TP JavaScript n°2 - DWWM - 10.04.2025

Mise en pratique des connaissances et compétences acquises en JavaScript

Mots clé : javascript, variables, structures de contrôle, boucles, objets littéraux, tableaux, fonctions, dom, événements, addEventListener, preventDefault()

groupe DWWM

- M.TOULOUSE -

TP - JAVASCRIPT - (TP n°2) -DWWM - Jeudi 10.04.2025

Codage d'une interface de gestion de cours de langages de développement web

1. 📖 Préambule

Ce TP est avant tout une opportunité pour vous de vous entraîner et de vous familiariser avec le JavaScript de manière progressive.
Ne soyez pas intimidé par les consignes : chaque étape a été pensée pour vous accompagner et vous permettre d'apprendre à votre rythme.
Rappelez-vous que vous êtes libre de personnaliser l'interface selon vos goûts et votre créativité !

Partez d'une interface très basique et améliorez-la au fur et à mesure de votre progression.
Vous pouvez également ajouter des fonctionnalités supplémentaires si vous le souhaitez, mais ne vous laissez pas déborder par la quantité de travail à réaliser.

Même si vous ne parvenez pas à réaliser toutes les fonctionnalités, l'important est de comprendre et de progresser.
N'hésitez pas à échanger avec vos camarades et à explorer les différentes possibilités offertes par le JavaScript.
Le plus important est de vous confronter aux concepts, d'expérimenter et de prendre confiance en vos capacités.
Vous avez toutes les clés en main pour réussir, alors amusez-vous à développer votre propre gestionnaire de cours !

---

2. 📨 Envoi des TP & conseils

Ce TP doit être rendu individuellement cependant l'entraide est la bienvenue !

Pensez à bien indenter et commenter tous vos codes !

- Merci de me retourner le TP avant 17h00 au courriel suivant : `frontend@netc.fr`
- Le dossier du TP devra être zippé.
- Le TP devra être compressé en respectant STRICTEMENT le format suivant qui utilise un underscore (_) entre chaque token écrit sans accents dans le nom de fichier : * `prenom_nom_tp2_javascript_100425.zip`

  Exemple :
  Je m'appelle Frédéric DUBOIS, mon fichier sera alors nommé : frederic_dubois_tp1_javascript_100425.zip
  Si vous ne parvenez pas à envoyer votre dossier .zip (gmail pose souvent des problèmes), utilisez l'excellent service français SMASH, gratuit, sans inscription ni abonnement : https://fr.fromsmash.com

> Prenez le temps d'analyser les tâches demandées avant de coder !
>
> - Le codage n'est pas uniquement une affaire de technologie mais de réflexion.
> - Il est suggéré de ne pas utiliser l'ordinateur durant la première heure du TP.
> - Passer directement à la phase de codage sur écran est souvent très sournois et contreproductif.
> - Travaillez avec du papier et un stylo afin d'envisagez votre stratégie de codage face à la problématique.
> - Faites des schémas, des croquis, envisagez des scénarios de codage, annotez vos idées
> - Formulez délicatement et consciencieusement sous forme de phrase en français vos étapes de codage

---

3. 📔 Informations Générales

- Titre du TP : Gestionnaire de Cours
- Date : Jeudi 10 avril 2025
- Durée : 7 heures
- Auteur : M. TOULOUSE
- Sujet : Développement d'une application web pour gérer des cours.
- Rendu : Avant 17h00, par email à `frontend@netc.fr` (format de fichier strictement respecté).

---

4. 🎯 Objectifs Pédagogiques

À la fin de ce TP, vous serez capable de :

1. Manipuler des tableaux et des objets pour gérer des données dynamiques.
2. Structurer un programme en JavaScript avec des fonctions claires et réutilisables.
3. Utiliser des méthodes de tableaux comme `findIndex`, `filter`, `icludes`, `sort`.
4. Appliquer des concepts de DOM manipulation pour créer une interface interactive.

---

5. 📜 Contexte

Vous devez développer un programme pour gérer des cours.

Chaque cours inclut :

- Un langage (HTML, CSS, Python, PHP).
- Un titre.
- Une description.
- Un niveau (Débutant, Intermédiaire, Avancé, Expert).

