

Introduction aux objets en JavaScript

Objectif : Comprendre ce qu'est un objet en JavaScript, comment le créer, accéder à ses propriétés, les modifier.

Sommaire

1. **C'est quoi un objet JS ?**
2. **Accéder à la propriété d'un objet JS**
3. **Différence entre objet et tableau**
4. **Exercice**
5. **Ressources**

C'est quoi un objet JavaScript ?

Un objet, c'est une structure qui permet de regrouper plusieurs informations dans une seule variable. Chaque information à l'intérieur d'un objet est appelée une propriété.

```
let chat = {  
  nom: "Minou",  
  age: 3,  
  race: "Siamois",  
  estAdopte: true  
};
```

Ici, on a créé **un objet** appelé chat avec **4 propriétés**.

Chaque propriété est une paire clé/valeur (ex : "nom": "Minou").

Accéder à une propriété d'un objet JS avec le point

```
let chat = {  
  nom: "Minou",  
  age: 3,  
  race: "Siamois",  
  estAdopte: true  
};
```

Avec un point :

```
console.log(chat.nom); // "Minou"  
console.log(chat.age); // 3
```

Accéder à une propriété d'un objet JS avec les crochets

```
let chat = {  
  nom: "Minou",  
  age: 3,  
  race: "Siamois",  
  estAdopte: true  
};
```

Avec des crochets (utile quand la clé est dynamique) :

```
console.log(chat[race]); // "Siamois"
```

Modifier une propriété d'un objet JS

```
let chat = {  
  nom: "Minou",  
  age: 3,  
  race: "Siamois",  
  estAdopte: true  
};
```

On peut changer la valeur d'une propriété existante :

```
chat.age = 4;  
chat.estAdopte = false;
```

Modifier une propriété d'un objet JS

```
let chat = {  
  nom: "Minou",  
  age: 3,  
  race: "Siamois",  
  estAdopte: true  
};
```

On peut ajouter une nouvelle valeur à une propriété existante :

```
chat.couleur = "Gris";
```

Pourquoi utiliser des objets en JavaScript ?

Les objets sont parfaits pour représenter des choses réelles dans un programme :

- un **utilisateur** : nom, email, mot de passe
- un **animal** : espèce, âge, taille, vacciné ou pas
- un **produit** : prix, nom, catégorie, stock

Plus l'application devient complexe, plus on a besoin de regrouper les données.

```
let utilisateur = {  
  prenom: "Alice",  
  email: "alice@gmail.com",  
  isAdmin: false,  
};
```

À retenir sur les objets JS

Élément	Exemple
Créer	<code>let obj = { clé: valeur }</code>
Lire	<code>obj.clé</code> ou <code>obj["clé"]</code>
Modifier	<code>obj.clé = nouvelleValeur</code>
Ajouter	<code>obj.nouvelleCle = valeur</code>

Quelle différence entre tableau et objet en JS ?

Un tableau sert à stocker une liste ordonnée de valeurs.

- Chaque valeur a un **index numérique** (0, 1, 2, ...)
- Parfait pour des **collections de données similaires** (ex : prénoms, scores, couleurs...)

```
let couleurs = ["rouge", "bleu", "vert"];  
console.log(couleurs[0]); // "rouge"
```

Un objet sert à regrouper des données organisées autour de propriétés nommées.

- Chaque propriété a un **nom (clé)** défini par vous-même.
- Parfait pour représenter des **“choses” complexes** : une personne, un animal, un produit...

```
let animal = {  
  nom: "Tigrou",  
  espece: "tigre",  
  age: 5  
};  
console.log(animal.nom); // "Tigrou"
```

Rappel des différences tableau / objet

	Tableau	Objet
Clés/index	Numériques (0, 1, 2...)	Nommées (“nom”, “age”, etc.)
Ordre	Oui, les éléments sont ordonnés	Non garanti (mais souvent respecté)
Utilisation typique	Liste de valeurs similaires	Description structurée d’un “objet” réel
Accès	tableau[0]	objet.clé ou objet["clé"]
Type JS	Array	Object

Un tableau, c’est comme une boîte d’œufs — Un objet, c’est comme une fiche d’identité

Ressources

reduce() en JavaScript

https://developer.mozilla.org/fr/docs/Web/JavaScript/Guide/Working_with_objects

https://developer.mozilla.org/fr/docs/Web/JavaScript/Reference/Global_Objects/Object

Vidéo (fr) Objets JS

<https://www.youtube.com/watch?v=W3dMVypqT3U>