Examen 2

Design Web – Automne 2025

Consignes

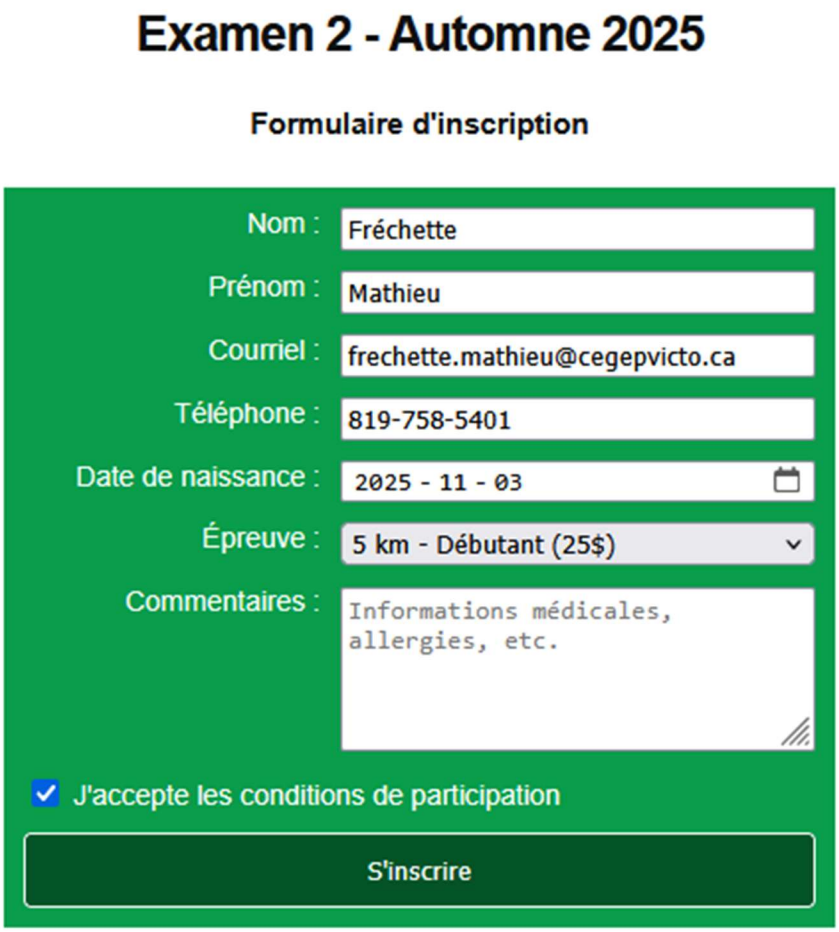
- Vous avez droit à toutes vos notes et Internet, mais pas aux outils de génération de code (Copilot, Chat GPT et autres.).
- Aucune communication n'est permise (messagerie, courriel, etc.).
- Un élève pris en flagrant délit de plagiat se verra sanctionné conformément à la procédure de gestion des cas de plagiat, de tricherie ou de fraude.
- La durée de l'examen est de 1h50.
- Une fois l'examen terminé, vous devez me remettre **tout le dossier de votre examen** via GitHub Classroom.
- Il est de votre responsabilité de vous assurer que tous vos fichiers sont inclus dans la remise.
- **Aucun fichier ne sera accepté** après avoir quitté le lieu de l'examen.

Installation

- Clonez le dépôt Github Classroom disponible à l'adresse que votre enseignant va vous donner.
- **Une fois votre travail terminé**, ajoutez vos fichiers (*git add --all*) et faites un commit de tous vos fichiers modifiés avec le message "**Version finale**" (*git commit -m "Version finale"*). Envoyez ensuite le tout sur Github en faisant un "push" (*git push*).
- Non évalué, mais doit être fait **correctement** AVANT de quitter le lieu de l'examen.

Question 1 – Formulaire d'inscription

Dans le fichier **question1.html** reproduisez le formulaire suivant :



Des règles css sont déjà présentent dans le fichier styles.css pour aligner le formulaire. La disposition ne sera pas évaluée.

