

TP 1 - Chiffrement de César

L'objectif de ce TP est de créer un outil de cryptage et de décryptage de texte à l'aide du chiffrement de César.

Notez que, bien qu'il existe de nombreuses façons de mettre en œuvre le Chiffrement de César, vous _ DEVEZ utiliser la méthode décrite ci-dessous pour ce TP.

Chiffrer et déchiffrer du texte à l'aide d'un alphabet de texte chiffré

Pour les besoins de cette mission, nous appellerons le texte non crypté "texte en clair" et le texte crypté "texte chiffré".

Le schéma de cryptage pour cette mission est un simple code de substitution. Le système fonctionne en remplaçant chaque caractère du texte en clair par le caractère correspondant dans un "alphabet chiffré", qui est toujours de la même longueur que l'alphabet en clair.

Par exemple :

l'alphabet en clair : ABCDEFGHIJKLMNOPQRSTUVWXYZ

alphabet chiffré : KLMNOPQRSTUVWXYZABCDEFGHIJ

Compte tenu de l'alphabet du texte chiffré, si nous chiffrions la chaîne "writing code is cool", elle deviendrait "gbsdsxq myno sc myyv". En effet, nous remplaçons "w" par "g", "r" par "b" et ainsi de suite. Le décryptage fonctionne exactement de la même manière, mais à l'inverse. Ainsi, "myyv" en utilisant les alphabets ci-dessus devient "cool".

Génération d'un alphabet de texte chiffré à partir d'une clé

Dans votre programme, plutôt que de demander à l'utilisateur de spécifier directement un alphabet de texte chiffré, vous lui demanderez plutôt d'entrer une clé qui sera utilisée pour générer l'alphabet de texte chiffré. Nous commencerons par mettre notre alphabet de texte chiffré à égalité avec l'alphabet de texte en clair. Ensuite, nous déplacerons chaque caractère vers la droite du nombre de lettres spécifié par la clé. Lorsque nous arrivons à la fin, les lettres s'enroulent autour.

Ainsi, par exemple, si notre clé est 3, nous commencerons par :

l'alphabet en clair : ABCDEFGHIJKLMNOPQRSTUVWXYZ

Ensuite, déplacez chaque lettre vers les 3 positions de droite :

alphabet chiffré : XYZABCDEFGHIJKLMNOPQRSTUVWXYZ

Notez que le A est décalé vers les 3 lettres de droite, et les 3 dernières lettres de l'alphabet (XYZ) se sont déplacées vers le début.

Exigences du programme

Règles de cryptage et de décryptage

Pour crypter ou décrypter le texte saisi par l'utilisateur, vous devez utiliser les règles suivantes pour chaque caractère :

- S'il s'agit d'une lettre, remplacez-la comme décrit ci-dessus par une lettre de la même casse (majuscule ou minuscule).
- S'il s'agit d'un espace, conservez l'espace.
- S'il s'agit d'autre chose, sautez cette étape. Il ne doit pas apparaître dans la sortie.

Par exemple, si nous avons les alphabets ci-dessous (la touche est 16) :

l'alphabet en clair : ABCDEFGHIJKLMNOPQRSTUVWXYZ

alphabet chiffré : KLMNOPQRSTUVWXYZABCDEFGHIJ

Et ils ont entré "Writing code is,!~;;?5555 Cool", nous obtenons "Gbsdsxq myno sc Myyv" comme résultat, car nous préservons les espaces, mais en supprimant tout autre caractère non trouvé dans l'alphabet.

Page HTML

Créez une page HTML avec un formulaire simple pour accepter la saisie d'une clé (numérique 1-25) et d'un message texte (de n'importe quelle longueur). Il doit y avoir 3 boutons : Crypter, Décrypter, & Réinitialiser. Ajoutez également des *instructions* de base à la page afin que toute personne qui navigue sur cette page sache à quoi elle sert et comment elle fonctionne.

