

# CRYPTOGRAPHIE : DECOUVERTE DE PYTHON

## 1. Ecriture de scripts :

Nous allons voir quelques commandes du langage de programmation Python et les utiliser pour crypter un message.

### Un premier script

Ouvrir Pyscripiter (eduPython).  
Dans la zone de saisie entrer :

```
texte= « Hello world »  
print(texte)
```

Ce petit bout de code que l'on tape dans la zone de saisie est appelé un script. Un script est donc une succession d'instructions que l'ordinateur va exécuter.

Cliquer sur l'icône *play*. Le résultat s'affiche dans la fenêtre "Python Interpreter".

1) Ecrire un script qui affiche l'addition de deux variables *a* et *b* initialisées à 2 et 5.

### Rang d'une lettre et réciproquement

Effacer le script précédent et taper le script suivant :

```
def rang(lettre):  
    return ord(lettre)-ord("A")  
  
print(" A : " , rang("A"))  
print(" B : " , rang("B"))  
print(" C : " , rang("C"))  
print(" Z : " , rang("Z"))
```

2) Exécuter le script.

**Remarque :** le mot clé *def* indique la création d'une fonction. Ici la fonction se nomme *rang* et prend comme paramètre une variable *lettre*.

3) Compléter le script pour qu'il affiche le rang des lettres *R* et *Y*.

Sauvegarder le script sur le bureau.

Compléter votre programme par le script suivant :

```
def lettre(x):  
    return chr(x+ord("A"))  
print("0: " , lettre(0))  
print("1: " , lettre(1))  
print("2: " , lettre(2))
```

4) Exécuter le script.

5) Modifier le script pour qu'il affiche la lettre de rang 12.

## Fonction affine

Compléter votre programme par le script suivant :

```
a=3
b=1

def f(x):
    return (a*x+b)%26

print("f(1)=", f(1))
print("f(2)=", f(2))
```

6) Exécuter le script.

7) Calculer  $f(9)$ ,  $f(12)$ . Comparer les résultats avec le tableau réalisé lors de la première séance de cryptographie.

Remarque : La commande %26 permet de calculer le reste de la division euclidienne par 26.

## Lire et écrire dans une fichier texte.

8) Ouvrir le bloc Note de Windows et écrire un texte en majuscule sans sauter de ligne et sans utiliser de ponctuation. Sauvegarder le fichier texte dans le même répertoire que votre programme sous le nom *monTexte.txt*.

9) Saisir le script à la suite de celui existant suivant :

```
texteRecup=open("monTexte.txt","r")
texte=texteRecup.read()
print(texte)
texteRecup.close()
```

Que contient la variable *texte* ?

3) Saisir ce script à la suite de votre programme :

```
compteur=0
for caractere in texte:
    if rang(caractere)==rang("E"):
        compteur=compteur+1
print("il y a ",compteur, "E dans le texte")
```

4) Modifier le script pour qu'il compte le nombre de « O » dans le texte.

5) Saisir ce script à la suite de votre programme :

```
texte="La MPS c'est vraiment trop bien!"
texteAEcrire = open("texteFichier.txt","w")
texteAEcrire.write(texte)
texteAEcrire.close()
```

Quel est le rôle de ce script ?

## 2. Le cryptage affine

On rappelle que pour coder un lettre L par cryptage affine, on procède en trois étapes :

- on calcule son rang  $x$  ;
- on calcule l'image  $y$  du rang  $x$  par la fonction affine  $f$  ;
- on trouve C la lettre située au rang  $y$ .

1) Compléter les `__` du script suivant pour que la fonction `crypte` réalise bien un cryptage affine associé à la fonction  $f$  déjà définie dans votre programme. Tester ensuite votre programme.

```
def crypte(L):
    x=rang( _ _ _ _ )
    y= _ _ _
    C= _ _ _
    return C

print("A->", crypte("A"))
print("B->", crypte("B"))
```

2) Saisir ce script à la suite de votre programme :

```
a=3
b=1
code=""
for L in "CRYPTOGRAPHIE":
    code=code+crypte(L)

print(code)
```

Que devient le mot CRYPTOGRAPHIE après avoir été codé ?

3) Modifier l'ensemble de votre programme pour qu'il ouvre et récupère votre texte sauvegardé dans le fichier `monTexte.txt`, qu'il crypte chaque lettre de votre texte à l'aide de la fonction affine définie par  $f(x)=4x+2$  et qu'il sauvegarde le texte crypté dans un fichier texte nommé `texteCrypte.txt`.

4) Modifier votre programme pour qu'il ne convertisse pas le caractère d'espace.