

# Cours JavaScript

# Introduction

JavaScript est le langage de script côté client le plus populaire et le plus utilisé. Les scripts côté client font référence aux scripts qui s'exécutent dans votre navigateur Web. JavaScript est conçu pour ajouter de l'interactivité et des effets dynamiques aux pages Web en manipulant le contenu renvoyé par un serveur Web.

JavaScript a été initialement développé sous le nom de **LiveScript** par **Netscape** au milieu des années **1990**. Il a ensuite été renommé **JavaScript** en **1995** et est devenu une norme **ECMA en 1997**.

Désormais, JavaScript est le langage de script côté client standard pour les applications Web, et il est pris en charge par pratiquement tous les navigateurs Web disponibles aujourd'hui, tels que Google Chrome, Mozilla Firefox, Apple Safari, etc.

JavaScript est un langage orienté objet et présente également certaines similitudes de syntaxe avec le langage de programmation Java. Mais JavaScript n'est en aucun cas lié à Java.

JavaScript est officiellement maintenu par l'ECMA (European Computer Manufacturers Association) en tant qu'ECMAScript. ECMAScript 6 (ou ES6) est la dernière version majeure de la norme ECMAScript.

# Ce que vous pouvez faire avec JavaScript

Il y a beaucoup plus de choses que vous pouvez faire avec JavaScript.

- Vous pouvez modifier le contenu d'une page Web en ajoutant ou en supprimant des éléments.
- Vous pouvez modifier le style et la position des éléments sur une page Web.
- Vous pouvez surveiller les événements tels que le clic de la souris, le survol, etc. et y réagir.
- Vous pouvez effectuer et contrôler des transitions et des animations.
- Vous pouvez créer des fenêtres contextuelles d'alerte pour afficher des informations ou des messages d'avertissement à l'utilisateur.
- Vous pouvez effectuer des opérations basées sur les entrées de l'utilisateur et afficher les résultats.
- Vous pouvez valider les entrées de l'utilisateur avant de les soumettre au serveur.

La liste ne s'arrête pas là, il y a beaucoup d'autres choses intéressantes que vous pouvez faire avec JavaScript. Vous en apprendrez plus sur chacun d'eux en détail dans les prochains chapitres.

# Ce que couvre ce cours

Ce cours de JavaScript couvre tous les concepts de programmation fondamentaux, y compris les types de données, les opérateurs, la création et l'utilisation de variables, la génération de sorties, la structuration de votre code pour prendre des décisions dans vos programmes ou pour boucler plusieurs fois sur le même bloc de code, créer et manipuler des chaînes. et des tableaux, définissant et appelant des fonctions, etc.

Une fois que vous êtes à l'aise avec les bases, vous passerez au niveau suivant qui explique l'idée des objets, le modèle d'objet de document (DOM) et le modèle d'objet de navigateur (BOM), ainsi que la façon d'utiliser le JavaScript natif des objets tels que Date, Math, etc., et effectuer des conversions de type.

Enfin, vous explorerez certains concepts avancés tels que les écouteurs d'événements, la propagation des événements, l'emprunt de méthodes à d'autres objets, le comportement de levage de JavaScript, l'encodage et le décodage des données JSON, ainsi qu'un aperçu détaillé des nouvelles fonctionnalités introduites dans ECMAScript 6 (ou ES6).

# JavaScript Mise en route

Ici, vous apprendrez à quel point il est facile d'ajouter de l'interactivité à une page Web à l'aide de JavaScript. Mais, avant de commencer, assurez-vous que vous avez des connaissances pratiques en HTML et CSS.

Eh bien, commençons par le langage de script côté client le plus populaire.

# Ajout de JavaScript à vos pages Web

Il existe généralement **trois façons** d'ajouter JavaScript à une page Web :

Incorporation du code JavaScript entre une paire de balises `<script>` et `</script>`

Créer un fichier JavaScript externe avec l'extension `.js`, puis le charger dans la page via l'attribut `src` de la balise `<script>`.

Placer le code JavaScript directement dans une balise HTML en utilisant les attributs de balise spéciaux tels que `onclick`, `onmouseover`, `onkeypress`, `onload`, etc.

