

December 20, 2020

1 Fonctions prédéfinies pour chaînes de caractères

1.1 D'une liste de (chaînes) de caractères à une chaîne: `.join()`

Rappel: même s'il y a des points communs entre chaînes de caractères et listes (en particulier le fait qu'on puisse parcourir les deux types de données avec `for x in`), il est utile de savoir passer d'une chaîne à une liste et vice-versa.

```
[2]: t = "maison"  
     t1 = list(t)  
     print(t)  
     print(t1)
```

```
maison  
['m', 'a', 'i', 's', 'o', 'n']
```

```
[3]: news = ""  
     for lettre in t1:  
         news += lettre  
     print(news)
```

```
maison
```

Pour passer d'une liste de caractères à une chaîne, on utilise la fonction pré-définie `.join()`.

```
[4]: x = "*".join(t1)  
     print(x)
```

```
m*a*i*s*o*n
```

`s.join(l)` produit une chaîne formée par la concaténation des éléments de `l` avec `s` entre chaque élément. Si `s == ''` alors l'opération produit la concaténation de tous les éléments de la liste `l`.

```
[5]: y = "".join(t1)  
     print(y)
```

```
maison
```

1.2 Découpage d'une chaîne : fonction .split()

```
[6]: z = "voyage".split('a')
     print(z)
```

```
['voy', 'ge']
```

```
[7]: b = "la tribu prophétique aux prunelles ardentes"
     bl = b.split('au')
     print(bl)
```

```
['la tribu prophétique ', 'x prunelles ardentes']
```

1.3 Nettoyage d'une chaîne: fonction .strip()

```
[8]: a = b.strip('la')
     print(a)
```

```
tribu prophétique aux prunelles ardentes
```

Exercice: Soit la chaîne t ci-dessous. Donner la liste des caractères qui la composent, et la liste des mots qui la composent ; puis triez ces listes

```
[9]: t = "je suis un cimetière abhoré de la lune"
```

```
[10]: y = list(t)
      print(y)
```

```
['j', 'e', ' ', 's', 'u', 'i', 's', ' ', 'u', 'n', ' ', 'c', 'i', 'm', 'e', 't',
'i', 'è', 'r', 'e', ' ', 'a', 'b', 'h', 'o', 'r', 'é', ' ', 'd', 'e', ' ', 'l',
'a', ' ', 'l', 'u', 'n', 'e']
```

```
[11]: z = []
      for let in t:
          z.append(let)
      print(z)
```

```
['j', 'e', ' ', 's', 'u', 'i', 's', ' ', 'u', 'n', ' ', 'c', 'i', 'm', 'e', 't',
'i', 'è', 'r', 'e', ' ', 'a', 'b', 'h', 'o', 'r', 'é', ' ', 'd', 'e', ' ', 'l',
'a', ' ', 'l', 'u', 'n', 'e']
```

```
[12]: a = t.split()
      print(a)
      print(t)
```

```
['je', 'suis', 'un', 'cimetière', 'abhoré', 'de', 'la', 'lune']
je suis un cimetière abhoré de la lune
```

N.B. Bien faire la différence entre les fonctions `.split()` et `.strip()` et la fonction `.sort()` ci dessous. Les deux premières méthodes retournent une copie altérée de la chaîne de départ, mais la chaîne de départ est inchangée :

```
[13]: y.sort()
      print(y)
```

```
[' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', 'a', 'a', 'b', 'c', 'd', 'e', 'e', 'e', 'e',
'e', 'h', 'i', 'i', 'i', 'j', 'l', 'l', 'm', 'n', 'n', 'o', 'r', 'r', 's', 's',
't', 'u', 'u', 'u', 'è', 'é']
```

```
[14]: x1 = "première chaîne"
      x2 = "deuxième chaîne"
      x3 = x1.split('e')
      print(x3)
      print(x1) # x1 ne change pas
      x4 = x2.strip('e')
      print(x4)
      print(x2) # x2 ne change pas
```

```
['pr', 'mièr', ' chaîne', '']
première chaîne
deuxième chaîn
deuxième chaîne
```

