1. Récupérer les Enfants et les Frères

Méthode / Propriété	Description
element.children	Retourne une HTMLCollection des éléments enfants (sans les nœuds texte/commentaires)
element.childNodes	Retourne une NodeList de **tous** les nœuds enfants (y compris texte/commentaires)
element.firstElementChild	Retourne le **premier enfant élément**
element.lastElementChild	Retourne le **dernier enfant élément**
element.nextElementSibling	Retourne l'**élément frère suivant**
element.previousElementSibling	Retourne l'**élément frère précédent**
element.parentElement	Retourne le **parent** d'un élément

Exemple : Lister les enfants et frères

```
<!DOCTYPE html>
<html lang="fr">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Manipulation du DOM</title>
</head>

<body>
  <h2>Liste des éléments</h2>
  <ul id="parentElement">
    <li>Item 1</li>
    <li id="item2">Item 2</li>
    <li>Item 3</li>
    <li>Item 4</li>
  </ul>
  <button onclick="afficherEnfants()">Afficher les enfants</button>
  <button onclick="afficherSiblings()">Afficher les frères et sœurs de Item 2</button>
  <script>
    const parent = document.getElementById("parentElement");
    const item2 = document.getElementById("item2");
    // Fonction pour afficher les enfants
    function afficherEnfants() {
      const children = [...parent.children]; // Convertir la collection en tableau
      console.log("Enfants :", children);
      alert("Enfants : " + children.map(child => child.textContent).join(", "));
    }
    // Fonction pour afficher les frères et sœurs de Item 2
    function afficherSiblings() {
      const siblings = [...parent.children].filter(el => el !== item2);
      console.log("Frères et sœurs de Item 2 :", siblings);
      alert("Frères et sœurs de Item 2 : " + siblings.map(sib => sib.textContent).join(", "));
    }
  </script>
</body>
</html>
```

2. Ajouter des éléments

Méthode	Description
parent.appendChild(element)	Ajoute à la fin d'un parent
parent.insertBefore(newElement, existingElement)	Insère avant un élément existant
element.insertAdjacentHTML(position, htmlString)	Insère du HTML avant ou après un élément
parent.append(newElement, otherElement, "text")	Ajoute plusieurs éléments ou texte
parent.prepend(newElement)	Ajoute au début d'un parent

Exemple : Ajouter un élément

```
const parent = document.getElementById("parentElement");

// 1. Créer un élément
const newItem = document.createElement("li");
newItem.textContent = "Nouvel Item";

// 2. Ajouter à la fin
parent.appendChild(newItem);

// 3. Ajouter au début
const firstChild = parent.firstElementChild;
parent.insertBefore(newItem, firstChild);
```

Exemple : Insérer du HTML

```
const list = document.getElementById("parentElement");
list.insertAdjacentHTML("beforeend", "<li>Item Inséré</li>");
```

Valeurs possibles de `insertAdjacentHTML` :

- `"beforebegin"` → Avant l'élément
- `"afterbegin"` → Au début de l'élément (avant son premier enfant)
- `"beforeend"` → À la fin de l'élément (après son dernier enfant)
- `"afterend"` → Après l'élément

3. Supprimer des éléments

Méthode	Description
`element.remove()`	Supprime l'élément lui-même
`parent.removeChild(childElement)`	Supprime un élément enfant
`parent.replaceChild(newElement, oldElement)`	Remplace un élément enfant

Exemple : Supprimer un élément

```
const itemToRemove = document.querySelector("#parentElement li");
itemToRemove.remove();  // Supprime l'élément
```

Exemple : Supprimer un enfant spécifique

```
const parent = document.getElementById("parentElement");
const firstChild = parent.firstElementChild;
parent.removeChild(firstChild);
```

4. Remplacer un élément

```
 const parent = document.getElementById("parentElement");
const newItem = document.createElement("li");
newItem.textContent = "Élément Remplacé";

const oldItem = parent.children[1];
parent.replaceChild(newItem, oldItem);
```

5. Cloner un élément

Si tu veux copier un élément et l'ajouter ailleurs :

```
const parent = document.getElementById("parentElement");
const clone = parent.children[0].cloneNode(true); // true = clone avec ses enfants
parent.appendChild(clone);
```

