

Exercice : Listes

Exercice 1

La suite de Fibonacci est une suite dont chaque valeur s'obtient en ajoutant les deux valeurs qui la précèdent, avec les nombres 0 et 1 comme valeurs initiales.

Soit `t` la liste python des premiers nombres de la suite de Fibonacci :

$$t = [0, 1, 1, 2, 3, 5, 8, 13, 21]$$

- 1) On saisit l'instruction `len(t)`. Quelle est la valeur renvoyée.
- 2) Comment peut-on afficher la dernière valeur de la liste ? Proposer plusieurs façons d'y parvenir.
- 3) Quelle est la valeur `t[5]` ? et `t[8]` ? Quelle sera la douzième valeur de cette liste si on la complète.
- 4) On souhaite compléter la liste des nombres de Fibonacci. Écrire la fonction **terme_suivant** qui prend en paramètre la liste des nombres de la suite de Fibonacci et ajoute le terme suivant en fin de liste.
- 5) La fonction `n_suivant(liste,n)` a pour paramètres une liste de nombres et un nombre entier `n`. Écrire cette fonction pour quelle ajoute `n` nombres supplémentaires à la liste de Fibonacci.

Exercice 2

- 1) Créer, en Python, une liste **panier** contenant les fruits **pomme**, **poire**, **prune** et **orange** où chaque fruit est une chaîne de caractères.
- 2) Écrire une fonction **dans_panier** qui prend en paramètre le panier de fruits et un fruit et renvoie un booléen qui affirme si le fruit est dans le panier ou non.
- 3) Écrire une fonction **ajoute_au_panier** qui prend en paramètre la liste de fruits et un fruit pour l'ajouter à la liste **panier** si celui-ci n'y est pas.
- 4) Écrire une fonction **supprime_du_panier** qui prend en paramètre la liste de fruits et un fruit pour le supprimer de la liste **panier** si celui-ci y est.
- 5) Créer une fonction **mélange** qui prend en paramètre la liste de fruits et mélange aléatoirement les fruits de la liste. Pour cette fonction, on pourra utiliser une fonction aléatoire du module **random**.
- 6) Pour vérifier si un fruit est dans le panier, on parcourt toute la liste. Est-il possible d'effectuer cette vérification sans parcourir toute la liste ?

Exercice 3

- 1) Écrire une fonction **liste_aleatoire()** qui renvoie une liste de 10 nombres entiers compris entre 1 et 50 choisis aléatoirement.
Créer la liste **alea** en appelant la fonction **liste_aleatoire()**.
- 2) Créer une fonction **max_liste** qui prend en paramètre une liste et qui renvoie son plus grand élément et sa position dans la liste.
- 3) Créer une fonction **dans_liste** qui prend en paramètres un nombre entier et une liste de nombres et renvoie **True** si le nombre est présent dans la liste et **False** dans le cas contraire.
- 4) Créer la fonction **decale** qui prend en paramètre la liste de nombres et la modifie en décalant toutes les nombres de 1 rang vers la droite.
La première position, laissée libre, sera occupée par la dernière valeur éjectée de la liste.