JavaScript – Session 2

Le code doit être écrit dans un fichier index.html (attention, un seul index par dossier).
Vous pouvez utiliser le template base disponible sur la page exos @ campus.ow-lab.com.
Le point de départ de vos programmes est la fonction init() activée avec un attribut onload sur body.
Le code JS sera rédigé dans un balise script, située dans le fichier HTML.

Attention => on n'utilisera jamais ici de variables globales. Utiliser le scope des fonctions

Exercice 1: Inverser les couleurs

Déclarer deux div HTML portant respectivement les id green et red.
Ces div ont un attribut style liant une couleur de fond, respectivement rouge et verte.
Placer un attribut onclick sur chaque div.
Le click doit lancer l'exécution d'une fonction invertColor().
Passer la valeur this en paramètre de cette fonction.
(test: afficher la valeur de ce paramètre dans le corps de la fonction).
Utiliser les id pour permettre à votre programme d'intervertir la couleur des div.

Avant click sur a ou b =>

Après click sur a ou b =>

Exercice 2: Inversion de lettres

Écrire une fonction swapLetters(str) inversant l'ordre des lettres de la chaîne passée en paramètre.
Elle retourne le résultat de l'inversion sous la forme d'une chaîne de caractères.
Afficher cette chaîne inversée dans la console.

Exemple: swapLetters('abcdefg') retourne 'gfedcba'.

Exercice 3: Lignes et colonnes

Écrire deux fonctions drawRow(row, col) et drawColumn(col).
DrawRow() prend 2 entiers en paramètres: nombre de lignes à dessiner, nombre de colonnes par ligne.
drawColumn() retourne une chaîne de caractères dessinant les colonnes remplissant une ligne.
drawRow() retourne une chaîne de caractères à afficher dans la console.

Appeler successivement drawRow(?,?) en jouant sur les paramètres pour obtenir ce résultat:

```
" | 0 | "
" | 0 | 1 | "
" | 0 | 1 | 2 | "
" | 0 | 1 | 2 | 3 | 4 | "
" | 0 | 1 | 2 | 3 | 4 | 5 | "
```

Exercice 4: Palindrome

Écrire la fonction estPalindrome(str) testant si la chaîne passée en paramètre est bien un palindrome.
Rappel: un palindrome est une chaîne pouvant se lire dans les deux sens.

EstPalindrome() retourne true si str est palindrome, false dans le cas contraire.
Tester avec les valeurs: 'non palindrome', 'bob', 'rotor', 'Kayak'.
Résultats attendus: false, true, true, true, true.

Pour aller plus loin : le test sur 'Sa nana snob porte de trop bons ananas' doit retourner true.

Exercice 5: Usine à Users

Créer une div HTML portant l'id res_user2
Créer un button ou input HTML portant l'id trigger_greetings.

Écrire un constructeur User() produisant des objets user.
Chaque objet a 3 membres:

1. name: string
2. age: number
3. sayHello: function(object) { /*ex: affiche dans la console: 'Hello Alfred!!!' */}

Coder la fonction createUser(count),
Cette méthode instancie *n* objet users (*n* est passé en paramètre).
Exemple: en lui passant 3 en paramètre, 3 objets user devront être créés.

Le programme demande à l'utilisateur de saisir les membres name et age pour chaque nouvel objet via une méthode prompt() fournie par JS. Une vérification de type doit être effectuée. Tant que le type est incorrecte, le programme redemandera à l'utilisateur de ressaisir.

Retourner un array contenant les *n* objets créés et le stocker.

Écrire une fonction helloUser(users), exécutée au click sur le button HTML.
Tester d'abord le contenu due l'array.
Au moyen d'une boucle, cette méthode doit afficher dans la console le résultat de la fonction sayHello(toUser) exécutée par chaque objet user et prenant en paramètre l'objet user contenu dans la case précédente de l'array.