

## JavaScript Base II – Le boucles ... suite

En HTML : déclarer 2 éléments buttons et leur attribuer les id action1 et action2,  
Sur body, placer l'attribut onload= init().  
La fonction init() contient 2 listeners : btn1.onclick = looper4 ; btn2.onclick = looper5 ;  
Un jeu de donnée de test dataset1.js est fourni dans le zip  
Note : consulter le cours (js/structures de contrôle) et la doc @ MDN !

### Exo 1 – Looper IV ( do ... while )

Réaliser le programme suivant, dans une fonction looper4:

Déclarer deux variables: **question** et **réponse**.

Affecter ces variables avec les données de votre choix (ex: **question**: 'JS is Fun', **reponse** : 'yes').

Utiliser la boucle **do ... while** :

Un **prompt()** affiche la variable question, **tant ... que** la réponse fournie n'est pas strictement égale au contenu de la variable reponse.

Le programme vérifie la saisie de l'utilisateur :

- Si la réponse est incorrecte, prompt() affiche « Mauvaise réponse » + la **question** initiale.
- Sinon, casser la boucle (**break**) => un message de succès est alors affiché.

### Exo 2 – Looper V ( forEach )

Cet exercice se déroule en deux parties.

L'objectif est d'obtenir un plateau et des cartes en HTML représentant des objets JS.

1 / Réaliser le programme suivant, dans une fonction looper5() :

Utiliser le jeu de donnée test dataset1.js fourni en annexe du zip.

Importer ces données avec une balise HTML script [src=route].

Comprendre la structure du tableau et les types de données présentes.

Télécharger des images correspondant aux différents objets du tableau.

Modifier ces objets :

Renseigner chaque propriété icon avec un **vers** une image (relatif à l'index.htm).

Option : améliorer dataset1, le modifier + y ajouter des objets et propriétés.

2 / Parcourir l'array dataset1 avec une boucle **forEach()**.

Passer **infos** en premier paramètre de forEach.

Dans le corps de la boucle, console.log(**infos**) permet d'afficher chaque objet dans la console.

À chaque tour de boucle, la fonction looper6(**infos**) est exécutée.

Option : l'appel de fonction démarre avec un **délai**.

Voir la suite ci-dessous.

### Exo 3 – Looper VI ( for ... in )

Réaliser le programme suivant, dans une fonction looper6(**infos**) :

Créer la fonction afficherCarte(**infos**) qui devra,

1 / Parcourir les propriétés de chaque objet **infos** avec une boucle **for ... in**

Note : vérifier le comportement de la boucle en les affichant temporairement dans la console.

[Aller plus loin.](#)

2 / Dans le corps de la boucle,

Créer une div HTML pour chaque objet **infos** ( ex : `div.carte#carte_id + i` ).

Insérer dans chaque `div.carte` créée, avec la même méthode:

- la propriété **infos**.nom dans une balise HTML `h2`.
- la propriété **infos**.icon dans une balise `img`.
- **infos**.hobbies sous forme de balise `ul>li`
- **infos**.genre sous forme de code couleur CSS
- **infos**.espece sous forme de balise `span`

3 / Insérer les divs créées dans un conteneur HTML (ex: `div.cartes-conteneur#carte_conteneur`).

Les classes `.carte` et `.cartes-conteneur` seront stylisées en CSS.

L'objectif est d'obtenir un plateau et des cartes (à jouer, mosaïque ou tuiles à la windows 8) ou autre... selon l'inspiration (color, background, opacity, fonts, blocks, grille flex-box, transitions, animation au hover ou au click, tri par propriété, etc).

4 / Vous avez carte blanche pour améliorer cette base.