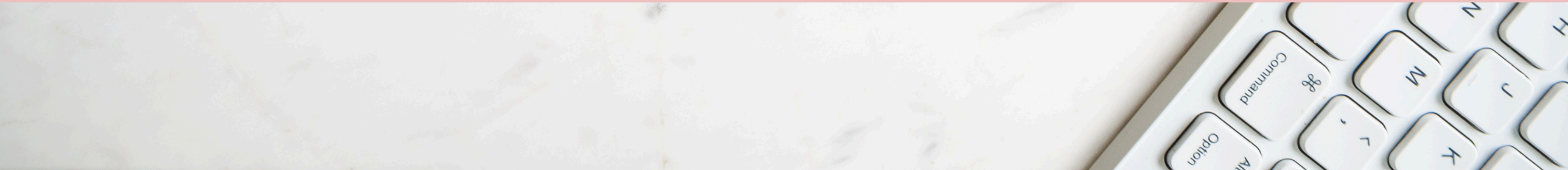
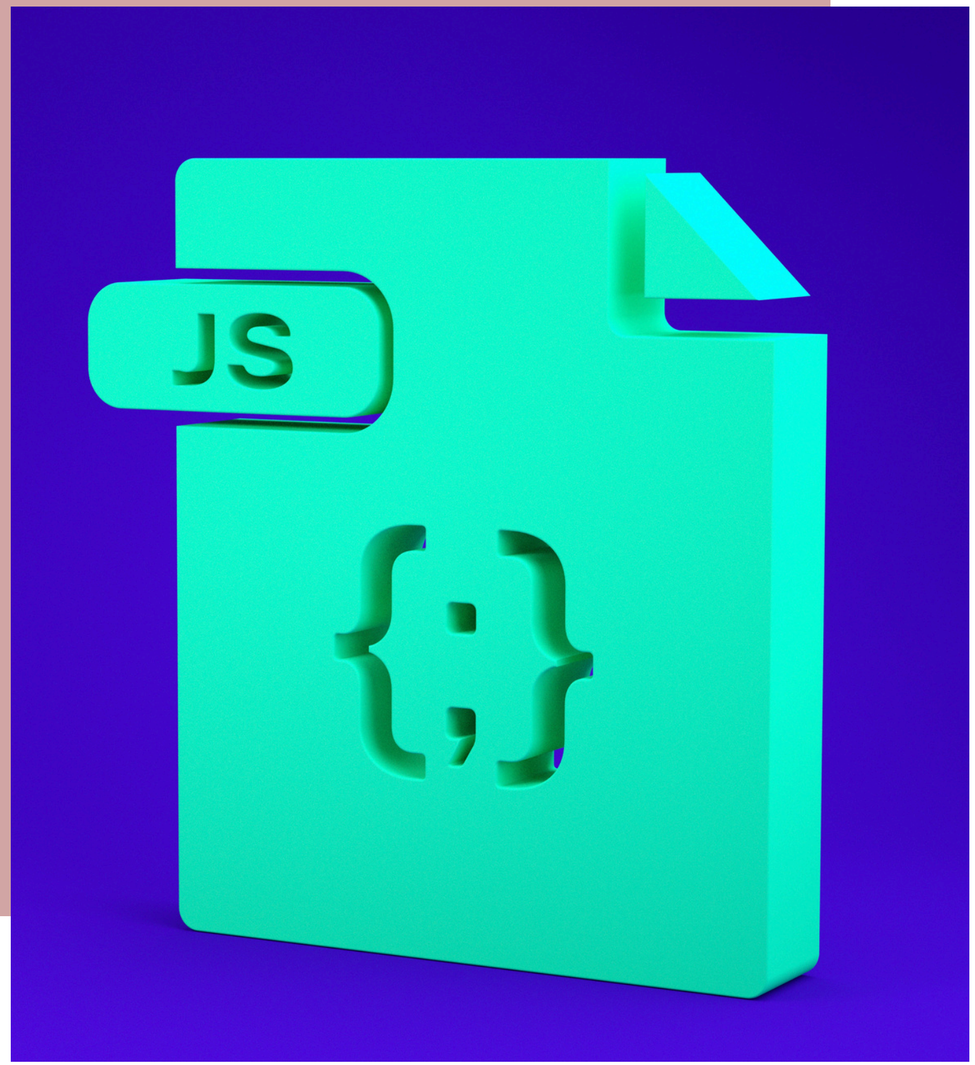
JavaScript côté Navigateur

