

# Exercices de JavaScript pour Débutants

## 1 Introduction

Ce document propose une série d'exercices pour pratiquer les bases de JavaScript : variables, opérateurs, conditions, boucles, fonctions, tableaux, objets, et manipulation du DOM. Les exercices sont classés par thème, augmentent en difficulté, et incluent des défis bonus. Des solutions commentées sont fournies à la fin pour vérifier vos réponses.

## 2 Exercices

### 2.1 Variables et Opérateurs

#### Exercice 1 : Calcul de l'âge

Créez une variable `anneeNaissance` avec une année (ex. 2000). Calculez l'âge en 2025 et affichez un message comme : "J'ai 25 ans."

**Indice** : Utilisez l'opérateur `-`.

#### Exercice 2 : Convertisseur de température

Écrivez un programme qui convertit une température en Celsius vers Fahrenheit. La formule est :  $F = C * 9/5 + 32$ . Déclarez une variable `celsius` et affichez le résultat, ex. : "20°C équivaut à 68°F."

**Indice** : Testez avec `celsius = 20`.

#### Exercice 3 : Vérification de parité

Créez une variable `nombre`. Affichez si le nombre est pair ou impair en utilisant l'opérateur modulo `%`. Exemple : "6 est pair." ou "7 est impair."

**Bonus** : Si le nombre est négatif, affichez "Nombre invalide."

### 2.2 Conditions

#### Exercice 4 : Catégorie d'âge

Créez une variable `age`. Affichez :

- "Enfant" si l'âge est inférieur à 13.
- "Adolescent" si l'âge est entre 13 et 17 inclus.
- "Adulte" si l'âge est 18 ou plus.
- "Invalide" si l'âge est négatif ou non numérique.

Testez avec `age = 15`, `age = -5`, `age = 30`.

### Exercice 5 : Feu tricolore

Créez une variable `couleur` avec "rouge", "orange", ou "vert". Affichez :

- "Arrêtez-vous !" pour rouge.
- "Préparez-vous." pour orange.
- "Avancez." pour vert.
- "Couleur inconnue." sinon.

Utilisez `switch`.

### Exercice 6 : Billet de cinéma

Un cinéma applique des tarifs selon l'âge :

- Moins de 12 ans : 5 euros.
- 12 à 17 ans : 8 euros.
- 18 ans ou plus : 12 euros.

Créez une variable `age` et affichez le prix, ex. : "Le billet coûte 8 euros."

**Bonus** : Si l'âge est invalide, affichez "Âge non valide."

## 2.3 Boucles

### Exercice 7 : Table de multiplication

Affichez la table de multiplication d'un nombre (ex. 7) de 1 à 10. Exemple :

```
7 x 1 = 7
7 x 2 = 14
...
7 x 10 = 70
```

Utilisez une boucle `for`.

### Exercice 8 : Compte à rebours

Affichez un compte à rebours de 10 à 1, suivi de "Décollage !". Utilisez une boucle `while`.

Exemple :

```
10
9
...
1
Décollage !
```

### Exercice 9 : FizzBuzz

Affichez les nombres de 1 à 20, mais :

- Remplacez les multiples de 3 par "Fizz".
- Remplacez les multiples de 5 par "Buzz".
- Remplacez les multiples de 3 et 5 par "FizzBuzz".

Exemple : 1, 2, Fizz, 4, Buzz, Fizz, 7, ....

## 2.4 Fonctions

### Exercice 10 : Calcul du carré

Créez une fonction `carre` qui prend un nombre et retourne son carré. Testez avec `carre(4)` (résultat : 16) et `carre(7)` (résultat : 49).

### Exercice 11 : Vérification de palindrome

Créez une fonction `estPalindrome` qui prend une chaîne et retourne `true` si c'est un palindrome (ex. "radar"), `false` sinon (ex. "bonjour"). Ignorez les majuscules/minuscules.

Testez avec "Radar", "Hello".

### Exercice 12 : Maximum de trois nombres

Créez une fonction `maxTrois` qui prend trois nombres et retourne le plus grand. Testez avec `maxTrois(5, 10, 3)` (résultat : 10).

**Bonus** : Utilisez `Math.max`.

## 2.5 Tableaux

### Exercice 13 : Liste de courses

Créez un tableau `courses` avec 4 aliments. Affichez chaque élément avec une boucle `for...of`. Ajoutez un aliment avec `push` et supprimez le dernier avec `pop`. Affichez le tableau final.

### Exercice 14 : Somme d'un tableau

Créez une fonction `sommeTableau` qui prend un tableau de nombres et retourne leur somme. Testez avec `[1, 2, 3, 4]` (résultat : 10).

**Bonus** : Vérifiez que le tableau ne contient que des nombres.

### Exercice 15 : Filtrer les positifs

Créez une fonction `filtrerPositifs` qui prend un tableau de nombres et retourne un nouveau tableau contenant uniquement les nombres positifs. Testez avec `[-2, 0, 5, -1, 3]` (résultat : `[5, 3]`).

