

TP : CODAGE DES CARACTÈRES

1. RAPPELS

Les caractères sont codés selon la norme utf-8 et les standards Unicode. En python, on peut ainsi utiliser les commandes suivantes :

```
>>> "\N{GREEK CAPITAL LETTER DELTA}" # utiliser la description sémantique d'un
caractère
'Δ'
>>> "\u0394" # utiliser un point de code sur 16 bits
'Δ'
>>> "\U00000394" # utiliser un point de code sur 32 bits
'Δ'
>>> ord("A") # obtenir le numéro d'ordre du caractère dans la table unicode
65
>>> ord('Δ')
916
>>> chr(65) # donne le caractère correspondant au caractère de numéro 65
'A'
>>> chr(916)
'Δ'
>>> pi="\u03c0" # codage d'un caractère par son point de code
>>> print(pi)
π
>>> ord(pi) # fonction renvoyant le numéro d'ordre d'un caractère
960
>>> chr(960) # fonction affichant le caractère correspondant à un numéro
d'ordre
'π'
```

EXERCICE 1

Compléter les deux fonctions suivantes pour lesquelles répondent à leurs docstring en n'utilisant pour le traitement des caractères que les fonctions présentées en rappel. On n'utilisera donc pas les méthodes `upper()` et `lower()` :

```
>>>"A".lower()
'a'
>>> a="M Laignel, 59000 Lille."
>>> a.upper()
'M LAIGNEL, 59000 LILLE.'
```

Fonctions à compléter :

```
def conversion_minuscules(texte):
    '''
    convertit les lettres majuscules d'un texte en minuscules
    texte : str
    return : str
    >>> a="M NSI, 59000 Lille."
    >>> conversion_minuscules(a)
    "m nsi, 59000 lille."
    '''
    texte_retour=""
    for lettre in texte:
        ordre_lettre=ord( ... )
        if ( ... <= ordre_lettre <= ... ):
            texte_retour+=chr(ordre_lettre + ... )
        else :
            texte_retour+=chr( ... )
    return texte_retour

def conversion_majuscules(texte):
    '''
    convertit les lettres majuscules d'un texte en minuscules
    texte : str
    return : str
    >>> a="M NSI, 59000 Lille."
    >>> conversion_minuscules(a)
    "M NSI, 59000 LILLE."
    '''
```

EXERCICE 2

Pour être relativement sécurisé, un mot de passe doit comporter au moins 8 caractères dont des minuscules, des majuscules, des chiffres et au moins un caractère spécial.

Compléter la fonction suivante :

```
def mdp_secure(mot):  
    '''  
    détermine si un texte contient au moins 8 caractères dont des minuscules,  
des  
majuscules, des chiffres et un caractère spécial  
return : True ou False  
>>> mdp_secure("Toto12@")  
False  
>>> mdp_secure("Toto12@fr")  
True  
>>> mdp_secure("toto12&fr")  
False  
'''
```

EXERCICE 3

On considère les commandes et retours suivants en python.

```
>>> "A"<"B"  
True  
>>> "A"<"b"<"z"  
False  
>>> "A"<"B"<"Z"  
True  
>>> "b"<"n"  
True  
>>> "A"<"a"  
True  
>>> "ba">"bc"  
False
```

D'après vous, comment Python compare deux textes avec > et < ?

EXERCICE 4

Chiffrement par décalage

Programmer une fonction permettant de coder et de décoder un texte donnée suivant le chiffrement de César : https://fr.wikipedia.org/wiki/Chiffrement_par_d%C3%A9calage