
	<b>Projet 2</b>  <b>Le code de César et de Vigenère</b> <b>Travail demandé Elève 1</b>	<b>1 / 3</b>  
---	---	---

## 1 Principe de codage

**Question 1.** Expliquer en quelques lignes et en utilisant une représentation graphique le principe du code de César.

## 2 Manipuler les fichiers texte

**Objectif :** Apprendre à manipuler des fichiers texte en Python de façon à :

- ouvrir, fermer un fichier
- lire et écrire du texte dans un fichier

### 2.1 Lecture d'un fichier texte

Vous disposez d'un fichier python qui permet de lire un fichier texte.

Nom du fichier : **liretexteEleve1.py**

Programme 1	
fichier="texte.txt"	#Nom du fichier traité
file = open(fichier, "r")	#Ouverture du fichier
contenu=file.read()	#Lecture du contenu du fichier
file.close()	#Fermeture du fichier
print (contenu)	
print(type(contenu))	#Affiche le type de variable

**Question 2.** Tester le programme mis à disposition en mode « Déboguer », voir Document Ressource DR1. Analyser l'évolution du contenu des variables notamment la variable « contenu ». Préciser le type de la variable « contenu ».

Votre camarade de projet a testé un programme qui permet aussi de lire un fichier texte mais qui est différent du votre.

**Question 3.** A partir des critères d'analyse définis, **comparer** les deux programmes.

*Pour la suite du projet on utilisera le programme 1.*

### 2.2 Ecriture dans un fichier texte

Vous disposez d'un fichier python qui permet de lire un fichier texte.

Nom du fichier : **ecrituretexteEleve1.py**

Elève 1
fichier="texteEcrire1.txt"
fichier = open(fichier, "w")
fichier.write("nouveau texte 1")
fichier.close()

**Question 4.** Tester le programme mis à disposition. Analyser le résultat en ouvrant le fichier texte modifié.

Votre camarade de projet a testé un programme qui permet aussi d'écrire dans un fichier texte mais qui est différent du votre.

**Question 5.** Comparer vos analyses et choisir le programme que vous conserveriez pour la suite du projet. Justifier votre réponse.

## 3 Format du fichier texte

Le texte contenu dans le fichier texte et qui devra ensuite être codé, ne comportera que des lettres minuscules non accentuées et des espaces et des retours à la ligne (saut de ligne).

## 4 Implémenter le codage

### 4.1 Fichiers texte

**Question 6.** Ecrire une fonction qui permet de récupérer le contenu d'un fichier texte :

- Nom de la fonction : **lireFichier()**
- Paramètre : **nomFichier**
- Renvoi : variable « **contenu** » de type chaîne de caractères contenant le texte

**Question 7.** Tester la fonction afin de vérifier son bon fonctionnement.

### 4.2 Caractères et nettoyage

Le travail fait précédemment nous a permis de récupérer l'ensemble du texte contenu dans un fichier. Il nous faut maintenant le préparer en vue de son codage. Deux étapes sont nécessaires :

- Décomposer la chaîne de caractères contenant le texte de façon à placer chacune des lettres dans une liste (« texte » → ['t', 'e', 'x', 't', 'e'])
- Supprimer les caractères particuliers, ici les espaces ' ' et les sauts de lignes matérialisés par \n ([ 't', 'e', 'x', 't', 'e', ' ', ' ', 'l', 'u', '\n' ] → ['t', 'e', 'x', 't', 'e', 'l', 'u' ])

Pour cela on va créer une **fonction** pour chaque **étape**. Chacune sera écrite depuis un algorithme fournis.

**Question 8.** A partir de l'algorithme donné, **écrire** une fonction qui permet de décomposer la chaîne de caractère en une liste de lettre :

- Nom de la fonction : **extraire()**
- Paramètre : **contenu**
- Renvoi : variable « **listeLettres** » de type liste contenant toutes les lettres et caractères autres du texte
- La liste **listeLettres** est stockée dans une variable de type liste de même nom

Algorithme
<b>DEBUT DE L'ALGORITHME</b> <b>Créer</b> une liste vide <b>POUR</b> chaque caractère <b>car</b> de la variable contenant le texte <b>DEBUT POUR</b> <b>Ajouter</b> le caractère à la liste contenant les lettres <b>FIN POUR</b> <b>RETOURNER</b> la liste contenant toutes les lettres <b>FIN DE L'ALGORITHME</b>