## 2.6 Objets

### Exercice 16 : Profil utilisateur

Créez un objet `utilisateur` avec les propriétés `nom`, `age`, et une méthode `presenter` qui affiche "Je m'appelle [nom] et j'ai [age] ans.". Appelez la méthode.

### Exercice 17 : Inventaire

Créez un objet `inventaire` avec des produits (ex. `pommes: 5`, `bananes: 3`). Créez une fonction `ajouterProduit` qui ajoute ou met à jour la quantité d'un produit. Testez en ajoutant 2 pommes.

## 2.7 DOM

### Exercice 18 : Changeur de couleur

Créez une page HTML avec un bouton et un paragraphe. Quand le bouton est cliqué, changez la couleur du paragraphe en rouge et son texte en "Couleur changée !".

### Exercice 19 : Compteur de clics

Créez une page avec un bouton et un paragraphe affichant 0. Chaque clic sur le bouton incrémente le compteur. Exemple : 1, 2, ...

### Exercice 20 : Liste dynamique

Créez une page avec un champ de texte, un bouton, et une liste `<ul>`. Quand le bouton est cliqué, ajoutez le contenu du champ comme nouvel élément `<li>` à la liste. Videz le champ après ajout.

## 3 Solutions Commentées

### 3.1 Exercice 1 : Calcul de l'âge

```
1 let anneeNaissance = 2000;
2 let age = 2025 - anneeNaissance;
3 console.log("J'ai " + age + " ans."); // J'ai 25 ans.
```

*On soustrait l'année de naissance à 2025 pour calculer l'âge.*

### 3.2 Exercice 2 : Convertisseur de température

```
1 let celsius = 20;
2 let fahrenheit = celsius * 9/5 + 32;
3 console.log(celsius + "°C équivaut à " + fahrenheit + "°F."); //
    20°C équivaut à 68°F.
```

*On applique la formule de conversion et concatène les chaînes.*

### 3.3 Exercice 3 : Vérification de parité

```
1 let nombre = 6;
2 if (nombre < 0) {
3     console.log("Nombre invalide.");
4 } else if (nombre % 2 === 0) {
5     console.log(nombre + " est pair.");
6 } else {
7     console.log(nombre + " est impair.");
8 }
```

*Le modulo % détermine la parité, avec une vérification des négatifs.*

### 3.4 Exercice 4 : Catégorie d'âge

```
1 let age = 15;
2 if (isNaN(age) || age < 0) {
3     console.log("Invalide");
4 } else if (age < 13) {
5     console.log("Enfant");
6 } else if (age <= 17) {
7     console.log("Adolescent");
8 }
```

```

8 } else {
9   console.log("Adulte");
10 }

```

*Les conditions vérifient les plages d'âge de manière séquentielle.*

### 3.5 Exercice 5 : Feu tricolore

```

1 let couleur = "vert";
2 switch (couleur) {
3   case "rouge": console.log("Arrêtez-vous !"); break;
4   case "orange": console.log("Préparez-vous."); break;
5   case "vert": console.log("Avancez."); break;
6   default: console.log("Couleur inconnue.");
7 }

```

*switch simplifie la gestion des cas pour les valeurs discrètes.*

### 3.6 Exercice 6 : Billet de cinéma

```

1 let age = 14;
2 if (isNaN(age) || age < 0) {
3   console.log("Âge non valide.");
4 } else if (age < 12) {
5   console.log("Le billet coûte 5 euros.");
6 } else if (age <= 17) {
7   console.log("Le billet coûte 8 euros.");
8 } else {
9   console.log("Le billet coûte 12 euros.");
10 }

```

*Les tarifs sont attribués selon les tranches d'âge.*

### 3.7 Exercice 7 : Table de multiplication

```

1 let nombre = 7;
2 for (let i = 1; i <= 10; i++) {
3   console.log(nombre + " x " + i + " = " + (nombre * i));
4 }

```

*La boucle for génère chaque ligne de la table.*

### 3.8 Exercice 8 : Compte à rebours

```

1 let i = 10;
2 while (i >= 1) {
3   console.log(i);
4   i--;
5 }
6 console.log("Décollage !");

```

La boucle *while* décrémente jusqu'à 1, suivie du message final.

### 3.9 Exercice 9 : FizzBuzz

```
1 for (let i = 1; i <= 20; i++) {  
2   if (i % 3 === 0 && i % 5 === 0) {  
3     console.log("FizzBuzz");  
4   } else if (i % 3 === 0) {  
5     console.log("Fizz");  
6   } else if (i % 5 === 0) {  
7     console.log("Buzz");  
8   } else {  
9     console.log(i);  
10  }  
11 }
```

Les conditions vérifient les multiples dans l'ordre de priorité.

