

Langage Javascript SN2 - Séance 3 - EXERCICES - Lille

Enoncés :

Le rendu attendu est un fichier compressé au format ZIP portant le nom de l'étudiant (prenom-nom.zip). À l'intérieur de ce fichier ZIP, chaque exercice devra être représenté par un fichier HTML portant le nom "exercice1.html" pour le premier exercice, "exercice2.html" pour le deuxième, et ainsi de suite. Si vous décidez de séparer les fichiers JavaScript, ils devront suivre la même convention de nommage que les fichiers HTML correspondants.

**ATTENTION CHAQUE EXERCICE NON RENDU
SERA PÉNALISÉ PAR SON NOMBRE DE POINT.
IL FAUT AU MOINS ESSAYER !**

Exercice 1 - Afficher un message au clic (1 point)

Créez une page HTML avec un bouton ayant un ID "bouton1" et une zone de texte vide. Utilisez JavaScript pour ajouter un gestionnaire d'événements au bouton. Lorsque l'utilisateur clique sur le bouton, insérer un paragraphe contenant le message "Hello, World!" dans la zone de texte

Exemple de sortie : Lorsque l'utilisateur clique sur le bouton, la zone de texte affiche : "Hello, World!".

Exercice 2 - Changement de style au survol (3 points)

Créez une page HTML classique.

Utilisez JavaScript pour créer dynamiquement une div, ajouter un paragraphe avec un texte "Survolez-moi" et ajouter un gestionnaire d'événements qui modifie le style de l'élément lorsque l'utilisateur le survole.

Lorsque l'utilisateur clique sur l'élément, permettez-lui de choisir parmi plusieurs options de style à appliquer, telles que la couleur de fond, la couleur du texte, ou la taille de la police.

Vous pouvez créer une interface utilisateur simple (par exemple, un menu déroulant, une liste, des tips etc....) pour permettre à l'utilisateur de sélectionner ces options.

Appliquez le style choisi à l'élément lorsque l'utilisateur clique sur le style choisi.

Exemple de sortie : Lorsque l'utilisateur survole l'élément, il peut choisir une nouvelle couleur de fond, une nouvelle couleur de texte ou une nouvelle taille de police, et le style de l'élément change en conséquence.

Exercice 3 - Liste interactive (1 point)

Créez une page HTML contenant une liste à puces (ul) avec plusieurs éléments de liste (li). Utilisez JavaScript pour ajouter un gestionnaire d'événements qui permet à l'utilisateur de cliquer sur un élément de la liste pour le marquer comme "terminé" en changeant son style (par exemple, en le mettant en gras ou en changeant la couleur). De plus, ajoutez un bouton pour supprimer un élément de la liste.

Exemple de sortie : Lorsque l'utilisateur clique sur un élément de la liste, cet élément devient gras et de couleur verte. Un bouton à côté de chaque élément permet de le supprimer de la liste lorsque l'utilisateur le souhaite.

Exercice 4 - Validation de formulaire (2 point)

Créez un formulaire HTML avec les champs requis : Nom, Prénom, Date de naissance, Lieu de naissance, Adresse, Email, Téléphone et un bouton "Envoyer".

Utilisez JavaScript pour ajouter un gestionnaire d'événements au bouton "Envoyer" du formulaire.

Lorsque l'utilisateur clique sur le bouton "Envoyer", vérifiez que tous les champs obligatoires sont remplis. Si l'un des champs obligatoires est vide, affichez un message d'erreur indiquant que les champs obligatoires doivent être complétés.

Vérifiez également que chaque champ est correctement formaté (par exemple, que l'adresse email est au bon format, que le numéro de téléphone ne contient que des chiffres, etc.). Affichez des messages d'erreur spécifiques pour chaque champ s'il y a des erreurs de format.

Si tous les champs sont remplis correctement, affichez un message de confirmation indiquant que le formulaire a été soumis avec succès.

Exemple de sortie : Lorsque l'utilisateur clique sur le bouton "Envoyer", s'il a laissé des champs vides ou mal formatés, il voit des messages d'erreur correspondants. S'il remplit correctement tous les champs, il voit un message de confirmation.

Exercice 5 - Galerie d'images (2 points)

Créez une galerie d'images avec des vignettes cliquables. Utilisez JavaScript pour créer un carrousel d'images qui permet à l'utilisateur de faire défiler les images à l'aide de boutons précédents/suivants. Ajoutez également un bouton pour afficher une image en taille réelle dans une boîte modale avec la possibilité de zoomer.

Exemple de sortie : Lorsque l'utilisateur clique sur une vignette d'image, une boîte modale s'ouvre avec l'image en taille réelle. L'utilisateur peut faire défiler les images à l'aide des boutons précédents/suivants et zoomer sur l'image.

