Exercices – Série 1

Pour cette série d’exercice, créez un repository spécifique /JS/SEQ4/s02/serie-01/ et publiez-le sur GITHUB en public.

Chaque exercice sera dans son propre sous-répertoire (exo-01…). Quand les exercices sont fini : COMMIT et PUSH. Créez pour chaque exercice un fichier html (exo-01.html) et joignez-y un fichier js (exo-01.js).

# Déclarer des classes

|  |
| --- |
| **Déclaration et manipulation d’une classe** |
| *Pour déclarer une classe, il faut définir le nom de la classe ainsi que ces attributs.*  class Stagiaire  {  nom;  prenom;  } |
| *Ici n’est pas précisé le type des attributs, il peut être une bonne idée de le noter en commentaire afin de rester cohérent. JavaScript n’est pas un langage fortement typé.*  *Pour manipuler une personne, il faut la créer, l’instancier :*  let personne = new Stagiaire();  *Pour modifier la valeur des attributs*  personne.nom = "DUPONT";  personne.prenom = "Martin";  *Pour accéder à la valeur d’un attribut :*  console.log(personne.nom);  console.log(personne.prenom); |

Écrire une page HTML avec un fichier js associé : main.js, contenant le code qui s’exécute directement.

## Personne

Ecrivez le fichier personne.js qui contient la définition de la classe Personne (nom, prénom). Écrire le script javascript dans main.js qui demande votre nom et prénom (prompt) et instancie un objet de la classe Personne. Affichez ensuite cette personne dans le log.

## Animal

Ecrivez le fichier animal.js qui contient la définition de la classe Animal (nom, espèce). Écrire le script javascript dans main.js qui demande le nom et l’espèce de l’animal (prompt) et instancie un objet de la classe Animal. Affichez ensuite cette personne dans le log.

## Véhicule

Ecrivez le fichier vehicule.js qui contient la définition de la classe Vehicule (marque, modèle, nombrePlaces). Écrire le script javascript dans main.js qui demande la marque, le modèle et le nombre de place du véhicule (prompt) et instancie un objet de la classe Animal. Affichez ensuite ce véhicule dans le log.

## Maison

Ecrivez le fichier maison.js qui contient la définition de la classe Maison (numeroRue, typeVoie, nomVoie, codePostal, ville). Écrire le script JavaScript dans main.js qui demande les informations et afficher l’objet créé dans le log.

## Constructeurs

Dans le main.js, ajoutez les lignes suivantes : assurez-vous de les rendre fonctionnelles.

let formateur = new Personne('BRU','Thierry');

let chatFormateur = new Animal ('Arwen','chat');

let vehiculeFormateur = new Vehicule('ninebot','NavBOT G30MAX','1');

let domicileFormateur = new Maison(1,'rue','du paradis','18999','Las Angelos');

|  |
| --- |
| **Création d’un constructeur** |
| *Le constructeur par défaut reste limité. Un constructeur personnalisé peut être créé.*  constructor(pPrenom="",pNom="")  {  this.prenom = pPrenom;  this.nom = pNom;  } |
| *Le « this » permet d’accéder à l’instance qui est en cours de manipulation, l’objet s’accède lui-même.* |

## Une méthode

Ajoutez une méthode afficher () pour chaque classe. Pour tester, ajoutez dans main.js les lignes suivantes.

formateur.afficher()

chatFormateur.afficher()

vehiculeFormateur.afficher()

domicileFormateur.afficher()

Vous devez obtenir dans le log les lignes suivantes :

* « La personne s’appelle Thierry BRU. »
* « L’animal s’appelle Arwen et est de l’espèce Chat. »
* « Le véhicule est de la marque ninebot et du modèle NavBOT G30MAX. Il peut transporter 1 personne(s). »
* « Le logement se situe 1 rue du paradis - 18999 Las Angelos »

|  |
| --- |
| **Création de méthodes** |
| *Des méthodes supplémentaires peuvent être créées, un peu comme des fonctions.*  salutation()  {  return `je m'appelle ${this.prenom} ${this.nom}`;  } |
| *Le « this » permet d’accéder à l’instance qui est en cours de manipulation, l’objet s’accède lui-même.* |

# Calculer la surface d’un rectangle

Écrire une page HTML avec deux fichiers js associé : main.js, contenant le code qui s’exécute directement et rectangle.js, contenant la définition de la classe Rectangle.

Créez un constructeur pour Rectangle prenant en paramètre la longueur et la largeur.

Ajoutez à la classe Rectangle une méthode retournant la surface du rectangle ainsi que la longueur du périmètre.

Dans le fichier main.js, demandez (prompt) à l’utilisateur une longueur et une largeur. Affichez avec un message d’alerte (alert) indiquant la surface et le périmètre du rectangle.

# Calcul de la surface d’un cercle

|  |
| --- |
| **Utiliser une propriété de la classe Math** |
| *La propriété* [*Math.PI*](https://developer.mozilla.org/fr/docs/Web/JavaScript/Reference/Global_Objects/Math/PI) *représente le ratio entre le périmètre d'un cercle et son diamètre. Elle vaut environ 3.14159. Pour élever au carré, utiliser* [*Math.Pow()*](https://developer.mozilla.org/fr/docs/Web/JavaScript/Reference/Global_Objects/Math/Pow)  console.log(Math.PI);// affiche la valeur de PI  let deuxPi = 2\*Math.PI; // affecte 2PI à la variable deuxPi  let carre = Math.pow(12,2)// affecter le carré de 12 à la variable carre |
|  |

Écrire un programme javascript et créez la classe Cercle. Le constructeur prend un seul paramètre : le rayon. Créez les méthodes calculerSurface() et calculerPerimetre.

Dans le programme « main.js » :

* Demandez à l’utilisateur le rayon du cercle
* Affichez dans une alerte: « le cercle de rayon … cm pour surface … cm2. Son périmètre est de … cm »
* Contrôlez les saisies et empêchez l’utilisateur de saisir autre chose qu’un entier positif pour le rayon du cercle.