Travaux pratiques développement web

JavaScript : Les éléments avancés.

Objectifs :

Le but de cette séance de travaux pratiques est d'approfondir sur les aspects avancés de JavaScript, en se focalisant à ce niveau sur l'interface HTML et du modèle orienté objet des documents.

- Approfondir les connaissances sur la notion des DOM.
- Découvrir les aspects avancés de JS avec les fonctions auto-appelantes.
- Développer des jeux sérieux avec JS et la balise canvas de HTML 5

Consignes :

- ***Vous devez rendre à la fin de ce TP un compte rendu avec des imprimes écran montrant toutes les étapes détaillées et validées.***

Exercice 1 : CONJUGAISON AUTOMATIQUE

1. Définir une variable contenant un verbe, soit du premier groupe, soit du deuxième groupe.
2. Définir un tableau à six cases contenant les pronoms personnels.
3. Définir un tableau contenant les marques de conjugaison au présent de l'indicatif pour les verbes du premier groupe, et un autre pour les terminaisons des verbes du deuxième groupe.
4. Écrire un test qui décide si le verbe défini appartient au premier ou au deuxième groupe.
5. Produire l'affichage des formes conjuguées de ce verbe au présent de l'indicatif, chaque forme étant précédée du ou des pronom(s) personnel(s) adéquat(s).

Exercice 2 : ANIMATION ET INTERACTIVITE

Le but de cet exercice de découvrir quelques aspects interactifs possibles avec JavaScript.

Etape 1 : EFFET SUR LES IMAGES

a- Alternance de deux images

1. Créer une page html5 contenant une image.
2. Programmer en JavaScript une alternance entre cette image et une autre, toutes les deux secondes.
3. Ajouter un bouton et le code JavaScript associé pour arrêter ce défilement.
4. Ajouter un bouton et le code JavaScript associé pour le reprendre.

b- Disparition d'une image et flash

1. Créer une page html5 faisant apparaître une image quelconque.
2. Ajouter un bouton et le code JavaScript associé qui la fait disparaître (utiliser la propriété CSS de l'image *visibility*).
3. Ajouter un bouton qui fait revenir l'image.

c- Variantes :

4. Programmer un effet de **fondu** pour que l'image apparaisse et disparaisse progressivement (utiliser cette fois la propriété CSS de l'image **opacity** et les fonctions JavaScript **setTimeout** ou **setInterval**).
5. Provoquer cette fois un **flash** lorsque la souris passe sur une image : fondu au blanc rapide puis réapparition de l'image.

d- Diaporama

1. Placer des noms d'images dans un tableau JavaScript.
2. Programmer l'affichage une par une de ces images dirigées par deux boutons, un précédent, un suivant.

e- Image aléatoire

3. Placer des noms d'images dans un tableau JavaScript.
4. Programmer l'affichage aléatoire de l'une de ces images, toutes les deux secondes.

f- Déplacement d'une image

5. Afficher un ensemble de photos dans une page html5.
6. Ajouter le code JavaScript qui fait circuler l'une des photos parmi les autres en fonction d'actions de l'utilisateur, au clavier ou à la souris.

Etape 2 : VARIATIONS AUTOUR DE TEXTES CHANGEANTS

a- Sommaire automatique

- Récupérer les titres de niveau h2 et les faire apparaître comme une liste à puces, à un endroit approprié.
- Faire en sorte que chaque item produit soit cliquable et amène à la section associée.

b- Dictionnaire visuel

On étiquette certains mots du texte, la définition apparaît,

- D'abord dans un popup au clic sur le mot,
- Puis dans un paragraphe dédié au passage de la souris sur mot.

Dictionnaire visuel amélioré : On peut faire défiler les mots au clavier (flèches gauche et droite).

Texte selon image

1. Créer une page html5 affichant plusieurs photos.
2. Définir un tableau JavaScript qui contienne un texte descriptif pour chacune des photos affichées.
3. Ajouter le code JavaScript qui, quand la souris passe sur une image, fait apparaître le texte associé.
4. Ajouter le code JavaScript qui fasse disparaître le texte lorsque la souris quitte l'image.
5. Programmer un effet de déroulement progressif pour l'affichage des textes.