Apprendre à utiliser JavaScript côté navigateur pour manipuler le DOM, gérer les événements, et créer des interactions dynamiques sur les pages web.

Introduction au JavaScript côté Navigateur et Manipulation du DOM





Qu'est-ce que JavaScript côté Navigateur ?

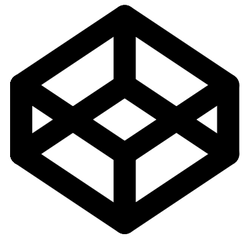
RÔLE :

Ajoute de l'interactivité aux pages web.

Manipule le contenu de la page en temps réel.

Réagit aux actions des utilisateurs (clics, saisie, défilement).

Pourquoi Utiliser JavaScript côté Navigateur ?



Interactivité

Crée des expériences utilisateur dynamiques.

Performance

Réduit le besoin d'interaction serveur pour certaines tâches.

Flexibilité

Permet des mises à jour en temps réel du contenu de la page sans rechargement complet.

Vue d'Ensemble des Fonctionnalités JavaScript côté Navigateur

MANIPULATION DU DOM (DOCUMENT OBJECT MODEL)

Modifier et mettre à jour le contenu HTML et CSS.

GESTION DES ÉVÉNEMENTS

Réagir aux actions de l'utilisateur telles que les clics et les saisies.

VALIDATION DES FORMULAIRES

Assurer la validité des données saisies avant l'envoi au serveur.

ANIMATIONS ET EFFETS

Créer des transitions fluides et des animations sur les éléments de la page.

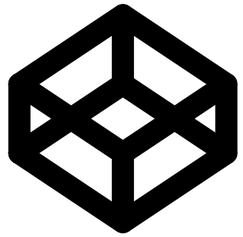
HTML CSS JS Result

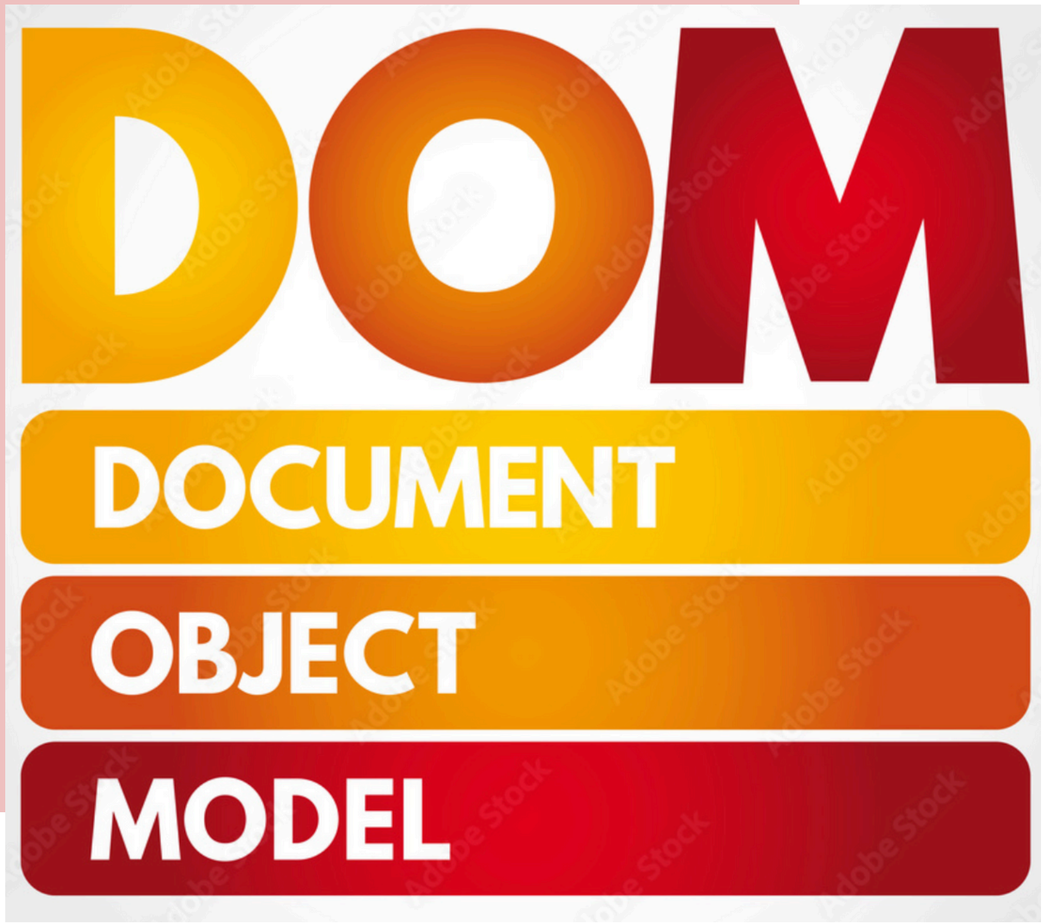
```
document.addEventListener('DOMContentLoaded',
function() {
    document.body.innerHTML += '<p>Hello, JavaScript!
</p>';
});
```

