

# Le projet : le chiffrement de César

Jean-Christophe Toussaint & Lionel Bastard

12 novembre 2025

## 1 Principe

En cryptographie, le chiffrement de César, également appelé décalage de César, est l'une des techniques de chiffrement les plus simples et les plus connues. Il s'agit d'un type de chiffrement par substitution dans lequel chaque lettre du texte en clair est remplacée par une lettre située à un nombre fixe de positions dans l'alphabet. Par exemple, avec un décalage vers la gauche de 3, D est remplacé par A, E devient B, et ainsi de suite. La méthode doit son nom à Jules César, qui l'utilisait dans sa correspondance privée.

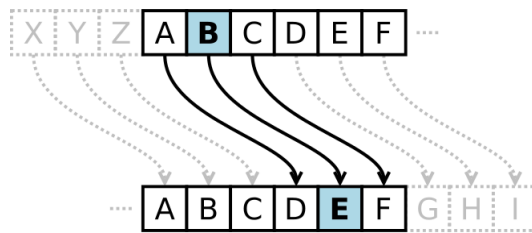


FIGURE 1 – Le chiffre de César fonctionne par décalage des lettres de l'alphabet. Source wikipedia.

L'étape de chiffrement effectuée par un chiffrement de César est souvent incorporée dans des schémas plus complexes, tels que le chiffrement de Vigenère, et a encore une application moderne dans le système ROT13. Comme tous les chiffres de substitution à un seul alphabet, le chiffre César est facilement cassé et, dans la pratique moderne, n'offre pratiquement aucune sécurité de communication.

## 2 Préambule

On donne ci-après quelques instructions de base python pour réaliser votre projet.

1. La fonction membre `index(element)` de la classe `list`, donne la position de la première occurrence d'un élément rencontré dans une liste.

Exemple : soit `abc=['A', 'B', 'C', 'D', 'E']`, `abc.index('A')` retourne 0, `abc.index('E')` retourne 4, tandis que `abc.index('Z')` génère une erreur du type « `ValueError` » car 'Z' n'est pas un élément de la liste.

2. La fonction python `chr(n)` renvoie le caractère associé au code ASCII  $n \in [0, 128[$ . Elle prend un seul entier comme argument. Si l'entier passé en paramètre est en dehors de la plage, la méthode retourne « `ValueError` ». Exemple : `chr(65)` retourne 'A'.
3. La fonction python `ord(c)` renvoie la valeur ASCII associé au caractère `c`. Si la longueur de la chaîne passée en paramètre est supérieure à un, une erreur « `TypeError` » est générée. Exemple : `ord('A')` retourne 65.
4. La fonction membre `upper()` de la classe `string` permet de transformer une chaîne comportant des caractères minuscules en majuscules. Exemple `"aBc".upper()` retourne `"ABC"`.
5. L'astuce suivante permet de transformer une liste de caractères en une chaîne de caractères : `".join(['A', 'B', 'C'])` où `"` est le caractère vide (2 fois simple quote) retourne la chaîne `"ABC"`.

### 3 Exercices

1. Générer la liste `abc` contenant les caractères allant de 'A' jusqu'à 'Z'. Ajouter-lui à droite le caractère blanc.
2. Générer la liste `abc_cipher` à partir de la liste précédente `abc`, décalé d'un entier `shift`. Avec `shift=3`, vous devez retrouver le résultat de la figure 1.

### 4 Développement

1. Développer une fonction `encrypt_string(string, shift)` qui à partir d'une chaîne de caractères `string` retourne sa version cryptée pour un décalage fixé `shift`.
2. Développer une fonction `decrypt_string(string, shift)` qui fait l'opération inverse.
3. Développer un programme principal, permettant de valider les deux fonctions précédentes, en prenant quelques exemples. N'oubliez pas de transformer la chaîne de caractères saisie en sa version en majuscules avant de l'encrypter. Quelles limitations identifiez-vous ?

### 5 Application à un document texte

On vous demande de développer une version plus élaborée qui fera le cryptage d'un texte contenu dans un fichier texte.

1. Développer d'abord une fonction `readfile(filename)` qui lit le texte au format ASCII depuis le fichier de nom `filename` et qui retourne une liste contenant chaque ligne du fichier sous la forme d'un élément de liste. Cette liste sera nommé `page`.  
*Indication* : on utilisera la fonction `readlines()`.
2. Développer une fonction `encrypt_page` qui encrypte la liste précédente `page` en fixant le décalage `shift` à 3. Elle retournera la liste des lignes cryptées. Pour éliminer le caractère de fin de ligne `'\n'`, on appliquera la fonction membre `rstrip('\n')` à chaque ligne lue.
3. Développer une fonction `decrypt_page` permettant de décrypter une liste cryptée de chaînes de caractères. La valeur de `shift` est fixée à 3.
4. Développer une fonction `savefile(filename, page)` pour sauvegarder cette liste encryptée dans un fichier dont le radical est celui d'origine complété par le suffixe `"_cipher"`. `page` est une liste de chaînes de caractères en majuscule.  
Par exemple : si `filename` est `"OnlyForYourEyes.txt"` alors le nom du fichier crypté s'appellera `"OnlyForYourEyes_cipher.txt"`.  
*Indication* : utiliser la fonction membre `split('.')` de la classe `string` pour découper le nom du fichier en utilisant le séparateur `'.'`