

Exercices – Javascript DOM

Version : 1.0

L'objectif de ces exercices est mettre en pratique les fonctionnalités permettant de manipuler le DOM.

Exercice 1 : Méthodes `getElementById`, `getElementsByName` et `getElementsByTagName`

1. Télécharger [exo01.html](#) dans lequel vous aurez à insérer votre code JavaScript. Notez en passant l'utilisation des attributs title pour informer sur les éléments de titre et paragraphe (pour les tester, laisser la souris au-dessus de l'élément).
2. Créer une fonction `modif_paragraphe`, appelée en cliquant sur le titre. Cette fonction sélectionne le paragraphe en utilisant son identifiant, puis le modifie avec la propriété `innerHTML`, en remplaçant le mot **original** en caractères droit par le mot **corrigé**, en italique.
3. Créer une fonction `centrage_h1`, appelée en cliquant sur le paragraphe. Cette fonction détecte d'abord les éléments portant le nom de balise `h1`. Elle sélectionne ensuite le premier d'entre eux, et modifie son attribut `align`, en lui affectant la valeur "center", à l'aide de la méthode `setAttribute`.

Exercice 2 : Changements d'attributs et méthode `getElementsByTagName`

1. Télécharger le fichier [exo02.html](#).
2. Récupérer l'objet de liste à l'aide d'une méthode `getElementsByTagName`.
3. Récupérer la collection d'items de liste à l'aide d'une méthode `getElementsByTagName`.
4. Changer la valeur de l'attribut de type de la liste en "circle".
5. Pour chaque élément de liste, changer le texte en "blablabla", et affecter à l'attribut `onclick` un appel à une boîte d'alerte avec un petit texte.

Exercice 3 : Accès aux enfants

1. Télécharger le fichier [exo03.html](#).
2. La fonction `change_enfants()` sélectionne le paragraphe à l'aide de son identifiant.
3. Affiche combien d'enfants a le paragraphe.
4. Changer la valeur du premier enfant de l'élément paragraphe en "On change le texte plein d'enfants".
5. Diviser par deux les hauteurs et largeur de l'image (le dernier enfant).
6. Changer le code HTML du deuxième enfant en "en bleu".
7. Changer la propriété `display` de style du quatrième enfant en "block".

Exercice 4 : Manipulation des styles des enfants

1. Télécharger le fichier [exo04.html](#).
2. Créer une variable globale `couleur` initialisée à 0.

3. Dans la fonction `anime()` :
 1. Identifier l'élément d'identifiant `citation`.
 2. Faire une boucle sur l'ensemble de la collection des éléments-enfants.
 3. Pour chaque élément-enfant, tester s'il s'agit d'un noeud de paragraphe. Si oui, appeler la fonction `anime_paragraphe` avec comme argument l'objet-élément.
4. Dans la fonction `anime_paragraphe(element)` :
 1. Faire en sorte que la couleur de l'élément vaille `"rgb("+String(couleur)+"%,"+String(couleur)+"%,"+String(couleur)+"%")"` (cela permet d'exprimer une couleur rgb en pourcentages de rouge, vert et bleu).
 2. Si `couleur` est inférieure à 90 :
 1. appeler la fonction `setTimeout()`, avec comme premier argument `function(){anime_paragraphe(element)}`, et un délai de 100 ms.
 2. Incrémenter `couleur`.
 3. Si `couleur` est supérieure ou égale à 90, faire apparaître `"Stop !"` dans le paragraphe d'identifiant `info`.

Exercice 5 : Créer et ajouter des éléments

1. Télécharger le fichier [exo05.html](#).
2. Sélectionner la liste d'identifiant `"listecommissions"`.
3. Créer un nouvel item de liste et lui affecter un texte.
4. Ajouter ce nouvel élément de liste à la liste

Modifier le code en ajoutant un champ de saisie de texte et un bouton de validation. Le texte ajouté à la liste devra être saisi dans le champ de... saisie.

Exercice 6 : Suppression d'éléments

1. Reprendre l'exercice précédent. Ajouter un bouton permettant, par un appel de fonction, de supprimer le dernier élément de la liste.
2. Que se passe-t-il quand on essaie de supprimer un élément de la liste lorsque celle-ci est vide ? Corriger le code pour éviter cette erreur.