Hello, JavaScript!

Premier Script JavaScript

MODIFIER UN ÉLÉMENT DU DOM.





Découverte du DOM

(DOCUMENT OBJECT MODEL)

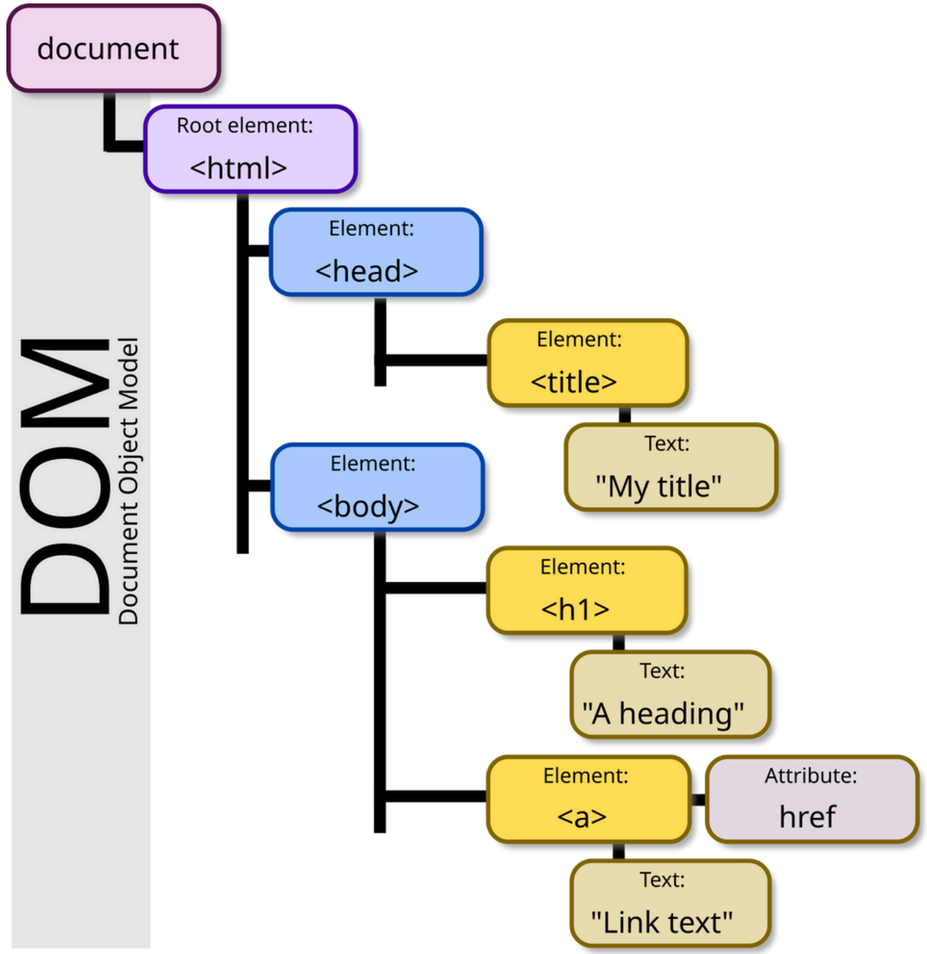
Comprendre et manipuler la structure des documents web avec JavaScript