Exercice 6 : Modification de texte avec jQuery (1point)

Créez une page HTML avec un bouton. Lorsque l'utilisateur clique sur le bouton, le texte d'un élément HTML (par exemple, une balise <p>) change en utilisant jQuery.

Écrivez un script jQuery pour ajouter un gestionnaire d'événements au bouton "Cliquez pour changer le texte". Lorsque le bouton est cliqué, le texte de l'élément avec l'ID "text-to-change" doit être modifié.

Lorsque l'utilisateur clique sur le bouton, le texte de l'élément <p> doit changer. Par exemple, vous pouvez le faire passer de "Texte initial" à "Nouveau texte !" ou à tout autre texte de votre choix.

Assurez-vous que votre script jQuery est inclus dans la page HTML.

Exemple de sortie : Lorsque l'utilisateur clique sur le bouton, le texte de l'élément <p> change instantanément.

Exercice 7 - Horloge en temps réel (2 points)

Créez une horloge en temps réel qui affiche l'heure actuelle.

Utilisez JavaScript pour mettre à jour l'heure en temps réel sans recharger la page. Ajoutez également la possibilité de choisir un fuseau horaire pour afficher l'heure dans différents endroits du monde.

Exemple de sortie :

L'horloge affiche l'heure actuelle et se met à jour en temps réel sans recharger la page.

L'utilisateur peut choisir un fuseau horaire parmi une liste déroulante pour afficher l'heure dans différents endroits du monde.

Exercice 8 - Compte à rebours (2points)

Créez une page HTML contenant un élément (par exemple, un paragraphe) qui affiche un compte à rebours à partir d'un nombre donné.

Utilisez JavaScript pour décrémenter le compte à rebours en temps réel et mettre à jour l'affichage toutes les secondes jusqu'à ce qu'il atteigne zéro. Ajoutez également un bouton pour démarrer/arrêter/reprendre le compte à rebours.

Exemple de sortie :

Le compte à rebours commence à partir d'un nombre donné et décrémente en temps réel toutes les secondes jusqu'à atteindre zéro. L'utilisateur peut démarrer/arrêter/reprendre le compte à rebours à l'aide d'un bouton.

Exercice 9 - Drag and Drop (3 points)

Créez une page HTML contenant plusieurs éléments (par exemple, des images) que l'utilisateur peut faire glisser et déposer ailleurs sur la page.

Utilisez JavaScript pour ajouter un gestionnaire d'événements qui permet le glisser-déposer. Lorsque l'utilisateur dépose un élément, affichez un message pour indiquer où il a été déposé, et ajoutez la possibilité de redimensionner les éléments déplacés.

Exemple de sortie :

L'utilisateur peut faire glisser et déposer des éléments (par exemple, des images, du texte) à différents endroits de la page. Un message s'affiche pour indiquer où l'élément a été déposé.

En bonus : l'utilisateur peut redimensionner les éléments déplacés.

Exercice 10 : Gestionnaire de tâches (regroupe plusieurs exercices - 3 Points)

Les utilisateurs peuvent ajouter, supprimer et marquer des tâches comme terminées. Les tâches doivent être stockées localement dans le navigateur.

Créez une page HTML avec une liste `` vide pour afficher les tâches. Ajoutez un formulaire avec un champ de texte pour ajouter de nouvelles tâches et un bouton "Ajouter".

Utilisez JavaScript pour ajouter un gestionnaire d'événements au bouton "Ajouter". Lorsque l'utilisateur soumet le formulaire, la nouvelle tâche doit être ajoutée à la liste.

Chaque tâche doit avoir une case à cocher à gauche. Les utilisateurs peuvent cocher ou décocher une tâche pour la marquer comme terminée ou non terminée.

Chaque tâche doit avoir un bouton "Supprimer" à droite. Lorsque l'utilisateur clique sur ce bouton, la tâche correspondante doit être supprimée de la liste.

Utilisez le stockage local (`localStorage`) pour sauvegarder les tâches. Lorsque la page est rechargée, les tâches précédemment ajoutées doivent toujours être visibles.

Assurez-vous que le style de la liste et des éléments est agréable, avec des différenciations visuelles pour les tâches terminées et non terminées.

Ajoutez également un bouton "Effacer tout" qui permet de supprimer toutes les tâches de la liste et de réinitialiser le stockage local.

Les utilisateurs doivent pouvoir renommer une tâche en double-cliquant dessus.

Exemple de sortie : Les utilisateurs peuvent ajouter, supprimer, cocher/décocher/renommer des tâches et voir les modifications persistantes même après avoir rechargé la page.