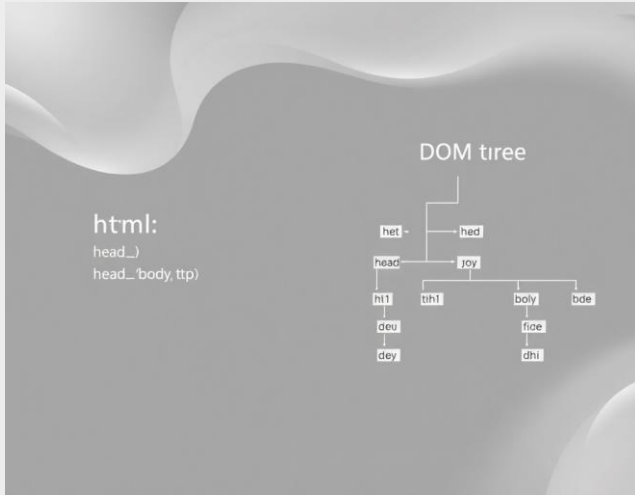




# Introduction à JavaScript

Une exploration des concepts fondamentaux pour débiter en programmation web.

# Qu'est-ce que le DOM (Document Object Model) ?



Le DOM est une représentation structurée de votre document HTML. Il permet à JavaScript d'interagir avec la page web et d'en modifier le contenu, la structure et le style. Pensez-y comme à un arbre généalogique de votre page, où chaque élément est un membre de la famille que vous pouvez manipuler.

# Manipuler le DOM : Exemple Pratique

Avec JavaScript, nous pouvons sélectionner des éléments et changer leurs propriétés. Par exemple, nous pouvons trouver un titre par son ID et modifier son texte.

## Code JavaScript

```
// Ce code sélectionne l'élément avec l'ID 'titrePrincipal' et  
change son contenu.
```

```
let titre = document.getElementById('titrePrincipal');  
titre.innerHTML = 'Bonjour le Monde !';
```

## Résultat



# Outils de débogage JavaScript



Le débogage est l'art de trouver et de corriger les erreurs dans votre code. Les navigateurs modernes incluent des outils de développement puissants pour vous aider. La console JavaScript est l'un des plus importants.

# La console JavaScript : Un outil essentiel

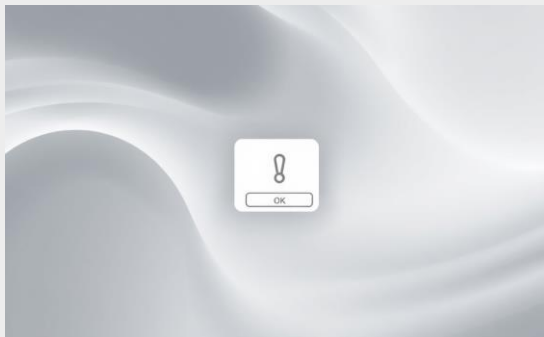
La console est une interface en ligne de commande dans votre navigateur qui vous permet de :

- Afficher des messages et des valeurs de variables avec `console.log()`.
- Tester des extraits de code JavaScript en direct.
- Voir les erreurs et les avertissements générés par votre script.
- Interagir avec le DOM de la page.

Pour l'ouvrir, faites un clic droit sur une page web et choisissez 'Inspecter', puis allez à l'onglet 'Console'.

# Interactions avec l'utilisateur

JavaScript fournit des boîtes de dialogue simples pour communiquer avec l'utilisateur directement depuis le navigateur.



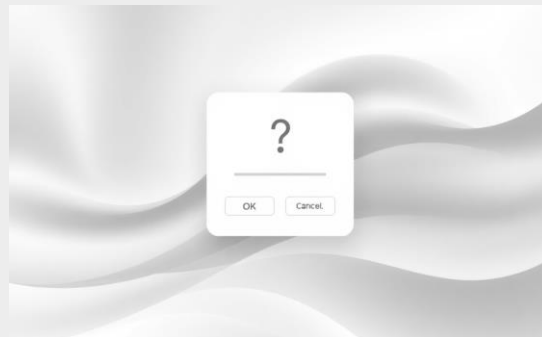
## **alert()**

Affiche une boîte de dialogue avec un message et un bouton 'OK'. Idéal pour notifier l'utilisateur.



## **prompt()**

Affiche une boîte de dialogue avec un message, un champ de saisie de texte et des boutons 'OK'/'Annuler'. Permet de récupérer une entrée de l'utilisateur.



## **confirm()**

Affiche une boîte de dialogue avec une question et des boutons 'OK'/'Annuler'. Permet de demander une confirmation (oui/non).

