MPSI-PCSI

TP6 : Chaînes de caractères et fonctions

L'objectif de cette séance est de travailler sur les chaînes de caractères rencontrées lors du TP1 avec la fonction d'affichage print. Pour rappel, une chaîne de caractères est un ensemble de caractères délimité par des :

- apostrophes : 'Une première chaîne de caractères' mais 'Attention à l'apostrophe' provoquera une erreur de syntaxe.
- guillemets : "Voici une seconde chaîne de caractères"
- triples guillemets : """Voici une chaîne de caractères qui peut s'étendre sur plusieurs lignes, pratique pour commenter une fonction :"""

Les chaînes de caractères sont des données textuelles de type str (string) et il est alors possible de convertir un entier ou un flottant en chaîne de caractères à l'aide de l'instruction str (cf TP1).

Opérations élémentaires sur les chaînes de caractère

Exercice 1 :

On considère les deux chaînes de caractères suivantes s_1 = "Bon" et s_2 ="Jour", que renvoient les instructions suivantes :

a. $s_1 + s_2$
b. $s_1 * 2$
c. len(s_2)
d. $s_1[0]$=”M”

1.

$s_1 + s_2$ concatène les chaînes s_1 et s_2,
$s * p$ duplique p fois (p est un entier) la chaîne s.
len(s) donne la longueur (nombre de caractères) de la chaîne de caractères s.
Une chaîne de caractères est immuable, autrement dit il est impossible de modifier un caractère ou la longueur de la chaîne.

Longueur et parcours d'une chaîne de caractères

2.

Si s désigne une chaîne de caractères, et $k \in \mathbf{N}$, $s[k]$ permet d'accéder au caractère d'indice k de s, c'est-à-dire le $(k+1)$ème caractère, s'il existe.

Exercice 2 :

On considère la chaîne de caractères suivante s = "Bonjour", à quel caractère correspond

1. $s[0]$
2. $s[6]$
3. $s[7]$
4. $s[\text{len}(s) - 1]$
5. $s[-1]$

Maintenant que nous savons écrire des boucles, on peut les utiliser pour parcourir une chaîne *mot* caractère par caractère. Pour cela, on peut énumérer les indices des différents caractères en utilisant une boucle de la forme

$$\text{for } k \text{ in range(len}(mot))$$

où k représente la position (c'est-à-dire l'indice) du caractère parcouru, auquel on peut alors accéder via $mot[k]$. On retrouve le fait que si la chaîne de caractères est de longueur n, les caractères sont numérotés de 0 à $n-1$.

Exercice 3 :

Écrire une fonction **compteI(mot)**, qui prend en argument une chaîne de caractères et qui renvoie le nombre de lettres "I" (majuscules ou minuscules) qu'elle contient.

Par exemple, **compteI("Vive l'informatique")** doit renvoyer 3.

Exercice 4 :

Écrire une fonction **placelettre(lettre,mot)** qui étant donnés un caractère lettre et une chaîne de caractères mot, renvoie l'indice de la première rencontre du caractère lettre dans mot et renvoie -1 si le caractère n'apparaît pas dans le mot.

Par exemple, **placelettre("m","J'aime l'informatique")** renvoie 4 car le premier **'m'** est le cinquième caractère du texte, donc d'indice 4, et **placelettre("d","J'aime l'informatique")** renvoie -1 car le caractère **'d'** n'apparaît pas dans le texte.

3 (Slicing)**.**

L'opérateur [] de la question précédente permet aussi de sélectionner une sous-chaîne d'une chaîne de caractères (on appelle cette technique le slicing).

Exercice 5 :

On considère la chaîne de caractère suivante $s =$ "Vive la MPCSI", que renvoient les commandes suivantes

1. $s[2:4]$
2. $s[:5]$
3. $s[7:]$
4. $s[::2]$
5. $t = s[:7] + "" + s[9:]$

4 (On retiendra que :)**.**

$s[i:j]$ est la sous chaîne contenant les caractère $s[i]$,$\cdots$,$s[j-1]$.
$s[0:len(s):2]$ équivalent à $s[::2]$ est la sous chaîne constituée d'un caractère sur deux de s à partir du premier
$s[len(s)-1::-1]$ équivalent à $s[::-1]$ est la sous chaîne constituée des caractères de s à l'envers.

Exercice 6 :

Écrire, sans utiliser de boucle, une fonction **verifie(k,mot,texte)** qui renvoie True si la chaîne **mot** se trouve dans le**texte** exactement à l'indice **k**.

Par exemple, **verifie(0,"MPCSI","Vive la MPCSI")** renvoie False tandis que **verifie(8,"MPCSI","Vive la MPCSI")** renvoie True.

Exercice 7 :

Un palindrome est une figure de style désignant un texte ou un mot dont l'ordre des lettres reste le même qu'on le lise de gauche à droite ou de droite à gauche, comme dans la phrase : "Esope reste ici et se repose ".

Les noms communs "été", "kayak", "rotor" et "ressasser" sont également des palindromes. Écrire une fonction **palindrome(mot)** qui renvoie True si la chaîne de caractères **mot** est un palindrome, et False sinon.

On pourra astucieusement utiliser ici la technique du slicing.

Exercice 8 :

Écrire une fonction **separe(mot)** qui renvoie une nouvelle chaîne de caractères constituée des caractères de la chaîne **mot** séparés par des caractères *.

Par exemple, **separe("Python")** doit renvoyer "P*y*t*h*o*n"