

JavaScript Base I – Les boucles (while et for)

En HTML : déclarer 3 boutons possédant chacun un attribut onclick et un id (btn1, btn2, btn3...)

Sur body, placer l'attribut onload= init().

init() contient 3 listeners déclenchant l'exécution des fonctions loop1, II et III.

Exemple : `btn1.onclick = loop1 ;`

Exercice 1 – Looper I

Dans une fonction loop1():

Déclarer un tableau tab1 et le remplir avec *n* valeurs primitives (tel que *n* >= 5).

Parcourir tab1 à l'aide d'une boucle while. Dans la console : afficher le contenu de chaque case et son type à chaque

itération (tour de boucle). Une fois sorti la boucle, afficher : 'succès ' + *n* + ' tours de boucle' dans la console.

Exercice 2 – Looper II

En HTML : déclarer une div avec l'identifiant result_looper2

Dans une fonction loop2() :

- Déclarer trois tableaux : tab2, resultBoolean et resultNumber
- Remplir tab2 avec *n* valeurs primitives (*n* >= 10).
- Note : resultBoolean et resultNumber sont vides pour le moment.

Avec une boucle for, afficher le contenu des cases de tab2 dans la console.

- Si le type de la valeur est de type boolean, ajouter la valeur à resultBoolean
- Si le type de la valeur est de type number, ajouter la valeur à resultNumber
- Pour cela : utiliser la méthode `array.push()` (MDN est votre ami).

Afficher la longueur de ces deux tableaux de résultat dans result_looper2 et y préciser lequel de ces deux tableaux contient le plus de cases.

Exercice 3 – Looper III

En HTML : déclarer 5 éléments div, 3 p et 2 éléments ul comprenant 3 et 8 li.

Dans une fonction loop3()

- Avec la méthode `getElementsByTagName`: remplir 3 arrays (divs, parags et lists)
- Dans loop3 , déclarer une fonction interne (closure) `parseArray(arr)`
- Dans loop3 déclarer une autre fonction interne `showItems(list)`
- Exécuter `parseArray` 3 fois en passant divs, puis parags, puis lists en paramètre

Rédiger le corps de `parseArray`. La fonction doit parcourir le tableau passé en paramètre avec une boucle. À chaque itération, afficher le contenu de la case, son type et sa **propriété** `nodeName` dans la console.

Si `nodeName` équivaut à 'DIV', modifier l'identifiant de l'élément (**propriété** `id`) pour respecter ce schéma : `div_1`, `div_2`, `div_3`, etc... Compléter l'algorithme pour modifier également l'identifiant des éléments p et ul en suivant la même logique.

Si `nodeName` équivaut à 'UL', passer l'élément à la fonction `showItems(list)`.

- Dans le corps de `showItems()` :
- Faire `console.log` de l'objet `list` localiser sa **propriété** `children` dans la console
- Parcourir les enfants (`children`) de `list` avec une boucle
- Les afficher dans la console
- Changer leur couleur de police avec la propriété `style`