**### 11. óra: DOM alapjai**

**- A DOM (Document Object Model) fogalma**

**- Node-ok és elemek megértése**

**- Gyakorlati példák: a DOM alapvető felépítésének áttekintése.**

**11. Óra: DOM Alapjai**

**1. Bevezetés (10 perc)**

* **Tanár mondja:** "Ma megismerkedünk a DOM fogalmával, amely a webfejlesztés egyik legfontosabb eleme. A DOM, vagyis a Document Object Model, egy strukturált hierarchia, amely a HTML dokumentumot reprezentálja. Ez azt jelenti, hogy minden elem, attribútum és szöveg JavaScript-ből elérhető, manipulálható, és dinamikusan módosítható."
* **Fontos kiemelni:**
  + **Miért hasznos a DOM?** A DOM segít interaktív weboldalakat készíteni, ahol az oldalak reagálnak a felhasználói műveletekre (pl. kattintás, űrlapkitöltés).
  + **Mi történik a háttérben?** A DOM egy modellt ad, amely lehetővé teszi, hogy közvetlenül hozzáadjunk, törüljük vagy megváltoztassuk az oldal tartalmát.
  + **Hol alkalmazható?** Interaktív elemek készítésekor, pl. gombokra kattintva tartalom megjelenítésére vagy dinamikus űrlapok kezeléséhez.

**2. A DOM alapjai (15 perc)**

* **Tanár mondja:** "A DOM-ot egy fás struktúra képviseli, amelynek gyökere a document objektum. Ez a fa minden HTML elemet, attribútumot és szöveget tartalmaz."
* **Egyszerű példa magyárázata:** "Képzeljük el a következő HTML dokumentumot:"

A képen szöveg, képernyőkép, képernyő, Betűtípus látható

Automatikusan generált leírás

<!DOCTYPE html>

<html>

<head>

<title>DOM Példa</title>

</head>

<body>

<h1 id="cim">DOM Alapok</h1>

<p class="leiras">Ez egy bevezetés a DOM-hoz.</p>

</body>

</html>

* + **Mi történik a DOM-ban?**
    - A gyökerelem a <html>.
    - A <body> tartalmazza a <h1> és a <p> elemeket.
    - Az id="cim" egyedi azonosítására használt, amely közvetlenül elérhetővé teszi a JavaScript számára.
* **Fontos gondolat:** "A DOM lehetővé teszi, hogy bármely elemhez hozzáférjünk, és azt a kódunkkal megmódosítsuk."

**3. DOM Node-ok és elemek részletesebben (15 perc)**

* **Tanár mondja:** "A DOM alapegysége a node, vagy csomópont. Ezeket a csomópontokat használjuk a dokumentum hierarchikus struktúrájának megértéséhez."
* **Node-ok típusai:**
  + **Elem node-ok:** pl. <h1>, <p> — ezek a HTML elemeket képviselik.
  + **Attribútum node-ok:** pl. az id vagy class attribútumok, amelyek kiegészítik az elemeket információval.
  + **Szöveg node-ok:** pl. a "DOM Alapok" szöveg, ami a <h1> elem belsejében található.
* **Gyakorlati példa:** "Nyissuk meg a böngésző fejlesztői eszközeit (F12), majd vizsgáljuk meg egy egyszerű oldal struktúráját. A DOM fa vizualizációja segít megérteni, hogyan kapcsolódnak egymáshoz az elemek."

**4. A DOM elérése és manipulálása JavaScript-tel (20 perc)**

* **Tanár mondja:** "JavaScript-tel közvetlenül elérhetjük a DOM elemeit, majd manipulálhatjuk azokat — hozzáadhatunk új tartalmat, módosíthatjuk a meglévőt, vagy eltávolíthatunk elemeket."
* **Konkrét példák:**

A képen szöveg, képernyőkép, Betűtípus látható

Automatikusan generált leírás

// Egy elem kiválasztása ID alapján

const cim = document.getElementById('cim');

console.log(cim.innerText); // Kiírja: DOM Alapok

// Elemek kiválasztása osztály név alapján

const leirasok = document.getElementsByClassName('leiras');

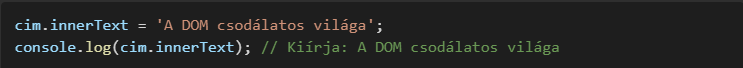
console.log(leirasok[0].innerText); // Kiírja: Ez egy bevezetés a DOM-hoz.

// Elemek kiválasztása CSS szelektorral

const pElem = document.querySelector('p');

console.log(pElem.innerText); // Kiírja: Ez egy bevezetés a DOM-hoz.

* **Részletes magyázat:**
  + **getElementById:** Ez a leggyorsabb módszer egy adott ID-val rendelkező elem elérésére.
  + **getElementsByClassName:** Egy adott osztály név alapján tömbszerű objektumot kapunk.
  + **querySelector:** CSS szelektorral precízen kereshetünk elemeket.
* **Gyakorlat:** "Változtassuk meg a <h1> elem szövegét JavaScript segíségével:"



cim.innerText = 'A DOM csodálatos világa';

console.log(cim.innerText); // Kiírja: A DOM csodálatos világa

**5. Gyakorlati Feladatok (10 perc)**

* **Feladat 1:** "Hozzatok létre egy HTML oldalt, amely tartalmaz egy <h1> elemet és egy <p> elemet. JavaScript-tel írjatok ki mindkét elem szövegét a konzolra!"
* **Feladat 2:** "Változtassátok meg a <h1> elem szövegét, hogy ez legyen: 'A DOM dinamikus kezelése.'"
* **Feladat 3:** "Adjatok hozzá egy új <p> elemet a DOM-hoz, amely ezt a szöveget tartalmazza: 'Ezt a bekezdést dinamikusan hoztuk létre JavaScript-ből.'"

**6. Óra zárása és kérdések (5 perc)**

* **Tanár mondja:** "A DOM alapjainak megértése elengedhetetlen a dinamikus webfejlesztéshez. Ha bármilyen kérdésetek van, most térjetek rá, és tárgyaljuk meg."
* **Otthoni feladat:** "Hozzatok létre egy HTML oldalt, amely tartalmaz egy listaelemet (<ul> vagy <ol>), és JavaScript-tel adjatok hozzá új elemeket dinamikusan a list