

Travail sur les fichiers texte et les chaînes de caractères

Séance 7

Remplir la fiche de cours à mesure que vous allez rencontrer les nouvelles fonctions. Vous indiquerez pour chaque fonction rencontrée l'utilité de la fonction et surtout un exemple permettant de mettre clairement en valeur la syntaxe à utiliser

Créer un programme affichant 8 lignes des caractères du fichier « les_animaux_malades_de_la_pestes.txt » 15 caractères à la fois.

Créer un programme copiant ligne à ligne le fichier « les_animaux_malades_de_la_pestes.txt » dans un autre fichier texte au nom de votre choix

Séance 8

Tester les fonctions restantes de la fiche de cours et la compléter.

En fin de séance les élèves ont pu commencer à chercher à créer des fonctions :

- 1) Découper le texte en phrase.
- 2) Majuscules/Minuscules
 - a. Mettre toutes les phrases du texte en minuscule.
 - b. Remettre une majuscule au début de chaque ligne
 - c. Mettre une lettre sur deux en majuscule, une lettre sur deux en minuscule.
 - d. Mettre des majuscules uniquement au début de chaque phrase.

Pour le dernier quart d'heure : **Retour sur Turtle**

Créer des fonctions dessinant des figures géométriques

En utilisant ces fonctions , des boucles (à l'intérieur d'elles ou à l'extérieur), et des test, créez un dessin le plus beau possible (vous avez au maximum 100 lignes)

Séance 9

Partie I

1h de turtle pour présenter les projets personnels au professeur, pendant l'attente début de recherche :

Créer des fonctions :

- 1) Stylisé voir le fichier « decodage-cerveau.jpg »
 - a. Objectif : Remplacer **dans un mot** systématiquement certaines lettres par d'autres pour l'écrire de manière stylisée.
 - i. Créer une fonction transformant tous les S d'une chaîne de caractères en 5
 - ii. Généraliser en créant Transform(chaine,initial,final) une fonction transformant dans une chaîne de caractère nommé « chaine » un caractère « initial » en sa version transformée nommée « finale » . La question précédente correspondra à l'utilisation de la fonction avec la commande : Transform('S','5')
 - iii. Après avoir créé deux listes : une des lettres à remplacer, une de leurs remplacements (par exemple [S,E,...] et [5,3,...]) créer une boucle qui transformera toutes les lettres de la première liste en lettre correspondant dans la seconde.
 - b. Faire une fonction étant capable de transformer notre fable en un texte de la même forme que « decodage-cerveau.jpg »
- 2) Dyslexie. voir le fichier « dyslexie-test.jpg »

- Permuter aléatoirement toutes les lettres d'un mot
- Permuter aléatoirement les lettres d'un mot en conservant sa première et sa dernière.
- Faire la même chose à l'échelle d'un texte.

Partie II

Créer un programme copiant les 5 premières lignes du fichier « les_animaux_malades_de_la_peste.txt » dans un fichier appelé « texte_court.txt »

On a comparé les sorties de deux variations du même programme (on a juste remplacer le paramètre 'r' par le paramètre 'rb' dans la fonction open.

```
fable = open('texte_court.txt','r')
truc=fable.read()
fable.close()
print(truc)
```

ça nous donne :

```
Un mal qui répand la terreur,
Mal que le Ciel en sa fureur
Inventa pour punir les crimes de la terre,
La Peste (puisqu'il faut l'appeler par son nom),
Capable d'enrichir en un jour l'Achéron,
Faisait aux Animaux la guerre.
```

et

```
fable = open('texte_court.txt','rb')
texteBinaire=fable.read()
fable.close()
print(texteBinaire)
```

ça nous donne :

```
b'Un mal qui r\xe9pand la terreur,\r\nMal que le Ciel en sa fureur\r\nInventa pour punir les
crimes de la terre,\r\nLa Peste (puisqu'il faut l'appeler par son nom),\r\nCapable d'enrichir en un jour
l'Ach\xe9ron,\r\nFaisait aux Animaux la guerre.'
```

On remarque que les caractères accentués on été remplacés par de drôles d'écritures , par exemple « é » a été remplacé par « \xe9 » , mais ce n'est pas tout, « ' » a été remplacé par \x92 et le saut à la ligne par « \r\n » par contre toutes les lettres non accentués (autrement dit pour les caractères de l'alphabet latin classique)et les espaces sont restés inchangés.

