



DOM Manipulation

Elementlarni qo'shish va o'zgartirish



Ishning Maqsadi

Ushbu amaliy mashg'ulotning asosiy maqsadi - JavaScript orqali HTML sahifasidagi elementlarni dinamik ravishda boshqarish ko'nikmalarini egallash.

- **DOM tushunchasini** o'rganish va uning ahamiyatini tushunish
- **Elementlarni tanlash** usullarini amalda qo'llash
- **Yangi elementlar yaratish** va sahifaga qo'shish
- **Mavjud elementlarni o'zgartirish** va o'chirish
- **Interaktiv veb-ilovalar** yaratish ko'nikmalarini rivojlantirish



Kirish

Zamonaviy veb-saytlar faqat statik matn va rasmlardan iborat emas. Foydalanuvchi bilan o'zaro ta'sir qiluvchi, dinamik sahifalar yaratish uchun JavaScript va DOM (Document Object Model) muhim rol o'ynaydi.

Tasavvur qiling, siz onlayn do'kon yaratapsiz. Foydalanuvchi mahsulotni savatchaga qo'shganda, sahifani qayta yuklashsiz savatchadagi elementlar soni o'zgarishi kerak. Aynan shu vazifalarni DOM manipulation orqali amalga oshiramiz.



Nazariy Qism

1. DOM (Document Object Model) nima?

DOM - bu HTML hujjatning dasturlash interfeysi. U HTML sahifani daraxt tuzilmasida ko'rsatadi va JavaScript orqali uni o'zgartirish imkonini beradi. Har bir HTML elementi - bu DOMda **obyekt** (object) hisoblanadi.

2. Elementlarni tanlash usullari:

```
// ID orqali tanlash let element =
document.getElementById('myId'); // Class orqali tanlash let
elements = document.getElementsByClassName('myClass'); //
Selector orqali tanlash (eng ko'p ishlatiladigan) let elem =
document.querySelector('.myClass'); let elems =
document.querySelectorAll('div');
```

3. Element yaratish va qo'shish:

```
// Yangi element yaratish let yangiDiv =
document.createElement('div'); // Matn qo'shish
yangiDiv.textContent = 'Salom dunyo!'; // Sahifaga qo'shish
document.body.appendChild(yangiDiv);
```

4. Elementni o'zgartirish:

```
// Matnni o'zgartirish element.textContent = 'Yangi matn'; //
HTML ni o'zgartirish element.innerHTML = '<strong>Qalin
matn</strong>'; // Style o'zgartirish element.style.color =
'red'; // Class qo'shish/olib tashlash
element.classList.add('yangiClass');
element.classList.remove('eskiClass');
```

5. Elementni o'chirish:

```
// Elementni o'chirish element.remove(); // Yoki ota-element
orqali parentElement.removeChild(childElement);
```



Amaliy Namuna - Vazifalar Ro'yxati

Quyidagi misolda DOM manipulation usullarini amalda ko'ramiz:

 Qo'shish



Rangni o'zgartirish



Hammasini tozalash

Vazifalar bu yerda ko'rinadi...

```
function addTask() { let input =
document.getElementById('taskInput'); let taskList =
document.getElementById('taskList'); if (input.value.trim()
!== '') { let taskDiv = document.createElement('div');
taskDiv.className = 'demo-item'; taskDiv.innerHTML = `
<span>${input.value}</span> <button class="delete-btn"
onclick="deleteTask(this)">O'chirish</button> `;
taskList.appendChild(taskDiv); input.value = ''; } }
```

✓ Xulosa

DOM manipulation - bu zamonaviy veb-dasturlashning asosi. Ushbu ko'nikmalarni egallash orqali siz:

- Interaktiv veb-sahifalar yarata olasiz
- Foydalanuvchi tajribasini yaxshilaysiz
- Dinamik kontent bilan ishlay olasiz
- Single Page Application (SPA) yaratish uchun tayyor bo'lasiz