

Le DOM et les évènements



1. Introduction

Le Document Object Model

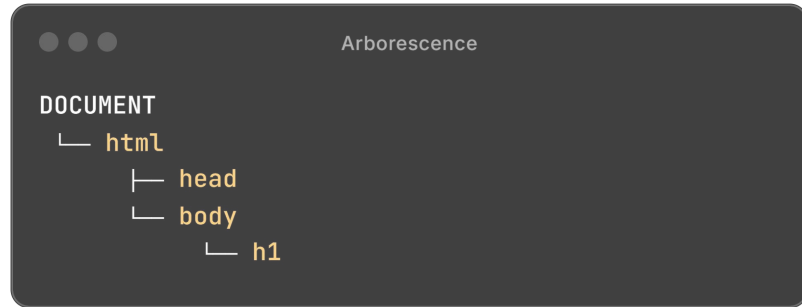
- Permet à JavaScript de lire et modifier une page HTML comme une arborescence
- Chaque élément HTML devient un objet JavaScript

```
DOM

<body>
  <h1>Titre</h1>
</body>

...

document.body // sélectionne <body> et ses enfants
```



Pourquoi utiliser le DOM ?

- Dynamiser une page sans la recharger
- Créer des interfaces interactives

2. Sélection des éléments

Sélectionner un élément

- Pour lire ou modifier son contenu
- Pour lui associer des événements (clic, survol, etc.)
- Pour l'inspecter ou le manipuler dynamiquement, lui appliquer un style ...

Méthodes de sélection

- Par id, classe, tag...

```
Exemples de selecteurs

// Sélectionne un élément unique par son ID
const titre = document.getElementById("monTitre");

// Sélectionne plusieurs éléments ayant une même classe
const boutons = document.getElementsByClassName("btn");

// Sélectionne tous les éléments d'un certain type (div, p, li, ...)
const paragraphes = document.getElementsByTagName("p");

// Sélectionne le premier élément qui correspond au sélecteur CSS
const titre = document.querySelector("h1");
```

3. Modification des éléments

Modifier une fois la sélection

```
Modification

// Modifie ou lit le texte brut de l'élément (sans HTML)

document.querySelector("h1").textContent = "Nouveau titre";

// Modifier le contenu HTML
document.querySelector("div").innerHTML = "<p>Paragraphe <strong>modifié</strong></p>";

// Modifier les attributs
const lien = document.querySelector("a");
lien.getAttribute("href"); // lit l'attribut
lien.setAttribute("href", "https://google.com"); // modifie l'attribut

// Modifier les styles
const bouton = document.querySelector("button");
bouton.style.backgroundColor = "blue";
bouton.style.fontSize = "20px";

// Ajouter, retirer ou alterner une classe CSS
element.classList.add("actif");
element.classList.remove("cache");
element.classList.toggle("highlight");
```

4. Création d'éléments

Création / Ajout dans le DOM



Gestion des éléments

```
// 1. Création de l'élément
const newDiv = document.createElement("div");

// 2. Ajout de contenu et d'une classe
newDiv.textContent = "Je suis un nouvel élément !";
newDiv.classList.add("message");

// 3. Insertion dans le DOM
const container = document.getElementById("conteneur");
container.appendChild(newDiv);
```

Exercise 1

4. Gestion des événements

Qu'est ce qu'un événement ?

- Une action utilisateur ou du navigateur détectée par le navigateur
- JavaScript permet d'écouter ces événements pour agir dynamiquement

Ecouter un évènement

- `element.addEventListener("type", fonction);`
- **type** : nom de l'évènement ("click", "mouseover", "submit", ...)
- **fonction** : code à exécuter lorsqu'il se produit



Un exemple

```
document.querySelector("button").addEventListener("click", () => {  
    alert("Bouton cliqué !");  
});
```



Un exemple

```
element.removeEventListener("click", fonction);
```

Événements courants

click	Clic sur un élément
submit	Soumission de formulaire
input	Saisie dans un champ de formulaire
keydown	Touche du clavier enfoncée
mouseover	Curseur survole un élément
change	Changement de valeur (input, select)

Exercise 2



TP 2 : Validation des acquis