

Cours de JavaScript pour Débutants

1 Introduction à JavaScript

JavaScript est un langage de programmation utilisé pour rendre les pages web interactives. Ce cours couvre les bases pour débutants, incluant les variables, les conditions, les boucles, les fonctions, et la manipulation du DOM.

1.1 Prérequis

Aucune connaissance préalable n'est requise, mais une familiarité avec HTML et CSS est un plus.

1.2 Exécuter du JavaScript

JavaScript peut être exécuté :

- Dans une balise `<script>` dans un fichier HTML.
- Dans la console du navigateur (F12).
- Dans un fichier `.js` lié à un HTML avec `<script src="script.js">`.

Exemple :

```
1 console.log("Hello, JavaScript !");
```

2 Les Bases

2.1 Variables

Les variables stockent des données. Utilisez `let` pour les variables modifiables, `const` pour les constantes, et évitez `var`.

```
1 let nom = "Alice";
2 const age = 25;
3 nom = "Bob"; // OK
4 // age = 30; // Erreur
5 console.log(nom, age); // Affiche : Bob 25
```

2.2 Types de Données

- **String** : Texte, ex. "Bonjour".
- **Number** : Nombres, ex. 42, 3.14.

- **Boolean** : true ou false.
- **Array** : Liste, ex. ["pomme", "banane"].
- **Object** : Données structurées, ex. { nom: "Alice", age: 25 }.
- **null** : Valeur vide.
- **undefined** : Valeur non définie.

2.3 Opérateurs

- Arithmétiques : +, -, *, /, %, **.
- Comparaison : ==, ===, !=, !==, >, <.
- Logiques : && (ET), || (OU), ! (NON).

Exemple :

```
1 let a = 10, b = 3;
2 console.log(a + b); // 13
3 console.log(a \% b); // 1
4 console.log(a > 5 && b < 5); // true
```

3 Contrôle du Flux

3.1 Conditions

Les structures conditionnelles (if, else if, else) exécutent du code selon une condition.

```
1 let age = 16;
2 if (age >= 18) {
3   console.log("Tu es majeur.");
4 } else if (age >= 13) {
5   console.log("Tu es ado.");
6 } else {
7   console.log("Tu es mineur.");
8 }
```

3.2 Switch

Alternative aux conditions pour tester plusieurs valeurs.

```
1 let jour = "lundi";
2 switch (jour) {
3   case "lundi":
4     console.log("Début de semaine !");
5     break;
6   case "vendredi":
7     console.log("Bientôt le week-end !");
8     break;
9   default:
10    console.log("Jour quelconque.");
11 }
```

3.3 Boucles

Les boucles répètent du code.

— **for** :

```
1 for (let i = 1; i <= 5; i++) {  
2   console.log(i); // Affiche 1, 2, 3, 4, 5  
3 }
```

— **while** :

```
1 let i = 1;  
2 while (i <= 5) {  
3   console.log(i);  
4   i++;  
5 }
```

— **for...of** (pour les tableaux) :

```
1 let fruits = ["pomme", "banane"];  
2 for (let fruit of fruits) {  
3   console.log(fruit);  
4 }
```

4 Fonctions

4.1 Déclarer une Fonction

Les fonctions regroupent du code réutilisable.

```
1 function saluer(prenom) {  
2   return "Bonjour, " + prenom + " !";  
3 }  
4 console.log(saluer("Alice")); // Bonjour, Alice !
```

4.2 Fonctions Fléchées

Syntaxe concise (ES6) :

```
1 const addition = (a, b) => a + b;  
2 console.log(addition(5, 3)); // 8
```

4.3 Paramètres par Défaut

```
1 function direBonjour(prenom = "Inconnu") {  
2   console.log("Bonjour, " + prenom);  
3 }  
4 direBonjour(); // Bonjour, Inconunknown
```

5 Tableaux et Objets

5.1 Tableaux

Les tableaux stockent des listes.

```
1 let fruits = ["pomme", "banane"];
2 fruits.push("kiwi"); // Ajoute
3 console.log(fruits[0]); // pomme
```

Méthodes utiles : push, pop, length, forEach.

5.2 Objets

Les objets stockent des données sous forme de paires clé-valeur.

```
1 let personne = {
2   nom: "Alice",
3   age: 25,
4   saluer: function() {
5     console.log("Salut, je suis " + this.nom);
6   }
7 };
8 personne.saluer(); // Salut, je suis Alice
```

6 Manipulation du DOM

Le DOM permet de modifier une page web.

6.1 Sélectionner des Éléments

```
1 let titre = document.querySelector("h1");
2 titre.textContent = "Nouveau titre";
```

6.2 Gérer les Événements

```
1 let bouton = document.getElementById("monBouton");
2 bouton.addEventListener("click", () => {
3   alert("Bouton cliqué !");
4 });
```

6.3 Modifier le Style

```
1 let paragraphe = document.querySelector("p");
2 paragraphe.style.color = "blue";
```

7 Gestion des Erreurs

Utilisez `try...catch` pour gérer les erreurs.

```
1 try {  
2   let resultat = uneVariableNonDéfinie;  
3 } catch (erreur) {  
4   console.log("Erreur : " + erreur.message);  
5 }
```

8 Conclusion

Ce cours couvre les bases de JavaScript : variables, types de données, opérateurs, conditions, boucles, fonctions, tableaux, objets, DOM, et gestion des erreurs. Pour progresser :

- Pratiquez avec des exercices simples.
- Consultez MDN Web Docs pour approfondir.
- Créez des projets comme une calculatrice ou une liste de tâches.