Les slides suivants décrivent chacune de ces procédures en détail

# Intégration du code JavaScript

Vous pouvez intégrer le code JavaScript directement dans vos pages Web en le plaçant entre les balises `<script>` et `</script>`

La balise `<script>` indique au navigateur que les instructions contenues doivent être interprétées comme un script exécutable et non comme du HTML. Voici un exemple :

```
<!DOCTYPE html>
<html lang="fr">
<head>
  <meta charset="UTF-8">
  <title>Mise en route JavaScript</title>
</head>
<body>
  <script>
    var greet = "Hello World!";
    document.write(greet); // Prints: Hello World!
  </script>
</body>
</html>
```

Le code JavaScript dans l'exemple ci-dessus imprime simplement un message texte sur la page Web. Vous apprendrez ce que chacune de ces instructions JavaScript signifie dans les prochains chapitres

# Appel d'un fichier JavaScript externe

Vous pouvez également placer votre code JavaScript dans un fichier séparé avec une extension `.js`, puis appeler ce fichier dans votre document via l'attribut `src` de la balise `<script>`, comme ceci :

```
<script src = " js/hello.js "></script>
```

Ceci est utile si vous souhaitez que les mêmes scripts soient disponibles pour plusieurs documents. Cela vous évite de répéter la même tâche encore et encore et rend votre site Web beaucoup plus facile à entretenir.

Eh bien, créons un fichier JavaScript nommé `"hello.js"` et plaçons-y le code suivant :

```
// Une fonction pour afficher un message
function sayHello() {
    alert("Hello World!");
}

// Appel de la fonction au clic du bouton
document.getElementById("myBtn").onclick = sayHello;
```

Désormais, vous pouvez appeler ce fichier JavaScript externe dans une page Web à l'aide de la balise `<script>`, comme ceci :



```
<!DOCTYPE html>
<html lang="fr">
<head>
  <meta charset="UTF-8">
  <title>Inclure un fichier JavaScript externe</title>
</head>
<body>
  <button type="button" id="myBtn">Click Me</button>
  <script src="hello.js"></script>
</body>
</html>
```

**Remarque :** généralement, lorsqu'un fichier JavaScript externe est téléchargé pour la première fois, il est stocké dans le cache du navigateur (tout comme les images et les feuilles de style), il n'aura donc pas besoin d'être téléchargé plusieurs fois depuis le serveur Web qui crée les pages Web. charger plus rapidement.

# Placer le code JavaScript en ligne

Vous pouvez également placer du code JavaScript en ligne en l'insérant directement dans la balise HTML à l'aide des attributs de balise spéciaux tels que **onclick**, **onmouseover**, **onkeypress**, **onload**, etc.

Cependant, vous devez éviter de placer une grande quantité de code JavaScript en ligne car cela encombre votre HTML avec JavaScript et rend votre code JavaScript difficile à maintenir. Voici un exemple :

```
<!DOCTYPE html>
<html lang="fr">
<head>
  <meta charset="UTF-8">
  <title>JavaScript en ligne</title>
</head>
<body>
  <button onclick="alert('Hello World!')">Click Me</button>
</body>
</html>
```

L'exemple ci-dessus vous montrera un message d'alerte lors d'un clic sur l'élément bouton.

# Positionnement du script dans le document HTML

L'élément `<script>` peut être placé dans la section `<head>`, ou `<body>` d'un document HTML. Mais idéalement, les scripts doivent être placés à la fin de la section body, juste avant la balise de fermeture `</body>`, cela accélérera le chargement de vos pages Web, car cela empêchera l'obstruction du rendu initial de la page.

Chaque balise `<script>` bloque le processus de rendu de la page jusqu'à ce qu'elle ait entièrement téléchargé et exécuté le code JavaScript. Par conséquent, les placer dans la section d'en-tête (c'est-à-dire l'élément `<head>`) du document sans aucune raison valable aura un impact significatif sur les performances de votre site Web.

Astuce : Vous pouvez placer n'importe quel nombre d'éléments `<script>` dans un seul document. Cependant, ils sont traités dans l'ordre dans lequel ils apparaissent dans le document, de haut en bas.

# Différence entre les scripts côté client et côté serveur

Les langages de script côté client tels que JavaScript, VBScript, etc. sont interprétés et exécutés par le navigateur Web, tandis que les langages de script côté serveur tels que PHP, ASP, Java, Python, Ruby, etc. s'exécutent sur le serveur Web et la sortie renvoyé au navigateur Web au format HTML.

Les scripts côté client présentent de nombreux avantages par rapport à l'approche traditionnelle des scripts côté serveur. Par exemple, vous pouvez utiliser JavaScript pour vérifier si l'utilisateur a saisi des données non valides dans les champs du formulaire et afficher les notifications d'erreurs de saisie en conséquence en temps réel avant de soumettre le formulaire au serveur Web pour la validation finale des données et un traitement ultérieur afin d'éviter les utilisations inutiles de la bande passante du réseau et l'exploitation des ressources système du serveur.

De plus, la réponse d'un script côté serveur est plus lente que celle d'un script côté client, car les scripts côté serveur sont traités sur l'ordinateur distant et non sur l'ordinateur local de l'utilisateur.