

### Exo1 : Les fonctions récursives

**1)** En mathématiques, la suite de Fibonacci est une suite d'entiers dans laquelle chaque terme est la somme des deux termes qui le précèdent :

$$U_0 = 0$$

$$U_1 = 1$$

$$U_n = U_{(n-1)} + U_{(n-2)} \text{ pour tout } n \geq 2$$

Dans la suite, penser à utiliser des « assert » pour tester les pré-conditions de vos fonctions.

**1-1)** En utilisant la boucle « for », écrire une fonction affichant le terme d'ordre « n » de la suite de Fibonacci.

**1-2)** En utilisant la boucle « while », écrire une fonction retournant la liste des termes de la suite de Fibonacci jusqu'au terme d'ordre « n ».

**1-3)** Écrire une fonction récursive pour le calcul du terme d'ordre « n » de la suite de Fibonacci.

**2)** Écrire une fonction récursive pour le calcul de la factorielle d'un nombre, rappel :  $4! =$

$$4 \times 3 \times 2 \times 1 = 24.$$

**3)** Écrire une fonction récursive pour le calcul du plus grand commun diviseur entre deux entiers a et b.

**4)** Écrire une méthode récursive `pyramide_inversee(n, s)` qui écrit sur la première ligne, n fois la chaîne « s », sur la deuxième, n-1 fois la chaîne « s », et ainsi de suite jusqu'à la dernière ligne, où il y aura 1 fois la chaîne « s ». Ainsi `pyramide(3,"tot");` donnera

tottottot

tottot

tot

**5)** Écrire une méthode récursive qui calcule  $x^n$  où x et n sont des entiers :

**5-1)** En utilisant la propriété  $x^n = x * x^{n-1}$

**5-2)** En utilisant l'exponentiation rapide qui repose sur les égalités  $x^n = x^{\frac{n}{2}} * x^{\frac{n}{2}}$  ou  $x^n = x * x^{\frac{n-1}{2}} * x^{\frac{n-1}{2}}$  selon la parité de « n ».

### Exo2 : Manipulation d'un dictionnaire et liste en compréhension

**Rappel :** Les dictionnaires sont parfois présents dans d'autres langages sous le nom de « mémoires associatives » ou « tableaux associatifs ». À la différence des séquences (listes, tuples), qui sont indexées par des nombres, les dictionnaires sont indexés par des « clés », qui peuvent être de n'importe quel type immuable ; les chaînes de caractères et les nombres peuvent toujours être des clés. Un dictionnaire est constitué d'un ensemble d'item « clé : valeur », la clé doit être unique.

**1)** Créer un dictionnaire vide. Ajoutez-y les items « 'nom' : 'toto' », « 'prénom' : 'titi' », « 'age':20 », « 'adresse':404 ».

**2)** Modifier puis supprimer l'âge.

**3)** Écrire une instruction pour récupérer la valeur 'toto' à partir de la clé correspondante.

**4)** Écrire une instruction qui vérifie l'existence de la clé 'prénom' et 'age' dans le dictionnaire.

**5)** Écrire une instruction pour récupérer toutes les clés. Puis une deuxième instruction pour récupérer toutes les valeurs. Puis une troisième pour récupérer tous les items.

**6)** En utilisant les listes en compréhension, créez une liste de nombre entiers aléatoire de 0 à 10. Puis une deuxième liste d'entiers de -10 à -1.

**7)** Créer un nouveau dictionnaire, dans lequel vous associez la clé « positif » à liste des entiers positifs. Même traitement pour les entiers négatifs.

**8)** Créer une fonction qui prend en paramètre le dictionnaire de la question 7). La fonction tire un entier aléatoire entre -5 et 5, puis ajoute la valeur aléatoire dans le dictionnaire selon le signe de celle-ci. Exemple -2 sera ajouté à la liste des entiers négatifs dans le dictionnaire.

9) Écrire une fonction qui élève au carré les éléments positifs dans le dictionnaire précédant, puis qui élève au cube les éléments négatifs. Utilisez les listes en compréhension.

### Exo3 : Utilisation d'un dictionnaire

1) Écrire une fonction « occurrence(chaine) » qui construit un dictionnaire dans lequel les clés sont les caractères du mot « chaîne » et les valeurs sont les occurrences de chaque caractère dans le « mot ».

2-1) Saisir la liste de mots suivante :

mots = ['edward', 'catelyn', 'robb', 'sansa', 'arya', 'brandon', 'rickon', 'theon', 'rorbert', 'cersei', 'tywin', 'jaime', 'tyrion', 'shae', 'bronn', 'lancel', 'joffrey', 'sandor', 'varys', 'renly']

2-2) Écrire une fonction « mots\_position1(liste, let, pos) » qui prend en paramètre une liste de mots, une lettre et une position. La fonction retourne la liste de mots dans « liste » possédant la lettre « let » à la position « pos ».

2-3) Écrire une fonction « mots\_position2(liste, let, pos) » qui fait le même traitement que « mots\_position1 » mais cette fois-ci en compréhension de liste.

3-1) Écrire une fonction « dictionnaire\_new(liste) » qui construit un dictionnaire dont les clés sont de la forme (pos,let) et la valeur est la liste des mots ayant la lettre « let » à la position « pos ».

3-2) Écrire une fonction « mots\_position3(dico, let, pos) » qui prend en paramètre le dictionnaire de la question 3-1), une lettre « let » et une position « pos ». La fonction retourne la liste des mots dans « dico » possédant la lettre « let » à la position « pos ».

### Exo4 : Slicing, listes et dictionnaire en compréhension

1) L'administration cherche à lister un groupe de 15 élèves dans un TD, quel est la meilleure structure de donnée pour les regrouper ? Générer une liste de 15 élèves ayant chacun une lettre distincte.

2) Importer la librairie « string » et utiliser la variable `ascii_lowercase` qui liste l'ensemble des lettres minuscules. Comment obtenir les 15 premiers éléments à partir d'une notation type slice ?

3) On les divise en 5 sous-groupe pour un projet, comment peut-on adapter la structure de données pour organiser les élèves en sous-groupes ? Commencer par diviser les étudiants par groupes de 5 consécutifs. Importer la librairie « random ». Utiliser la fonction « choice » pour développer une fonction qui prend la liste originale et qui renvoie 5 groupes aléatoires. Utiliser la fonction « shuffle » pour organiser les groupes plus simplement.

4) Réorganiser la liste d'étudiants en 5 sous-groupes dans un dictionnaire en réutilisant la variable originale.

5) On veut maintenant réorganiser le dictionnaire pour assigner une note à chacun des étudiants du sous-groupe. Le nouveau dictionnaire contient le numéro de groupe comme clé et la valeur correspond à un dictionnaire {liste\_groupe : liste\_notes}.

6) Ajouter une clé 'moyenne' à chaque dictionnaire contenant les notes. Comment arrondir la moyenne à deux décimales ?

7) Ecrire les instructions permettant de mélanger les items du dictionnaire construit dans la question 6) (pensez à appliquer un « shuffle » sur les clés).

8) Trouver un moyen de déterminer le groupe avec la moyenne minimale, maximale ?