**Sujet**

**Exercice JavaScript - Gestion des étudiants**

**Objectif :**

Le but de cet exercice est de créer un programme JavaScript qui gère les informations et les opérations sur les étudiants. Le programme permettra à l'utilisateur d'entrer les informations d'un étudiant (prénom, nom, notes) et d'effectuer des opérations telles que le calcul de la moyenne, la recherche de la note la plus élevée, etc. Nous allons utiliser **Node.js** et **npm** pour exécuter et gérer notre programme.

**Instructions :**

1. Création du projet
   * Ouvrez votre éditeur de texte et créez un nouveau fichier.
   * Initialisez le projet en exécutant la commande suivante dans votre terminal : **npm init -y**
   * Cela va créer un fichier **package.json** qui contiendra les informations sur votre projet.
2. Installation de lodash
   * Dans votre terminal, exécutez la commande suivante pour installer le package lodash : npm install lodash
   * Cela va installer **lodash** et mettre à jour votre fichier **package.json** avec les informations de dépendance.
3. Demande des informations de l'étudiant
   * Utilisez la fonction **readline** pour demander à l'utilisateur d'entrer le *prénom*, le *nom* et les *notes* de l'étudiant.
   * Utilisez les fonctions **split** et **map** pour convertir la chaîne de notes en un tableau de nombres.
4. Création de l'objet étudiant et calcul de la moyenne
   * Créez un objet **etudiant** avec les propriétés *prenom*, *nom* et *notes*.
   * Ajoutez une méthode moyenne à l'objet **etudiant** pour calculer la moyenne des notes en utilisant la fonction **reduce** pour additionner les notes et la propriété **length** pour obtenir le nombre de notes.
   * Affichez la moyenne de l'étudiant.
5. Recherche de la note la plus élevée
   * Utilisez la fonction **Math.max** avec l'opérateur de déconstruction (...) pour trouver la note la plus élevée dans le tableau de notes.
   * Affichez la note la plus élevée.
6. Comparaison de deux notes
   * Créez une fonction **compareNumbers** qui prend deux nombres en paramètres et renvoie le plus grand des deux ou une égalité.
   * Testez la fonction en comparant les deux premières notes de l'étudiant.
7. Calcul de la factorielle
   * Créez une fonction factorielle qui calcule la factorielle d'un nombre en utilisant une récursion.
   * Testez la fonction en calculant la factorielle du nombre total de notes de l'étudiant.
8. Utilisation d'un ensemble (**Set**)
   * Créez un ensemble (**Set**) avec les notes de l'étudiant.
   * Utilisez la méthode **has** pour vérifier si une note spécifique (par exemple, 10) est présente dans l'ensemble.
   * Affichez le résultat de la vérification.
9. Utilisation d'une table de hachage (**Map**)
   * Créez une table de hachage (**Map**) pour associer le nom de l'étudiant à ses notes.
   * Utilisez la méthode set pour ajouter une entrée à la table de hachage.
   * Utilisez la méthode **get** pour récupérer les notes de l'étudiant à partir de son nom.
   * Affichez les notes de l'étudiant.
10. Utilisation de **lodash** pour mélanger les notes
    * Utilisez la fonction **shuffle** de **lodash** pour mélanger les notes de l'étudiant.
    * Affichez les notes avant et après le mélange.