Si au départ le codage des caractère pouvait se faire facilement et occupait peu de mémoire (parce que les caractères anglais étaient la référence) on a commencé a avoir des problèmes quand on a du intégrer les caractères venant d'autres pays, russie, asie, mais aussi les caractères accentués français, espagnols, tchèques) Chaque pays a développé ses conventions, son codage propre, mais ça génère pas mal de problème de compatibilité. Pour permettre à tout le monde de coder de la même manière, on a développé une norme commune : unicode.

Si on ajoute la ligne

```
for car in texteBinaire :
    print(car, end=' ')
```

```
on obtiendra : 85 110 32 109 97 108 32 113 117 105 32 114 233 112 97 110 100 32 108 97 32 116 101 114 114 101
117 114 44 13 10 77 97 108 32 113 117 101 32 108 101 32 67 105 101 108 32 101 110 32 115 97 32 102 117 114 101
117 114 13 10 73 110 118 101 110 116 97 32 112 111 117 114 32 112 117 110 105 114 32 108 101 115 32 99 114
105 109 101 115 32 100 101 32 108 97 32 116 101 114 114 101 44 13 10 76 97 32 80 101 115 116 101 32 40 112 117
105 115 113 117 146 105 108 32 102 97 117 116 32 108 146 97 112 112 101 108 101 114 32 112 97 114 32 115 111
110 32 110 111 109 41 44 13 10 67 97 112 97 98 108 101 32 100 146 101 110 114 105 99 104 105 114 32 101 110 32
117 110 32 106 111 117 114 32 108 146 65 99 104 233 114 111 110 44 13 10 70 97 105 115 97 105 116 32 97 117
120 32 65 110 105 109 97 117 120 32 108 97 32 103 117 101 114 114 101 46
```

ça nous permet de voir que même si tout à l'heur le print(texteBinaire) nous donnait une écriture presque lisible du texte, en fait en mémoire on a dans le fichier de type byte une suite de nombre, un par caractère du texte.

Ce que l'on voit sur l'écran et ce qu'on a stocké en mémoire sont deux choses différentes. Chaque caractère a un code.

On peut y accéder avec les fonctions chr et ord , inverses l'une de l'autre. chr(85) va me donner le caractère dont le code est 85 c'est-à-dire « U » alors que ord(« U ») me donnera 85.

Accéder au

chr(numéro) et ord(« caractère ») sont des fonctions inverses l'une de l'autre.

fin de séance (à terminer durant la séance 10)

dicté et commentaire du programme :

```
def Code(textebin,décalage) :  
    texteavantcode=bytearray(textebin)  
    texteaprescode=bytearray(b'')  
    for i in range(len(texteavantcode)) :  
        texteaprescode.append(i,texteavantcode[i]+ décalage)  
    return texteaprescode
```

après réflexion j'ai effectué quelques changements dans le programme , ce sont les zones surlignées

Séance 10

Partie I

Reprise de la fonction « Code »

La tester avec un décalage puis avec le décalage contraire pour être sûr de bien revenir sur nos pieds

Puis créer un programme qui va prendre un fichier et le crypter en utilisant le même principe

Partie II

Programmation orientée objet

Partie III

Retour sur les exercices à faire pour cette séance (voir séance 9) en parallèle début de la recherche du **projet à rendre le 13 décembre** :

Adapter l'exemple de la fonction cryptfichier(entrée,sortie,décalage) (voir plus haut) du cours pour ne pas décaler toutes les lettres de la même manière, par exemple en utilisant un mot comme clé, (chaque lettre du mot correspond à un décalage, par exemple bca correspondrait aux décalages 2,3 et 1 , et lors du codage la première lettre sera décalée de 2 la suivante de 3 la suivante de 1 , puis la suivante de 2 , puis de 3 , puis de 1 puis ...)

attention il faudra être capable de revenir en arrière (de décoder le message)