**Question 9.** Tester la fonction afin de vérifier son bon fonctionnement.



**Question 10.** Envoyer votre travail via le travail à faire de l'ENT. Le format du fichier est le suivant : **cesarE1-J1.py**

### 4.3 Mise en commun des fonctions

**Question 11.** Regrouper dans un seul fichier les différentes fonctions implémentées, soit :

<ul style="list-style-type: none"> <li>• <b>lireFichier()</b></li> <li>• <b>extraire()</b></li> </ul>	<ul style="list-style-type: none"> <li>• <b>ecrireFichier()</b></li> <li>• <b>supCarPar()</b></li> </ul>
---	--

**Question 12.** Tester le programme afin de vérifier son bon fonctionnement.



**Question 13.** Envoyer votre travail via le travail à faire de l'ENT. Le format du fichier est le suivant : **cesarE1&2-J2.py**

#### 4.4 Implémentation sans utilisation du code Ascii

**Question 14.** Faire une copie du fichier dont le nom est `cesarE1&2-J2.py`.

**Question 15.**

- Créer un fichier texte, dont le nom est `alphabet.txt`, contenant toutes les lettres de l'alphabet.
- Stocker l'ensemble des lettres de l'alphabet dans une variable de type chaîne de caractère et nommée `alphabet`. Pour cela utiliser la fonction **lireFichier()** déjà écrite précédemment.
- En vous aidant de l'algorithme donné, écrire une fonction qui permet de prendre en compte la clé de codage (décalage) :
  - Nom de la fonction : **decalage()**
  - Paramètres : **alphabet** et **cle**
  - Renvoi : variable « **copieAlphabet** » de type liste contenant l'alphabet décalé en fonction de la valeur de la clé de codage
  - La liste **copieAlphabet** est stockée dans une variable de type liste nommée **alphabetDecale**

Algorithme
<b>DEBUT DE L'ALGORITHME</b> Créer une copie de la liste <b>alphabet</b> nommée <b>copieAlphabet</b> <b>POUR</b> la variable <b>index</b> allant de <b>0</b> à <b>cle</b> <b>DEBUT POUR</b> Affecter le caractère traité à la variable <b>car</b> Supprimer le premier caractère de la liste <b>copieAlphabet</b> Ajouter le caractère contenu dans la variable <b>car</b> à la fin de la liste <b>copieAlphabet</b> <b>FIN POUR</b> RETOURNER la liste contenant toutes les lettres avec prise en compte de la clé <b>FIN DE L'ALGORITHME</b>

**Question 16.** Tester la fonction.

**Question 17.** En vous aidant de l'algorithme donné, écrire une fonction qui permet de coder le texte :

- Nom de la fonction : **codage()**
- Paramètres : **listeLettres** et **alphabet** et **alphabetDecale**
- Renvoi : variable « **texteCode** » de type liste contenant l'alphabet décalé en fonction de la valeur de la clé de codage
- La liste **texteCode** est stockée dans une variable de type liste de même nom

Algorithme
<b>DEBUT DE L'ALGORITHME</b> Créer une liste vide nommée <b>texteCode</b> <b>POUR</b> chaque caractère <b>car</b> de la liste contenant toutes les lettres <b>listeLettres</b> <b>DEBUT POUR</b> Affecter la valeur 0 à la variable <b>indice</b> de type entier <b>TANT QUE</b> le caractère n'est pas trouvé dans l'alphabet <b>DEBUT TANT QUE</b> Incrémenter la variable <b>indice</b> <b>FIN TANT QUE</b> Ajouter le caractère codé correspondant à la fin de la liste <b>texteCode</b> <b>FIN POUR</b> Créer une variable de type chaîne de caractères vide nommée <b>texteMystere</b> <b>POUR</b> chaque caractère <b>car</b> de la liste contenant toutes les lettres codées <b>texteCode</b> <b>DEBUT POUR</b> Ajouter à la variable <b>texteMystere</b> chaque lettre codée contenu dans <b>texteCode</b> <b>FIN POUR</b> RETOURNER le texte codé contenu dans la variable <b>texteMystere</b> <b>FIN DE L'ALGORITHME</b>

**Question 18.** Ecrire le texte codé dans un fichier texte dont le nom est `texteMystere.txt`.

**Question 19.** Tester l'ensemble.

**Question 20.** Créer un fichier compacté au format `cesarE1-J3.zip` contenant tous les fichiers nécessaires au fonctionnement de votre programme et le rendre via l'ENT.

Jalon

Etape 1

Etape 2