Faites le nécessaire pour que chaque champ du formulaire puisse répondre aux exigences suivantes :

- **Nom et prénom :**
  - Champ texte et obligatoire
  - Doit être d'une longueur comprise entre 2 et 30 caractères
- **Courriel :**
  - Champ obligatoire
  - Doit être une adresse courriel valide

- **Téléphone :**
  - Champ obligatoire
  - Doit être testé avec l'expression régulière suivante :
    - **[0-9]{3}-[0-9]{3}-[0-9]{4}**
- **Date de naissance :**
  - Champ obligatoire
  - Ne doit être supérieur à la date d'aujourd'hui (vous pouvez l'écrire en clair)
- **Épreuve :**
  - Champ obligatoire
  - Doit être représenté sous forme de liste avec 5 options :
    - Option 1: « --Sélectionnez une épreuve-- », valeur vide
    - Option 2: « 5 km - Débutant (25$) », valeur égale à 5km
    - Option 3: « 10 km - Intermédiaire (35$) », valeur égale à 10km
    - Option 4: « 21 km - Semi-marathon (50$) », valeur égale à 21km
    - Option 5: « 42 km - Marathon (75$) », valeur égale à 42km
- **Commentaires:**
  - Pas obligatoire
  - Zone de texte de 30 colonnes et 5 rangées.
- **Case à cocher "J'accepte les conditions de participation" :**
  - Case obligatoire

Précisions

- Le formulaire doit envoyer les données par la méthode **POST**.
- Les données sont envoyées à l'adresse suivante :

https://design-web.onrender.com/submit

Question 2 – Affichage des résultats

Dans le fichier question2.html vous avez déjà des éléments en place pour permettre d'afficher les résultats d'un coureur selon son numéro de participant.

Dans le fichier script.js vous avez un tableau d'objets (résultats) qui contient les résultats de 70 participants.

Vous devez programmer une fonctionnalité de recherche et d'affichage du résultat selon un numéro de participant en respectant les consignes suivantes.

- Vous ne pouvez pas modifier le fichier html. Tout le traitement doit uniquement se faire dans le fichier script.js.
- Le tableau résultats ne peut pas être modifier non plus.
- Quand l'utilisateur clique sur le bouton « Afficher », faites une recherche du numéro du participant dans le tableau résultats.
- Un participant **peut avoir fait plus d'une épreuve**, donc vous pourriez avoir plus d'un résultat (Par exemple le participant 42 a 3 résultats).
- Avant d'afficher les résultats, effacez le contenu de la section <tbody> du tableau.

Si vous trouvez un ou plusieurs résultats

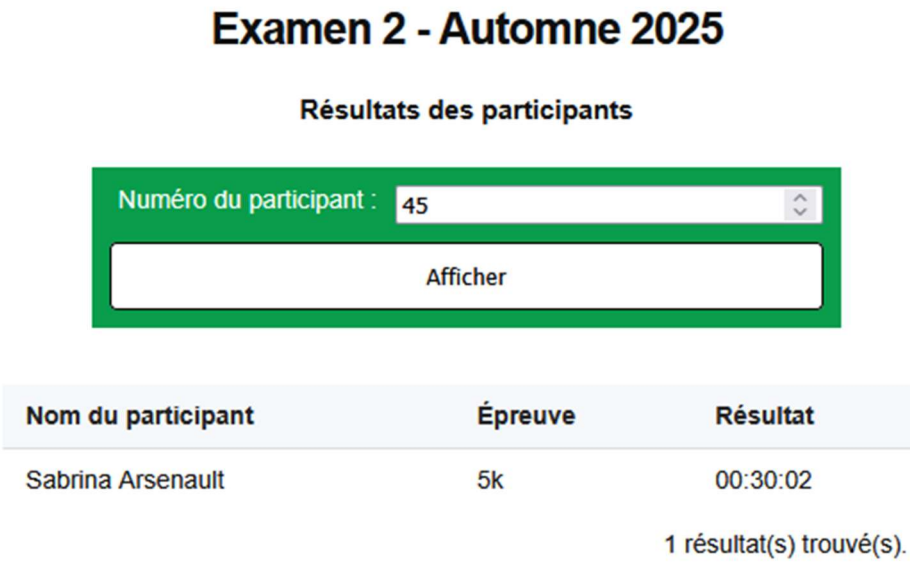
Pour chacun des résultats :

