

# Cours JavaScript - Développement Web Complet

## FORMATION COMPLÈTE JAVASCRIPT

### 1. Qu'est-ce que JavaScript ?

JavaScript est un langage de programmation principalement utilisé pour rendre les pages web interactives. Il permet de manipuler les éléments HTML et CSS, et de réagir aux événements utilisateurs.

### 2. Insertion de JavaScript dans une page HTML

- JavaScript en ligne : Dans l'attribut onclick, onmouseover, etc.
- JavaScript interne : Dans la balise `<script>` dans le `<head>` ou avant la fermeture du `<body>`.
- JavaScript externe : Dans un fichier .js lié à la page HTML via `<script src="file.js"></script>`.

Exemple de code JavaScript interne :

```
<head>

  <script>

    function greet() {

      alert("Bonjour, bienvenue !");

    }

  </script>

</head>

<body>

  <button onclick="greet()">Cliquez pour saluer</button>

</body>
```

### 3. Variables et Types de données

- String : Texte, exemple 'Bonjour'.
- Number : Nombres, exemple 123.
- Boolean : Valeurs true ou false.
- Array : Tableau de données, exemple [1, 2, 3].
- Object : Objet avec des paires clé-valeur, exemple { nom: 'Jean', age: 30 }.

Exemple :

```
let age = 30;
```

```
const name = 'Jean';
```

### 4. Fonctions en JavaScript

Les fonctions permettent de regrouper un ensemble d'instructions et de les exécuter plusieurs fois.

Exemple de fonction simple :

```
function greet() {  
    alert("Bonjour, bienvenue !");  
}
```

```
greet(); // Appel de la fonction
```

### 5. Les Conditions (if, else, switch)

Les structures conditionnelles permettent d'exécuter certaines instructions uniquement si une condition est remplie.

Exemple avec if :

```
let age = 20;
```

```
if (age >= 18) {  
    alert("Vous êtes majeur.");  
} else {  
    alert("Vous êtes mineur.");  
}
```

Exemple avec switch :

```
let fruit = 'pomme';  
switch(fruit) {  
    case 'pomme':  
        alert('C'est une pomme');  
        break;  
    case 'orange':  
        alert('C'est une orange');  
        break;  
    default:  
        alert('Fruit inconnu');  
}
```

## 6. Les Boucles (for, while, do-while)

Les boucles permettent de répéter un bloc de code plusieurs fois.

Exemple avec for :

```
for (let i = 0; i < 5; i++) {  
    console.log(i);  
}
```

Exemple avec while :

```
let i = 0;

while (i < 5) {

    console.log(i);

    i++;

}
```

## 7. Manipulation du DOM

Le DOM est une représentation en mémoire de la structure d'un document HTML. JavaScript peut interagir avec le DOM pour modifier le contenu, la structure et le style d'une page.

Exemple pour modifier le contenu d'un élément :

```
document.getElementById("myElement").innerHTML = "Nouveau contenu";
```

## 8. Les Événements en JavaScript

JavaScript permet de gérer les événements qui se produisent sur une page, comme un clic de souris, la saisie d'un texte, etc.

Exemple de gestion d'un clic :

```
document.getElementById("myButton").addEventListener("click", function() {

    alert("Vous avez cliqué sur le bouton !");

});
```

## 9. Les Formulaires et leur Validation

JavaScript est largement utilisé pour valider les formulaires avant leur soumission, afin de s'assurer que les données sont correctes.

Exemple de validation d'un champ de texte :

```
function validateForm() {  
    let name = document.getElementById("name").value;  
  
    if (name == "") {  
        alert("Le champ Nom ne peut pas être vide");  
        return false;  
    }  
}
```

## 10. Asynchronous JavaScript (AJAX)

AJAX permet de charger des données en arrière-plan sans recharger la page. Il est utilisé pour améliorer l'expérience utilisateur.

Exemple avec fetch :

```
fetch('https://api.example.com/data')  
    .then(response => response.json())  
    .then(data => console.log(data))  
    .catch(error => console.log("Erreur:", error));
```

## 11. Les Promesses (Promises)

Les promesses sont utilisées pour gérer les opérations asynchrones en JavaScript.

Exemple d'utilisation d'une promesse :

```
let promise = new Promise((resolve, reject) => {  
    let success = true;  
  
    if (success) {  
        resolve("Opération réussie");  
    }  
});
```

```
    } else {  
        reject("Échec de l'opération");  
    }  
});  
  
promise.then(result => console.log(result))  
    .catch(error => console.log(error));
```

## 12. Les Fonctions Fléchées (Arrow Functions)

Les fonctions fléchées sont une syntaxe plus concise pour déclarer des fonctions.

Exemple de fonction fléchée :

```
const greet = () => {  
    console.log("Bonjour");  
};  
  
greet();
```

Fin du cours JavaScript.