

Programmation 5

JavaScript – TP

Partie 1 : Traitement de données

1.1 - En JavaScript, créez un tableau à 2 dimensions de chaînes de caractères correspondant au tableau de profil-utilisateur suivant :

Prénom	Nom	Age	Ville	Adresse mail	Identifiant
Mathias	Ritrono	34	Nantes	mr@yopmail.com	1
Salif	Bamogo	25	Brest	sb@yopmail.com	2
Kristina	Pavlov	20	Villeurbanne	kp@yopmail.com	3
Elise	Seti	30	Grenoble	es@yopmail.com	4
Pierrick	Di Angelo	18	Strasbourg	pd@yopmail.com	5
Mohamed	Khaoui	23	Lille	mh@yopmail.com	6
Paige	Anderson	42	Evry	pa@yopmail.com	7

1.2 - En JavaScript, créez une fonction permettant d'ajouter au tableau un profil-utilisateur. Cette fonction doit demander paramètres 5 chaînes de caractères :

- Une chaîne correspondant au prénom
- Une chaîne correspondant au nom
- Une chaîne correspondant à l'âge
- Une chaîne correspondant à la ville
- Une chaîne correspondant à l'adresse mail

Lors de l'ajout dans le tableau, l'identifiant doit être généré automatiquement. Cet ID doit être unique (deux utilisateurs ne peuvent pas partager le même identifiant).

1.3 - En JavaScript, créez une fonction qui prend en paramètre une chaîne de caractères correspondant à une ville. Cette fonction retourne, sous forme de tableau à 2 dimensions, les profils des utilisateurs habitant dans cette ville.

1.4 - En JavaScript, créez une fonction qui prend en paramètre une chaîne de caractères correspondant à un prénom ou un nom. Cette fonction retourne, sous forme de tableau à 2 dimensions, les profils des utilisateurs ayant ce prénom ou ce nom.

1.5 - En JavaScript, créez une fonction qui prend en paramètre une chaîne de caractères correspondant. Cette fonction retourne, sous forme de tableau à 2 dimensions, les profils des utilisateurs dans lesquelles apparaissent la chaîne de caractères passée en paramètre.

Partie 2 : Affichage graphique

2.1 – En JavaScript, créez une fonction prenant en paramètres un tableau à 2 dimensions et un booléen. Cette fonction génère une table HTML (balise <table>) à partir du tableau passé en paramètre. Si le booléen passé en paramètre vaut *VRAI*, alors, on considère que la 1ère valeur du tableau correspond à une en-tête ; la fonction génère dans ce cas des éléments HTML de type <th>.

2.2 – Via un fichier CSS, faites en sorte que toutes les l'en-tête de table HTML affichée à l'écran aient :

- une couleur de fond noire
- une couleur de police de caractère blanche

2.3 – Créez un fichier HTML, et liez-le avec les fichiers JavaScript et CSS créés précédemment. Au chargement de la page HTML, une table HTML doit être affichée. Celle-ci doit représenter le tableau de profils-utilisateurs crée en **1.1**.

2.4 – Faites apparaître à l’écran un bouton “Ajouter un utilisateur”. Au clic sur ce bouton, cachez la table HTML, et affichez une interface permettant d’ajouter un utilisateur. Cette interface doit contenir :

- un champ-texte correspondant au nom du nouvel utilisateur
- un champ-texte correspondant au prénom du nouvel utilisateur
- un champ-numérique correspondant à l’âge du nouvel utilisateur
- un champ-texte correspondant à la ville du nouvel utilisateur
- un champ-texte correspondant à l’adresse mail du nouvel utilisateur
- un bouton “OK”

Lorsque le bouton “OK” est cliqué, vérifiez que les champs-textes et champ-numérique ne soit pas vide. Si un des champ est vide, affichez un message d’erreur, sinon :

- utilisez la fonction créée en **1.2** pour ajouter ce nouvel utilisateur au tableau JS
- utilisez la fonction créée en **2.1** pour afficher la liste des profils-utilisateurs

2.5 – Lorsque la table correspondant aux profils-utilisateurs est visible, une interface de recherche doit apparaître à l’écran. Cette interface doit permettre d’afficher des vues filtrées :

- on doit pouvoir rechercher et afficher des utilisateurs en fonction de leur ville (utilisez les fonctions créées en **1.3** et **2.1**)
- on doit pouvoir rechercher et afficher des utilisateurs en fonction de leur prénom/nom (utilisez les fonctions créées en **1.4** et **2.1**)
- on doit pouvoir rechercher et afficher des utilisateurs en fonction d’un terme (utilisez les fonctions créées en **1.5** et **2.1**)