

Licence Culture Humaniste et Scientifique

Informatique

Feuille TD n°3

Cette séance utilise le langage Python.

Exercice 1. Echauffement

Récupérez le fichier `start.py`. Ouvrez-le avec Thonny et exécutez-le (avec F5). Expliquez les résultats obtenus.

Exercice 2. Corrections

Récupérez le fichier `erreurs.py`. Ce fichier contient un certain nombre d'erreurs très courantes. Corrigez-les !

Le but des exercices suivants est la manipulation de textes.

Exercice 3. Echauffement texte

Dans le shell, tapez les instructions suivantes :

```
s = "azerty"
s
print(s)
s = "azer\nty"
s
print(s)
print(s[1:3])
for c in s :
    print(c)
for i in range(len(s)) :
    print(s[i])
```

Expliquez les résultats obtenus.

Exercice 4. Caractère dans une chaîne

Ecrire une fonction `contient(s)` qui teste si la chaîne `s` contient le caractère 'z'.

Modifier votre résultat pour que la fonction `contient(s, c)` vérifie si la chaîne `s` contient le caractère `c` passé en paramètre.

Modifier votre résultat pour que la fonction `contient(s, c)` renvoie le nombre de fois que la chaîne `s` contient le caractère `c`.

Modifier votre résultat pour que la fonction `contient(s, c)` renvoie l'indice du premier emplacement du caractère `c` dans la chaîne `s`. Si le caractère `c` n'est pas présent, la fonction retournera la valeur -1.

Exercice 5. Sous-chaînes

Ecrire une fonction `sous_chaine(s1, s2)` qui teste si la chaîne `s2` contient la chaîne `s1`.

Modifier votre fonction pour qu'elle retourne le nombre de fois où la chaîne `s2` contient la sous-chaîne `s1`. Attention, 'aaa' contient deux fois la sous-chaîne 'aa'.

Exercice 6. Comptage

Écrire une fonction `nLignes(t)` qui compte le nombre de lignes du texte `t`.

Écrire une fonction `nmots(t)` qui compte le nombre de mots du texte `t`. On pourra être amené à utiliser les fonctions `isspace(c)` qui teste si le caractère `c` est un espace (soit ' ', soit '\t', soit '\n'), et `isalpha(c)` qui teste si `c` est une lettre de l'alphabet, majuscule ou minuscule, et éventuellement accentuée.

Pour tester les deux fonctions ci-dessus, on pourra utiliser un fichier texte `Ulysse.txt` et la suite d'instructions suivante, après avoir remplacé les '`...`' par le chemin d'accès au fichier `Ulysse.txt` :

```
import os
os.getcwd()
os.chdir('... ')

f = open('Ulysse.txt', 'r', encoding = "UTF8")
t = f.read()
print("nombre de lignes : ", nLignes(t), "nombre de mots : ", nmots(t))
f.close()
```

En déduire une fonction `compter(t)` qui pour un texte `t` compte son nombre de caractères, de mots et de lignes.

Exercice 7. *(A faire à la maison)* Palindromes

Écrire une fonction `en_minuscule(t)` qui retourne le texte `t` converti en minuscules.

Écrire une fonction `supprimer_ponct(t)` qui retourne le texte `t` dans lequel les espaces et la ponctuation ont été supprimés.

Un palindrome est un texte qui peut se lire dans un sens ou dans l'autre, si on ignore les majuscules, les espaces ou la ponctuation. Par exemple, les phrases suivantes sont des palindromes :

- Tu l'as trop écrasé, César, ce Port-Salut.
- Ésope reste ici et se repose.
- Et la marine va venir à Malte.

En utilisant les fonctions écrites dans les questions 1 et 2, écrire une fonction `est_palindrome(t)` qui teste si le texte `t` est un palindrome.