Générateur de mots de passe

TP complémentaire 1

Avant de démarrer ce TP, il convient d'avoir suivi les vidéos des modules 1, 2, 3 et 4.

Durée estimée
1 heure / 2 heures

Énoncé

Le travail se fait uniquement dans le fichier script.js

Aucune modification n'est autorisée dans les fichiers HTML, CSS et JavaScript fournis.

En vous appuyant sur l'ensemble des concepts abordés dans les vidéos de cours, mettre en place un générateur de mots de passe.

Niveau 1

Mettre en place les abonnements à l'évènement **clic** sur les deux boutons. Le premier sert à générer le mot de passe*. Le second à copier celui-ci une fois qu'il aura été généré.

*Pour être précis, l'appui sur le premier bouton ne générera pas directement le mot de passe, mais appellera une méthode principale. En effet, il y a plusieurs étapes avant que celui-ci ne soit généré.

La fonction principale devra vérifier quels sont les critères retenus par l'utilisateur (minuscules, majuscules…). Vous pourrez utiliser la console du navigateur pour vérifier la bonne lecture des éléments.

Niveau 2

À partir des critères lus, il faudra générer aléatoirement le mot de passe et afficher celui-ci dans le fichier HTML dans la zone prévue.

Concernant les caractères spéciaux, voici ceux qui ont été retenus :

! " # $ % & ' ( ) * + , - . /

Si aucun critère n'est choisi, il faudra indiquer un message d'information dans la zone du mot de passe.

Niveau 3

Dans cette dernière étape, il faudra gérer le niveau de sécurité du mot de passe en fonction des critères choisis.

Afin de vous aider, l'affichage est déjà codé. Plusieurs fonctions sont présentes dans le fichier **canvas.js**. Nous y reviendrons après.

Nous avons déterminé 5 niveaux caractérisés chacun par une couleur :

- Mauvais
- Moyen
- Correct
- Bon
- Très bon

Pour déterminer dans quel groupe se situe le mot de passe, nous vous proposons 4 paliers. Ceux-ci se déterminent en termes de combinaisons possibles.

Par exemple, si le choix retenu est de deux caractères composés uniquement de chiffres, le nombre de combinaisons est de 10^2 soit 100.

Si l'utilisateur y ajoute les minuscules et augmente la taille du mot de passe pour atteindre 4 caractères, le nombre de combinaisons évolue $(10 + 26)^4$ soit 1679616.

La formule est donc : $\mathbf{x^n}$.

x : le nombre de caractères possibles

n : la taille du mot de passe.

Évidemment, plus le nombre de combinaisons est élevé, plus le mot de passe sera sécurisé.

On considère donc que si le mot de passe est :

- **inférieur ou égal à 8503056** combinaisons alors il est **mauvais**.
- **compris entre 8503056 et 62523502209** combinaisons incluses alors il est **moyen**.
- **compris entre 62523502209 et 1235736291547681** combinaisons incluses alors il est **correct**.
- **compris entre 1235736291547681 et 7326680472586201000** combinaisons incluses alors il est **bon**.
- **Au-delà de 7326680472586201000** combinaisons, il est **très bon**.

Une fois le nombre de combinaisons calculé, il faudra appeler la fonction qui convient pour mettre à jour la partie graphique. Le fichier **canvas.js** possède les fonctions utiles pour cela :

step0() : Etat initial.

step1() : Mauvais

step2() : Moyen

step3() : Correct

step4() : Bon

step5() : Très bon

Enfin, il faudra finaliser la fonction permettant la copie du code dans le **presse-papier**. Pour s'assurer que le code est bien copié, il faudra l'afficher dans une boîte de dialogue. Il est possible d'afficher directement la valeur ou de faire le choix de lire le contenu du presse-papier (plus difficile car nécessite des connaissances non abordées durant ce cours). Vous trouverez les deux solutions dans la correction.

Solution

Une solution est proposée pour ce TP. Ces éléments sont disponibles dans les ressources à télécharger.