Chaînes de caractères
Les chaînes sont entre ' ou " et les \n, \t sont toujours évalués dedans !
On peut aussi faire commencer et finir une chaîne par un triple **double** quotes """ ou simple quote ''' (permet d'inclure des retours ch
concaténation de chaînes : x = 'aaa' + 'bbb' donne aaabbb.
répétition d'une chaîne : x = 'ab' * 5 donne abababababab
Extraction de sous-chaînes :

x = 'abcdef' : définition de la chaîne.
print(x[2]) : 3ème caractère (indice commence à 0), ici c.
print(x[0:3]) : caractères d'indices 0 à 3 - 1, ici abc.
print(x[1:]) : caractères à partir de l'indice 1, ici bcdef.
print(x[:3]) : caractères jusqu'à l'indice 3 - 1, ici abc.
print(x[-2:]) : les 2 derniers caractères, ici ef.
print(x[0:-2]) : toute la chaîne sauf les 2 derniers caractères.
si l'index de fin est > à la longueur, c'est la longueur qui est utilisée.
print(len(x)) : longueur, ici 6.
x[::2] : chaîne avec un caractère sur deux (en commençant par le premier), ici ace.
x[1::2] : chaîne avec un caractère sur deux en commençant par le deuxième, ici bdf.
x[::-1] : la chaîne renversée, ici fedcba.

Les chaînes sont read-only (non mutables), donc on ne peut pas faire x[1] = 'x'
Eclater une chaîne en liste de caractères : l = list(myString)
Fonctions sur les chaînes :

'N' in s : renvoie True si N est un caractère de s.
s.count('ab') : retourne le nombre de chaîne "ab" (non overlappant).
isalnum, isalpha, isdigit, islower, isupper, isspace : tests sur le type des caractères de la chaîne (False si chaîne vide).
s.startswith('ab') : renvoie True si commence par "ab" (idem avec endswith).
s.find('ab') : retourne le plus petit index correspondant à la chaîne "ab" (-1 si pas trouvé).
s.find('ab', 3) : retourne le plus petit index correspondant à la chaîne "ab" dans s[3:] (-1 si pas trouvé).
s.find('ab', 3, 15) : retourne le plus petit index correspondant à la chaîne "ab" dans s[3:15] (-1 si pas trouvé).
s.rfind('ab') : retourne le plus grand index correspondant à la chaîne "ab" (-1 si pas trouvé), i.e commence par la recherche par la
s.index('ab') comme find, mais si sous-chaîne non trouvé, lève une ValueError.
s.lower() : renvoie la chaîne convertie en minuscules (idem avec s.upper() pour les majuscules), sans affecter s
s.capitalize() : met en majuscule la première lettre et en minuscules toutes les autres quelque soit la casse de départ.
s.title() : renvoie une chaîne où toutes les premières lettres de chaque mot sont en majuscule et le reste en minuscules.
s.replace('old', 'new') : remplace toutes les occurrences de "old" par "new", sans affecter s.
s.replace('old', 'new', 1) : remplace seulement la première occurrence.
s.translate(str.maketrans('ACGT', 'TGCA')) : renvoie la chaîne avec les A remplacés par des T, les C par des G, etc (maketrans c
s.translate(None, '\n') : enlève les retours chariots, sans affecter s.
s.translate(None, 'aeiouy') : enlève les voyelles.
str.split(s) : retourne une liste de chaînes en coupant sur les caractères blancs (espace, tabulation, retours chariot). Si s est vide,
on peut aussi faire s.split(). Par exemple, ' a b c '.split() donne ['a', 'b', 'c']. Donc attention, split se comporte différemment sans ar
' a b'.split(' ') : renvoie ['', 'a', '', 'b']
' a b'.split() : renvoie ['a', 'b'] : les chaines vides sont éliminés !
s.split('\t') : splitte sur les tabulations, et se comporte bien par défaut si à la fin de la chaîne, il y a des tabulations consécutives (c
str.split(s, 'a') : splitte sur les "a" (chaîne). Si s est vide, la liste comporte un élément, la chaîne vide.
sep.join(stringList) : concatène les chaînes stringList avec la chaîne sep comme séparateur.
s.strip() : retourne un chaîne avec les espaces à gauche et à droite retirés (trim), y compris les retours chariot (sinon, rstrip ou lstr
s.strip('a') : retourne une chaîne avec les 'a' des deux côtés enlevés (pareil avec lstrip et rstrip)
s.center(15) : retourne une chaîne de longueur 15 avec des espaces pour centrer la chaîne (si s est de longueur > 15, retourn s).
s.ljust(15) : justification à gauche (rjust pour à droite), sans affecter s
ord('A') : renvoie 65 (conversion d'un caractère en code décimal).
chr(65) : renvoie 'A' (conversion d'un code décimal en caractère, fonction inverse de la fonction ord).

Voir la section Unicode / Encodage pour les problèmes de character set.
Formattage :

'Hello %s' % 'World' : donne la chaîne Hello World
'%s a comme age %d ans' % ('Jean', 34) : donne la chaîne Jean a comme age 34 ans
utiliser %% si on veut avoir le signe %.