Manipulacja DOM w JavaScript

Wprowadzenie do DOM

Document Object Model (DOM) to reprezentacja struktury dokumentu HTML lub XML, którą przeglądarka interpretuje jako drzewo obiektów. Każdy element na stronie (tag HTML) staje się węzłem w tym drzewie, a JavaScript może manipulować tymi węzłami, dodawać nowe elementy, modyfikować istniejące lub usuwać niepotrzebne.

Wyszukiwanie elementów DOM

JavaScript oferuje kilka metod do wyszukiwania elementów w DOM:

getElementById: Wyszukuje element o określonym atrybucie id.
getElementsByClassName: Wyszukuje elementy o określonej klasie.
getElementsByTagName: Wyszukuje elementy o określonym tagu (np. div, p).
querySelector: Wyszukuje pierwszy element pasujący do selektora CSS.
querySelectorAll: Wyszukuje wszystkie elementy pasujące do selektora CSS.

Przykład:

```
let elementId = document.getElementById('mojeId');
let elementKlasa = document.getElementsByClassName('mojaKlasa');
let elementTag = document.getElementsByTagName('p');
let pierwszyDiv = document.querySelector('div');
let wszystkieDivy = document.querySelectorAll('div');
```

Tworzenie nowych elementów

Za pomocą JavaScript można dynamicznie tworzyć nowe elementy w DOM i dodawać je do dokumentu. Służy do tego metoda createElement:

Przykład:

```
let nowyElement = document.createElement('p');
nowyElement.textContent = "To jest nowy paragraf!";
document.body.appendChild(nowyElement); // dodanie elementu do ciała dokumentu
```

Modyfikacja istniejących elementów

Za pomocą JavaScript możemy modyfikować istniejące elementy na stronie. Można zmieniać ich treść, atrybuty, klasy CSS lub style bezpośrednio w JavaScript.

Przykład modyfikacji treści i stylu:

```
let naglowek = document.getElementById('naglowek');
naglowek.textContent = "Nowy nagłówek!";
naglowek.style.color = "blue";
naglowek.style.fontSize = "24px";
```

Usuwanie elementów

Aby usunąć element z DOM, można użyć metody removeChild lub remove. Przykład usunięcia elementu:

Przykład usunięcia elementu:

```
let elementDoUsuniecia = document.getElementById('usuwanyElement');
elementDoUsuniecia.remove(); // Usuwa element z DOM
```

Manipulacja atrybutami

Możemy dodawać, modyfikować lub usuwać atrybuty elementów za pomocą metod:

setAttribute(): Dodaje lub modyfikuje atrybut.
getAttribute(): Zwraca wartość atrybutu.
removeAttribute(): Usuwa atrybut.

Przykład manipulacji atrybutami:

```
let obraz = document.getElementById('mojeZdjecie');
obraz.setAttribute('src', 'nowyObrazek.jpg'); // Zmiana atrybutu 'src'
obraz.removeAttribute('alt'); // Usunięcie atrybutu 'alt'
```

Zadania powtórkowe
Zadanie 1:
Napisz program, który tworzy tablicę zawierającą liczby od 1 do 10, a następnie wyświetla wszystkie liczby większe od 5.

Zadanie 2:
Napisz funkcję, która przyjmuje tablicę liczb i zwraca największą wartość.

Zadanie 3:
Zdefiniuj obiekt uczen, który będzie posiadał następujące właściwości: imie, wiek, oraz tablicę oceny. Napisz funkcję, która zwróci średnią ocen ucznia.

Zadania do samodzielnego wykonania
Zadanie 4:
Napisz program, który znajdzie element o id="naglowek" i zmieni jego tekst na „Witaj w JavaScript!”.

Zadanie 5:
Stwórz nowy element div z tekstem „To jest nowy blok” i dodaj go na końcu body dokumentu.

Zadanie 6:
Znajdź wszystkie elementy o klasie item na stronie i zmień ich kolor tekstu na czerwony.

Zadanie 7:
Napisz program, który usunie element o id="staryElement" z dokumentu.

Zadanie 8:
Utwórz nowy przycisk na stronie i dodaj mu atrybut type="button" oraz value="Kliknij mnie".

Zadanie 9:
Stwórz funkcję, która zmienia atrybut src obrazu o id="obrazek" na nowy obrazek nowyObrazek.jpg.

Zadanie 10:
Napisz funkcję, która doda klasę aktywny do wszystkich elementów li na stronie.