Feuille d'exercices n°5 : «Chaînes de caractères»

Premières manipulations

Exercice 1 (Fonction nombre d'occurrences) Écrire une fonction `nombreOcurrences(caractere, mot)` qui prend en argument un caractère x et une chaîne de caractère `mot` et qui renvoie le nombre de fois où le caractère x est présent dans `mot`. Par exemple, si `mot` est le mot «java», `nombreOcurrences('a', mot)` vaut 2.

Exercice 2 (Fonction premier mot) Écrire une fonction `premierMot(chaine)` qui renvoie le premier mot d'une chaîne de caractère. Par exemple si ma chaîne est «samedi soir, je vais au cinéma», on renverra «samedi».

Exercice 3 (Un peu d'unicode) Afficher les caractères dont le code unicode est compris entre les nombres hexadécimaux $(3B1)_{16}$ et $(3E1)_{16}$. Commenter.

Exercice 4 (Codage de César) Écrire une fonction `codage` qui prend en argument une chaîne de caractères et décale ses lettres de trois crans dans l'alphabet. Par exemple, «bac» sera transformée en «edf». On utilisera pour cela les fonctions `ord` et `chr` qui permettent la conversion des caractères en code ascii (ou unicode). Tester avec «zenattitude», qui doit devenir «chqdwwlwxgh» ... On écrira aussi une fonction de décodage.

Exercice 5 (Exo Google-Class) Create a function called `both_ends` : given a string s, return a string made of the first 2 and the last 2 chars of the original string, so 'spring' yields 'spng'. However, if the string length is less than 2, return instead the empty string.

Exercice 6 (Exo Google-Class) Create a function called `match_ends` : given a list of strings, return the count of the number of strings where the string length is 2 or more and the first and last chars of the string are the same.

Exercice 7 (Quelques caractères spéciaux) Afficher les textes suivants à l'aide d'une seule instruction print :

- ```
  un peu
  beaucoup
  passionnément
  ```
- `"je t'aime" dit-elle`
- `le fichier à importer est Z:\mpsi\file.txt`
- `le fichier à importer est Z:\nouveau\file.txt`
- `le fichier à importer est Z:\temp\file.txt`
  ```

Plus difficile

Exercice 8 (Écrire en verlan)

1. Écrire une fonction `verlan(mot)` qui prend en argument un mot et renvoie le mot écrit à l'envers. Par exemple si `mot` est la chaîne `'volley'`, verlan(mot) renverra `'yellov'`.
2. Généraliser au cas d'une phrase. Par exemple, si la phrase est : «je joue au volley», on devra renvoyer «ej euoj ua yellov».

Exercice 9 (Une suite sans nom) On considère la suite u suivante :

$u_0 = 1$.

$u_1 = 11$ car dans u_0 il y a un 1.

$u_2 = 21$ car dans u_1 il y a deux 1.

$u_3 = 1211$ car dans u_2 il y a un 2 et un 1.

$u_4 = 111221$ car dans u_3 il y a un 1, un 2, deux 1.

$u_5 = 312211$ car dans u_4 il y a trois 1, deux 2, un 1. Etc...

Écrire un script qui affiche les 20 premiers termes de la suite.

On pourra écrire une fonction `image(mot)` qui prend en argument une chaîne de caractères constituée seulement de chiffres parmi $0, 1, 2, 3, 4, 5, 6, 7, 8, 9$ et qui renvoie une chaîne de caractère fabriquée selon le procédé ci-dessous. Par exemple, `image('1')` renverra `'11'`, `image('111221')` renverra `'312211'`. Plus généralement `image('22255788')` renverra `'32251728'` car il y a trois 2, deux 5, un 7 et deux 8.

On rappelle que l'instruction `str(7)` convertit l'entier 7 en la chaîne de caractère `'7'`, et réciproquement, l'instruction `int('245')` convertit la chaîne `'245'` en l'entier 245.

Découverte de quelques méthodes sur les chaînes de caractères

Les méthodes suivantes sont hors-programme, mais très pratiques dès que l'on manipule des chaînes de caractères.

Exercice 10 (Méthode upper) On découvre la méthode `upper` (et `lower` sa réciproque).

1. Taper sur la console et commenter le script suivant :

```
chaine = "Je mange un kiwi. C'est bon!"
chaine.upper()
```

2. On considère `chaine = "abcdeàéèù"`. Appliquer la méthode `upper` à l'objet `chaine`, puis pour chaque caractère `c` de `chaine` calculer `ord(c.upper()) - ord(c)`. Commenter.

Exercice 11 (Quelques méthodes) Chercher de l'aide sur la documentation officielle de Python à l'adresse `http://docs.python.org`) sur :

- la méthode `strip`

- la méthode `split`
- la méthode `join`

Exercice 12 (Méthode title) Écrire une fonction `majuscule_mot` qui met en majuscules la première lettre de chaque mot d'une chaîne de caractères. Par exemple, «je mange du fromage» donnera «Je Mange Du Fromage» (on suppose que chaque mot est séparé par un espace et qu'il n'y a pas de symboles de ponctuation). On utilisera pour cela la méthode `upper`.

Remarque : votre fonction `majuscule_mot` existe déjà en Python sous forme de la méthode `title`.

Exercice 13 (Reverse words) Exercice tiré de «Google Code Jam», Africa 2010, Qualification Round. Given a list of space separated words, reverse the order of the words. Each line of text contains L letters and W words. A line will only consist of letters and space characters. There will be exactly one space character between each pair of consecutive words. Par exemple si la phrase est «J'en suis tout retourné», cela devra afficher «retourné tout suis J'en».