

## Mini projet : Les fondamentaux de Python

Ce projet consiste à coder un programme permettant à un utilisateur de crypter ses fichiers confidentiels en s'identifiant et en utilisant sa propre méthode de chiffrement.

### Consigne :

Ce que doit faire votre programme (partie **clé**) :

- L'utilisateur exécute votre programme (double clic sur le fichier python).
- Le programme lui demande de s'authentifier en lui demandant une combinaison d'informations personnelles aux choix (mot de passe/email, mot de passe/nom utilisateur...).
- Afin de se protéger de l'intrusion d'une potentielle machine. Si l'utilisateur échoue 5 fois à fournir la bonne combinaison d'informations lors de l'authentification, le programme doit se verrouiller et afficher un message d'erreur. Dans ce cas-là, le programme doit également prévenir l'utilisateur qu'il y a eu une tentative d'intrusion lorsque le programme est relancé.
- Si l'utilisateur réussit à s'authentifier, le programme lui demande de saisir le nom d'un fichier texte à crypter.
- Une fois que l'utilisateur a saisi le nom du fichier, le programme va ouvrir et lire le contenu du fichier présent dans le dossier, puis va crypter ce contenu avant de l'enregistrer dans un nouveau fichier texte dans le dossier du projet (pour simplifier le projet, on ne traite pas l'erreur ou le fichier n'est pas présent dans le dossier).
- Tant que l'utilisateur ne ferme pas le programme, celui-ci continue à lui demander un nom de fichier texte à crypter.

Ce que doit faire votre programme (partie **bonus**) :

- Créer un programme qui vérifie que le fichier à crypter est bien dans le dossier du projet.
- Créer un programme permettant de décrypter le fichier crypté.
- S'il y a une tentative d'intrusion, enregistrer l'heure de l'intrusion et verrouiller le programme pour une durée de 24 à partir de la date d'intrusion.

Pour se challenger :

- Prévenez l'utilisateur par email en cas d'intrusion.
- Modifier votre programme pour que celui-ci fonctionne également avec des images de types png et jpeg.

### Remarque :

Pour réaliser ce programme, vous pouvez utiliser soit des méthodes(programmation procédurale), soit des classes (programmation orientée objets).

Pour **crypter le contenu d'un fichier texte**, vous devez réaliser **par vous-même** le code et il n'est pas autorisé d'utiliser une librairie déjà disponible. Une méthode de cryptage **simple** est suffisante (soyez créatif).

### Exemple de méthodes de cryptage simples (à titre d'exemple) :

Une potentielle solution pourrait être de transformer les lettres d'un fichier en nombres :

"Test test test test" => 10114871 15114871 15114871 15114871

Une potentielle solution pourrait être de décaler certaines lettres d'un mot :

"Test test test test" => eTst etst etst etst

Une potentielle solution pourrait être d'interchanger les lettres d'un mot :

"Test test test test" => Ocer bcer bcer bcer

Une potentielle solution pourrait être de crypter un mot en fonction d'une suite de chiffre :

"Test test test test" => 185484fe5s1fe4sf

124587

185484fe5s1fe4sf => "Test test test test"

124587

### Rappel :

Pour ouvrir un fichier texte et lire son contenu ou éditer son contenu :

- On commence par utiliser la fonction **open()** de python en mode écriture ou lecture
- Puis on utilise les fonctions **read()** ou **readlines()** pour lire le contenu du fichier.
- On utilise la fonction **write()** pour écrire le contenu du fichier.
- Enfin on termine en utilisant la fonction **close()** pour fermer le fichier (tant que l'on a pas utilisé cette fonction python n'écrira pas de contenu dans notre fichier).

Pour en savoir plus :

- <https://python.doctor/page-lire-ecrire-creer-fichier-python>
- [https://python.sdv.univ-paris-diderot.fr/07\\_fichiers/](https://python.sdv.univ-paris-diderot.fr/07_fichiers/)

### Conseil :

Pour simplifier le code, je vous conseille de créer un dossier pour le projet et de **placer les fichiers cryptés et les fichiers textes que l'utilisateur souhaite crypter dans le même dossier.**

### Notation :

La notation prendra en compte les éléments suivants :

- Le fonctionnement du code.
- La lisibilité du code.
- La compartimentation du code (utilisation de fonctions ou classes).
- La créativité (choix de la méthode de chiffrement et idées additionnelles).

- La complexité du code.