

Cours Explicatifs sur les bases de JavaScript

JavaScript est l'un des langages de programmation les plus utilisés pour le développement web. Il est principalement utilisé pour rendre les sites web interactifs et dynamiques. Ce cours est conçu pour les débutants et vous fournira une base solide en JavaScript, en couvrant les concepts fondamentaux comme les variables, les types, la syntaxe, les fonctions, les objets, et la programmation orientée objet (POO).

1. Introduction à JavaScript

JavaScript est un langage de script léger et interprété qui est exécuté directement dans les navigateurs web. Il peut être utilisé pour manipuler les éléments HTML, gérer les événements, effectuer des calculs, envoyer des requêtes HTTP, etc.

Intégration de JavaScript

Pour écrire du JavaScript dans une page HTML, il suffit d'utiliser la balise `<script>` :

Html (fichier index.html)
Copier le code

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Introduction à JavaScript</title>
</head>
<body>
  <h1>Exemple de JavaScript</h1>
  <script>
    // Ici on écrit du JavaScript
    console.log("Bonjour, JavaScript!");
  </script>
</body>
</html>
```

2. Les Variables et Types de base

En JavaScript, les variables peuvent être déclarées à l'aide de `var`, `let`, ou `const`.



2.1 Variables

- `let` : permet de déclarer des variables à portée de bloc (au sein des accolades `{}`).
- `const` : déclare des constantes, c'est-à-dire des variables dont la valeur ne peut pas être modifiée.
- `var` : ancienne façon de déclarer des variables, avec une portée fonctionnelle.

Exemple :

javascript
Copier le code

```
let nom = "Marc";
const age = 30;
var ville = "Yaoundé";

console.log(nom); // Affiche : Marc
console.log(age); // Affiche : 30
console.log(ville); // Affiche : Yaoundé
```

2.2 Types de données

Les types de base en JavaScript incluent :

- `string` : chaînes de caractères.
- `number` : nombres (entiers et décimaux).
- `boolean` : vrai ou faux (`true/false`).
- `null` : valeur vide.
- `undefined` : non défini.
- `object` : objets, tableaux, etc.

Exemple :

javascript
Copier le code

```
let nom = "Jean"; // string
let age = 25; // number
let estActif = true; // boolean
let taille = null; // null
let poids; // undefined
```

3. Les Opérateurs et Conditions



3.1 Opérateurs arithmétiques

Les opérateurs arithmétiques en JavaScript incluent :

- + (addition),
- - (soustraction),
- * (multiplication),
- / (division),
- % (modulo, pour obtenir le reste de la division).

Exemple :

javascript
Copier le code

```
let x = 10;
let y = 5;
console.log(x + y); // 15
console.log(x - y); // 5
console.log(x * y); // 50
console.log(x / y); // 2
console.log(x % y); // 0
```

3.2 Les structures conditionnelles

Les conditions en JavaScript sont définies avec if, else if et else. L'opérateur ternaire condition ? valeurSiVrai : valeurSiFaux est aussi utilisé pour des conditions plus simples.

Exemple :

javascript
Copier le code

```
let age = 20;

if (age < 18) {
  console.log("Vous êtes mineur");
} else if (age >= 18 && age <= 65) {
  console.log("Vous êtes adulte");
} else {
  console.log("Vous êtes senior");
}
```



4. Les Boucles

Les boucles permettent de répéter des actions.

4.1 La boucle `for`

La boucle `for` est utilisée pour répéter un bloc de code un certain nombre de fois.

Exemple :

javascript
Copier le code

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

4.2 La boucle `while`

La boucle `while` répète le code tant qu'une condition est vraie.

Exemple :

javascript
Copier le code

```
let compteur = 0;  
  
while (compteur < 3) {  
  console.log("Compteur : " + compteur);  
  compteur++;  
}
```

5. Les Fonctions

Les fonctions sont des blocs de code réutilisables qui peuvent être appelés à plusieurs endroits.

5.1 Déclaration de fonction

Les fonctions en JavaScript peuvent être déclarées de plusieurs façons. Voici la syntaxe classique :



javascript
Copier le code

```
function direBonjour(nom) {  
    return "Bonjour " + nom + "!";  
}  
  
console.log(direBonjour("Alice")); // Affiche : Bonjour Alice!
```

5.2 Fonctions fléchées (Arrow Functions)

Les fonctions fléchées sont une façon plus concise d'écrire les fonctions.

Exemple :

javascript
Copier le code

```
const addition = (a, b) => a + b;  
  
console.log(addition(2, 3)); // Affiche : 5
```

6. Les Objets

Les objets en JavaScript sont des collections de propriétés et méthodes.

6.1 Déclaration d'objets

Un objet peut être défini comme une paire clé-valeur.

Exemple :

javascript
Copier le code

```
let personne = {  
    nom: "Alice",  
    age: 30,  
    direBonjour: function() {  
        console.log("Bonjour, je m'appelle " + this.nom);  
    }  
};  
  
console.log(personne.nom); // Affiche : Alice  
personne.direBonjour(); // Affiche : Bonjour, je m'appelle Alice
```



6.2 Accès et modification des propriétés d'un objet

Les propriétés d'un objet peuvent être accédées et modifiées via la notation par point ou par crochets.

Exemple :

javascript
Copier le code

```
personne.nom = "Bob"; // Modification via la notation par point
console.log(personne.nom); // Affiche : Bob

personne["age"] = 35; // Modification via la notation par crochets
console.log(personne.age); // Affiche : 35
```

