

1. Définition et Structure du DOM

✓ Qu'est-ce que le DOM ?

Le **DOM** (Document Object Model) est :

✓ Une **API (Interface de Programmation)** fournie par le navigateur.

✓ Un **standard du W3C** qui définit comment JavaScript peut accéder et modifier une page web.

➡ En résumé : **le DOM est une représentation de ta page HTML sous forme d'arbre d'objets manipulables en JavaScript.**

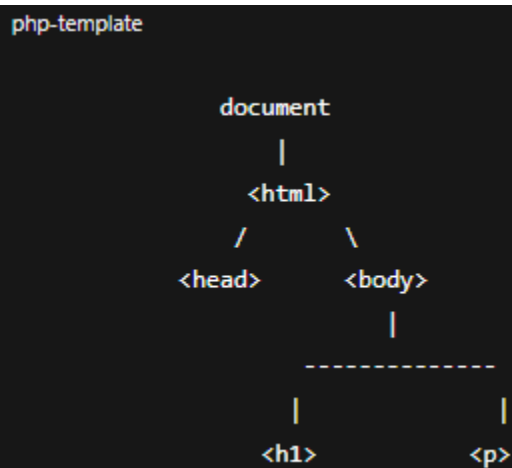
✓ Structure en Arbre

Le DOM transforme ceci :

```
html

<!DOCTYPE html>
<html>
  <body>
    <h1>Titre</h1>
    <p>Un paragraphe</p>
  </body>
</html>
```

En ceci :



- ✓ Nœud racine → document
 - ✓ Chaque balise HTML est un nœud.
 - ✓ Chaque texte est aussi un nœud texte.
-

2. Pourquoi le DOM est essentiel à JavaScript ?

Sans DOM, JavaScript **ne pourrait pas interagir avec la page**.

Grâce au DOM, JavaScript peut :

- ✓ Modifier le contenu (texte, images...)
- ✓ Modifier le style (couleurs, taille...)
- ✓ Ajouter ou supprimer des éléments
- ✓ Réagir aux actions utilisateur (clic, saisie, scroll...)

Ce processus s'appelle :

La manipulation du DOM

C'est l'art de modifier une page web dynamiquement via JavaScript.

3. Méthodes Clés d'Interaction avec des Exemples

Voici un **mini-exemple simple pour chaque point, comme demandé**.

A. Sélection d'Éléments

1) document.getElementById()

HTML :

```
html

<h1 id="titre">Bonjour</h1>
```

JS :

```
js

let p = document.querySelector(".info");
p.textContent = "Texte modifié !";
```

3) document.querySelectorAll()

JS :

```
js

let items = document.querySelectorAll("li");
items[0].textContent = "Premier élément modifié";
```

B. Gestion des Événements — addEventListener()

HTML :

```
html

<button id="btn">Clique-moi</button>
<p id="msg"></p>
```

JS :

```
js

document.getElementById("btn").addEventListener("click", function () {
    document.getElementById("msg").textContent = "Bouton cliqué !";
});
```

C. Modification du contenu

1) textContent

```
js

element.textContent = "Nouveau texte";
```

2) innerHTML

```
js

element.innerHTML = "<strong>Texte en gras</strong>";
```

D. Modification du style et des classes

1) Ajouter / retirer une classe

```
js  
  
element.classList.add("active");  
element.classList.remove("active");
```

2) Modifier le style en direct

```
js  
  
element.style.color = "red";  
element.style.backgroundColor = "yellow";
```

E. Manipulation de la structure (Ajouter / Supprimer)

1) Créer un élément

```
js  
  
let newDiv = document.createElement("div");  
newDiv.textContent = "Bonjour !";
```

2) L'ajouter dans la page

```
js  
  
document.body.appendChild(newDiv);
```

3) Supprimer un élément

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

En conclusion , Le DOM constitue le cœur de l'interactivité moderne sur le web. En représentant une page HTML sous forme d'un arbre d'objets accessibles via JavaScript, il permet aux développeurs de **modifier dynamiquement le contenu, la structure et le style d'une page web**, même après son chargement.

Grâce aux méthodes essentielles comme `querySelector`, `addEventListener`, `classList`, ou encore `createElement`, JavaScript peut :

- sélectionner précisément n'importe quel élément,
- réagir aux actions de l'utilisateur,
- mettre à jour du texte, des classes ou du CSS,
- ajouter ou supprimer des éléments du document.

Cette capacité à **manipuler en temps réel l'interface** transforme une simple page statique en une application web riche, fluide et interactive.

C'est ce qui rend aujourd'hui possible les menus dynamiques, les formulaires intelligents, les animations visuelles, ou encore les SPA (Single Page Applications).

En maîtrisant le DOM et ses événements, le développeur acquiert l'un des piliers indispensables pour créer des expériences web modernes, ergonomiques et réactives.