L'application doit permettre :

1. D'ajouter un cours.
2. De modifier un cours existant.
3. De supprimer un cours.
4. De rechercher un cours par titre ou description.

5. De trier les cours par titre ou niveau.

---

6. 📊 Données de départ

6.1. Liste des émojis

- Niveaux :
  - 0 : Débutant 🌱
  - 1 : Intermédiaire 🌿
  - 2 : Avancé 🌳
  - 3 : Expert 🎓
- Langages :
  - html : 🌐
  - css : 🎨
  - python : 🐍
  - php : 🐘

6.2. Code de base

> Il vous est fourni un code de base nommé `starter.html` qui contient la structure HTML de l'interface.
>
> A reste à vous de l'enrichir et de le personnaliser selon vos besoins en respectant l'arborescence du projet et les consignes données dans ce TP.

---

7. Arborescence du projet

> Vous devez créer un dossier nommé `prenom_nom_tp2_js` (vous remplacerez `prenom` et `nom` par votre prénom et nom respectifs) contenant les fichiers suivants :
>
> Le fichier `html` contiendra la structure de votre interface.
>
> Le fichier `css` contiendra le design de votre interface.
>
> Le fichier `js` contiendra la logique (code javascript) de votre interface.

```
prenom_nom_tp2_js
    ├── courses.html
    ├── courses.css
    └── courses.js
```

---

8. 📦 Organisation du TP

8.1. Partie 1 : Structure de base et style

- Utiliser la structure HTML fournie et/ou l'enrichir selon vos besoins
- Comprendre et utiliser les styles CSS existants
- Se familiariser avec l'organisation du code

8.2. Partie 2 : Affichage des cours

- Créer la fonction d'affichage des cours
- Utiliser les classes CSS pour le style des cartes
- Afficher les émojis correspondants aux langages et niveaux

8.3. Partie 3 : Ajout de cours

- Gérer le formulaire d'ajout
- Valider les données saisies
- Ajouter le nouveau cours à la liste

8.4. Partie 4 : Suppression de cours

- Implémenter la fonction de suppression
- Gérer le clic sur le bouton supprimer
- Mettre à jour l'affichage

---

9. 🚀 Prolongement & Enrichissement

Si vous terminez avant le temps imparti, vous pouvez ajouter les fonctionnalités suivantes :

1. Pagination : Afficher un nombre limité de cours par page.
2. Filtres avancés : Ajouter des filtres pour afficher uniquement les cours d'un certain niveau ou langage.
3. Statistiques : Afficher le nombre total de cours, le nombre de cours par niveau, etc.
4. Animations : Ajouter des animations CSS pour les cartes (ex. hover, transitions).

---

10. ⚙️ Critères d'Évaluation du TP

> Votre travail sera évalué sur les critères suivants :

1. Fonctionnalité : Chaque partie du TP fonctionne-t-elle correctement ?
2. Lisibilité : Le code est-il clair, bien structuré, indenté et commenté ?
3. Respect des consignes : Les consignes et la mise en forme sont-elles respectées ?
4. Utilisation des concepts : Les boucles, fonctions et manipulations DOM sont-elles bien utilisées ?

5. Interactivité : L'interface est-elle intuitive, agréable à utiliser, facile à appréhender et réactive ?
6. Bonus : Des fonctionnalités bonus ont-elles été intégrées, et si oui, sont-elles fonctionnelles ?

> Bien vérifier que votre production comporte les éléments suivants :

- 📄 Les fichiers `courses.html`, `courses.css` et `courses.js` sont bien présents.
- 🌈 Utilisation des variables css
- 📱 Le code css utilise la philosophie mobile-first .
- ✅ Le tableau `courses` est correctement affiché dans la section `.course-list`.
- 🎨 Les cartes sont stylisées avec CSS BEM.
- 📝 Chaque carte affiche le titre, la description, le niveau et le langage.
- 🗑️ Les boutons "Supprimer" sont présents sous chaque carte.
- 🔄 Les cours s'affichent dynamiquement via JavaScript.

