

Programmation – TP 6

Traitements séquentiels et interfaces graphiques

JavaScript

Exercice 1

Créez une fonction qui prend en paramètre un tableau de chaînes de caractères, et qui, via un traitement séquentiel, affiche sur un navigateur l'ensemble des éléments du tableau.

Les éléments doivent être affichés **horizontalement**, en générant des éléments HTML `<p>`.

Exemple :

Si la fonction prend en paramètre le tableau :

`["foo", "bar", "baz", "qux", "quux", "corge"]`

La fonction doit générer l'affichage suivant :

foo bar baz qux quux corge

Note : Pour savoir pourquoi il est courant d'utiliser des noms de variables tels que *foo*, *bar*, *baz*, etc., voir la notion de *variable métasyntaxique* (https://fr.wikipedia.org/wiki/Variable_m%C3%A9tasyntaxique)

Exercice 2

Créez une fonction qui prend en paramètre un tableau de chaînes de caractères, et qui, via un traitement séquentiel, affiche sur un navigateur l'ensemble des éléments du tableau.

Les éléments doivent être affichés **verticalement**, en générant des éléments HTML `<label>`.

Exemple :

Si la fonction prend en paramètre le tableau :

`["foo", "bar", "baz", "qux", "quux", "corge"]`

La fonction doit générer l'affichage suivant :

*foo
bar
baz
qux
quux
corge*

Exercice 3

Créez une fonction qui prend en paramètre un tableau de chaînes de caractères, et qui, via un traitement séquentiel, affiche sur un navigateur l'ensemble des éléments du tableau.

Les éléments doivent être affichés **horizontalement**, en générant les éléments HTML `<table>`, `<tbody>`, `<tr>`, `<td>`.

Exemple :

Si la fonction prend en paramètre le tableau :

`["foo", "bar", "baz", "qux", "quux", "corge"]`

La fonction doit générer l'affichage suivant :

foo	bar	baz	qux	quux	corge
-----	-----	-----	-----	------	-------

Note : Pour plus d'information sur l'élément HTML `<table>`, voir la documentation MDN (<https://developer.mozilla.org/en-US/docs/Web/HTML/Element/table>)

Exercice 4

Créez une fonction qui prend en paramètre un tableau de chaînes de caractères, et qui, via un traitement séquentiel, affiche sur un navigateur l'ensemble des éléments du tableau. Les éléments doivent être affichés **verticalement**, en générant les éléments HTML `<table>`, `<tbody>`, `<tr>`, `<td>`.

Exemple :

Si la fonction prend en paramètre le tableau :

`["foo", "bar", "baz", "qux", "quux", "corge"]`

La fonction doit générer l'affichage suivant :

foo
bar
baz
qux
quux
corge

Exercice 5

Créez une fonction qui prend en paramètre un tableau de chaînes de caractères, et qui, via un traitement séquentiel, affiche sur un navigateur l'ensemble des éléments du tableau.

Les éléments doivent être affichés **horizontalement**, en générant les éléments HTML `<table>`, `<thead>`, `<tbody>`, `<tr>`, `<td>`, `<th>`.

La 1ère valeur du tableau passé en paramètre doit correspondre à l'en-tête de la table HTML.

Exemple :

Si la fonction prend en paramètre le tableau :

`["table test", "foo", "bar", "baz", "qux", "quux", "corge"]`

La fonction doit générer l'affichage suivant :

table test					
foo	bar	baz	qux	quux	corge

Exercice 6

Créez une fonction qui prend en paramètre un tableau à 2 dimensions de chaînes de caractères, et qui, via un traitement séquentiel, affiche sur un navigateur l'ensemble des éléments du tableau. Les éléments doivent être affichés, en générant les éléments HTML `<table>`, `<tbody>`, `<tr>`, `<td>`.

Exemple :

Si la fonction prend en paramètre le tableau :

```
[
    ["foo", "bar", "baz", "qux", "quux", "corge"],
    ["grault", "garply", "waldo", "fred", "plugh", "xyzzy"]
]
```

La fonction doit générer l'affichage suivant :

foo	bar	baz	qux	quux	corge
grault	garply	waldo	fred	plugh	xyzzy

Exercice 7

Créez une fonction qui prend en paramètre un tableau à 2 dimensions de chaînes de caractères, et qui, via un traitement séquentiel, affiche sur un navigateur l'ensemble des éléments du tableau. Les éléments doivent être affichés, en générant les éléments HTML `<table>`, `<thead>`, `<tbody>`, `<tr>`, `<td>`, `<th>`.

La 1ère valeur du tableau passé en paramètre doit correspondre à l'en-tête de la table HTML.

Exemple :

Si la fonction prend en paramètre le tableau :

```
[
    ["col1", "col2", "col3", "col4", "col5", "col6"],
    ["foo", "bar", "baz", "qux", "quux", "corge"],
    ["grault", "garply", "waldo", "fred", "plugh", "xyzzy"]
]
```

La fonction doit générer l'affichage suivant :

col1	col2	col3	col4	col5	col6
foo	bar	baz	qux	quux	corge
grault	garply	waldo	fred	plugh	xyzzy

Exercice 8 :

Partie 1 : Traitement de données

8.1 - 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

8.2 - 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).

8.3 - 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.

8.4 - 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.

8.5 - En JavaScript, créez une fonction qui prend en paramètre une chaîne de caractères. 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

8.6 – 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>.

8.7 – 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

8.8 – 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 **8.1**.

8.9 – 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 **8.2** pour ajouter ce nouvel utilisateur au tableau JS
- utilisez la fonction créée en **8.1** pour afficher la liste des profils-utilisateurs

8.10 – 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 **8.3** et **8.6**)
- on doit pouvoir rechercher et afficher des utilisateurs en fonction de leur prénom/nom (utilisez les fonctions créées en **8.4** et **8.6**)
- on doit pouvoir rechercher et afficher des utilisateurs en fonction d’un terme (utilisez les fonctions créées en **8.5** et **8.6**)