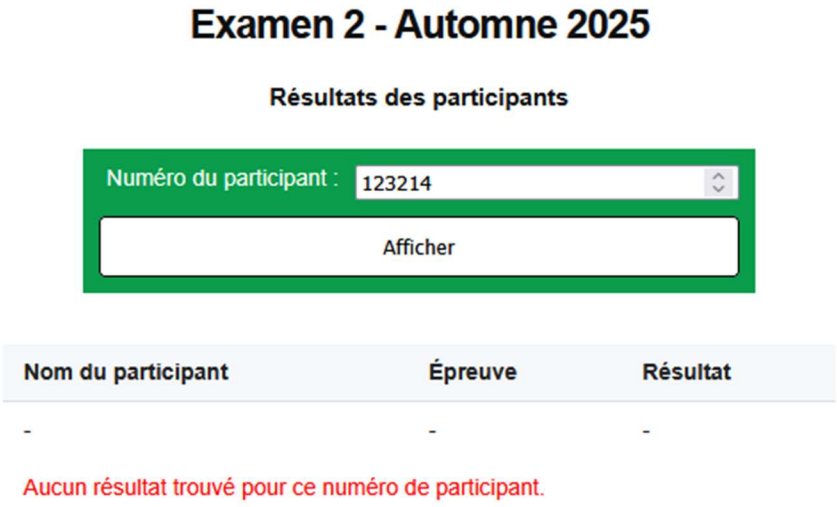
- Créez une nouvelle ligne de tableau.
- Créez trois nouvelles cellules de tableau qui contiendront :
  - Le prénom et le nom du participant
  - Le nom de l'épreuve
  - Le résultat
- Ajoutez ces cellules en ordre dans la nouvelle ligne
- Ajoutez ensuite la nouvelle ligne dans la balise <tbody> du tableau.
- Dans la cellule avec le id **resultat-message**, inscrivez le nombre de résultats trouvés : x résultat(s) trouvé(s).
- Ajoutez à cette cellule la classe **recherche-trouve** et enlever la classe **erreur**.

Si vous ne trouvez pas de résultats

- Créez une nouvelle ligne de tableau.
- Créez trois nouvelles cellules de tableau qui contiendront chacune un tiret ('-').
- Ajoutez ces cellules en ordre dans la nouvelle ligne
- Ajoutez ensuite la nouvelle ligne dans la balise <tbody> du tableau.
- Dans la cellule avec le id **resultat-message**, inscrivez « Aucun résultat trouvé pour ce numéro de participant ».
- Ajoutez à cette cellule la classe **erreur** et enlever la classe **recherche-trouve**.

Exemple d’affichage





Grille d'évaluation

CRITÈRES	INDICATEURS	ÉCHEC (0%)	ACCEPTABLE (60%)	BIEN (80%)	EXCELLENT (100%)
Utilisation appropriée du langage de balisage	• Associer correctement un libellé à chaque contrôle. • Utiliser les attributs appropriés aux contrôles du formulaire. • Valider correctement le formulaire selon les exigences. • Envoyer convenablement les données du formulaire	**Peu ou aucuns** contrôles ont un libellé. **Peu** d'attributs requis sont présents dans le formulaire. **Aucune** validation n'est conforme ou présente. L'adresse **et** la méthode ne sont pas conforme.	**Certains** contrôles ont un libellé. **Certains** attributs requis sont présents dans le formulaire. **Certaines** validations sont conformes et présentes. L'adresse **ou** la méthode n'est pas conforme.	La **majorité** des attributs requis sont présents dans le formulaire. **La majorité** des validations sont conformes et présentes.	**Tous** les contrôles ont un libellé. **Tous** les attributs requis sont présents dans le formulaire. **Toutes** les validations sont conformes et présentes. L'adresse **et** la méthode sont conforme.
	Résultat / 50	**0-29**	**30-39**	**40-44**	**45-50**
Manipulation adéquate des objets du modèle DOM	• Sélectionner adéquatement un élément du DOM • Modifier correctement les attributs d'un élément • Créer et supprimer convenablement un élément.	**Aucun** élément n'est sélectionné. **Aucun** attribut est modifié. **Aucun** élément est créé selon les exigences.	**Certains** éléments sont sélectionnés. **Certains** attributs sont bien modifiés. **Certains** éléments sont créés selon les exigences.		**Tous** les éléments sont sélectionnés. **Tous** les attributs sont bien modifiés. **Tous** les éléments sont créés selon les exigences.
	Résultat / 40	**0-23**	**24-35**		**36-40**
Programmation correcte des interactions entre l'interface Web et l'utilisateur	• Déclencher correctement un script lors d'un événement. • Respecter la syntaxe du langage JavaScript	**Aucun** script est lancé en réaction à un événement. Il n'y a beaucoup d'erreurs de syntaxe dans le script.	Le script est **parfois** lancé correctement en réaction à un événement. Il n'y a quelques erreurs de syntaxe dans le script.		Le script est **toujours** lancé correctement en réaction à un événement. Il n'y a aucune erreur de syntaxe dans le script.
	Résultat / 10	**0-5**	**6-8**		**9-10**
Total	**/ 100**				