



TP n° 5
Programmation orientée objet



On rappelle qu'un *polynôme formel* (ou simplement *polynôme*) est une suite $(a_n)_{n \in \mathbb{N}}$ de scalaires nuls à partir d'un certain rang. On distingue donc les *polynômes* des *fonctions polynomiales*.

Le polynôme $X^3 + 2X - 5$ pour donc être formalisé par la liste $[-5, 2, 0, 1]$ tout comme la liste $[-5, 2, 0, 1, 0, 0]$.

1. Créer une fonction `meme_poly(L,M)` qui renvoie `true` si les listes L et M représentent le même polynôme, et `false` sinon.
2. Créer une classe `Polynome` dont les objets seront des listes de flottants (des polynômes formels, donc). Programmer la méthode spéciale `__init__` de telle façon à ce que l'attribut d'un polynôme soit une liste de coefficients.
3. Adapter la fonction de la question 1 pour qu'elle devienne une méthode de la classe `Polynome`.
4. Programmer une méthode `deg` qui renvoie le degré d'un polynôme.
5. Surcharger¹ la méthode spéciale `__add__` pour que l'appelle
`>>> P+Q`
renvoie la somme des polynômes P et Q (supposés instanciés). Tester votre méthode sur des exemples.
6. Créer la classe `FracRat` des fractions rationnelles afin qu'elle hérite des méthodes de la classe `Polynome`, puis surcharger les méthodes `deg` et `__add__` afin qu'elles soient utilisables dans la classe `FracRat`.
7. On souhaiterait représenter un polynôme formel tel que $[3, -2, 0, 1, 0, 0]$ comme on le fait habituellement en mathématiques c'est-à-dire par l'affichage de la chaîne de caractères

$$3 + (-2)X + X^3.$$

On notera que

- les parenthèses sont souhaitées autour des flottants négatifs, pour ne pas obtenir d'expressions désagréables de la forme « $3 + -2X$ ».
- les termes de coefficient nul ne doivent pas s'afficher (il n'y a pas de terme en X^2).
- on écrit X et non X^1 et on n'écrit pas X^0 .
- on écrit X^3 et non $1X^3$.

Surcharger la méthode spéciale `__repr__` afin d'obtenir ce que l'on souhaite.

1. C'est-à-dire reprogrammer.