Atelier 2 : Javascript et POO

Objectif : L'objectif principal de ce Travail pratique et de se familiariser avec les conceptes POO de langage javascript.

Exercice 1 :

1. Ecrire une fonction constructeur voiture avec les attributs « model , marque, année, type, carburant » .
2. créer une liste des voitures.
3. implémenter le mécanisme d'héritage entre la classes voitures et deux autre classes Hyndai (attributs : série(string) , hybride (booléen) / méthodes : alarmer())et Ford (options (tableau)).
4. Trier puis afficher les voiture selon un ordre croissant des année.

Exercice 2 :

1. Creer deux objet native Etudiant (nom : string , prenom : string , age : number , cne : string) et Professeur (nom : string , age : number , cin : string ,)
2. Ajouter une méthode étudier() a l'objet Etudiant, puis ajouter une méthode enseigner() a l'objet Etudiant Professeur.
3. Trier les étudiants selon l'ordre alphabétique « nom et prénom ».

Exercice 3 :

En utilisant le javascript avec le standard EcmaScript 6 :

Définir une classe Vecteur2D avec un constructeur fournissant les coordonnées par défaut d'un vecteur du plan (par exemple : x = 0 et y = 0).

Dans le programme principal, instanciez un Vecteur2D sans paramètre, un Vecteur2D avec ses deux paramètres, et affichez-les.

Enrichissez la classe Vecteur2D précédente en lui ajoutant une méthode d'affichage et une méthode de surcharge d'addition de deux vecteurs du plan.

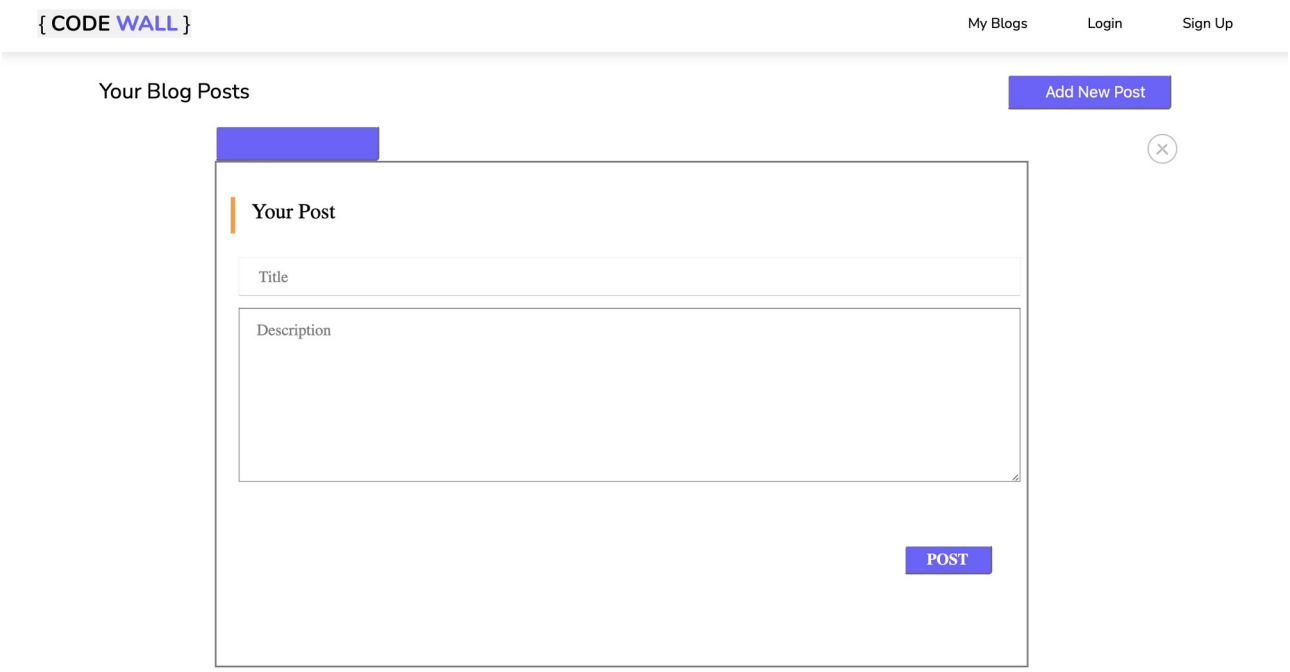
Dans le programme principal, instanciez deux Vecteur2D, affichez-les et affichez leur somme.
Définir une classe Rectangle avec un constructeur donnant des valeurs (longueur et largeur) par défaut et un attribut nom = "rectangle", une méthode d'affichage et une méthode surface renvoyant la surface d'une instance.

Définir une classe Carre héritant de Rectangle et qui surcharge l'attribut d'instance : nom = "carré".
Dans le programme principal, instanciez un Rectangle et un Carre et affichez-les.
Définir une classe Point avec un constructeur fournissant les coordonnées par défaut d'un point du plan (par exemple : x = 0.0 et y = 0.0).

Définir une classe Segment dont le constructeur possède quatre paramètres : deux pour l'origine et deux pour l'extrémité. Ce constructeur définit deux attributs : orig et extrem, instances de la classe Point. De cette manière, vous concevez une classe composite : la classe Segment est composée de deux instances de la classe Point.

Exercice 4 :

En se basant sur la paradigme POO essayer de mettre en place le mini blog suivant (Ajout , Liste , User) :



Note : Les informations « Post , Use » doivent êtres enregistrées dans un fichier JSON « Blog.json », penser a implémenter le mécanisme qui va convertir Class vers JSON et vice vers ça.

Livrable : Github avec un mini rapport sur read me.