Exemple Complet

```
<ul id="parentElement">
  <li>Item 1</li>
  <li>Item 2</li>
  <li>Item 3</li>
</ul>
<button onclick="ajouterElement()">Ajouter</button>
<button onclick="supprimerElement()">Supprimer</button>

<script>
  const parent = document.getElementById("parentElement");

  function ajouterElement() {
    const newItem = document.createElement("li");
    newItem.textContent = "Nouvel Item";
    parent.appendChild(newItem);
  }

  function supprimerElement() {
    if (parent.children.length > 0) {
      parent.removeChild(parent.lastElementChild);
    }
  }
</script>
```

6. Méthodes principales pour manipuler un tableau HTML

a. Ajouter une ligne : insertRow(index)

Ajoute une nouvelle ligne dans un tableau à l'index donné (**0** pour ajouter au début, `-1` pour ajouter à la fin).

b. Supprimer une ligne : deleteRow(index)

Supprime une ligne à l'index donné.

c. Ajouter une cellule : insertCell(index)

Ajoute une cellule dans une ligne donnée.

d. Supprimer une cellule : deleteCell(index)

Supprime une cellule d'une ligne donnée.

e. Accéder aux lignes et cellules :

- `table.rows` → Récupère toutes les lignes du tableau.
- `row.cells` → Récupère toutes les cellules d'une ligne.

Exemple récapulatif :

```
<!DOCTYPE html>
<html lang="fr">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width,
initial-scale=1.0">
    <title>Liste des Équipes Marocaines</title>
    <style>
        table {
            width: 100%;
            border-collapse: collapse;
            margin-top: 20px;
        }
        th, td {
```

```
            border: 1px solid black;
            padding: 8px;
            text-align: center;
        }
        img {
            width: 50px;
            height: 50px;
        }
    </style>
</head>
<body>

    <h2>Liste des Équipes Marocaines</h2>

    <table id="teamsTable">
        <thead>
            <tr>
                <th>Logo</th>
                <th>Nom de l'Équipe</th>
                <th>Action</th>
            </tr>
        </thead>
        <tbody>
            <!-- Les équipes seront ajoutées dynamiquement
ici -->
        </tbody>
    </table>

    <script>
        // Liste des équipes marocaines
        let teams = [
            { id: 1, name: "Wydad Casablanca", logo:
"https://logowik.com/content/uploads/images/wydad-
casablanca1494.logowik.com.webp" },
            { id: 2, name: "Raja Casablanca", logo:
"https://th.bing.com/th/id/R.27d69a994f8de456fc4c3c8f8a48475b
?rik=5k%2bMCMiavnRb6g&pid=ImgRaw&r=0" },
            { id: 3, name: "FUS Rabat", logo:
```

```
"https://vectorseek.com/wp-content/uploads/2023/07/Fus-Rabat-Logo-Vector.jpg" }
        ];

        const tableBody = document.querySelector("#teamsTable tbody");

        // Fonction pour afficher les équipes
        function renderTeams() {
            tableBody.innerHTML = ""; // Nettoyer le tableau avant de le remplir

            teams.forEach((team, index) => {
                let row = tableBody.insertRow();

                // Colonne du logo
                let cellLogo = row.insertCell(0);
                let img = document.createElement("img");
                img.src = team.logo;
                img.alt = team.name;
                cellLogo.appendChild(img);

                // Colonne du nom
                let cellName = row.insertCell(1);
                cellName.textContent = team.name;

                // Colonne du bouton supprimer
                let cellAction = row.insertCell(2);
                let button = document.createElement("button");
                button.textContent = "Supprimer";
                button.onclick = () => supprimerEquipe(index);
                cellAction.appendChild(button);
            });
        }

        // Fonction pour supprimer une équipe
```

```
        function supprimerEquipe(index) {
            teams.splice(index, 1); // Supprimer l'équipe du
tableau
            renderTeams(); // Mettre à jour l'affichage
        }

        // Affichage initial des équipes
        renderTeams();
    </script>

</body>
</html>
```