11

Chaînes de caractères

◇ Le type chaîne de caractères (*string*) permet de représenter des suites de caractères, c'est à dire du texte.Manipuler du texte est très fréquent en informatique : affichage des réponses à l'écran, traitement des informations entrées au clavier et, souvent, les données textuelles constituent un format d'échange commun entre deux programmes.

```
>>> chaine1="Langage"
>>> chaine2="Python"
>>> chaine1[1],chaine1[-1]
('a', 'e')
>>> chaine1[1:5]
'anga'
>>> chaine1*2
'LangageLangage'
>>> chaine1+chaine2
'LangagePython'
>>> len(chaine1)
7
```

⚠ On retrouve des constructions semblables aux listes, cependant les chaînes ne sont pas *modifiables* : on ne peut par exemple pas écrire **chaine[3:5]=...** par contre on peut écrire :

```
>>> ch="Un texte"
>>> ch[3:]
'texte'
>>> ch=ch[0:3]+"essai"
>>> print ch
Un essai
```

◇ Deux fonctions intéressantes : **repr** convertit un objet en une chaîne de caractères (par exemple pour l'enregistrer dans un fichier) et **eval** convertit une chaîne de caractères en un objet (par exemple pour traiter des informations entrées au clavier) :

```
>>> a=1e-3
>>> type(a)
<type 'float'>
>>> ch_a=repr(a)
>>> type(ch_a)
<type 'str'>
>>> print ch_a
0.001
```

```
>>> ch_b="[1,2,[3,4]]"
>>> type(ch_b)
<type 'str'>
>>> b=eval(ch_b)
>>> type(b)
<type 'list'>
>>> print b
[1, 2, [3, 4]]
```

◇ L'opérateur `%` (appelé opérateur de *formatage*) est très pratique pour présenter des résultats. Le fonctionnement a été expliqué en cours. Un exemple récapitulatif :

```
>>> print "Int %i Float %f Autre chose %s" % (12,4e-5,[1,3])
Int 12 Float 0.000040 Autre chose [1, 3]
>>> print "Int %i Float %e Autre chose %s" % (12,4e-5,[1,3])
Int 12 Float 4.000000e-05 Autre chose [1, 3]
```

Exemple Python. Les chaînes de caractères interviennent typiquement lorsqu'il faut demander des informations à l'utilisateur par l'intermédaire du clavier. Le programme suivant demande un nombre (entier) et affiche le carré de ce nombre :

```
x=int(raw_input("Entrez un nombre : "))
print "Le carré de ce nombre est %i" % i**2
```

Noter que le programme produira une erreur si on entre, par exemple, un nombre flottant tel que 1.2.

Exemple Python. Un algorithme (naïf) de recherche d'un texte donné à l'intérieur d'une chaîne. On considère une chaîne de caractères *txt* ainsi qu'un motif *mot* (c'est également une chaîne de caractères). On veut déterminer l'ensemble des indices i pour lesquels on retrouve *mot* dans *txt* à la position i.

```
def positions(txt,mot):
    """
    Construit la liste des indices i t.q.
    mot se retrouve dans txt à la position i
    """
    Pos=[]
    for i in range(len(txt)-len(mot)+1):
        nb_egaux=0
        for j in range(len(mot)):
            if mot[j]==txt[i+j]:
                nb_egaux=nb_egaux+1
        if nb_egaux==len(mot):
            Pos.append(i)
    return Pos
txt="1101001011101"
mot="101"
Pos=positions(txt,mot)
print "%s apparait dans %s aux positions %s" % (mot,txt,Pos)
```

```
101 apparait dans 1101001011101 aux positions [1, 6, 10]
```

Complexité : si n est la longueur de *txt* et ℓ celle de *mot*, alors $T_{\texttt{positions}}(n,\ell) = O(n\ell)$.