# Variables en JavaScript

Une variable est un conteneur pour stocker des données. En JavaScript, nous utilisons principalement les mots-clés `let`, `const` et (anciennement) `var` pour les déclarer.

## 1 **var**

L'ancienne manière de déclarer des variables. A une portée de fonction (function scope), ce qui peut parfois prêter à confusion. Son utilisation est aujourd'hui déconseillée.

## 2 **let**

Introduit en ES6. Permet de déclarer des variables dont la valeur peut changer. A une portée de bloc (block scope), ce qui est plus prévisible.

## 3 **const**

Introduit en ES6. Permet de déclarer des constantes, c'est-à-dire des variables dont la valeur ne peut pas être réaffectée. A également une portée de bloc.

# Règles de nommage des variables

Pour que votre code soit lisible et valide, suivez ces conventions :

## À faire

- Commencer par une lettre, un underscore (\_) ou un dollar (\$).
- Utiliser le 'camelCase' pour les noms composés (ex: `nomUtilisateur`).
- Choisir des noms descriptifs et clairs (ex: `ageClient` plutôt que `ac`).

## À ne pas faire

- Commencer par un chiffre.
- Utiliser des mots-clés réservés par JavaScript (ex: `let`, `if`, `for`).
- Utiliser des espaces ou des tirets dans les noms.



# Types de données en JavaScript

JavaScript est un langage à typage dynamique, ce qui signifie qu'une même variable peut contenir différents types de données. Voici les principaux types primitifs :

## **String (Chaîne de caractères)**

Représente du texte. Doit être entouré de guillemets simples (' '), doubles (" ") ou obliques (` `). Ex: 'Bonjour'.

## **Number (Nombre)**

Représente les nombres, qu'ils soient entiers ou à virgule flottante. Ex: 42, 3.14.

## **Boolean (Booléen)**

Représente une valeur logique. Ne peut avoir que deux valeurs : `true` (vrai) ou `false` (faux).

## **Object (Objet)**

Un type complexe qui permet de stocker des collections de données et des entités plus complexes.

# Concaténation : Unir des chaînes

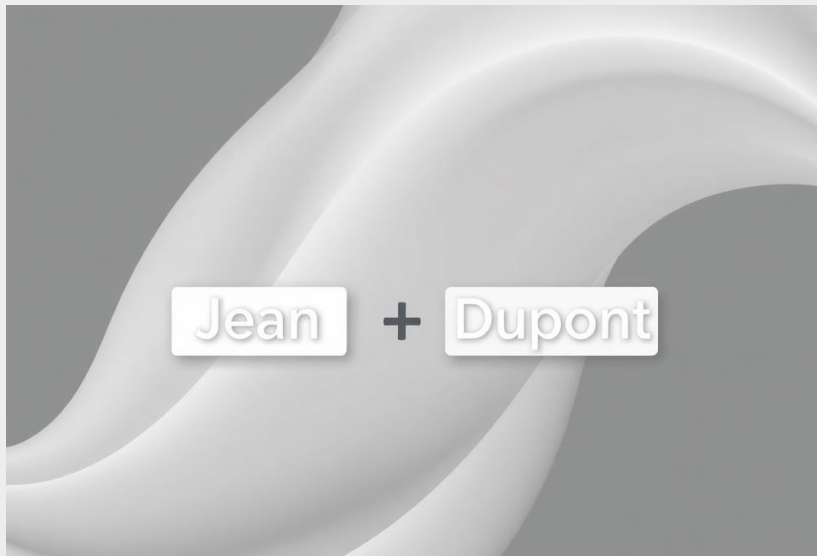
La concaténation est l'opération qui consiste à assembler plusieurs chaînes de caractères pour n'en former qu'une seule. En JavaScript, l'opérateur `+` est le plus souvent utilisé à cette fin.

## Exemple de code

```
let prenom = 'Jean';  
let nom = 'Dupont';  
  
let nomComplet = prenom + ' ' + nom;  
  
console.log(nomComplet);
```

## Sortie dans la console

Jean Dupont



# Résumé des points clés

- Le DOM est l'interface de JavaScript pour interagir avec le HTML.
- La console est votre meilleur ami pour le débogage.
- Utilisez ``alert``, ``prompt`` et ``confirm`` pour les interactions simples.
- Préférez ``let`` et ``const`` à ``var`` pour déclarer vos variables.
- Chaque donnée a un type : String, Number, Boolean, etc.
- Utilisez l'opérateur ``+`` pour concaténer des chaînes de caractères.

Continuez à pratiquer pour maîtriser ces bases !