Sous le formulaire, ajoutez un élément `div` pour la sortie. Il sera utile d'ajouter des attributs `id` sur les `div`, les boutons et les contrôles d'entrée pour les identifier facilement et y accéder à partir de JavaScript.

CSS de base

Utilisez le CSS pour que le formulaire s'affiche bien. Si vous le souhaitez, vous pouvez utiliser Bootstrap ou une autre bibliothèque pour vous aider dans votre affichage. Sinon, vous devrez écrire vous-même quelques CSS de base.

Code JS

N'oubliez pas d'écrire un bon code JavaScript ES6. J'attends de vous que vous créiez et utilisiez des fonctions avec des paramètres d'entrée et des valeurs de retour pour structurer votre code. Ce problème est *très* complexe si vous ne le décomposez pas en petites fonctions pour traiter les tâches individuelles.

Obtenir les valeurs d'entrée

Tout d'abord, nous devons configurer nos boutons pour obtenir la valeur des contrôles de saisie afin de pouvoir les utiliser dans le code de cryptage/décryptage. Configurer un gestionnaire de clic pour chaque bouton qui obtiendra chacun des champs du formulaire et enregistrera la valeur dans la console.

Astuce : Utilisez la méthode `String trim()` pour supprimer les espaces du début et de la fin de vos valeurs de saisie.

Réinitialisation du formulaire

Configurez un autre gestionnaire de clic pour effacer les valeurs des champs de saisie. Vous pouvez définir la valeur des contrôles de la même manière que vous obtenez la valeur, il suffit de mettre l'expression à gauche du signe égal.

Par exemple : `MonElement.value = ""`

Note : Si vous utilisez un type de saisie de formulaire "reset", vous n'avez pas besoin d'écrire de code pour le faire. Toutefois, pour ce TP, je veux m'assurer que vous êtes en mesure de définir les valeurs des champs de saisie par le biais de votre code JS.

Crypter et décrypter

Une fois que vous avez reçu les données de votre formulaire, vous pouvez travailler sur le code pour crypter ou décrypter le message. Pour effectuer le cryptage ou le décryptage, vous devrez utiliser des tableaux pour créer les alphabets "texte clair" et "texte chiffré" comme indiqué ci-dessus.

Je vous recommande de commencer par créer une variable de tableau pour l'alphabet "texte clair", car cela ne devrait jamais changer. (Soyez très prudent dans la configuration, car le fait de manquer une lettre ou d'échanger des lettres entraînera des traductions incorrectes). Ensuite, écrivez une fonction pour générer l'alphabet "texte chiffré" en décalant les lettres de la quantité spécifiée par la clé.

Une fois que vous avez l'alphabet en clair et les alphabets en cryptogramme corrects, vous pouvez commencer à parcourir le message de l'utilisateur, caractère par caractère. Trouvez l'emplacement de la lettre dans le tableau de l'alphabet en clair, puis utilisez l'index du tableau pour obtenir la lettre appropriée de l'alphabet de texte chiffré.

Conseils utiles

Pour que ce code fonctionne correctement, vous voudrez savoir si une lettre est en majuscule ou en minuscule. Sachez qu'une chaîne de caractères a une méthode `toUpperCase()`. Nous pouvons évidemment l'utiliser pour changer une lettre (ou une chaîne) en majuscule, mais nous pouvons également l'utiliser pour déterminer si la lettre était majuscule ou non au départ.

```
let lettre = "a" ;
if (letter === letter.toUpperCase()) {
  console.log ("C'était une lettre majuscule !") ;
} else {
  console.log ("Ce n'était pas une majuscule !") ;
}
```

Si la lettre était en majuscule, vous voulez que la lettre traduite soit également en majuscule. Sinon, vous devrez utiliser la méthode `String.toLowerCase()` pour changer la lettre en minuscules avant de l'écrire dans la sortie.

Rappelez-vous que vous aurez besoin d'une lettre en majuscule lorsque vous chercherez une correspondance dans l'alphabet en clair.__