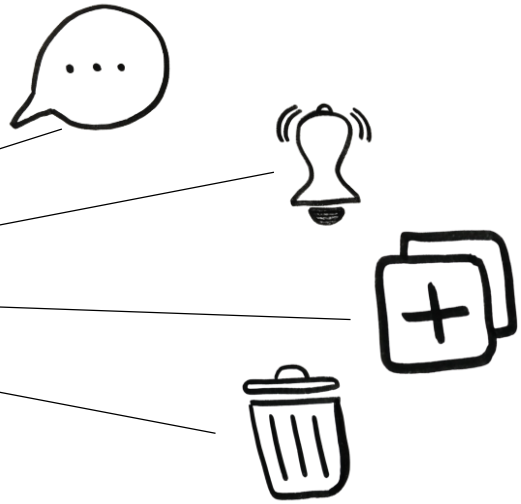
Objectif du module

À la fin de ce module, l'étudiant sera capable de :

- Créer dynamiquement des éléments HTML
- Les insérer dans la page
- Les supprimer
- Vider une zone d'affichage
- Comprendre la différence entre innerHTML et createElement()
- Construire une interface entièrement générée par JavaScript

Table des matières

1. Pourquoi aller plus loin ?



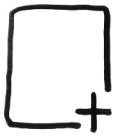
Jusqu'ici, on modifiait des éléments existants.

Maintenant on veut :

- Ajouter un commentaire
- Ajouter une notification
- Ajouter un élément dans une liste
- Supprimer un élément

Donc manipuler la structure elle-même.

2. Créer un élément



```
const div = document.createElement("div");
```

Créer ne signifie pas encore l'afficher.

3. L'ajouter à la page

```
parent.appendChild(div);
```

appendChild() ajoute l'élément à la fin du parent.

Exemple : ajouter un message

```
<!DOCTYPE html>
<html lang="fr">
<head>
    <meta charset="UTF-8">
    <title>Compteur de caractères</title>
    <link rel="stylesheet" href="css/bootstrap.css">
</head>
<body>
    <button id="ajouter" class="btn btn-primary">
        Ajouter un message
    </button>
    <div id="zone" class="mt-3"></div>
    <script>
        const bouton = document.querySelector("#ajouter");
        const zone = document.querySelector("#zone");
        bouton.addEventListener("click", function() {
        const p = document.createElement("p");
        p.textContent = "Nouveau message";
        p.classList.add("alert", "alert-info");
        zone.appendChild(p);
        });
    </script>
</body>
```

```
</html>
```

4. Supprimer un élément

On peut utiliser :

```
element.remove();
```

Exemple : bouton supprimer

```
<!DOCTYPE html>
<html lang="fr">
<head>
    <meta charset="UTF-8">
    <title></title>
    <link rel="stylesheet" href="css/bootstrap.css">
</head>
<body>
    <button id="ajouter" class="btn btn-primary">
        Ajouter un message
    </button>
    <div id="zone" class="mt-3"></div>

    <script>
        const bouton = document.querySelector("#ajouter");
        const zone = document.querySelector("#zone");

        bouton.addEventListener("click", function() {
            const p = document.createElement("p");
            p.classList.add("alert", "alert-info", "d-flex", "justify-content-between");

            p.textContent = "Nouveau message";

            // bouton supprimer
            const btnSupprimer = document.createElement("button");
            btnSupprimer.textContent = "Supprimer";
            btnSupprimer.classList.add("btn", "btn-danger", "btn-sm");

            btnSupprimer.addEventListener("click", function() {
                p.remove();
            });

            p.appendChild(btnSupprimer);
            zone.appendChild(p);
        });
    </script>
</body>
</html>
```

5. Vider une zone

```
zone.innerHTML = "";
```

Supprime tout le contenu.

innerHTML	createElement

`zone.innerHTML += "<p>Message</p>";`	`const p = document.createElement("p");`
	`p.textContent = "Nouveau message";`
	`zone.appendChild(p);`

✓ Rapide ✗ Moins sécurisé ✗ Moins contrôlé 	✓ Plus propre ✓ Plus contrôlé ✓ Meilleure pratique 

Exercice — Liste de tâches – todo.html

Créer une liste de tâches dynamique.

