

Chaînes de caractères

Fichiers externes

1 Chaîne de caractères

1.1 Généralité

Une chaîne de caractères peut être assimilée à une liste de caractères, à ceci près qu'elle n'est pas modifiable (famille des *tuples*). Les opérateurs de concaténation $+$ et $*$ restent valables. Les fonctions aussi.

1.2 Méthodes

Néanmoins, les méthodes ne sont pas les mêmes. Certaines permettent de modifier une chaîne de caractères. On ne présente ici que celles utiles dans le cadre de ce cours :

1. `S.isalpha()` : renvoie `True` si tous les caractères de `S` sont des lettres, `False` sinon.
2. `S.isdigit()` : renvoie `True` si tous les caractères de `S` sont des chiffres, `False` sinon
3. `S.upper()` : renvoie une chaîne en remplaçant les minuscules de `S` par des majuscules
4. `S.lower()` : renvoie une chaîne en remplaçant les majuscules de `S` par des minuscules
5. `S.find(sub)` : renvoie le plus petit indice de la chaîne de caractères `sub` recherchée dans la chaîne principale `S`. Cette fonction renvoie -1 si la recherche n'a pas abouti
6. `S.count(sub)` : renvoie le nombre d'occurrences de la chaîne de caractères `sub` dans la chaîne principale `S`
7. `S.replace(old,new)` : renvoie une copie de la chaîne de caractères en remplaçant toutes les occurrences de la chaîne `old` par `new`.
8. `S.split(sep)` : renvoie une liste de mots correspondante à `S` découpée selon le séparateur `sep`
9. `S.lstrip(car)` : renvoie une copie de la chaîne `S` où les caractères `car` situé en début de liste ont été retirés


Enfin, l'accès par indexation aux éléments d'une chaîne de caractères respectent la même syntaxe que celle pour les listes. L'affectation d'un élément de chaîne n'est pas autorisée.

1.3 Caractères spéciaux

Parmi les caractères spéciaux de la table `ASCII`, nous n'en retiendrons que 2 :

`\n` retour à la ligne *line feed*
`\t` tabulation *tabulation*

1.4 Algorithme de recherche d'un mot dans une chaîne



```
def index_of_word_in_text (mot, texte):
2  for i in range(1 + len(texte) - len(mot)):
3      j = 0
4      while j < len(mot) and mot[j] == texte[i + j]:
5          j += 1
6      if j == len(mot):
7          return i
8  return None
```

2 Fichiers textes externes

2.1 Présentation

Un fichier externe est un fichier sauvegardé sur un support de masse. Il contient des données préenregistrées selon une convention (*format*). Sous Python, il faut travailler de la façon suivante :

1. On ouvre le fichier dans un premier temps : on précise le nom du fichier avec son chemin d'accès et le mode d'accès (lecture seule ou lecture & écriture);
2. On lit ou on écrit des données dans ce fichier ;
3. On ferme le fichier une fois qu'on a terminé de travailler avec.

2.2 Ouverture et fermeture d'un fichier

Ouvrir un fichier correspond à :

- définir en mémoire une variable correspondant à un lien vers le fichier ;
- signifier au système d'exploitation que le fichier est en cours d'utilisation.

```
1 mon_fichier=open(Chemin,par)
```

où :

- **Chemin** est le nom du fichier précédé de son chemin d'accès : `U:/Python/Projet1/fichier.txt`
- **par** spécifie le mode d'ouverture (`r` pour lecture seule, `w` pour lecture et écriture)

Enfin, pour fermer le fichier, la méthode est la suivante :

```
1 mon_fichier=close()
```

2.3 Lecture et écriture du contenu textuel

La lecture se fait par méthodes, qui dépendent évidemment du contenu. Dans le cas d'un fichier texte, nous n'en retiendrons que trois :

1. `mon_fichier.read()` : renvoie la chaîne de caractères contenue dans le fichier
2. `mon_fichier.readline()` : renvoie la première ligne contenue dans le fichier. Un nouvel appel de cette méthode renvoie la seconde ligne, etc.
3. `mon_fichier.readlines()` : renvoie une liste de chaînes de caractères où chaque élément est une ligne du contenu du fichier.

L'écriture se fait par la méthode suivante, qui remplace le contenu actuel du fichier par la chaîne `s` :

```
1 mon_fichier.write(s)
```