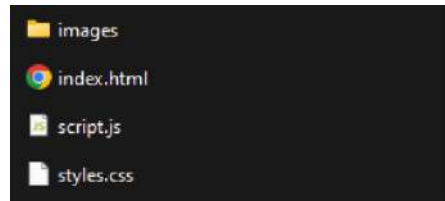

Projet de fin de formation

Le but de ce projet est de créer trois modes de jeux sur le tirage au sort.

Vous débuterez sur le premier mode. Pour les personnes ayant terminé pourront débuter le deuxième mode. Si ce dernier est également terminé, vous pourrez vous lancer sur le troisième mode.

Ce projet disposera donc d'un répertoire comprenant :

1. 1 fichier HTML (index.html)
2. 1 fichier CSS (styles.css)
3. 1 fichier JavaScript (script.js)
4. 1 dossier 'images' contenant les images (qui vous seront fournies)



PS : Le contenu du body dans le fichier HTML devra être vide (à part l'appel au script).

Tout se fera du côté JavaScript! 😊

Avant de vous lancer directement dans JavaScript, il est recommandé de créer tout d'abord votre structure HTML/CSS. Cela vous donnera une meilleure vision pour la manipulation du DOM lorsque vous travaillerez avec JavaScript. Une fois que vous avez donc mis en place votre code HTML/CSS et que vous êtes satisfait du visuel, vous pouvez mettre en commentaire le contenu du "body" et commencer à le mettre en place dans JavaScript.

Pour l'utilisation des symboles, vous pouvez récupérer le CDN de « font-awesome » afin de pouvoir utiliser les différents symboles.

CDN : <https://cdnjs.com/libraries/font-awesome>

Site : <https://fontawesome.com>

Mode 1 : Election

Le but de ce mode est de choisir de manière aléatoire un seul vainqueur.

- L'utilisateur devra insérer 2 joueurs avant de pouvoir lancer le jeu.
(Affichez un message d'avertissement avec le nombre de joueur minimum requis tant que ce n'est pas le cas). Lors de l'ajout d'un joueur :
 1. Vérifiez que le nom du joueur n'existe pas déjà dans la partie (maj non sensitive).
 2. Le nom ne doit pas être une chaîne de caractère vide.
- L'utilisateur peut ajouter/supprimer des joueurs tant que le jeu n'est pas lancé.
- Lors du lancement de tirage, aucune modification peut être faite. Tous les boutons devront être désactivés.
- Une fois que le tirage au sort est terminé, un message de victoire avec une animation de droite à gauche (boucle infini) apparaît. Le joueur peut seulement à ce moment :
 1. Stopper → Arrêter le jeu dans le but de pouvoir ajouter ou supprimer des joueurs ou bien de changer de mode

2. Relancer → Relance le tirage sans aucune modification.
- Le tirage au sort se comporte comme une roulette, c'est-à-dire que la rapidité du random diminue au fur et à mesure du temps pour qu'au finale il s'arrête et choisisse un joueur.

Mode 2 : Classement

Le but de ce mode est de créer un classement. Le random débutera par le dernier du classement pour finir par le premier.

- L'utilisateur devra rentrer 4 joueurs au minimum avant de pouvoir lancer le jeux.
- Une fois qu'un joueur a été sélectionné pour l'intégrer au classement, Il ne pourra plus être sélectionné jusqu'au relancement du tirage.
- Les différents boutons « play » « stop » « add » « retry » ainsi que l'ajout d'utilisateur devront se comporter de la même manière que le premier mode.
- On ne peut pas changer de mode en cours de jeu.
- Dans ce mode, il n'y a pas de message animé mais la position du classement sera indiqué sur la carte du joueur même une fois choisi. Affichez un trophée (or/argent/bronze) pour les personnes étant sur le podium

Mode 3 : Questions

Le but de ce mode est de répondre à des questions (vrai/faux). Il y aura l'utilisation de deux randoms... Un pour choisir le joueur qui devra répondre à la question, et l'autre qui permettra de choisir une question aléatoirement. Après chaque réponse, une nouvelle question et un nouveau joueur seront choisis par random (après un délai d'une seconde).

- Une fois qu'un joueur a répondu incorrectement à une question, il sera éliminé et ne pourra plus être sélectionné avant le redémarrage du mode.
- Les différents boutons « Play » « stop » « add » « retry » ainsi que l'ajout d'utilisateur devront se comporter de la même manière que les autres modes.
- On ne peut pas changer de mode en cours de jeu.
- Lorsqu'une réponse est donnée, le bouton sur lequel le joueur a cliqué change de couleur en fonction de la réponse :
 - rouge : Si la réponse n'est pas correct.
 - vert : Si la réponse est correct.

N'oubliez pas de gérer les différents cas possible et d'afficher un message adéquat dans le cas où :

- Tous les joueurs sont éliminés.
- Si il n'y a plus de questions disponible et qu'il reste encore au moins deux joueurs non éliminé dans le jeux.
- Si il n'y a plus de questions disponible et que tous les joueurs éliminés.

NB : Prenez l'habitude de développer en anglais lors du nommage des fonctions, des classes, des variables, des commentaires.

Il ne faut également pas hésiter à utiliser le site <https://www.w3schools.com> pour surmonter des blocages en HTML, CSS et JavaScript. Ce site est une ressource précieuse pour les débutants et les professionnels, offrant des explications claires et des exemples pratiques pour apprendre et maîtriser ces langages de programmation. Cependant, il est important de ne pas se limiter à une seule source d'information et de chercher d'autres ressources pour obtenir une compréhension complète et approfondie de ces langages.

BONUS : Si les trois modes sont terminés

Lors de l'ajout d'un joueur, mettez en place des RegExp (expression régulière) qui permettra de vérifier:

1. Que le nom indiqué par l'utilisateur dispose de lettre de A à Z, peut disposer d'un nombre à la fin mais peut également disposer d'un/des tiret(s) pour les mots composé ou bien avoir des espaces entre les mots.
2. Dans le cas d'un joueur déjà existant dans le jeu, vérifiez que le nouveau nom indiqué par l'utilisateur dispose obligatoirement un nombre à la fin (exemple : Jean2). Si c'est toujours invalide, reboot de la demande avec le RegExp du point 1.

IMPORTANT :

Je tiens à vous rappeler l'importance de travailler toujours avec des fonctions dans votre fichier JavaScript. Les fonctions sont des éléments essentiels de la programmation, et elles offrent de nombreux avantages. Voici quelques raisons pour lesquelles vous devriez les utiliser :

1. Modularité : Les fonctions vous permettent de découper votre code en petits blocs réutilisables. Vous pouvez les appeler à plusieurs endroits de votre programme, ce qui favorise une approche modulaire et facilite la maintenance de votre code. Cela améliore également la lisibilité et la compréhension de votre script.
2. Réutilisabilité : En encapsulant des morceaux de code dans des fonctions, vous pouvez les réutiliser à travers votre application. Cela évite la duplication de code, ce qui vous permet de gagner du temps et d'éviter les erreurs.
3. Encapsulation : Les fonctions vous permettent d'encapsuler des variables locales à l'intérieur de leur portée. Cela évite les conflits de noms et les effets de bord indésirables, en garantissant que les variables utilisées dans une fonction n'interfèrent pas avec d'autres parties de votre programme.
4. Testabilité : Travailler avec des fonctions rend votre code plus facile à tester. Cela facilite le débogage et la détection des erreurs.

En résumé, l'utilisation de fonctions dans votre fichier JavaScript améliore la modularité, la réutilisabilité, l'encapsulation et la testabilité de votre code. Cela vous permet de développer des applications plus robustes, maintenables et évolutives. Alors, n'oubliez pas d'adopter cette pratique dès le début de votre projet.