

IT 238– Technologie de Web scripting interactif - TRAVAUX PRATIQUES 6_7_8

Etudiant : DANDJA Kandifayi Faïz Basile

Filière : DWM

Manuel de présentation détaillé des éléments et interfaces réalisés - TP6

Introduction

Ce manuel a pour objectif de présenter de manière détaillée les différents éléments et interfaces réalisés dans le cadre du TP6. Il s'agit d'un mini-dashboard des signalements voirie, développé avec HTML, CSS et JavaScript, utilisant également la bibliothèque jQuery pour faciliter la manipulation du DOM et la gestion des événements.

1. Analyse du fichier HTML (index.html)

Le fichier HTML constitue la structure de base de l'application web.

- La déclaration `<!DOCTYPE html>` indique que le document est un fichier HTML5.
- La balise `<html lang="fr">` définit la langue principale du document comme étant le français.
- Dans la section `<head>`, on trouve :
 - `<meta charset="UTF-8" />` qui spécifie l'encodage des caractères en UTF-8, garantissant la prise en charge des caractères spéciaux français.

- `<meta name="viewport" content="width=device-width, initial-scale=1" />` qui assure une bonne adaptation de la page sur les appareils mobiles en réglant la largeur de la fenêtre d'affichage.

- `<title>Mini-dashboard des signalements voirie</title>` qui définit le titre affiché dans l'onglet du navigateur.

- `<link rel="stylesheet" href="style.css" />` qui lie la feuille de style CSS externe pour la mise en forme.

- `<script src="jquery.min.js"></script>` qui inclut la bibliothèque jQuery, utilisée pour simplifier la manipulation du DOM et la gestion des événements.

- `<script src="script.js" defer></script>` qui inclut le fichier JavaScript principal, chargé de la logique dynamique, avec l'attribut `defer` pour différer son exécution après le chargement du HTML.

- Dans la section `<body>` :

- `<header>` contient un titre principal `<h1>` affichant "Mini-dashboard des signalements voirie".

- La section `<section class="filters">` contient deux filtres :

- Un label et un menu déroulant `<select id="filter-type">` pour filtrer par type de signalement.

- Un label et un menu déroulant `<select id="filter-quartier">` pour filtrer par quartier.

- Chaque menu contient une option par défaut "Tous" avec la valeur "all".

- La section `<section class="report-list">` contient un tableau `<table id="signalements-table">` :

- L'en-tête `<thead>` définit les colonnes Description, Quartier, Type et Statut.

- Le corps `<tbody>` est vide au départ, les signalements y seront insérés dynamiquement via JavaScript.

2. Analyse du fichier JavaScript (script.js)

Ce fichier gère la logique dynamique et interactive du mini-dashboard.

- Le code est encapsulé dans `$(document).ready(function() { ... });` pour s'assurer que le DOM est entièrement chargé avant d'exécuter le script.

- Un tableau constant signalements contient des objets représentant les signalements, chacun avec les propriétés description, quartier, type et statut.
- La fonction remplirFiltres() :
 - Extrait les types et quartiers uniques des signalements à l'aide de Set.
 - Pour chaque type et quartier, elle ajoute une option correspondante dans les menus déroulants des filtres.
- La fonction afficherSignalements() :
 - Récupère les valeurs sélectionnées dans les filtres.
 - Filtre le tableau signalements selon ces valeurs, en incluant tous si la valeur est "all".
 - Vide le corps du tableau avec une animation de fondu (fadeOut), puis insère les lignes correspondant aux signalements filtrés.
 - Si aucun signalement ne correspond, affiche un message indiquant "Aucun signalement trouvé."
 - Fait réapparaître le tableau avec une animation de fondu (fadeIn).
- Les événements de changement sur les filtres déclenchent l'appel à afficherSignalements() pour mettre à jour l'affichage.

3. Analyse du fichier CSS (style.css)

Ce fichier définit le style visuel de l'application.

- Le corps (body) utilise une police sans-serif moderne, un fond en dégradé sombre et une couleur de texte claire, avec des transitions douces pour les changements de couleur.
- Le header est centré avec un espacement en bas.
- Le titre principal (h1) est en couleur dorée (#d4af37) avec une ombre portée pour le mettre en valeur.
- La section des filtres utilise flexbox pour disposer les éléments horizontalement avec un espacement et un retour à la ligne si nécessaire.
- Les labels des filtres sont en gras et dorés avec une ombre portée.
- Les menus déroulants (select) ont un style personnalisé avec bordure dorée, fond noir, texte clair, ombre portée, et effets au survol et au focus.

- La section report-list a une largeur maximale, un fond en dégradé sombre, des coins arrondis et une ombre portée dorée.
- Le tableau utilise un espacement entre les lignes, des cellules avec fond transparent doré, ombres portées et coins arrondis.
- L'en-tête du tableau est fixe (sticky) en haut avec fond doré et texte noir.
- Les lignes du tableau ont une transition pour l'agrandissement et l'ombre au survol, améliorant l'interactivité.

4. Utilisation de jQuery

La bibliothèque jQuery est utilisée pour simplifier la sélection des éléments DOM, la manipulation du contenu, la gestion des événements et les animations (fadeIn, fadeOut). Elle permet d'écrire un code plus concis et compatible avec plusieurs navigateurs.

Conclusion

Ce TP6 m'a permis de concevoir une interface web complète et fonctionnelle, en combinant la structure HTML, le style CSS et la logique JavaScript. Chaque partie du code a été pensée pour offrir une expérience utilisateur fluide, intuitive et agréable.