



Objectif du module

À la fin de ce module, l’étudiant sera capable de :

- Comprendre ce qu’est un objet JavaScript
- Créer un objet avec plusieurs propriétés
- Utiliser un tableau d’objets
- Générer une interface à partir de données structurées
- Appliquer des conditions sur les propriétés d’un objet

Table des matières

1. Qu’est-ce qu’un objet ?..... 2

 Exemple simple 2

 Accéder aux propriétés 2

2. Tableau d’objets..... 2

3. Générer dynamiquement une liste de produits 3

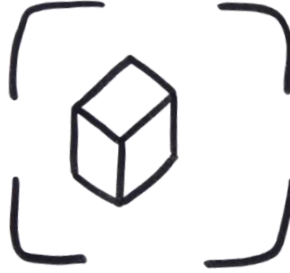
 Exercice — Fiche utilisateur – Fiche.html 4

 Exercice — Mini catalogue structuré – catalogue.html..... 4

 Exercice — Mini catalogue interactif – catalogueinteractif.html 4

1. Qu'est-ce qu'un objet ?

Un objet regroupe plusieurs informations liées.



Exemple simple

```
const produit = {  
  nom: "Chaussures",  
  prix: 79,  
  disponible: true  
};
```

Un objet = une fiche descriptive.

Accéder aux propriétés

```
produit.nom  
produit.prix  
produit.disponible
```

2. Tableau d'objets

Dans une vraie interface :

```
const produits = [  
  { nom: "Chaussures", prix: 79, disponible: true },  
  { nom: "T-shirt", prix: 25, disponible: false },  
  { nom: "Casquette", prix: 15, disponible: true }  
];
```

Chaque élément du tableau est un objet.

3. Générer dynamiquement une liste de produits

```
<!DOCTYPE html>
<html lang="fr">
<head>
  <meta charset="UTF-8">
  <title>Catalogue</title>
  <link rel="stylesheet" href="css/bootstrap.css">
</head>
<body class="p-4">

  <h2>Catalogue</h2>
  <div id="catalogue" class="mt-3"></div>

<script>
  const produits = [
    { nom: "Chaussures", prix: 79, disponible: true },
    { nom: "T-shirt", prix: 25, disponible: false },
    { nom: "Casquette", prix: 15, disponible: true }
  ];

  const catalogue = document.querySelector("#catalogue");

  for (let i = 0; i < produits.length; i++) {
    const bloc = document.createElement("div");
    bloc.classList.add("border", "p-2", "mb-3");

    const titre = document.createElement("h4");
    titre.textContent = produits[i].nom;

    const prix = document.createElement("p");
    prix.textContent = "Prix : " + produits[i].prix + " €";

    bloc.appendChild(titre);
    bloc.appendChild(prix);

    if (produits[i].disponible === false) {
      const rupture = document.createElement("p");
      rupture.textContent = "Rupture de stock";
      rupture.classList.add("text-danger", "fw-bold");
      bloc.appendChild(rupture);
    }

    catalogue.appendChild(bloc);
  }
</script>
</body>
</html>
```

Exercice — Fiche utilisateur – Fiche.html

Afficher une dynamiquement à partir d'un objet



Données imposées :

```
const user = {  
  prenom: "Alex",  
  age: 22,  
  membrePremium: true  
};
```

Créer :

Afficher dans la page :

- Prénom
- Âge
- Un badge "Premium" si l'utilisateur est membrePremium = true

Contraintes

- Ne pas écrire les valeurs en dur dans le HTML
- Utiliser les propriétés de l'objet
- Utiliser une condition pour afficher le badge
- Utiliser classList pour le style

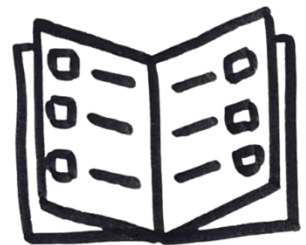
Exercice — Mini catalogue structuré – catalogue2.html

Reprenez le catalogue.html (tableau de 3 produits nom/prix/disponible (true/false))

Ajouter un bouton "Ajouter au panier" pour chaque produit mais Désactiver le bouton

si disponible = false

Changer la couleur du bloc si indisponible



Exercice — Mini catalogue interactif – catalogueinteractif.html

Combiner tableau d'objets/génération dynamique/événements

Interface attendue

Pour chaque produit :

- Nom
- Prix
- Bouton "Ajouter au panier"

Comportement attendu



Chaque clic affiche dans la console :

- “Produit ajouté : [nom du produit]”

Contraintes

- L'événement doit être ajouté dynamiquement dans la boucle
- Le nom du produit doit provenir de l'objet
- Ne pas utiliser de variable globale pour stocker le nom

Pour aller plus loin...

Créer un compteur global du panier.