

TP n° 6 : fichiers et chaînes de caractères

Une *chaîne de caractères* (*string* en anglais) est délimitée par des guillemets doubles " ou simples ' (il faut utiliser des guillemets doubles pour une chaîne contenant une apostrophe). Par exemple :

```
chaine1 = 'Il faut travailler'  
chaine2 = "l'informatique."
```

Les opérations possibles sur les chaînes (concaténation, accès à un caractère, longueur) sont expliquées dans le cours en ligne. Par exemple

```
chaine1+" "+chaine2
```

donne

```
Il faut travailler l'informatique.
```

Elles fonctionnent à peu près comme des listes à ceci près qu'elles *ne sont pas modifiables* : on ne peut par exemple pas modifier un caractère de la chaîne `chaine1` ci-dessus en écrivant `chaine1[0] = "i"` (à tester pour voir l'erreur renvoyée). Pour ce faire, il faut utiliser la concaténation : `chaine1 = "i"+ chaine1[1:]` (remarquer la notation identique à celles des listes : `unechaine[début:fin:pas]`).

1 La Chartreuse de Parme

Le but de cette partie est de déterminer quelques informations concernant le texte de *la Chartreuse de Parme* de Stendhal à l'aide de Python.

1. Télécharger les fichiers `ChartreuseParme.txt` et `TP6.py` dans un même dossier à partir de la page de M. Schleck (pour rappel : <https://tsi-troyes-physique.netlify.com/#Info>) puis ouvrir `TP6.py` dans Pyzo.

2. Que renvoie `alph[0]` ? `alph[2]` ? `len(ALPH)` ?

3. Écrire des commandes Python pour ouvrir le fichier `ChartreuseParme.txt` en mode lecture et stocker son contenu dans une variable `chartreuse`.

Pour rappel, toutes les commandes concernant les fichiers sont résumées dans le cours en ligne : https://tsi-troyes-physique.netlify.com/info/info_tsi1#fichiers.

4. a) Écrire une fonction `occurrences(lettre, texte)` qui prend en argument une chaîne d'un caractère `lettre` et une chaîne de caractères `texte` et qui renvoie le nombre d'occurrences de `lettre` dans `texte` (on ne cherchera pas à regrouper les minuscules et les majuscules).

Par exemple `occurrences("i", "Informatique")` renvoie 1.

- b) Se servir de cette fonction pour déterminer le nombre de `F` et de `w` contenus dans *la Chartreuse de Parme*.

5. a) Écrire une fonction `nbLettres(texte)` qui prend en argument une chaîne de caractères `texte` et qui renvoie le nombre de lettres qui la composent.

Par exemple `nbLettres("Je travaille l'informatique")` renvoie 24.

- b) Combien de lettres contient le texte de *la Chartreuse de Parme* ?

6. Écrire une fonction `toutcompter(texte)` qui prend en argument une chaîne de caractères `texte` et crée un fichier `Lettres.txt` dans lequel il écrit chaque lettre et son nombre d'occurrences.

On pourra utiliser la fonction `str(objet)` qui transforme `objet` en une chaîne de caractères. Par exemple, `str(3)` renvoie `'3'`.

7. Dans Pyzo, ouvrir ce fichier en mode lecture puis afficher son contenu.

2 Problèmes de chaînes

Exercice 1.

Écrire une fonction `presentation(prenom, age)` qui prend en arguments une chaîne de caractères `prenom` et un entier `age` et qui renvoie la chaîne de caractères « Je m'appelle `prenom` et j'ai `age` ans. ». Par exemple, `presentation("Guido", 62)` renvoie "Je m'appelle Guido et j'ai 62 ans.".

Exercice 2. *Palindromes*

1. Épreuve 3.01 « Chaîne de caractères palindrome » du challenge « Chaîne de caractères palindrome ».
2. Écrire une fonction `PhrasePalindrome(chaine)` qui prend en argument une chaîne de caractères `chaine` et qui renvoie `True` si `chaine` est un palindrome à condition de ne pas tenir compte des éventuels espaces et `False` sinon. Par exemple, `PhrasePalindrome("engage le jeu que je le gagne")` renvoie `True`.

Exercice 3.

1. Épreuve 3.08 « Rechercher un mot dans un texte » du challenge.
2. Épreuve 4.03 « Caractère le plus fréquent » du challenge.
3. Épreuve 4.11 « Anagramme » du challenge.