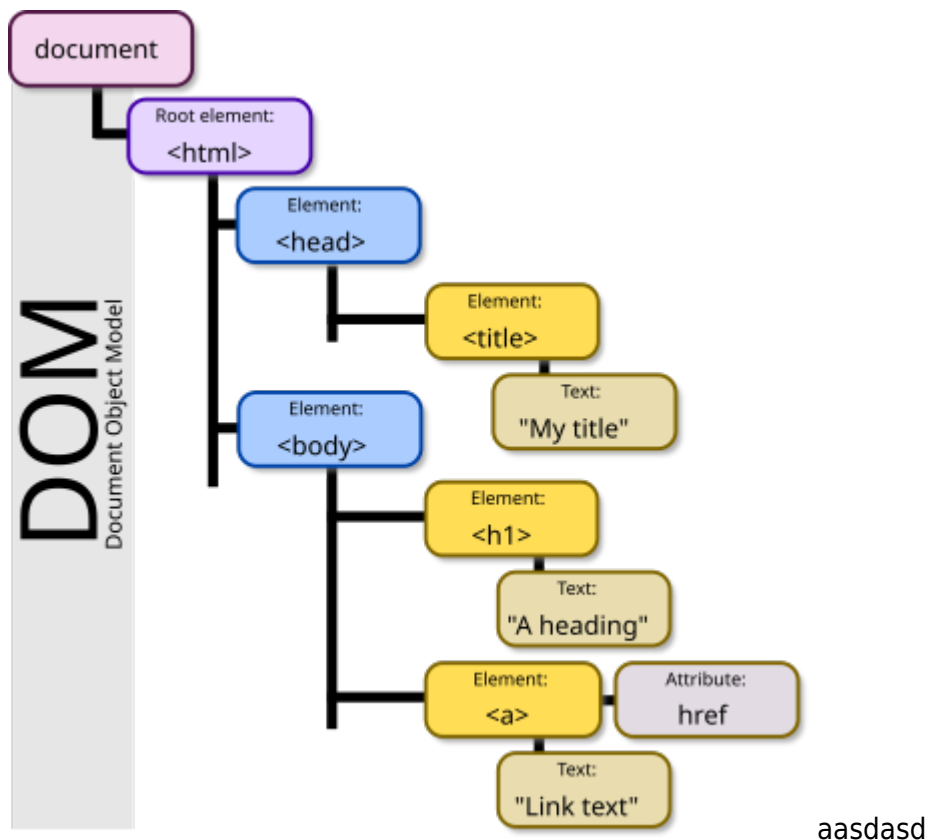


DOM manipuláció JavaScript segítségével



1. Bevezetés (5 perc) Mi a DOM (Document Object Model)? Miért fontos a DOM manipuláció a webfejlesztésben? Alapfogalmak: document objektum DOM-fák Elemtípusok (node, element, attribute, text)

2. Alapvető DOM manipulációs technikák bemutatása (25 perc) Elemek kiválasztása

getElementById getElementsByClassName getElementsByTagName querySelector querySelectorAll
Tartalom módosítása

innerText, textContent innerHTML Attribútumok módosítása

setAttribute, getAttribute classList.add, classList.remove, classList.toggle Új elemek létrehozása és hozzáadása

createElement appendChild, prepend insertBefore Eseménykezelés

addEventListener Példa: gombra kattintáskor szöveg megváltoztatása Stílus módosítása JavaScriptből

element.style.property Dinamikus osztályváltás classList-tel

3. Példa (10 perc) Én mutatok egy példát, amely jól illusztrálja az eddig tanultakat.

Bemutatott feladat: Egyszerű interaktív lista □ Feladat:

Van egy lista pár kezdeti elemmel. Egy megnyomására egy új elem hozzáadódik a listához. Ha egy listaelemre kattintanak, az pirosra változik.

```
<!DOCTYPE html>
<html lang="hu">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>DOM Manipuláció</title>
  <style>
    .highlight { color: red; }
  </style>
</head>
<body>
  <ul id="myList">
    <li>Első elem</li>
    <li>Második elem</li>
  </ul>
  <button id="addItem">Új elem hozzáadása</button>

  <script>
    document.getElementById('addItem').addEventListener('click',
function() {
    let newItem = document.createElement('li');
    newItem.textContent = "Új elem";
    document.getElementById('myList').appendChild(newItem);
});

    document.getElementById('myList').addEventListener('click',
function(event) {
    if (event.target.tagName === 'LI') {
      event.target.classList.toggle('highlight');
    }
  });
  </script>
</body>
</html>
```

Feladat

A feladat egy egyszerű, dinamikus teendő lista készítése.

- Legyen egy <input> szöveges mező, ahova a felhasználó beírhat egy új teendőt.
- Legyen egy „Hozzáadás” gomb, melynek lenyomásakor a beírt szöveg kerüljön fel egy listára.
- A listán szereplő elemekre kattintva azok legyenek áthúzva (CSS: text-decoration: line-through).
- Legyen egy „Törlés” gomb minden elem mellett, amely törli az adott teendőt listából.
- Az oldal elemeinek formázásához használj CSS stílusokat!

Az elkészült megoldást töltsd fel a GitHub repository-dba!

From:

<https://edu.iit.uni-miskolc.hu/> - **Institute of Information Science - University of Miskolc**

Permanent link:

https://edu.iit.uni-miskolc.hu/tanszek:oktatas:web_technologia_alapjai:js_dom?rev=1741610544

Last update: **2025/03/10 12:42**

