

Corrigé sujet 21 - Année : 2023

[Sujet 21 - 2022](#) ↴

1. Exercice 1

Script Python

```

1 def delta(liste):
2     codage=[liste[0]]
3     for i in range(1,len(liste)):
4         codage.append(liste[i]-liste[i-1])
5     return codage

```

Commentaires

- On construit le codage en partant du premier élément de la liste. Les autres éléments sont les différences entre deux éléments consécutifs de la liste de départ.
- L'écriture de cette fonction peut aussi se faire (de façon plus concise) en utilisant les listes par compréhension :

Script Python

```

def delta(liste):
    return [liste[i]-liste[i-1] if i>0 else liste[i] for i in
range(len(liste))]

```

2. Exercice 2

Script Python

```

1 class Noeud:
2     """
3         classe implémentant un noeud d'arbre binaire
4     """
5
6     def __init__(self, g, v, d):
7         """
8             un objet Noeud possède 3 attributs :
9             - gauche : le sous-arbre gauche,
10                - valeur : la valeur de l'étiquette,

```

```

11     - droit : le sous-arbre droit.
12     """
13     self.gauche = g
14     self.valeur = v
15     self.droit = d
16
17     def __str__(self):
18         """
19         renvoie la représentation du noeud en chaîne de caractères
20         """
21         return str(self.valeur)
22
23     def est_une_feuille(self):
24         """
25         renvoie True si et seulement si le noeud est une feuille
26         """
27         return self.gauche is None and self.droit is None
28
29
30     def expression_infixe(e):
31         s = "" #(1)
32         if e.gauche is not None: #(2)
33             s = '(' + s + expression_infixe(e.gauche)
34         s = s + str(e.valeur)
35         if e.droit is not None: #(3)
36             s = s + expression_infixe(e.droit) + ")"
37         return s

```

1. La variable `s` va contenir l'expression arithmétique
2. Si le noeud contient un fils gauche, on construit l'expression associée et on ajoute la valeur du noeud à la suite.
3. On construit la partie droite de l'expression (si elle existe)
4. Si on atteint cette ligne, l'expression a été construite en totalité, il reste à la renvoyer. Il ne devrait pas y avoir de `if !`

Attention

Sujet difficile avec diverses notions du programme (arbre, récursivité) et qui présente de plus un aspect mathématique.