---

11. Grille d'évaluation approximative

Critères	Description	Points
1. Fonctionnalité		**/10**
- Ajout de cours	Le formulaire permet d'ajouter un cours avec toutes les informations requises.	4
- Suppression de cours	Les cours peuvent être supprimés via un bouton dédié.	2
- Affichage dynamique	Les cours s'affichent dynamiquement sous forme de cartes.	4
2. Interface utilisateur		**/4**
- Design et esthétique	L'interface est attrayante, bien organisée et agréable à utiliser.	2
- Responsive design	L'interface s'adapte correctement aux différentes tailles d'écran.	2
3. Code et organisation		**/4**
- Lisibilité du code	Le code est bien structuré, indenté et facile à lire.	2
- Utilisation de BEM	Les classes CSS suivent la méthodologie BEM pour une meilleure organisation.	2
4. Méthodologie		**/2**
- Commentaires	Le code est bien commenté pour expliquer les parties importantes.	1
- Respect des consignes	Le projet respecte les consignes données dans le sujet.	1
5. Bonus (pris en compte si réussi)		**/2**
- Fonctionnalités supplémentaires	Des fonctionnalités non demandées mais pertinentes ont été ajoutées.	1
- Effets et animations	Des animations ou transitions CSS/JS ont été ajoutées pour améliorer l'UX.	1
TOTAL		**/20**

---

12. 💡 Conseils

1. Planifiez avant de coder :
   - Lisez attentivement l'énoncé.
   - Faites des schémas sur papier.
   - Découpez le problème en sous-tâches.
2. Codez méthodiquement :
   - Commencez par les fonctionnalités de base.
   - Testez chaque fonction avant de passer à la suivante.
   - Utilisez la console pour déboguer.
3. Organisez votre code :
   - Commentez les sections importantes.
   - Indentez proprement.
   - Nommez clairement vos variables et fonctions.
4. Gérez votre temps :
   - Concentrez-vous d'abord sur les fonctionnalités essentielles.
   - Gardez les bonus pour la fin.
   - Prévoyez du temps pour les tests.

---

13. 👋 Le mot de la fin

Ce TP est une excellente opportunité pour mettre en pratique vos connaissances en JavaScript et DOM.
Prenez le temps de bien structurer votre code et amusez-vous à personnaliser l'interface !

