

TP JavaScript : L'Interactivité, C'est Facile !

Durée : Environ 3 heures (ou plus si vous explorez !)

Introduction : Rendre une Page Web Vivante

Ce TP va vous montrer comment faire réagir une page web. On va créer des petites choses qui bougent, changent de couleur ou réagissent à vos clics et à votre clavier.

Les 3 Langages du Web

Une page web, c'est 3 choses :

1. HTML : Le Contenu. C'est le texte, les images, les boutons. Ce qu'on voit.
2. CSS : Le Style. C'est la couleur, la taille, la position. Comment ça rend joli.
3. JavaScript : L'Action. C'est ce qui fait que la page bouge, réagit, et fait des choses intelligentes.

Le DOM : La Carte de la Page

JavaScript utilise le DOM (Document Object Model) pour trouver et changer les éléments HTML. Le DOM, c'est comme une liste de tous les éléments de la page.

Comment trouver un élément ?

- `document.getElementById('son_id')` : On cherche un élément par son nom unique (id).

Les Événements : Quand l'Utilisateur Agit

Un événement, c'est quand l'utilisateur fait quelque chose : clique, tape, bouge la souris.

Comment réagir à un événement ?

- `element.addEventListener('nom_evenement', fonction_a_faire)` : On dit à JavaScript d'écouter un événement sur un élément et de lancer une fonction quand ça arrive.

localStorage : La Mémoire du Navigateur

localStorage permet à la page de se souvenir de choses même si vous la fermez. C'est pratique pour sauvegarder des réglages ou des listes.

Le TP : Nos Petites Applications

Nous allons faire 5 petites applications sur une seule page.

Fichiers :

- `index.html` : La structure.

- style.css : Le style (déjà fait, très simple).
 - script.js : Toute la logique JavaScript.
-

Partie 1 : Le Compteur

Un nombre qui augmente, diminue, et se remet à zéro. Il est maintenant plus joli !

Exercice 1.1 : La Structure HTML (Déjà fournie)

Ouvrez index.html. Vous verrez une section pour le compteur avec :

- Un texte () pour montrer le nombre.
- Trois boutons (`<button id="moins">`, `<button id="plus">`, `<button id="reset">`).

Tâche : Regardez ces éléments. Comprenez comment leurs id vont nous aider à les trouver avec JavaScript.

Exercice 1.2 : Le Style CSS (Mis à jour)

Ouvrez style.css. Il contient les règles pour rendre le compteur joli et centré, avec un fond, une bordure et des coins arrondis. Il utilise aussi des "variables CSS" (comme `--couleur-fond-principal`) pour que le thème sombre puisse changer facilement les couleurs.

Tâche : Observez comment les couleurs et les tailles sont définies. Notez les variables CSS qui commencent par `--` et les styles ajoutés pour rendre le compteur plus beau.

Exercice 1.3 : La Logique JavaScript (À comprendre et tester)

Ouvrez script.js. C'est là que la magie opère. Cherchez la section `// Partie 1: Compteur`.

Tâches :

1. Variable `c` : C'est le nombre que notre compteur garde en mémoire.
2. `getElementById` : Voyez comment on "attrape" l'affichage du compteur et les boutons.
3. `majC()` : Met à jour l'affichage du nombre.
4. `addEventListener('click', ...)` : Chaque bouton change `c` et appelle `majC()`.
5. `msg()` : Affiche un petit message.

Test : Ouvrez index.html dans votre navigateur. Cliquez sur les boutons du compteur. Voyez le nombre et les messages changer. Le compteur devrait maintenant avoir un aspect plus soigné.

Partie 2 : Formulaire

Un champ où vous tapez votre nom, et on vérifie s'il est assez long.

HTML (index.html) :

- Regardez la section "Formulaire". Il y a un champ de texte (`<input type="text">`) avec l'ID (id="nom") et un bouton.

JavaScript (script.js) :

- Cherchez // Partie 2: Formulaire.
- valNom() : Vérifie si le nom a au moins 3 lettres. Change le style du champ si erreur.
- formU.addEventListener('submit', ...) : Quand on valide le formulaire.
 - e.preventDefault() : Empêche la page de se recharger.
- nomIn.addEventListener('input', ...) : Vérifie le nom en direct quand vous tapez.

Test : Tapez un nom court, puis un nom long. Voyez l'erreur et le succès.

Partie 3 : Tâches

Une liste où vous pouvez ajouter, marquer comme fait, et supprimer des tâches. Elles sont sauvegardées !

HTML (index.html) :

- Regardez la section "Tâches". Il y a un champ de texte (`<input type="text">`) avec l'ID (id="tache-txt"), un bouton, et une liste vide ().

JavaScript (script.js) :

- Cherchez // Partie 3: Tâches.
- loadT() / saveT() : Fonctions pour localStorage.
- creerT(tache) : Crée un élément de tâche () avec ses boutons "Terminer" et "Supprimer".
- addT() : Ajoute une tâche à la liste et la sauvegarde.
- Les boutons "Terminer" et "Supprimer" sur chaque tâche mettent à jour son état et la sauvegarde.

Test : Ajoutez des tâches, marquez-les, supprimez-les. Rechargez la page : vos tâches sont toujours là !

Partie 4 : Thème

Changez la page en mode clair ou sombre.

HTML (index.html) :

- Regardez la section "Thème". Il y a un bouton (id="toggle-theme").

JavaScript (script.js) :

- Cherchez // Partie 4: Thème.
- applyTh() : Applique le thème sauvegardé au démarrage.

- `btnTheme.addEventListener('click', ...)` : Quand on clique, on ajoute/enlève la classe `theme-sombre` au `body` et on sauvegarde le choix.

Test : Cliquez sur le bouton. Voyez la page changer de thème. Rechargez : le thème reste le même.

Partie 5 : Glisser-Déposer

Faites bouger un carré avec votre souris. Le problème de non-affichage a été corrigé !

HTML (`index.html`) :

- Regardez la section "Glisser-Déposer". Il y a un `div` avec l'ID correct : `element-deplacable`.

JavaScript (`script.js`) :

- Cherchez // Partie 5: Glisser-Déposer.
- Correction importante : L'ID de l'élément déplaçable dans le HTML (`element-deplacable`) correspond maintenant à l'ID utilisé dans le JavaScript (`const dragEl = document.getElementById('element-deplacable');`).
- `dragEl.addEventListener('mousedown', ...)` : Quand vous cliquez sur le carré, on commence à le "glisser".
- `document.addEventListener('mousemove', ...)` : Quand la souris bouge, le carré la suit.
- `document.addEventListener('mouseup', ...)` : Quand vous relâchez le clic, le glisser s'arrête.
- `initDrag()` : Positionne le carré en haut à gauche au démarrage.

Test : Cliquez sur le carré et essayez de le déplacer. Il devrait maintenant suivre votre souris et rester dans sa zone.

Conclusion

Vous avez maintenant vu how JavaScript rend les pages web interactives. C'est la base de tout ce que vous voyez sur le web !