

1 JavaScript (10 points)

1.1 Question JavaScript (1 point)

Soit une liste nommée *lis*, donnez le fragment de code qui affiche toutes les valeurs de cette liste dans la console.

1.2 Problème JavaScript (9 points)

Vous développez un mécanisme qui permet de faire apparaître dans une zone de saisie d'un formulaire le nom de la ville qui correspond au code postal que vient de saisir l'internaute dans une autre zone de saisie.

1.2.1 Structure de données (1 point)

Les codes postaux (par exemple *34090*) et les noms des villes correspondantes (par exemple *Montpellier*) sont stockés dans une structure de données de votre choix nommée *codesVilles* (liste, dictionnaire...) et directement accessible dans votre code côté serveur : `var codesVilles = ...`

Pour illustrer cette structure de données, remplacer les ... précédents avec quelques valeurs (mêmes imaginaires).

1.2.2 Côté serveur (2.5 points)

Soit un code JavaScript utilisant les frameworks *Node.js* et *express*, donnez le code de l'endpoint qui sur une route comportant un code postal renvoie le nom de la ville associée.

1.2.3 Côté client (5.5 points)

Donnez le code HTML/JavaScript qui affiche un formulaire contenant une zone de saisie du code postal et une zone de saisie où apparaîtra le nom de la ville. Ce code permet d'invoquer le serveur lors de la validation de la zone de saisie du code postal (sur l'événement *onChange*), de récupérer le nom de la ville renvoyée par le serveur et puis de l'insérer dans le DOM comme valeur de la zone de saisie de la ville.

2 PHP (10 points)

2.1 Question PHP (1 point)

Soit un tableau nommé *tab*, donnez le code qui affiche chaque valeur de ce tableau entre deux balises `<li>`.

2.2 Problème PHP (9 points)

Vous développez une application de locations de voitures. Une base de données nommée *VOILOC* a déjà été créée et comporte :

une table *voitures* comportant (entre autres) comme attributs un identifiant unique (*idVoiture*), le type de véhicule (*type* de valeurs possibles *citadine* ou *routière*), la marque (*marque*) et le modèle (*modèle*).

une table *locations* comportant (entre autres) comme attributs un identifiant unique (*idLocation*), l'identifiant de la voiture louée (*idVoitureLouee*), la date de début de location (*dateDebut*) et celle de fin (*dateFin*). Les dates sont stockées sous la forme d'entiers *aaaammjj* (ex. : 20240117).

Donnez le code PHP qui permet d'afficher un formulaire comportant la liste des catégories de voitures, deux zones de saisie pour les dates de début et de fin de location et un bouton de validation.

Ces trois informations étant sélectionnées (vous n'avez pas à contrôler qu'ils le soient), lors de la soumission de ce formulaire, le script PHP doit se rappeler et afficher la liste des véhicules disponibles correspondant aux critères choisis par l'internaute (avant de réafficher le formulaire de recherche).

*Si vous ne savez pas gérer la vérification qu'un véhicule a pu être déjà loué sur plusieurs dates, une version simplifiée du problème est de considérer qu'une voiture est louée pour l'année entière : dans ce cas un seul enregistrement aura été créé dans la table *locations* du 20240101 au 20241231.*