

I Algorithmes

1.1 Chaînes de caractères

Une chaîne de caractère est une liste de lettres (une lettre est aussi appelée caractère). On les crée par

```
s = 'ceci est une chaîne''
s2 = 'ou avec des simples guillemets' # attention aux apostrophes !
```

1.2 Opérations

Sur les listes.

```
L1 = [2, 4, 5, 6, 7, -4]
L2 = [0, 1, 2, 3]
L1[1] # on obtient 4, le deuxième élément
len(L1) #nombre d'éléments, ici 6
L1 + L2 #concatène les listes et on obtient une nouvelle liste : [2, 4, 5, 6, 7, -4, 0, 1, 2, 3]
```

Sur les chaînes de caractères, les opérations sont les mêmes.

1.3 Slice

On peut sélectionner toute une partie d'une chaîne ou d'une liste avec la syntaxe suivante

```
s[1:4] # retourne la chaîne dont les caractères sont les
# caractères de s d'indices range(1, 4)
s[:6] # commence la sélection au début
s[3:] # sélectionne jusqu'à la fin
```

1.4 Au boulot

Créer un fichier “chaines.py” dans votre espace personnel.

1.4.1 Renverser

Ecrire une fonction qui renverse la chaîne passée en paramètre.

Donnée : une chaîne de caractère *s*

Sortie : *s* renversée.

Exemple : `reverse('chaîne')` retourne `'eniahc'`

1.4.2 Palindrome

Une chaîne est un palindrome ssi on peut la lire de gauche à droite ou de droite à gauche de la même façon. Par exemple, “radar” est un palindrome. Créer une fonction qui teste si une chaîne donnée est un palindrome.

Donnée : une chaîne de caractère *s*

Sortie : True ou False suivant que *s* est un palindrome ou non.

1.4.3 Occurrences

Créer une fonction qui renvoie le nombre d'occurrences d'un motif dans une chaîne. Par exemple le motif “aba” apparaît 2 fois dans “ababa”.

Donnée : une chaîne de caractère *s* et un motif *mot*

Sortie : Le nombre de fois où *mot* apparaît dans *s*.

1.4.4 Nombre de mots

Créer une fonction qui compte le nombre de mots dans une chaîne. Deux mots sont séparés par un ou plusieurs espaces.

Donnée : une chaîne de caractère *s*

Sortie : le nombre de mots de *s*.

Indication si *s* est une chaîne, tapez `help(s.split)`

II Lecture de fichiers

2.1 Accéder à un fichier texte

On souhaite lire le contenu du fichier “monfichier.txt”. Si le fichier n’est pas dans le répertoire courant, il faut préciser le chemin absolu d’accès au fichier (qui commence sous windows par une lettre de lecteur).

```
fichier = open('monfichier.txt')
```

Maintenant fichier est un objet python représentant un fichier. Il existe plusieurs manière d’accéder à ce fichier :

- fichier.read() retourne tout le contenu du fichier sous forme d’une chaîne de caractères.
- fichier.readlines() retourne la liste de toutes les lignes du fichier (une ligne se termine par une frappe sur Entrée, le caractère correspondant est ‘\n’)
- for ligne in fichier : est équivalent à for ligne in fichier.readlines() : (mais plus efficace d’un point de vue mémoire)

Quand vous avez fini d’utiliser le fichier (c’est à dire une fois toute les données lues) il faut exécuter

```
fichier.close()
```

pour que le fichier puisse être réutilisé par d’autres programmes (voire par le même programme...).

2.2 Wikipedia

Vous disposez d’une version modifiée de l’article Wikipedia sur Python dans le fichier “wiki-py.txt”. Les modifications apportées sont les suivantes : on a transformé toutes les lettres en une minuscule de l’alphabet.

1. Trouver le nombre de lignes dans cet articles.
2. Trouver le nombre de mots.
3. Trouver le nombre d’occurrences du mot “python”.
4. Parmi les mots de la première ligne, lequel apparaît le plus souvent dans le RESTE de l’article ? Le moins souvent ?

III Pour aller plus loin

1. Créer un fonction qui retourne le mot le plus long d’une chaîne.
2. Créer une fonction qui vérifie si un expression (donnée sous forme de chaîne) est bien parenthésée (autant de parenthèse ouvrante que fermante, avec toujours plus de parenthèse ouvertes que de fermée quand on lit de gauche à droite).
3. Créer une fonction qui renvoie le nombre de caractères minimum à retirer pour faire de la chaîne donnée un palindrome. Par exemple, il suffit de retirer le “o” de “baobab”