Les bases de Python

Fonctions, listes, chaines, dicts, sets, compréhensions

Dans ce TD, nous rafraichissons nos connaissances de Python en étudiant entre autres les fonctions, les listes et les dictionnaires

Exercice 1. Palindromes

- 1) Ecrire une fonction palindrome ayant pour argument une chaîne de caractères et qui retourne True si la chaîne est un palindrome et False sinon. Un palindrome est une chaîne qui est égale à son inverse. Proposer une solution en utilisant une itération (boucle). Ensuite vous pourrez utiliser les tranches de séquences de syntaxe s[i :j :k] qui retourne la sous-séquence de s comprise entre le ième élément compris et le jème non compris de pas k. s[: :-1] renvoie les éléments de s dans l'ordre inverse.
- 2) Ecrire une fonction `affichePalindromes(txt,n)` qui recherche et affiche les palindromes de taille n dans txt. Ces palindromes devront être exclusiment composées de caractères alphabétiques.
- conseil : utiliser la fonction split sur la chaine txt pour obtenir une liste des mots de txt (nous considererons que le séparateur est l'espace)

Exercice 2. On va construire et manipuler une liste de chaines comme par exemple la liste des prénoms de vos amis ou collègues.

- Créez une liste vide (2 méthodes)
- Ajoutez des éléments 1 par 1
- Ajoutez une autre liste à la première
- Triez la liste : tester avec la méthode sort et la fonction sorted
- Affichez la taille de la liste
- Affichez le nombre d'occurences d'un prénom
- Supprimez un élément
- Insérez un élément à un index donné
- Essayez les opérateurs + et * sur les listes
- Visualiser des exemples sur les listes et votre code sur `http://pythontutor.com`

Exercice 3. Recherche

— 1)Ecrire une fonction Python qui parcourt les éléments d'une liste et retourne un booléen indiquant si l'elément appartient ou non à celle-ci.

— 2)Modifier cette fonction pour retourner l'indice de l'élément s'il est présent (sa première occurrence) et `None` s'il n'est pas présent. Que faut-il modifier à cette fonction pour retourner l'indice de la dernière occurrence si l' élément est présent ?

Exercice 4. Compréhensions de listes

Que représentent les listes suivantes ?

```
1 x = [a for a in [1, 2, 3]]
2 x = [a for a in x if a == 2]
3 x = [a for a in range(0, 21) if a % 2 == 0]
4 x = [a * 2 for a in range(0, 11)]
```

— Ecrire une fonction de profil *impair(n)*

— En déduire à l'aide d'une compréhension la liste des nombres impairs inférieurs à n.

Exercice 5. Tri de trois éléments

— Définir maintenant une fonction prenant 3 entiers en entrée et renvoyant le Tuple trié correspondant.

Exercice 6. Les dictionnaires

— Écrivez une fonction *decompte(L)* prenant en entrée une liste de mots L et renvoyant le dictionnaire associant à chaque mot son nombre d'occurences dans L. Par exemple : `decompte([ "blanc","bleu","blanc","noir","bleu","bleu","rouge", "rouge" ])` doit renvoyer le dictionnaire :
`{'blanc': 2, 'bleu': 3, 'noir': 1, 'rouge': 2}`.

— Écrivez une fonction *abrege(L,N)* prenant en entrée une liste de mots L et un entier N et renvoyant le dictionnaire associant à chaque mot de L sa version abbrégée à N lettres. Par exemple :
abrege(["maison","immeuble","parking","hopital", "rue"], 3) doit renvoyer :
`{'hopital': 'hop.', 'immeuble': 'imm.', 'maison': 'mai.', 'parking': 'par.', 'rue': 'rue'}`

Exercice 7. Compréhensions de dicts Que représentent les dicts suivants ?

```
import string
 {c: 0 for c in string.ascii_lowercase}
 {k: v for (k, v) in zip(string.ascii_lowercase, range(26))}
```

Ecrire une compréhension donnant un dictionnaire des fréquences des lettres dans une chaine s. Comment le faire directement avec l'objet `Counter` du module `collections` ?

Exercice 8. Compréhensions d'ensembles

On peut écrire de la même façon une compréhension donnant l'ensemble des carrés des nombres entre 0 et 100. Faites de même pour les cubes des nombres entre 0 et 20. Comment en déduire l'ensembles des nombres inférieurs à 10000 quis sont à la fois des carrés et des cubes ?