

Boucles Imbriquées : Ctrl+F

Pour faire ce TP on utilisera Spyder (Documentation Spyder). On utilisera PythonTutor pour tester les fonctions et observer la mémoire.

On téléchargera l'archive `ressources_activite_5.zip` et on l'extraira dans le dossier de votre choix sur l'ordinateur (Extraire un fichier sous Windows 10). Vous ouvrirez dans Spyder le fichier `activite_5.py` contenu dans l'archive.

Lire la section sur les fichiers dans le cours .

1 Fichiers

Le but de cette section est d'implémenter une fonction `remplacer`, qui remplace une chaîne de caractère par une autre dans un fichier de texte.

Exercice 1. *Lecture d'un fichier*

1. Quel est le chemin absolu du fichier `activite_5.py` dans lequel vous travaillez ?
2. Quel est le chemin absolu du fichier `star_wars.txt` disponible dans le dossier `txt` du dossier de ressources extrait sur votre ordinateur ?
3. Quel est le chemin relatif du fichier `star_wars.txt` si on se situe dans le dossier de ressources extrait sur votre ordinateur ?

Implémentez les fonctions suivantes :

1. `nb_c` prend en entrée un chemin de fichier et renvoie le nombre de caractères écrits dans le fichier.
2. `nb_m` prend en entrée un chemin de fichier et renvoie le nombre de mots écrits dans le fichier.
3. `nb_l` prend en entrée un chemin de fichier et renvoie le nombre de lignes écrites dans le fichier.

On testera les fonctions sur le fichier `star_wars.txt` qui contient 15353 caractères (14939 sous linux), 2092 mots et 73 lignes.

Exercice 2. *Écriture d'un fichier*

Implémentez une fonction `voyelles` qui prend en entrée le chemin vers un fichier et renvoie le texte original où seules les voyelles sont conservées.

Vous proposerez deux implémentations :

1. Une avec un schéma de compréhension et la fonction `read`. On construira donc une liste de caractère, `v` qu'on convertira en chaîne de caractère à l'aide de l'instruction `"".join(v)`.
2. Une seconde avec une boucle imbriquée et le parcours du fichier ligne par ligne.

Dans la console Spyder, vous récupérez la sortie de la fonction `voyelles` sur le fichier `citation.txt` dans le dossier `txt`. Vous écrivez la chaîne obtenue dans un fichier `voyelles.txt` dans le dossier `txt`.

2 Rechercher

Il existe un grand nombre d'algorithmes ayant pour but rechercher de chaîne de caractères dans un texte. On se propose d'utiliser celui de la fenêtre glissante.

Pour chercher un mot de taille n dans un texte de taille m , on vérifie pour chaque position $i < m - n$ du texte si les caractères $i, i + 1, \dots, i + n - 1$ forment le mot recherché.

Exercice 3. Rechercher à une position

Implémentez une fonction `rechercher_i` qui prend trois entrées : une chaîne de caractère `s`, une chaîne de caractère `w` et un entier `i` et qui renvoie un booléen vrai si et seulement si le mot `w` est une sous-chaîne de `s` à l'indice `i` de `s`.

Testez votre fonction pour les entrées suivantes :

1. `s = "Nadine aime la grenadine", w = "dine", i = 2`
2. `s = "Nadine aime la grenadine", w = "dine", i = 20`
3. `s = "Nadine aime la grenadine", w = "dine", i = 1`
4. `s = "Nadine", w = "Nadine", i = 0`
5. `s = "Yooooohoooo !", w = "ooo", i = 1`
6. `s = "Yooooohoooo !", w = "ooo", i = 2`
7. `s = "Yooooohoooo !", w = "ooo", i = 8`

Exercice 4. Rechercher à l'aide de `recherche_i`

Implémentez une fonction `rechercher_0` qui prend trois entrées : une chaîne de caractère `s`, une chaîne de caractère `w` et un entier `d` et qui cherche le mot `w` dans le texte `s` à partir du caractère d'indice `d`. La sortie de la fonction sera l'indice le plus petit de `s` où se trouve la première lettre du mot `w` dans `s`, -1 si `w` n'est pas dans `s`.

