



Nom & Prénom :

Rotation - Miroir

On se propose de concevoir une page web permettant de saisir une chaîne de caractères **ch** et de la crypter en utilisant le principe suivant :

▲ Étape 1 : Rotation13

Effectuer, pour chaque caractère de **ch**, une rotation de 13 caractères dans l'ordre alphabétique en appliquant la formule $\text{Chr}(97 + (\text{Ord}(\text{ch}[i]) - 97 + 13) \bmod 26)$ pour obtenir le caractère correspondant.

▲ Étape 2 : Miroir

Transformer la chaîne de caractères obtenue à l'issue de l'Étape 1 en son miroir. C'est-à-dire permuter le premier caractère avec le dernier, le deuxième caractère avec l'avant dernier et ainsi de suite.

Exemple :

Pour **ch** = "algo"

▲ Étape 1 : Rotation13

- le caractère "a" sera crypté en "n".
En effet, $\text{Chr}(97 + (\text{Ord}(\text{"a"}) - 97 + 13) \bmod 26) = \text{Chr}(110)$ qui correspond au caractère "n"
- le caractère "l" sera crypté en "y".
En effet, $\text{Chr}(97 + (\text{Ord}(\text{"l"}) - 97 + 13) \bmod 26) = \text{Chr}(121)$ qui correspond au caractère "y".
- le caractère "g" sera crypté en "t".
En effet, $\text{Chr}(97 + (\text{Ord}(\text{"g"}) - 97 + 13) \bmod 26) = \text{Chr}(116)$ qui correspond au caractère "t".
- le caractère "o" sera crypté en "b".
En effet, $\text{Chr}(97 + (\text{Ord}(\text{"o"}) - 97 + 13) \bmod 26) = \text{Chr}(98)$ qui correspond au caractère "b".

D'où le résultat de l'étape 1 est **"nytb"**

▲ Étape 2 : Miroir

Le miroir de la chaîne **"nytb"** est **"btyn"**.

Donc, le cryptage de la chaîne **"algo"** en appliquant le principe ci-dessus est **"btyn"**

Travail demandé

- Créer le formulaire présenté en figure 1. Placer le code :
 - HTML dans un fichier nommé **"miroir.html"**.
 - CSS dans le fichier **"miroir.css"**.
 - JavaScript dans le fichier **"miroir.js"**.
- Développer la fonction **estValide(ch)** qui vérifie si la chaîne **ch** est constituée uniquement par des lettres minuscules.
- Développer la fonction **miroir(ch)** qui retourne le miroir d'une chaîne de caractères **ch**.
- Implémenter la fonction **rotation(ch)** qui permet de retourner le résultat de l'étape 1.

Fonction Rotation(ch : Chaîne):Chaîne

Debut

res ← ""

Pour i de 0 à Long(ch)-1 Faire

res ← res + Chr(97+(Ord(ch[i])-97+13) mod 26)

Fin Pour

Retourner res

Fin

Équivalence

Algorithme	JavaScript
chr(n)	String.fromCharCode(n)
ord(car)	car.charCodeAt(0)

Exemples

String.fromCharCode(97) ⇒ "a"

const car = "b"; car.charCodeAt(0) ⇒ 98

5. Développer la fonction **crypter()** qui s'exécute suite à un clic sur le bouton "**Crypter**", permettant :
- de récupérer la chaîne **ch** saisie. La chaîne **ch** doit être non vide, de longueur inférieure à 10 et contient seulement des lettres alphabétiques minuscules.
 - de retrouver la chaîne cryptée en utilisant les fonctions **rotation(ch)** et **miroir(ch)**.
 - et d'afficher le résultat adéquat comme le montrent les figures 2, 3 et 4.

	succession.html	succession.css	succession.js	
Barème	Squelette	0.75		Note
	Formulaire	0.75		
	Composants + Label	3.75		
	Gestion évènement	0.75		
	Liaison CSS	0.75		
	Liaison JS	0.75		
		4 règles de style	3	
			estValide() 2.5 miroir() 2.5 rotation() 2 crypter() 2.5	

Figure 1

Rotation - Miroir

Une chaîne =

Crypter

-

Figure 2

Rotation - Miroir

Une chaîne =

Crypter

Veuillez introduire une chaîne.

Figure 3

Rotation - Miroir

Une chaîne =

Crypter

La chaîne cryptée est : btyl

Figure 4

Rotation - Miroir

Une chaîne =

Crypter

Introduire une chaîne alphabétique.