

Exercices sur les événements en JavaScript

Exercice 1 – Réagir à un clic sur un bouton

Créez une page HTML contenant un bouton avec l'attribut `id="monBouton"`.

Dans un script JavaScript, utilisez un écouteur d'événement pour détecter le clic sur ce bouton. Lorsque l'utilisateur clique sur le bouton, affichez une boîte de dialogue (`alert`) contenant le texte suivant :

"Vous avez cliqué sur le bouton."

Exercice 2 – Modifier le contenu d'un paragraphe au clic

Ajoutez dans la page un paragraphe avec `id="texte"` et un bouton avec `id="modifierTexte"`.

Écrivez un script qui, lorsqu'on clique sur le bouton, modifie le contenu du paragraphe pour qu'il affiche :

"Le texte a été modifié avec succès."

Exercice 3 – Changer la couleur d'une zone avec la souris

Ajoutez une `<div>` de 200 pixels de large et 100 pixels de haut avec l' `id="zone"`.

Lorsque l'utilisateur passe la souris sur cette zone (`mouseover`), changez la couleur de fond de la div en **bleu clair**.

Quand la souris quitte la zone (`mouseout`), la couleur de fond doit revenir à **blanc**.

Exercice 4 – Compter les clics d'un bouton

Ajoutez un bouton avec l' `id="compter"` et un paragraphe avec l' `id="resultat"`.

À chaque fois que le bouton est cliqué, le paragraphe doit afficher un message sous la forme :

"Vous avez cliqué X fois."

où `X` est le nombre de clics effectués depuis le chargement de la page.

Exercice 5 – Afficher la valeur d'un champ texte

Ajoutez un champ de texte avec l' `id="champNom"` et un bouton avec l' `id="afficherNom"` .

Lorsqu'on clique sur le bouton, le contenu du champ doit être affiché dans un paragraphe `id="message"` , sous la forme suivante :

"Bonjour, [valeur du champ]"

(par exemple : "Bonjour, Sophie").

Exercice 6 – Activer et désactiver un champ

Ajoutez un champ texte avec l' `id="champTexte"` (initialement activé) et un bouton `id="toggleChamp"` .

Lorsqu'on clique sur le bouton, le champ doit alterner entre les états **activé** et **désactivé**.

Si le champ est désactivé, le bouton doit l'activer. S'il est activé, le bouton doit le désactiver.

Exercice 7 – Ajouter dynamiquement un élément à une liste

Ajoutez un bouton avec `id="ajouter"` et une liste non ordonnée (``) avec `id="listeElements"` .

Lorsque l'utilisateur clique sur le bouton, un nouvel élément `` doit être ajouté à la liste, contenant le texte suivant :

"Nouvel élément ajouté"

Exercice 8 – Supprimer l'élément cliqué dans une liste

Créez une liste (``) contenant 5 éléments `` ayant tous la classe `"cliquable"` .

Lorsque l'utilisateur clique sur un de ces éléments, celui-ci doit être supprimé de la liste.

Exercice 9 – Détecter la touche Entrée dans un champ

Ajoutez un champ texte avec l' `id="champClavier"` et un paragraphe avec `id="messageClavier"` .

Quand l'utilisateur appuie sur la touche **Entrée** dans ce champ, le paragraphe doit afficher ce message exact :

"Vous avez appuyé sur la touche Entrée."

Si une autre touche est pressée, ne rien faire.

Exercice 10 – Filtrer dynamiquement une liste à partir d'un champ

Créez une liste (``) avec les éléments suivants : "chat", "chien", "cheval", "cochon", "cerf". Donnez-leur tous la classe `animal` .

Ajoutez un champ de texte avec l' `id="filtreAnimaux"` .

À mesure que l'utilisateur tape du texte dans ce champ (événement `"input"`), seuls les éléments de la liste contenant ce texte doivent rester visibles.

Les autres doivent être cachés (`display: none`).