Pour implémenter cette fonction vous utiliserez la fonction `rechercher_i` de l'exercice précédent.

Testez votre fonction pour les entrées suivantes :

1. `s = "Nadine aime la grenadine", w = "dine", d = 0`
2. `s = "Nadine aime la grenadine", w = "dine", d = 20`
3. `s = "Nadine aime la grenadine", w = "dine", d = 2`
4. `s = "Nadine", w = "Nadine", d = 0`
5. `s = "Yooooohoooo !", w = "ooo", d = 0`
6. `s = "Yooooohoooo !", w = "ooo", d = 2`
7. `s = "Yooooohoooo !", w = "ooo", d = 8`

Exercice 5. Rechercher avec boucle imbriquée

Implémentez une fonction `rechercher` qui prend trois entrées : une chaîne de caractère `s`, une chaîne de caractère `w` et un entier `d` et qui cherche le mot `w` dans le texte `s` à partir du caractère d'indice `d`. La sortie de la fonction sera l'indice le plus petit de `s` où se trouve la première lettre du mot `w` dans `s`, -1 si `w` n'est pas dans `s`.

Testez votre fonction pour les entrées données dans la question précédentes.

Exercice 6. *Rechercher tout*

En utilisant la fonction `rechercher` de la question précédente, implémentez une fonction `rechercher_tout` qui prend deux entrées : une chaîne de caractère `s` et une chaîne de caractère `w`. La sortie de la fonction sera la liste des indices de `s` où se trouve la première lettre d'un mot `w` dans `s`.

On fera attention que deux mots w ne se superpose pas. Si c'est le cas on prendra uniquement l'indice le plus petit. *e.g* pour le texte : "nananadine", si on cherche "nana" on trouvera deux positions, 0 et 2. Mais la lettre en position 2 est déjà une lettre de la première occurrence de "nana", on renverra uniquement la position 0. Cela évitera les conflits lorsqu'il s'agira de remplacer le mot "nana"

Testez votre fonction pour les entrées suivantes :

1. `s = "Nadine aime la grenadine"`, `w = "dine"`
2. `s = "Nadine"`, `w = "Nadine"`
3. `s = "Yooooohoooo !"`, `w = "ooo"`

Exercice 7. *Recherche dans un fichier*

En utilisant la console Spyder et la fonction `rechercher_tout` donnez la liste des positions où on peut trouver le mot "nadine" dans le fichier "txt/texte_interessant.txt".

3 Remplacer

Exercice 8. *Remplacer*

En utilisant la fonction `rechercher_tout` de la question précédente, implémentez une fonction `remplacer` qui prend trois entrées : une chaîne de caractère `s`, une chaîne de caractère `w` et une chaîne de caractère `v`. La sortie de la fonction sera le texte dans lequel toutes les occurrences de `w` seront remplacées par celle de `v`.

Il existe plusieurs manière de résoudre ce problème. Les indications ci-dessous ne sont pas forcément utiles pour toutes les solutions.

On pourra commencer par supposer que w et v sont de même longueurs. Pour ensuite gérer le cas où `v` et `w` sont de longueurs différentes on conservera le décalage induit sur les indices par les remplacements déjà effectués.

On rappelle qu'on peut passer d'une chaîne de caractère à une liste de caractère grâce à `list(s)`. On peut aussi passer d'une liste de caractère à une chaîne de caractère en faisant `"".join(L)`. On rappelle qu'on peut remplacer plusieurs éléments consécutifs d'une liste par une autre liste grâce aux slices : `L[12:21] = [1,2,4]`.

Testez votre code à l'aide de l'exemple suivant :

```
s8="Après un premier jeu, les enfants jouent à un nouveau jeu"
w8="jeu"
remplacer(s8,w8,"joujou")
```

Exercice 9. *Remplacer dans un fichier*

En utilisant la console Spyder et la fonction `remplacer` remplacer toutes les occurrences du mot "Force" par "Nadine" dans le fichier "txt/star_wars.txt" et écrivez un fichier "txt/nadine_wars.txt" avec le résultat.