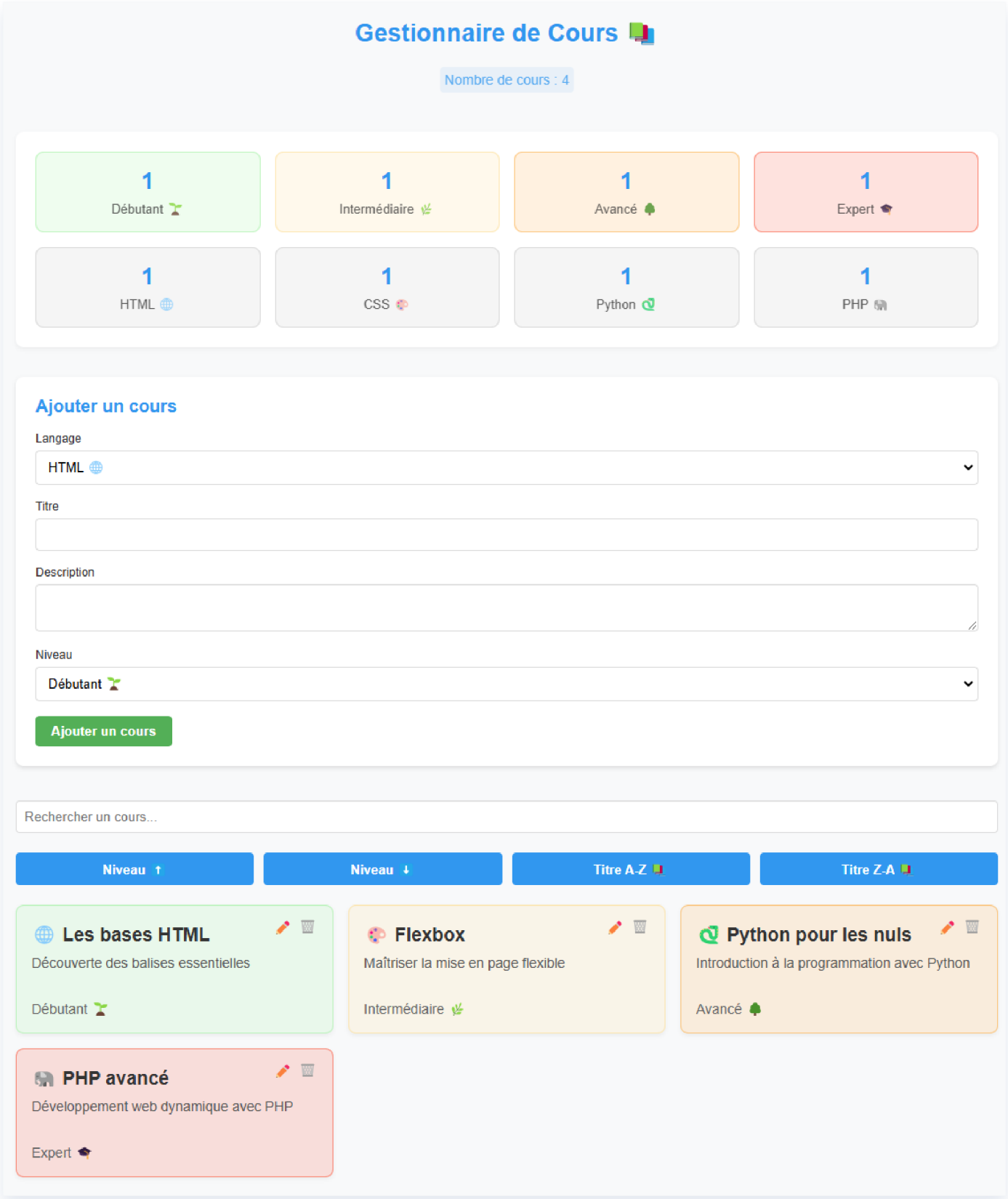
Bon courage et bon codage ! 😊

M.TOULOUSE

---

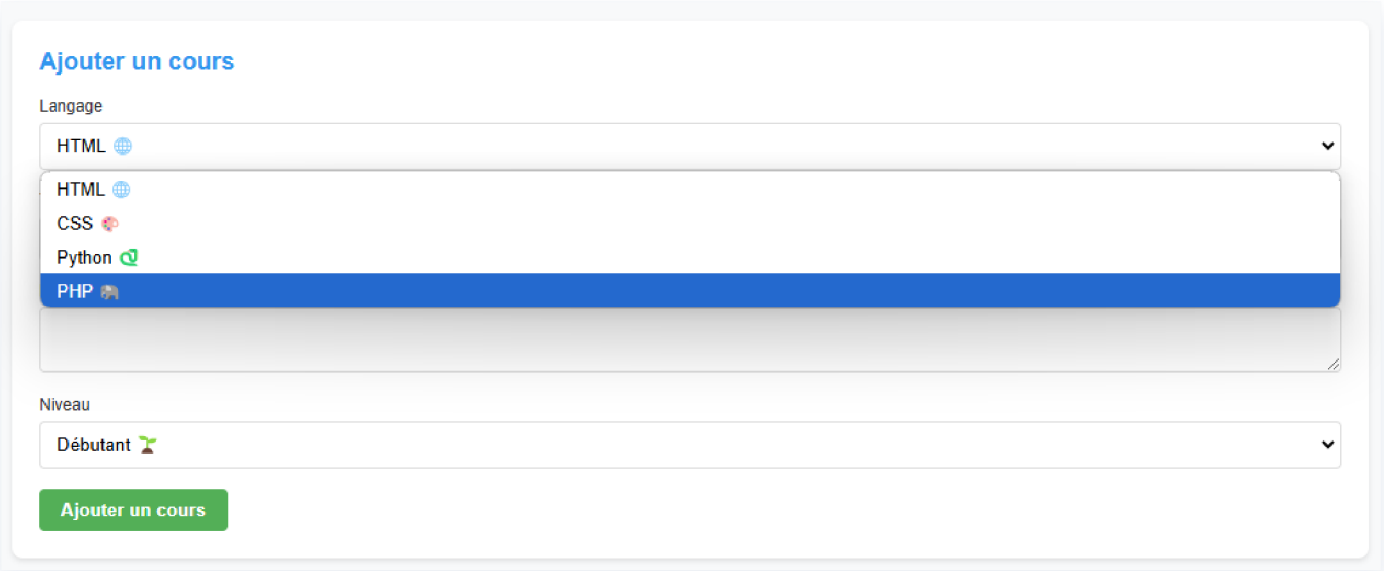
14. ANNEXES

14.1. Exemple d'interface



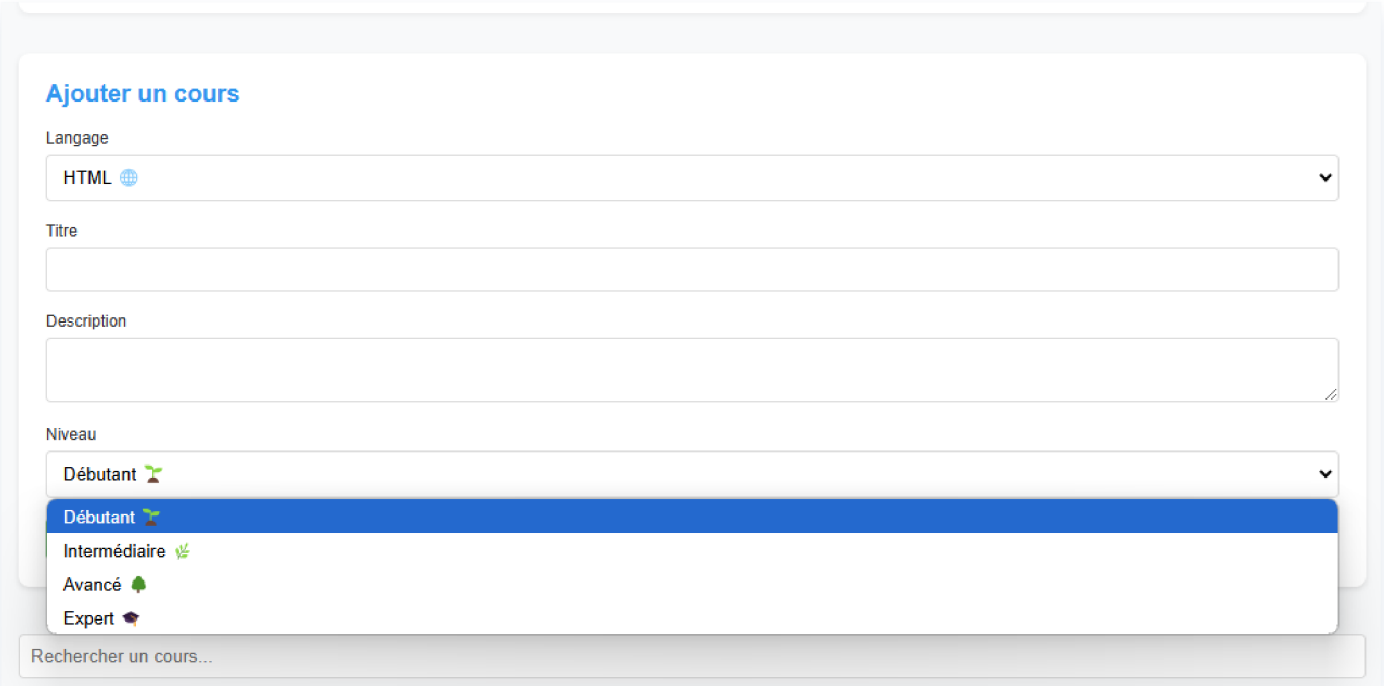
14.2. Sélection du langage du cours

Voici une proposition pour la sélection du langage.
Vous pouvez cependant choisir d'avoir recours à un champ de texte classique si cela vous convient mieux.



14.3. Sélection du niveau de difficulté du cours

Je vous propose un exemple de sélection des niveaux de difficulté.
Vous êtes libre de choisir la méthode de sélection qui vous convient le mieux (champ texte, liste déroulante, bouton radio, etc.).



14.4. Liste des cours

> Par soucis d'esthétisme, les cours (codé sous forme de cards flexbox) seront disposés sous forme de grilles css auto responsives.

14.5. Affichage de statistiques relatives aux cours

> Enfin, vous trouverez un modèle pour déployer des statistiques relatives aux cours.
> Vous pouvez choisir d'afficher ces statistiques sous forme de texte, de graphiques...

15. Fin du sujet de TP ... 😵 😅 😪 !!!