

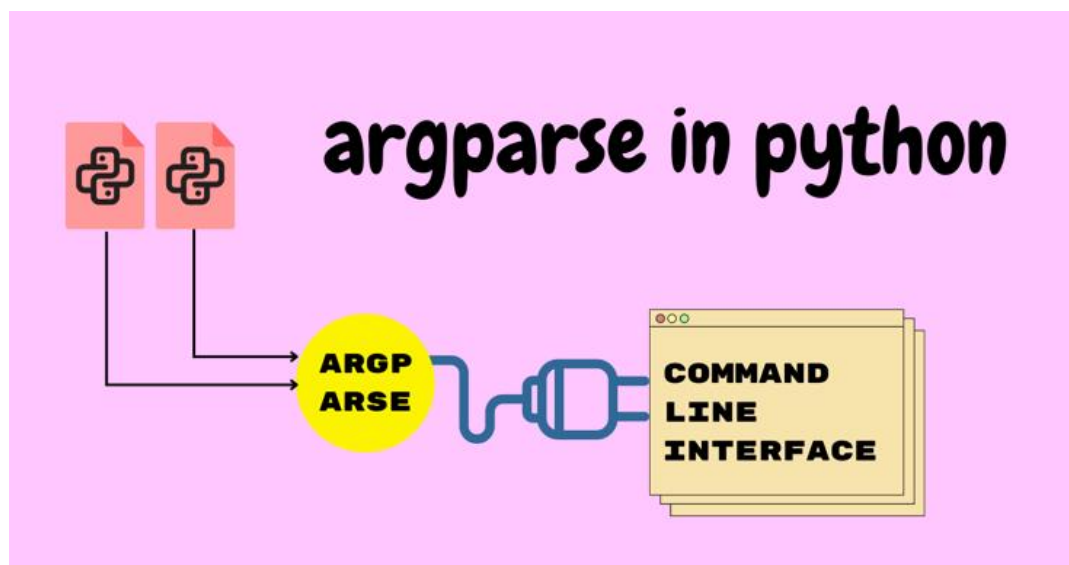
Module 122

Introduction :

Dans le cadre du module 122, l'objectif était d'explorer le chiffre de Vigenère en utilisant Python pour créer un projet de Scripting. Le chiffre de Vigenère est une méthode de chiffrement par substitution poly alphabétique (qui veut dire que chaque lettre peut être remplacée par différentes lettres selon sa position dans le message et la clé utilisée pour le chiffrement. Cela rend le cryptage plus robuste et moins vulnérable aux attaques de cryptanalyse, améliorant ainsi la sécurité du message chiffré). Contrairement au chiffrement César qui est aussi très connu, où chaque lettre est remplacée par une autre lettre avec un décalage fixe, le chiffre de Vigenère utilise une série de décalages basée sur une clé.

Librairie Argparse

La librairie argparse est utilisée pour faciliter la création de lignes de commande avec des arguments dans les programmes Python. Elle permet de définir assez facilement les options, les arguments positionnels et les aides pour les programmes en ligne de commande. En résumé, argparse simplifie la gestion des entrées utilisateur et rend les programmes Python interactifs plus conviviaux en permettant aux utilisateurs de spécifier des options et des arguments lorsqu'ils exécutent le programme à partir de la ligne de commande.



Explication de toutes les fonctions utilisées dans le script

`encrypt_decrypt(text, key, mode) :`

Cette fonction prend en entrée un texte, une clé de chiffrement et un mode (chiffrement ou déchiffrement).

Elle chiffre ou déchiffre le texte en fonction de la clé et du mode spécifiés.

Les caractères alphabétiques du texte sont décalés dans l'alphabet en fonction de la clé et du mode.

Les caractères spéciaux et les chiffres sont ignorés et conservés inchangés.

La fonction renvoie le texte chiffré ou déchiffré.

`main() :`

Cette fonction est la fonction principale du script.

Elle analyse les arguments de la ligne de commande pour déterminer le mode (chiffrement ou déchiffrement), le fichier d'entrée, la clé de chiffrement et le fichier de sortie.

Elle vérifie l'existence du fichier d'entrée et lit son contenu.

Elle appelle la fonction `encrypt_decrypt` pour chiffrer ou déchiffrer le texte.

Enfin, elle écrit le résultat dans le fichier de sortie et affiche un message de succès.

Fonction pour le chiffrement

La fonction de chiffrement (`encrypt_decrypt`) prend en entrée un texte, une clé de chiffrement et un mode. Elle chiffre le texte en déplaçant chaque caractère alphabétique dans l'alphabet selon la clé et le mode spécifiés. Les caractères alphabétiques en minuscules et en majuscules sont traités de manière distincte. Les caractères spéciaux et les chiffres sont ignorés et conservés inchangés. Le résultat est le texte chiffré.

Exécution :

Message original

```
message-original.txt
1  Bonjour ! Voici un message chiffré ! 123456 @!!
2  |
```

Commande pour chiffrer le message

```
122-python python3 main.py -c message-original.txt 123 message-chiffré.txt
L'opération c'est bien passée !
```

Le résultat chiffré (les caractères spéciaux ont été ignorés)

```
message-chiffré.txt
1  Uhgchnk ! Ohbvb ng fxlltzx vabyyké ! 123456 @!!
```

Fonction pour le déchiffrement

La fonction de déchiffrement (`encrypt_decrypt`) est essentiellement la même que la fonction de chiffrement, mais avec un mode différent. Elle prend en entrée un texte chiffré, une clé de déchiffrement et un mode de déchiffrement. Elle décale chaque caractère alphabétique dans l'alphabet en sens inverse selon la clé et le mode spécifiés pour retrouver le texte original. Tout comme la fonction de chiffrement, elle ignore les caractères spéciaux et les chiffres. Le résultat est le texte déchiffré.

Exécution :

Message à déchiffrer :

```
message-chiffré.txt
1 Uhgchnk ! 0hbvb ng fxlltzn vabyyké ! 123456 @!!
```

Commande pour déchiffrer :

```
122-python python3 main.py -d message-chiffré.txt 123 message-déchiffré.txt
L'opération c'est bien passée !
```

Le résultat déchiffré :

```
message-déchiffré.txt
1 Bonjour ! Voici un message chiffré ! 123456 @!!
```

Conclusion

J'ai vraiment apprécié ce projet. Nous avons bien réparti les tâches, comme pour le dernier projet, et nous l'avons réalisé avec plaisir.

José Pedro Gomes Cunha

J'ai bien aimé ce LABO, il m'a permis de me remettre un peu à faire du Python car je n'en avais pas fait depuis longtemps. Je pense que ce travail a été très utile car il permet aux gens qui le font d'avoir quelques notions du langage le plus utilisé en 2024.

Gabriel Dovat