title>
4 </head>
5 <body>
6   <header>
7     <ul class="menu">
8       <li>Главная</li>
9       <li>Статьи</li>
10      <li>Контакты</li>
11    </ul>
12  </header>
13  <article id="12">
14    <h1>Как выучить джаваскрипт?</h1>
15    <p>Нужно начать учиться.</p>
16  </article>
17 </body>
```

Браузер превращает эту разметку в JavaScript представление, сохраняя при этом вложенность и свойства каждого HTML-элемента. Этого удаётся достичь с помощью специальной структуры данных – дерева.

Каждый тег представляет собой узел дерева. У каждого узла могут быть дочерние узлы – таким образом сохраняется информация о вложенности тегов.

Правда, слово тег в этом контексте не используют и говорят элемент.

Например, HTML-код выше превратится в такое дерево:

