**EXERCICE : Réaliser le jeu du pendu.**

Réaliser le jeu du pendu : il y a un mot à trouver, l’utilisateur entre des lettres jusqu’à trouver le mot.  
L’exercice que je vous propose comporte plusieurs niveaux de difficulté.  
Votre objectif principal est de réaliser le premier niveau de complexité pour notre prochain cours. C’est le niveau qui est accessible pour tout le monde.  
Les niveaux de complexité suivant sont à réaliser si vous désirez pousser plus loin vos connaissances en programmation.  
  
**BUT DE L’EXERCICE :** savoir parcourir le contenu d’un tableau grâce aux méthodes apprises durant la session 04 du module LPB, qui sont listées dans les slides 63 à 69 du pdf théorique que je vous fournis. Vous trouverez les exemples pratiques dans le repo github que j’ai mis à jour.  
  
  
**NIVEAU 1 :**

**TU AFFICHES LES INFORMATIONS / INTERACTIONS AVEC L’UTILISATEUR AVEC DES BOITES DE DIALOGUES (alert, prompt, ou confirm)**

* Choisir un mot que votre joueur devra trouver.
* Créer une variable pour le mot à trouver et un tableau qui va accueillir autant de valeur ‘-‘ qu’il n’y a de lettre à trouver, une variable pour le score du joueur et une variable pour accueillir la lettre donnée par le joueur.
* Lancez le jeu.
* Informez votre joueur du nombre de lettres contenue dans votre mot, afficher cette information en représentant un tiret pour chaque lettre (exemple pour trouver le mot ‘bonjour’, afficher la chaine de caractère ‘-------‘).
* Demander au joueur de donner une lettre.
* Vérifier si cette lettre se trouve dans votre mot.
* Si la lettre existe remplacer le ‘–‘ par la lettre trouvée et afficher la variable avec les tirets et la/les lettres trouvée par le joueur.
* Si la lettre n’existe pas, ajouter 1 à la variable score et informer le joueur du nombre d’essais qu’il lui reste
* Si le joueur trouve toutes les lettres, il a gagné.
* Si le score du joueur atteint 10, le jeu est terminé.

Tips utiles tu trouveras dans le fichier exemplePendu.js un exemple pour parcourir une chaine de caractère avec une boucle, et insérer des données quand la lettre proposée est trouvée ?  
  
**NIVEAU 2 :**

On améliore le projet, Tu trouveras dans le fichier fourni dans le zip, le fichier mot.js  
Dans ce fichier tu trouveras 6 tableaux contenant chacun une liste de mot de 5 à 10 lettres.

* Choisis un tableau de mot que tu inséreras dans ton projet.
* Le mot à trouver sera un mot tiré au sort dans le tableau de mot que tu auras choisis.  
  Pour générer un nombre aléatoirement en javascript, tu peux utiliser cette fonction :

Let nombreAléatoire = parseInt(Math.random() \* (maximum - 0) + 0);

* Dans cette fonction, la variable maximum aura pour valeur le nombre total de valeurs incluses dans le tableau mot que tu auras choisi.
* Utilise le nombre tiré au sort pour définir le mot à trouver et met ce mot dans une variable.
* Le jeu est prêt à être lancé, tu peux donc utiliser ce que tu as codé durant le niveau 1
* Que le joueur gagne ou perde, à la fin du jeu, propose-lui de rejouer
* Si le joueur accepte, tu relances le jeu en ayant pris soin de réinitialiser toutes les variables si cela est nécessaire.
* Grâce à la fonction de génération aléatoire, tu vas tirer un autre nombre au sort, et par extension, le mot à trouver sera différent.

**NIVEAU 3 :**

Nouvelle amélioration, On va proposer au joueur le choix de la difficulté. tu vas maintenant utiliser 3 ou tous les tableaux de mots à trouver contenu dans le fichier mot.js

* Demande au joueur de choisir le nombre de lettres que doit contenir le mot à découvrir (tu donnes les uniquement le nombre de lettres des tableaux que tu as décidés d’utiliser)
* Tu tires au sort le mot en fonction du nombre de lettre choisie par le joueur
* Tu adaptes ton code produit durant la niveau pour implémenter les nouvelles fonctionnalité .

**FINAL BOSS :**

**UNIQUEMENT POUR LES CODEURS SOLIDES :**

**On a travaillé jusqu’à présent avec des boites de dialogue. On peut maintenant jouer avec le dom pour afficher les lettres à trouver directement sur la page html plutôt que dans les boites de dialogues. Je vous fourni en plus 10 images qu’il faudra afficher au fur et à mesure que l’utilisateur se trompe.**

* **Quand le joueur trouve une lettre, la page html se met à jour avec les lettres trouvées et celle qui restent à découvrir.**
* **Quand le joueur se trompe on change l’image en fonction du scoring. Les images sont nommées pendu000.png à pendu010.png. Si le score du joueur = 0 on afficher pendu000.png, si le score = 5 on affiche pendu005.png et ainsi de suite jusqu’au game over.**

**Objectif à réaliser :**

**SI TU N’AS JAMAIS FAIT DE JS, TON BUT EST D’ARRIVER A CODER LE NIVEAU 1 POUR COMPRENDRE LA LOGIQUE ; SI TU AS UN PEU PLUS DE TEMPS, TU ES LARGEMENT CAPABLE AVEC LA THEORIE ET LA PRATIQUE A TA DISPOSTION DE CODER LE NIVEAU 2.  
CEUX QUI ONT DÉJÀ UN TOUT PETIT PEU D’XP EN JS DEVRAIENT ETRE CAPABLE DE CODER LE NIVEAU 3**

**CEUX QUI SONT A L’AISE DEVRAIENT POUVOIR DÉJÀ JOUER AVEC LE DOM ET DONC TENTER LE FINAL BOSS ;**On reparle de tout ça dans notre prochaine session, aborde l’exercice avec sérénité. Ce n’est pas une corvée, juste un petit jeu pour te permettre de monter en compétence. BON AMUSEMENT.