### 3.10 Exercice 10 : Calcul du carré

```
1 function carre(nombre) {  
2   return nombre * nombre;  
3 }  
4 console.log(carre(4)); // 16  
5 console.log(carre(7)); // 49
```

La fonction retourne le produit du nombre par lui-même.

### 3.11 Exercice 11 : Vérification de palindrome

```
1 function estPalindrome(chaine) {  
2   chaine = chaine.toLowerCase();  
3   let inverse = chaine.split("").reverse().join("");  
4   return chaine === inverse;  
5 }  
6 console.log(estPalindrome("Radar")); // true  
7 console.log(estPalindrome("Hello")); // false
```

On normalise la chaîne et compare avec son inverse.

### 3.12 Exercice 12 : Maximum de trois nombres

```
1 function maxTrois(a, b, c) {  
2   return Math.max(a, b, c);  
3 }  
4 console.log(maxTrois(5, 10, 3)); // 10
```

*Math.max* est une méthode efficace pour trouver le maximum.

### 3.13 Exercice 13 : Liste de courses

```
1 let courses = ["pain", "lait", "oeufs", "fruits"];
2 for (let item of courses) {
3   console.log(item);
4 }
5 courses.push("chocolat");
6 courses.pop();
7 console.log(courses); // ["pain", "lait", "oeufs"]
```

*La boucle affiche les éléments, et **push/pop** modifient le tableau.*

### 3.14 Exercice 14 : Somme d'un tableau

```
1 function sommeTableau(tableau) {
2   let somme = 0;
3   for (let num of tableau) {
4     if (typeof num !== "number") return "Erreur : tableau
5       invalide.";
6     somme += num;
7   }
8   return somme;
9 }
10 console.log(sommeTableau([1, 2, 3, 4])); // 10
```

*La fonction parcourt le tableau et accumule la somme, avec une vérification.*

### 3.15 Exercice 15 : Filtrer les positifs

```
1 function filtrerPositifs(tableau) {
2   let positifs = [];
3   for (let num of tableau) {
4     if (num > 0) positifs.push(num);
5   }
6   return positifs;
7 }
8 console.log(filtrerPositifs([-2, 0, 5, -1, 3])); // [5, 3]
```

*On crée un nouveau tableau avec les nombres positifs uniquement.*

### 3.16 Exercice 16 : Profil utilisateur

```
1 let utilisateur = {
2   nom: "Alice",
3   age: 25,
4   presenter: function() {
5     console.log("Je m'appelle " + this.nom + " et j'ai " + this.
6       age + " ans.");
7   }
8 };
9 utilisateur.presenter(); // Je m'appelle Alice et j'ai 25 ans.
```

*L'objet contient une méthode qui utilise `this` pour accéder aux propriétés.*

### 3.17 Exercice 17 : Inventaire

```
1 let inventaire = { pommes: 5, bananes: 3 };
2 function ajouterProduit(produit, quantite) {
3   if (inventaire[produit]) {
4     inventaire[produit] += quantite;
5   } else {
6     inventaire[produit] = quantite;
7   }
8 }
9 ajouterProduit("pommes", 2);
10 console.log(inventaire); // { pommes: 7, bananes: 3 }
```

*La fonction met à jour ou ajoute un produit dans l'objet.*

### 3.18 Exercice 18 : Changeur de couleur

```
1 <!DOCTYPE html>
2 <html>
3 <body>
4   <p id="para">Texte initial</p>
5   <button onclick="changerCouleur()">Changer</button>
6   <script>
7     function changerCouleur() {
8       let para = document.getElementById("para");
9       para.style.color = "red";
10      para.textContent = "Couleur changée !";
11    }
12  </script>
13 </body>
14 </html>
```

*Le bouton déclenche une fonction qui modifie le style et le texte.*

### 3.19 Exercice 19 : Compteur de clics

```
1 <!DOCTYPE html>
2 <html>
3 <body>
4   <p id="compteur">0</p>
5   <button onclick="incrementer()">Cliquer</button>
6   <script>
7     let compte = 0;
8     function incrementer() {
9       compte++;
10      document.getElementById("compteur").textContent = compte;
11    }
12  </script>
13 </body>
```



```
14 </html>
```

*Chaque clic incrémente une variable et met à jour l’affichage.*

### 3.20 Exercice 20 : Liste dynamique

```
1 <!DOCTYPE html>
2 <html>
3 <body>
4   <input id="input" type="text" placeholder="Ajouter un item">
5   <button onclick="ajouterItem()">Ajouter</button>
6   <ul id="liste"></ul>
7   <script>
8     function ajouterItem() {
9       let input = document.getElementById("input");
10      let liste = document.getElementById("liste");
11      let texte = input.value;
12      if (texte) {
13        let li = document.createElement("li");
14        li.textContent = texte;
15        liste.appendChild(li);
16        input.value = "";
17      }
18    }
19  </script>
20 </body>
21 </html>
```

*La fonction crée un nouvel élément `<li>` et vide le champ.*