

# 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 quelles 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 les fonctions suivantes :

```

def presence_majuscules(texte):
'''
détermine si un texte contient au moins une majuscule
return : True ou False
>>> presence_majuscules("Toto12@")
True
>>> presence_majuscules("toto12@fr")
False
'''

def presence_minuscules(texte):
'''
détermine si un texte contient au moins une minuscule

```

```

return : True ou False
>>> presence_minuscules("Toto12@")
True
>>> presence_minuscules("TOTO12&")
False
'''

def presence_chiffres(texte):
    '''
    détermine si un texte contient au moins un chiffre
    return : True ou False
    >>> presence_chiffres("Toto12@")
    True
    >>> presence_chiffres("TOTO&")
    False
    '''

def presence_speciaux(texte):
    '''
    détermine si un texte contient au moins un caractère spécial
    return : True ou False
    >>> presence_speciaux("Toto12@")
    True
    >>> presence_speciaux("Aa12")
    False
    '''

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 donné suivant le chiffrement de César : [https://fr.wikipedia.org/wiki/Chiffrement\\_par\\_d%C3%A9calage](https://fr.wikipedia.org/wiki/Chiffrement_par_d%C3%A9calage)

## Exercice 5

---

La **stéganographie** est un domaine où l'on cherche à dissimuler discrètement de l'information dans un media de couverture (typiquement un [signal](#) de type texte, son, image, vidéo, etc.)

<https://fr.wikipedia.org/wiki/St%C3%A9ganographie>

Mettre au point un programme permettant de cacher un texte dans une image et de retrouver ce texte à partir de l'image originale.