Qu'est-ce que le DOM ?

Définition

Le DOM est une interface de programmation qui permet aux scripts de mettre à jour le contenu, la structure et le style des documents web.

Le DOM représente la page de manière arborescente, permettant aux développeurs d'accéder et de manipuler ses éléments.



Les composants du DOM

DOCUMENT

Le nœud racine représentant le document entier.

ÉLÉMENTS (ELEMENT NODES)

Les balises HTML comme , <p>, <a>, etc.

ATTRIBUTS (ATTRIBUTE NODES)

Les propriétés des éléments, comme class, id, src, etc.

TEXTE (TEXT NODES)

Le contenu textuel à l'intérieur des éléments.

Accéder au DOM avec JavaScript

SÉLECTION DES ÉLÉMENTS

GetElementById

Sélectionne un élément par son ID.

getElementsByTagName

Sélectionne tous les éléments avec un certain nom de balise.

getElementsByClassName

Sélectionne tous les éléments avec une classe spécifique.

querySelector et querySelectorAll

Sélectionne des éléments en utilisant des sélecteurs CSS.

```
var elementById = document.getElementById('exampleId');
var elementsByClass = document.getElementsByClassName('exampleClass');
var elementsByTag = document.getElementsByTagName('p');
var elementByQuery = document.querySelector('.exampleClass');
var elementsByQueryAll = document.querySelectorAll('.exampleClass');
```

Accéder au DOM avec JavaScript

MODIFIER LE CONTENU

innerHTML et textContent

Modifier le HTML ou le texte d'un élément.

MODIFIER LES CLASSES

classList

Ajouter, enlever ou basculer des classes CSS.

MODIFIER LES ATTRIBUTS

setAttribute et getAttribute

Modifier ou obtenir les attributs des éléments.

```
var element = document.getElementById('exampleId');
element.innerHTML = 'Nouveau contenu';
element.setAttribute('src', 'nouvelle-image.jpg');
element.classList.add('nouvelle-classe');
```

Ajouter et Supprimer des Éléments

AJOUTER DES ÉLÉMENTS

createElement

Créer un nouvel élément.

appendChild et insertBefore

Ajouter un élément à la page.

SUPPRIMER DES ÉLÉMENTS

removeChild

Supprimer un élément enfant.

```
var newElement = document.createElement('div');
newElement.textContent = 'Je suis un nouveau div';
document.body.appendChild(newElement);

var parentElement = document.getElementById('parent');
var childElement = document.getElementById('child');
parentElement.removeChild(childElement);
```

Traverser le DOM

NAVIGUER DANS LE DOM

parentNode et childNodes

Accéder au parent et aux enfants d'un nœud.

firstChild, lastChild

Accéder au premier et au dernier enfant.

nextSibling et previousSibling

Accéder aux nœuds voisins.

```
var parent = element.parentNode;
var children = element.childNodes;
var firstChild = element.firstChild;
var nextSibling = element.nextSibling;
```

Pratique - 1

Créer une page de type FAQ avec 4 questions et 4 boutons “en savoir plus”
A l’état initial les 4 réponses sont vides
Lorsque l’on clique sur un bouton, la réponse correspondante s’affiche (bonus et les autres réponses se vident)