Par contraste, la fonction `.sort()` ne renvoie pas une nouvelle liste: elle trie “sur place” la liste sur laquelle on l’appelle.

```
[15]: x = y.sort()
      print(x)
```

None

Exercice Etant donné une lettre et une liste de mots, donnez le nombre de mots qui contiennent cette lettre. Faire une fonction correctement paramétrée (et avec utilisation de `return`)

```
[16]: def contient(lettre, mot):
      return lettre in mot
      print(contient('e', 'maison'))
      print(contient('i', 'maison'))
```

False

True

```
[17]: print(a)
      c = 0
      for mot in a:
          if contient('e', mot):
              c += 1
```

```
print(c)
```

```
['je', 'suis', 'un', 'cimetière', 'abhoré', 'de', 'la', 'lune']  
4
```

```
[18]: def nb_mots_lettre(listem, lettre):  
      c = 0  
      for mot in listem:  
          if contient(lettre, mot):  
              c += 1  
      return c
```

```
[19]: print(nb_mots_lettre(a, 'i'))
```

```
2
```

2 Instructions sur les fichiers

```
[20]: f = open("demo-file-latin.txt", "r")  
      text = f.read()  
      f.close()
```

```
[21]: print(type(text))
```

```
<class 'str'>
```

```
[22]: print(len(text))
```

```
497719
```

```
[23]: print(text[:100])
```

```
JULES VERNE
```

```
CINQ SEMAINES EN BALLON
```

```
VOYAGE DE DÉCOUVERTES EN AFRIQUE PAR 3 ANGLAIS
```

```
CHAPITRE PRE
```

```
[24]: # Suite à la démo, j'ai modifié le code pour n'afficher que 20 lignes et éviter  
      ↳ de charger le notebook  
      f = open("demo-file-latin.txt", "r")  
      c = 0  
      for line in f:  
          c += 1
```

```
print(line[:10])
if c == 20:
    break
f.close()
```

JULES VERN

CINQ SEMAI

VOYAGE DE

CHAPITRE P

La fin d'u

Il y av

Ce rare

« L'Ang

```
[25]: f = open("demo-file-latin.txt", "r")
for line in f:
    if len(line) > 800:
        print(line[:25])
f.close()
```

Ils étaient là pourtan
Des toasts nombreux fu
Aussitôt mille encoura
--Pourquoi ? répondit le

```
--Non pas, mon cher Dick.  
    Un jour, après avoir r  
--Eh bien ! le voici. La  
--Enfin, reprit le docteu  
    Pendant les longues h  
    Il se trouvait alors à  
--Sans doute, mon cher Di  
    Le missionnaire était  
    Le docteur ne put rien  
--Volontiers. James Bruce  
--Oui, cette contrée est  
    Que de fois ses regard  
    « Je me croyais perdu,  
--Il y a longtemps, répon  
--C'est alors qu'un coura  
--D'ailleurs, si je ne me  
--Le voici. En 1854, un m
```

```
[26]: f = open("demo-file-latin.txt", "r")  
ll = []  
for line in f:  
    ll.append(line.strip())  
f.close()
```

```
[27]: print(ll[:10])
```

```
['JULES VERNE', '', 'CINQ SEMAINES EN BALLON', '', 'VOYAGE DE DÉCOUVERTES EN  
AFRIQUE PAR 3 ANGLAIS', '', '', 'CHAPITRE PREMIER', '', '']
```

```
[28]: f = open("demo-file-latin.txt", "r")  
lm = []  
for line in f:  
    lm.extend(line.strip().split())  
f.close()
```

```
[29]: print(lm[:25])
```

```
['JULES', 'VERNE', 'CINQ', 'SEMAINES', 'EN', 'BALLON', 'VOYAGE', 'DE',  
'DÉCOUVERTES', 'EN', 'AFRIQUE', 'PAR', '3', 'ANGLAIS', 'CHAPITRE', 'PREMIER',  
'La', 'fin', "d'un", 'discours', 'très', 'applaudi.--Présentation', 'du',  
'docteur', 'Samuel']
```