

Esempio Completo di Pagina HTML con CSS e JavaScript che Manipola il DOM

Descrizione Funzionale:

L'esempio è una pagina web che consente all'utente di interagire con un elenco di articoli. L'utente può:

1. **Aggiungere un nuovo articolo** all'elenco.
2. **Modificare il contenuto** di un articolo esistente.
3. **Rimuovere un articolo** dall'elenco.

Ogni articolo avrà un titolo e una descrizione. Verranno anche applicati alcuni stili CSS per migliorare l'aspetto della pagina.

Struttura del Progetto:

- **HTML:** Crea la struttura base della pagina con i vari elementi come input, bottoni e lista.
- **CSS:** Stili per migliorare l'aspetto visivo della pagina.
- **JavaScript:** Gestisce la logica per aggiungere, modificare e rimuovere articoli nell'elenco.

Codice Completo

HTML (index.html)

```
<!DOCTYPE html>
<html lang="it">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Gestione Articoli</title>
  <link rel="stylesheet" href="styles.css">
</head>
<body>
  <div class="container">
    <h1>Gestione Articoli</h1>

    <div class="form">
      <input type="text" id="titolo" placeholder="Titolo dell'articolo">
      <textarea id="descrizione" placeholder="Descrizione
dell'articolo"></textarea>
      <button id="aggiungi">Aggiungi Articolo</button>
    </div>

    <ul id="articoli">
      <!-- Gli articoli aggiunti dinamicamente appariranno qui -->
    </ul>
  </div>

  <script src="script.js"></script>
</body>
</html>
```

CSS (styles.css)

```
body {
  font-family: Arial, sans-serif;
  margin: 0;
  padding: 0;
  background-color: #f4f4f9;
}

.container {
  width: 80%;
  margin: 0 auto;
  padding: 20px;
  background-color: white;
  box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
  border-radius: 8px;
}

h1 {
  text-align: center;
  color: #333;
}

.form {
  display: flex;
  flex-direction: column;
  margin-bottom: 20px;
}

input, textarea {
  padding: 10px;
  margin-bottom: 10px;
  border: 1px solid #ddd;
  border-radius: 4px;
  font-size: 16px;
}

button {
  padding: 10px;
  background-color: #4CAF50;
  color: white;
  border: none;
  border-radius: 4px;
  cursor: pointer;
}

button:hover {
  background-color: #45a049;
}

ul {
  list-style-type: none;
  padding: 0;
}

li {
  padding: 15px;
  background-color: #e1e1e1;
  margin-bottom: 10px;
  border-radius: 4px;
  display: flex;
  justify-content: space-between;
```

```

        align-items: center;
    }

    button.edit, button.delete {
        background-color: #f1c40f;
        border: none;
        padding: 5px 10px;
        cursor: pointer;
        border-radius: 4px;
    }

    button.delete {
        background-color: #e74c3c;
    }

    button.edit:hover, button.delete:hover {
        opacity: 0.8;
    }

```

JavaScript (script.js)

```

// Variabili per i selettori
const btnAggiungi = document.getElementById("aggiungi");
const listaArticoli = document.getElementById("articoli");
const titoloInput = document.getElementById("titolo");
const descrizioneInput = document.getElementById("descrizione");

// Funzione per aggiungere un articolo
function aggiungiArticolo() {
    const titolo = titoloInput.value.trim();
    const descrizione = descrizioneInput.value.trim();

    if (titolo === "" || descrizione === "") {
        alert("Titolo e descrizione non possono essere vuoti!");
        return;
    }

    const li = document.createElement("li");
    li.classList.add("articolo");

    li.innerHTML = `
        <div>
            <h3>${titolo}</h3>
            <p>${descrizione}</p>
        </div>
        <div>
            <button class="edit">Modifica</button>
            <button class="delete">Rimuovi</button>
        </div>
    `;

    // Aggiungi il nuovo articolo alla lista
    listaArticoli.appendChild(li);

    // Aggiungi gli event listener per modificare o rimuovere
    li.querySelector(".edit").addEventListener("click", function() {
        modificaArticolo(li, titolo, descrizione);
    });
    li.querySelector(".delete").addEventListener("click", function() {
        rimuoviArticolo(li);
    });
}

```

```

    // Pulisce i campi di input
    titoloInput.value = "";
    descrizioneInput.value = "";
}

// Funzione per modificare un articolo
function modificaArticolo(li, titolo, descrizione) {
    titoloInput.value = titolo;
    descrizioneInput.value = descrizione;

    // Rimuove l'articolo dalla lista e permette di modificare
    rimuoviArticolo(li);
}

// Funzione per rimuovere un articolo
function rimuoviArticolo(li) {
    listaArticoli.removeChild(li);
}

// Aggiungi un articolo quando il bottone viene cliccato
btnAggiungi.addEventListener("click", aggiungiArticolo);

// Aggiungi un articolo quando l'utente preme "Enter" dopo aver compilato i
campi
titoloInput.addEventListener("keypress", function(e) {
    if (e.key === "Enter") {
        aggiungiArticolo();
    }
});

```

Analisi Top-down e Bottom-up

1. Analisi Top-down:

In un approccio top-down, si parte dalla visione complessiva del sistema e si scende nei dettagli:

1. **Struttura generale (HTML):** La pagina contiene un titolo, un modulo per inserire dati e una lista di articoli.
2. **Interazione dell'utente (CSS e JavaScript):** L'utente interagisce con il modulo, quindi un bottone "Aggiungi Articolo" inserisce un nuovo articolo nella lista. Inoltre, per ogni articolo, vengono mostrati due pulsanti: uno per modificarlo e uno per eliminarlo.
3. **Comportamento del programma (JavaScript):**
 - Quando l'utente preme "Aggiungi Articolo", viene eseguita la funzione `aggiungiArticolo()`, che crea un nuovo elemento nella lista.
 - La funzione `modificaArticolo()` consente all'utente di modificare un articolo esistente, prelevando il titolo e la descrizione.
 - La funzione `rimuoviArticolo()` consente di rimuovere un articolo dalla lista.

2. Analisi Bottom-up:

In un approccio bottom-up, si inizia dai dettagli implementativi per costruire il sistema completo.

1. **Eventi di JavaScript:**
 - L'evento `click` sul bottone "Aggiungi Articolo" chiama la funzione `aggiungiArticolo()`.

- All'interno di questa funzione, creiamo dinamicamente l'elemento `li`, lo aggiungiamo alla lista e associamo i gestori di eventi per modificare o eliminare l'articolo.
- La funzione `modificaArticolo()` prepopola i campi di input con i dati esistenti, mentre `rimuoviArticolo()` elimina l'elemento dalla lista.

2. Manipolazione del DOM:

- La creazione del nuovo articolo viene gestita tramite `document.createElement()`, e l'inserimento nella lista tramite `appendChild()`.
- Il contenuto dell'articolo viene inserito utilizzando `innerHTML`.

Flusso Operativo:

1. L'utente compila il modulo (titolo e descrizione) e preme il pulsante "Aggiungi Articolo".
2. La funzione `aggiungiArticolo()` crea un nuovo articolo e lo inserisce nell'elenco.
3. L'utente può fare clic su "Modifica" per modificare un articolo esistente, che precompila i campi di input.
4. L'utente può fare clic su "Rimuovi" per eliminare un articolo dalla lista.