

Exercices sur les Classes en JavaScript (ES6)

Ces exercices sont conçus pour vous aider à comprendre et à pratiquer la création et la manipulation de classes en JavaScript en utilisant la syntaxe ES6.

Introduction aux Classes en ES6

En ES6, les classes offrent une syntaxe plus claire et concise pour définir des objets et gérer l'héritage par rapport à la syntaxe ES5. Voici un exemple simple de déclaration de classe en ES6 :

```
class Personne {
  constructor(nom, prenom) {
    this.nom = nom;
    this.prenom = prenom;
  }

  afficherNomComplet() {
    console.log(`${this.prenom} ${this.nom}`);
  }
}

let personne = new Personne('Dupont', 'Marie');
personne.afficherNomComplet(); // Affiche "Marie Dupont"
```

Exercices

Exercice 1 : Création d'une Classe Simple

Objectif :

- Déclarer une classe `Personne` avec des attributs `nom` et `prenom`.
- Instancier un objet de cette classe et afficher ses attributs.

Instructions :

- Dans un fichier `personne.js`, déclarez la classe `Personne` avec les attributs `nom` et `prenom`.
- Dans un fichier `main.js`, importez ou incluez la classe `Personne`.
- Utilisez `prompt` pour demander à l'utilisateur son nom et son prénom.
- Créez une instance de `Personne` avec les valeurs saisies.
- Affichez les attributs de l'objet créé dans la console avec `console.log`.

Exercice 2 : Ajout d'un Constructeur

Objectif :

- Ajouter un constructeur à la classe `Personne` pour initialiser les attributs lors de la création de l'objet.

Instructions :

- Modifiez la classe `Personne` pour inclure un constructeur qui prend en paramètres `nom` et `prenom`.
- Dans `main.js`, utilisez ce constructeur pour créer l'objet `Personne` avec les valeurs saisies par l'utilisateur.
- Affichez les attributs de l'objet dans la console pour vérifier qu'ils ont été correctement initialisés.

Exercice 3 : Ajout de Méthodes

Objectif :

- Ajouter une méthode `afficher()` à la classe `Personne` qui affiche une phrase décrivant la personne.

Instructions :

- Dans la classe `Personne`, ajoutez une méthode `afficher()` qui affiche : `La personne s'appelle [prenom] [nom]`.
- Dans `main.js`, appelez cette méthode sur l'objet `Personne` créé précédemment.

Exercice 4 : Création de Classes Supplémentaires

Objectif :

- Créer les classes `Animal`, `Vehicule` et `Maison` avec des attributs spécifiques.
- Ajouter des constructeurs et des méthodes `afficher()` pour chaque classe.

Instructions :

- Classe `Animal` (`animal.js`):**
 - Attributs : `nom`, `espece`.
 - Constructeur pour initialiser les attributs.
 - Méthode `afficher()` qui affiche : `L'animal s'appelle [nom] et est de l'espèce [espece]`.
- Classe `Vehicule` (`vehicule.js`):**
 - Attributs : `marque`, `modele`, `nombrePlaces`.
 - Constructeur pour initialiser les attributs.
 - Méthode `afficher()` qui affiche : `Le véhicule est de la marque [marque] et du modèle [modele]. Il peut transporter [nombrePlaces] personne(s)`.
- Classe `Maison` (`maison.js`):**
 - Attributs : `numeroRue`, `typeVoie`, `nomVoie`, `codePostal`, `ville`.
 - Constructeur pour initialiser les attributs.
 - Méthode `afficher()` qui affiche : `Le logement se situe [numeroRue] [typeVoie] [nomVoie] - [codePostal] [ville]`.

Exercice 5 : Utilisation des Objets Créés

Objectif :

- Créer des instances des classes précédentes avec des valeurs prédéfinies.
- Appeler les méthodes `afficher()` pour vérifier le bon fonctionnement.

Instructions :

- Dans `main.js`, ajoutez les lignes suivantes :

```
let formateur = new Personne('Estezet', 'Nicolas');
let chatFormateur = new Animal('Mathilda', 'chat');
let vehiculeFormateur = new Vehicule('Tesla', 'Model X', 7);
let domicileFormateur = new Maison(1, 'rue', 'du paradis', '18999', 'Las Angelos');
```

- Assurez-vous que ces lignes fonctionnent correctement en appelant la méthode `afficher()` sur chacun de ces objets.
- Vous devez obtenir dans le log les lignes suivantes :

```
La personne s'appelle Nicolas Estezet.
L'animal s'appelle Mathilda et est de l'espèce chat.
Le véhicule est de la marque Tesla et du modèle Model X. Il peut transporter 7 personne(s).
Le logement se situe 1 rue du paradis - 18999 Las Angelos.
```

Exercice 6 : Calcul de la Surface d'un Rectangle

Objectif :

- Créer une classe `Rectangle` avec des méthodes pour calculer la surface et le périmètre.
- Interagir avec l'utilisateur pour saisir les dimensions et afficher les résultats.

Instructions :

- Classe `Rectangle` (`rectangle.js`):**
 - Attributs : `longueur`, `largeur`.
 - Constructeur pour initialiser les attributs.
 - Méthodes :
 - `calculerSurface()` : retourne la surface du rectangle.
 - `calculerPerimetre()` : retourne le périmètre du rectangle.
- Dans `main.js` :
 - Demandez à l'utilisateur de saisir la longueur et la largeur via des `prompt`.
 - Créez une instance de `Rectangle` avec les valeurs saisies.
 - Affichez avec un `alert` la surface et le périmètre sous la forme :

```
Le rectangle de longueur ... cm et de largeur ... cm a une surface de ... cm² et un périmètr
e de ... cm.
```

Exercice 7 : Calcul de la Surface d'un Cercle avec Héritage

Objectif :

- Créer une classe parent `Figure` et une classe enfant `Cercle` qui hérite de `Figure`.
- Implémenter des méthodes de calcul redéfinies dans la classe enfant.

Instructions :

- Classe `Figure` (`figure.js`):**
 - Méthodes :
 - `calculerSurface()` : retourne 0 par défaut.
 - `calculerPerimetre()` : retourne 0 par défaut.
- Classe `Cercle` (`cercle.js`):**
 - Hérite de `Figure`.
 - Attributs : `rayon`.
 - Constructeur pour initialiser le `rayon`.
 - Redéfinit les méthodes `calculerSurface()` et `calculerPerimetre()` pour le cercle.
- Dans `main.js` :
 - Demandez à l'utilisateur de saisir le rayon du cercle.
 - Vérifiez que la saisie est un nombre positif.
 - Créez une instance de `Cercle` avec le rayon saisi.
 - Affichez avec un `alert` :

```
Le cercle de rayon ... cm a une surface de ... cm² et un périmètre de ... cm.
```