Créer :

- Un champ texte
- Un bouton “Ajouter”
- Une zone liste

Comportement :

- Ajouter une tâche à la liste
- Chaque tâche peut être supprimée en cliquant dessus

Contraintes :

- Utiliser createElement()
- Utiliser appendChild()
- Utiliser trim()
- Ne pas utiliser innerHTML pour ajouter les tâche

Exercice — Système de notifications

Créer un système de notifications dynamiques.

Créer :

- Un bouton “Afficher notification”
- Un bouton “Supprimer tout”
- Une zone notifications

Comportement :

- Chaque clic ajoute une notification

- Chaque notification peut être supprimée individuellement
- “Supprimer tout” vide la zone

Contraintes

- Utiliser `createElement()`
- Utiliser `appendChild()`
- Utiliser `remove()`
- Utiliser `innerHTML = ""` uniquement pour vider

Exercice — Mini galerie dynamique – galerie.html

Ajouter des images dynamiquement.

Créer :

- Un bouton “Ajouter image”
- Une zone galerie

Comportement :

- Ajouter une image
- Bouton pour supprimer chaque image
- Supprimer enlève uniquement l’image concernée

Contraintes

- Utiliser `createElement()`
- Utiliser `appendChild()`
- Ne pas utiliser `innerHTML +=`
- Ajouter un événement sur chaque bouton créé

Préparation

1. Créer un dossier nommé **img** dans le dossier du projet.
2. Placer au minimum 3 images dedans.
3. Nommer les images exactement :

```
image1.jpg
image2.jpg
image3.jpg
```

Penser à respecter :

- la casse (image ≠ Image)
- l’extension (.jpg)
- l’orthographe exacte

Le chemin sera construit dynamiquement :

```
img.src = "img/image" + indexImage + ".jpg";
```

Exercice — diaporamaimage.html

Créer, un diaporama d'image avec deux boutons (previous, next)

Créer :

- Bouton "previous"
- Bouton "next"
- Zone image

Comportement :

- À chaque clic → nouvelle image
- Images stockées dans tableau

Contraintes :

- Utiliser tableau
- Utiliser boucle ou index
- Utiliser src dynamique

Exercice – afficher une image au clic - tache.html

Créer une page contenant :

- Une page vide (ou un titre), sans image au départ.

Préparation :

- Créer un dossier tache/
- Ajouter une image : tache /rouge..png (ou img tache image1.jpg)

Comportement attendu :

- Au clic n'importe où dans la page :
  - créer un élément <img>
  - lui donner un src vers l'image du dossier img/
  - l'ajouter dans la page

Contraintes :

- Utiliser document.addEventListener("click", …)
- Utiliser createElement("img")
- Utiliser appendChild
- Le fichier image doit être local (dossier img/)

Variante (bonus) :

- Placer l'image à l'endroit du clic avec event.clientX / event.clientY.

event est un objet

event.clientX → position horizontale du clic

event.clientY → position verticale du clic

position: absolute permet de placer un élément précisément

Exercice —Tache aléatoire (4 images) - tache2.html

Objectif : choisir une image aléatoire depuis un tableau.

Préparation :

- Dossier img/ contenant exactement :
  - tache1.png, tache2.png, tache3.png, tache4.png

Créer une page contenant :

- Une page vide (ou un titre).

Comportement attendu :

- À chaque clic :
  - choisir une image aléatoire parmi les 4
  - créer <img> et l'ajouter à la page

Contraintes :

- Utiliser un tableau : const taches = ["tache1.png", ...]
- Utiliser un tirage aléatoire
- Utiliser createElement + appendChild
- Ne pas utiliser innerHTML +=