Tartalom

Mi is az a DOM?

Document **O**bject **M**odel kifejezés rövidítése. A DOM segítségével elérhetünk HTML elemeket, újakat hozhatunk létre, módosíthatjuk a már meglévők attribútumait, vagy épp figyelhetjük az eseményeket, tehát segítségével manipulálhatjuk a weboldalunkat, hogy az interaktívvá válhasson.

Elemek értékeinek lekérdezése

Ahhoz, hogy az elemeket elérjük valamilyen módon, rendelkezniük kell egy azonosító értékkel, ami segítségével „utalhatunk rá”.

---

getElementById()

Ilyen azonosító lehet az id, ilyenkor egy elemre hivatkozunk, mivel ugye id egy van egy oldalon, ha jól dolgoztunk…

document.getElementById(”idErtek”);

---

getElementsByName()

Ha több elemet akarunk kijelölni

document.getElementsByName(”nevErtek”);

---

getElementsByClassName()

Ebben az esetben a több elem egy tömb szerkezetet ad vissza, amit mondjuk for ciklussal bejárhatunk.

document.getElmentsByClassName(”osztalyNeve”);

---

getElementsByTagName()

lehetőségünk van html tag-ek alapján is kijelölni elemeket

document.getElementsByTagName(”htmlTagNeve”);

Ha szeretnénk úgyis hivatkozhatunk az elemre, mint ahogy azt már CSS-ben megszoktuk, ehhez az alábbi függvényt kell használnunk, manapság ezek a legelterjedtebbek.

querySelector()

Manapság már lehetőségünk van úgy nevezett Query Selectort írni

document.querySelector(”#azonositoNeve”);

Az első osztalyNeve elemet adja vissza, ha az összeset szeretnénk akkor a…

querySelectorAll()

document.querySelectorAll(”.osztalyNeve”);

Az adott osztálynévvel rendelkező összes elemet adja vissza…

Amennyiben nem jól adjuk meg az értéket, az eredmény null visszatérési értékű

Elemek értékeinek lekérdezése módosítása

Tulajdonságok Lekérdezése

Összes tulajdonság lekérdezése

elemNeve.attributes

Egy adott tulajdonság értékének lekérdezése

elemNeve.getAttribute(tulajdonsagNeve);

pl.: elemNeve.getAttribute(”style”);

Tulajdonságok Módosítás

elemNeve.setAttribute(”mit”, ”mire”);

Mindig két paramétert kell megadni, hatásásra az adott html elem attribútuma beállítódik az általunk megadott értékre. (ha létezik)

Elem CSS tulajdonságának módosítása

Konkrét CSS tulajdonság lekérdezése

let elemMentettNeve = window.getComputedStyle(elemNeve);

let elemTulajdonsagAllapot= elem.getPropertyValue("tulajdonság"); //kimenti a tulajdonság értékét

vagy

let elemTulajdonsagAllapot = document.getElementById("elemNeve").style.elemTulajdonsaga;

Konkrét CSS tulajdonság módosítása

document.getElementById("elemNeve").style.elemTulajdonsaga = "tulajdonsagErteke";

Például.:Konkrét elem tulajdonságának beállítása példa, elrejtett elem megjelenítése

```
let elem = window.getComputedStyle(leiras);
let allapot = elem.getPropertyValue("display");
if (allapot != 'block') {
        document.getElementById("leiras").style.display = "block";
}
```

Több html elem „bejárása”…

először válasszunk ki az elemeket, amiket szeretnék módosítani

```
let elemLista=document.querySelectorAll(”ElemekNeve”); //Elemneve lehet pl.: input
for(let i=0;i<elemLista.length;i++){
        elemLista[i].style.backgroundColor=”red”;
}
```

Több elem módosító függvény

```
function tobbElemModosito(szelektor, tulajdonsag, ertek)
{
        let elemLista=document.querySelectorAll(szelektor);
        for(let i=0;i<elemLista.length;i++)
        {
                elemLista[i][tulajdonsag]=ertek; //Elemlista[i].tulajdonsag=ertek; -nek felel meg
        }
}
```

addEventListener() – funkció(k) hozzárendelése elemhez

Lehetőségünk van a html-ben meglévő kódok helyett JS-ben is eventeket hozzárendelni egy html elemhez.

pl.:

```
let kuldesGomb=document.querySelector(” form #kuldes”);

kuldesGomb.onclick=function(){
        alert(„üzenet”);
}
```

```
kuldesGomb.addEventListener(”click”, function(){
        alert(„üzenet”);
}
```

Az első és a második között az a különbség, hogy míg az első felülírja, a második hozzáfűzi az eventet a már korábban létezőhöz.

Osztály módosító műveletek

Osztályok **hozzáadása** DOM segítségével, egy adott elemhez.

Mielőtt megpróbálunk osztályt hozzáadni egy elemhez érdemes azt kiválasztani

```
let modositandoElem = document.querySelector("elemHivatkozasa");

modositandoElem.classList.add("hozzaadando-osztaly");
```

Osztályok **eltávolítása** DOM segítségével, egy adott elemhez.

Mielőtt megpróbálunk osztályt hozzáadni egy elemhez érdemes azt kiválasztani, csak hogy tudjuk pontosan hol kell törölni azt.

```
let modositandoElem = document.querySelector("elemHivatkozasa");

modositandoElem.classList.remove("eltavolitando-osztaly");
```

DOM Műveletek táblázattal

Új sor beillesztése, adatmezőkkel

```
let tabla = document.querySelector("table"); //adott tábla kiválasztása tabla néven

let sor = tabla.insertRow(); //új sor beillesztése az adott táblába

let tablaAdat1 = sor.insertCell(); //tabla elemének beillesztése az adott sorba

tablaAdat1.innerHTML = cellaErtek; //adott táblázat cella értékének megadása
```

Adott sor törlése

```
document.querySelector("tablaNeve").deleteRow(SorAzonosito);
```