Variante :

6. Créer une page html5 avec un menu contenant plusieurs items.
7. Programmer en JavaScript l'affichage d'un texte explicatif lorsque la souris passe sur l'un des items.

Etape 3 : SLIDE PUZZLE OU TAQUIN

Premières étapes :

1. Créer l'apparence du jeu : trois cases par trois cases, chaque case étant un bouton HTML.
2. Associer des styles CSS à ces boutons.
3. Écrire le code JavaScript qui, lorsque l'on clique sur un bouton, l'échange avec la case vide.

Étapes suivantes :

1. N'autoriser que les déplacements valides.
4. Compter et afficher le nombre de déplacements effectués.

Travail restant :

1. Mélanger aléatoirement les cases au début du jeu.
2. Détecter la fin de partie.
3. Adapter le jeu à une version avec les morceaux d'une photo, plutôt qu'avec des boutons.

Projet : DEVELOPPEMENT D'UN JEU

Consigne : Vous devez choisir de travailler sur un de ces deux projets et préparer un compte rendu détaillé de votre rendu, avec des imprimes écran annotés et commentés.

PROJET 1 : JEU « MEMORY »

Nous voulons implémenter un jeu de « mémoire ». Au début du jeu les cartes sont faces cachées, l'interface doit permettre d'en retourner deux. Si les deux cartes sont identiques elles restent visibles, sinon elles disparaissent après un court laps de temps. Le jeu se termine quand toutes les paires ont été découvertes.

1. Récupérer huit images de même taille et coder en HTML/CSS la disposition des seize cartes.
2. Programmer en JavaScript le mélange des cartes, par exemple en répétant une centaine de fois le processus suivant : tirer deux cartes au hasard et les échanger. Ce mélange doit avoir lieu au chargement de la page.
3. Utiliser la CSS pour rendre les cartes invisibles. Programmer en JavaScript le retournement d'une carte quand on clique sur elle. *optionnel : Produire un joli effet visuel quand la carte apparaît.*
4. Permettre de cliquer sur deux cartes cachées. Tester si les deux images dévoilées sont les mêmes. Si les images sont différentes, les faire disparaître après une seconde d'affichage.
5. Interdire le clic sur une image déjà apparente. Interdire le clic sur une troisième carte pendant que les deux cartes cliquées sont apparentes.
6. Compter le nombre de paires découvertes, compter les nombres de clics utilisés. Compléter en produisant des messages en cours de partie (découverte d'une paire, nombre de clics utilisés, message de victoire, etc.).
7. Ajouter deux boutons : l'un qui permet de commencer une nouvelle partie, l'autre qui donne la solution en découvrant toutes les cartes.

PROJET 2 « JEU DE LE PUZZLE DE YANN »

1. Choisir deux photos d'un même objet (portraits d'une personne à des âges différents, même paysage à des saisons différentes, etc.). L'une des versions est désignée comme cible, c'est-à-dire que c'est elle qui devra finalement s'afficher. Découper chaque photo en 9 morceaux. Choisir les noms des images pour permettre de connaitre aisément l'emplacement d'un morceau et la version dont il provient.
2. Préparer l'apparence en HTML/CSS : neuf images disposées en 3x3, chacune avec un identifiant.
3. Écrire une fonction JavaScript qui, à partir d'un emplacement, choisit aléatoirement et affiche un morceau destiné à cet emplacement.
4. Faire en sorte que la fonction précédente s'exécute 50 fois, une fois chaque quart de seconde, à chaque fois sur un emplacement choisi aléatoirement.
5. Les 50 itérations passées, on continue les tirages aléatoires mais uniquement pour les morceaux qui ne sont pas affichés dans la bonne version.

6. Détecter quand l'image cible est complètement affichée. Signaler que le but est atteint et cesser alors toute modification.

Améliorations possibles :

7. Généraliser votre code pour gérer plus de deux versions de l'image.
8. Généraliser votre code pour utiliser des découpages autres que 3x3.
9. Provoquer un effet de fondu lors des changements d'images.
10. Ajouter des boutons pour stopper ou lancer une nouvelle animation, pour contrôler la cible ou l'effet de fondu, etc.