Projet : Créer un Calendrier de l'Avent interactif

Objectif du projet

Créer un calendrier de l'Avent interactif en JavaScript, HTML et CSS, avec des fenêtres à ouvrir chaque jour.

Chaque fenêtre doit afficher un message et un effet visuel (par exemple, un fade-in), et seules les fenêtres suivantes peuvent être ouvertes après la précédente.

De plus, les couleurs des fenêtres doivent être attribuées aléatoirement, et les numéros des jours doivent être mélanges à chaque rechargement de la page.

Niveau 1 : Bases de HTML et CSS (Facile)

Objectif

Construire l'interface visuelle du calendrier de l'Avent.

1. HTML

- Créez une page web contenant un conteneur pour le calendrier. Ce conteneur contiendra 24 fenêtres (une pour chaque jour).
- Chaque fenêtre doit avoir un numéro de 1 a 24 (les numéros seront mélanges a chaque rechargement de la page).
- Ajoutez un titre "Calendrier de l'Avent" au-dessus du calendrier.

2. CSS

- Utilisez du CSS pour styliser le calendrier et ses fenêtres.
- Les fenêtres doivent avoir une forme carrée ou rectangulaire (par exemple, 100px par 100px).
- Utilisez des couleurs de fond différentes pour chaque fenêtre (les couleurs seront attribuées aléatoirement pour chaque fenêtre).

3. Interactivité (facultatif)

- Ajoutez un effet visuel au survol des fenêtres (par exemple, changez la couleur de fond lorsqu'un utilisateur passe la souris dessus).

Niveau 2 : Ajouter de l'interactivité avec JavaScript (Intermédiaire)

Objectif

Rendre les fenêtres interactives, de sorte que l'utilisateur puisse cliquer dessus pour "ouvrir" chaque jour.

1. JavaScript

- Créez un tableau contenant des messages pour chaque jour (par exemple, "Jour 1 :
- Préparez-vous pour un mois festif !").
- Implémenter une fonctionnalité JavaScript qui affiche un message spécifique lorsque l'utilisateur clique sur une fenêtre.
- Lorsque l'utilisateur clique sur une fenêtre, elle doit changer d'apparence pour montrer qu'elle est "ouverte" (par exemple, en changeant la couleur de fond ou en la retournant).

2. CSS pour l'interactivité

- Appliquez des styles CSS pour rendre l'ouverture de la fenêtre visible (par exemple, utilisez transforme: rotâtes(180deg); pour simuler l'ouverture de la fenêtre).

3. Mélanger les numéros des jours

- Mélangez les numéros de jours dans un tableau a chaque rechargement de la page, de sorte qu'ils ne suivent pas un ordre numérique fixe. Ce mélange doit être effectue a chaque chargement de la page.

4. Attribution aléatoire des couleurs

- Assignez une couleur aléatoire a chaque fenêtre lors de son affichage (utilisez un tableau de couleurs prédéfinies et choisissez une couleur aléatoire pour chaque fenêtre).

Niveau 3 : Stocker l'etat des fenêtres et gestion de l'etat (Avancé)

Objectif

Enregistrez l'etat du calendrier dans le local Storage afin que les fenêtres ouvertes persistent après un rechargement de la page.

1. JavaScript (local Storage)

- Lorsque l'utilisateur ouvre une fenêtre, enregistrez cet etat dans le local Storage (par exemple, un tableau contenant les jours ouverts).
- Lors du chargement de la page, vérifiez les données dans le local Storage et affichez les fenêtres déjà ouvertes.
- Assurez-vous que les fenêtres ne peuvent être ouvertes que dans l'ordre, c'est-a-dire que l'utilisateur ne peut pas ouvrir le jour 2 avant le jour 1.

2. CSS pour l'interactivité

- Ajoutez un effet de transition fluide lorsque l'utilisateur ouvre une fenêtre (par exemple, un effet fade-in ou une transition sur la rotation).

3. Mélanger les numéros des jours et les couleurs

- Les numéros des jours doivent être mélanges a chaque rechargement de la page.
- Les couleurs des fenêtres doivent être attribuées aléatoirement pour chaque fenêtre.

Niveau 4 : Ajout de fonctionnalités avancées et personnalisation (Expert)

Objectif

Améliorer l'interactivité et l'expérience utilisateur avec des effets visuels, un bouton d'initialisation, et un calendrier dynamique.

1. JavaScript (effets visuels)

- Implémenter un effet fade-in ou zoom-in lorsque le popup apparait avec le message du jour.
- Ajoutez un bouton pour réinitialiser le calendrier (par exemple, pour vider le local Storage et recommencer depuis le jour 1). Assurez-vous que le calendrier est entièrement réinitialisé, y compris la suppression des fenêtres ouvertes.

2. CSS (animations et transitions)

- Créez des animations CSS pour rendre l'interface plus fluide et attrayante (par exemple, un effet
- de transition fluide sur l'ouverture des fenêtres).
- Ajoutez des styles visuels pour le popup qui affiche les messages (par exemple, un fond légèrement floute derrière le popup).

3. JavaScript (logique de validation)

- Vérifiez que l'utilisateur peut seulement ouvrir les fenêtres dans l'ordre (si le jour 1 n'est pas encore ouvert, le jour 2 doit être désactive, etc.).
- Le bouton de réinitialisation doit remettre toutes les fenêtres a leur etat initial.

4. Bonus (facultatif)

- Créez une version responsive du calendrier afin qu'il s'adapte sur différentes tailles d'écran (mobiles, tablettes, ordinateurs de bureau).

Critères d'évaluation

1. Niveau 1 (HTML et CSS)

- Structure correcte du calendrier.
- Utilisation appropriée des balises HTML et du CSS pour styliser les fenêtres.

2. Niveau 2 (Interactivité avec JavaScript)

- Gestion correcte des évènements de clic.
- Affichage du message du jour de manière claire et dynamique.
- Mélange des numéros et attribution des couleurs aléatoires.

3. Niveau 3 (local Storage et gestion de l'etat)

- Enregistrement et récupération de l'etat des fenêtres ouvertes.
- Vérification de l'ordre des fenêtres ouvertes.
- Couleurs et numéros mélanges aléatoirement a chaque rechargement.

4. Niveau 4 (Fonctionnalités avancées)

- Implémentation d'effets visuels fluides.
- Réinitialisation du calendrier avec un bouton.
- Interface responsive et expérience utilisateur améliorée.
- Conseils et ressources :
- HTML et CSS : Explorez des tutoriels sur la mise en page avec Flexion ou Gri.
- JavaScript : Pratiquez les évènements (addEventListener), la manipulation du DOM et l'utilisation de local Storage.
- CSS : Découvrez les transitions et animations pour améliorer les effets visuels.