

ÉPREUVE FRONT-END — Interfaces adaptatives et interactives

Développement sans framework

Durée : 4 heures (5 heures 20 minutes avec tiers temps)

Mode : Individuel

Technologies autorisées : HTML5, CSS3, JavaScript (ES6+)

Technologies interdites : Frameworks et librairies externes

Contexte

Une structure pédagogique souhaite disposer d'une application web simple permettant de consulter et gérer des ressources pédagogiques (documents, articles, cours ou liens). L'application doit être accessible, responsive et interactive, et développée exclusivement en HTML, CSS et JavaScript natif.

Objectifs

- 1 Afficher une liste de ressources pédagogiques.
- 2 Ajouter de nouvelles ressources via un formulaire.
- 3 Interagir avec les données sans recharger la page.
- 4 Adapter l'interface aux différents supports.

Données & API simulée

Les données doivent être chargées de manière asynchrone depuis une API simulée fournie sous la forme d'un fichier JSON local nommé `resources.json`. Ce fichier représente une API fictive et sert uniquement à charger les données initiales. Le fichier JSON ne doit pas être modifié dynamiquement.

Fonctionnalités attendues

1. Structure et accessibilité

- 1 Structure sémantique complète.
- 2 Titres hiérarchisés.
- 3 Formulaire accessible.

2. Interface utilisateur

- 1 Design cohérent et lisible.
- 2 Palette de couleurs sobre.
- 3 Footer positionné en bas de page.

3. Responsive design

- 1 Interface utilisable sur mobile, tablette et desktop.
- 2 Navigation adaptée aux petits écrans.

4. Interactions côté client

- 1 Affichage dynamique des ressources.
- 2 Ajout d'une ressource via un formulaire.
- 3 Validation simple des champs.

5. Communication asynchrone

- 1 Chargement initial des ressources via une requête asynchrone.
- 2 Gestion basique des erreurs.

Livrable attendu

Un dossier contenant l'ensemble des fichiers nécessaires au fonctionnement de l'application (HTML, CSS, JavaScript et fichier JSON). Le code doit être structuré, lisible et ne présenter aucune erreur bloquante.

Critères d'évaluation

- 1 Qualité de la structure HTML.
- 2 Cohérence du design et de l'UX.
- 3 Adaptation responsive.
- 4 Fonctionnalités JavaScript.
- 5 Utilisation correcte des données asynchrones.

La priorité est donnée à la cohérence, à la lisibilité du code et à la stabilité de l'application.