

Exercice 1 : En Python, la fonction `__init__` est nommée *constructeur*. Elle est appelée automatiquement lors de l'instanciation de l'objet.

En pratique elle est réalisée par la composition de deux opérations : l'allocation et l'initialisation. L'allocation consiste à réserver un espace mémoire au nouvel objet. L'initialisation consiste à fixer l'état du nouvel objet.

Il existe un certain nombre d'autres fonctions particulières, chacune avec un nom prédéfini et entourée d'une paire de `_`. Tous les opérateurs classiques de Python peuvent être ainsi redéfinis pour un objet :

- `__eq__` définit l'opérateur `==`
- `__lt__` définit l'opérateur `<`
- `__contains__` définit l'opérateur `in`

`__le__` `<=`, `__gt__` `>`, `__ge__` `>=`, `__ne__` `!=`

1. Définir une classe *Loto*. Cette classe possède un attribut *numeros* de type *list* qui contiendra les 6 numéros du tirage du loto et un attribut *complementaire* qui désignera le numéro complémentaire.
2. Définir la méthode `__str__` qui renverra une chaîne de caractères des numéros du loto.
3. Définir la méthode `__contains__` qui vérifiera si le numéro donné en argument est dans les 6 numéros du loto.
4. Définir la méthode *est_gagnant* qui possède deux arguments : une liste d'entiers et un entier. Elle renverra `True` si le tirage correspond exactement à la proposition.

Exercice 2 : On définit une classe *Fraction* pour représenter un nombre rationnel. Cette classe possède deux attributs *numérateur* et *dénominateur*. Le dénominateur doit être strictement positif.

1. Écrire le constructeur de cette classe. Il doit lever une `ValueError` si le dénominateur n'est pas strictement positif.
2. Définir la méthode `__str__` qui renvoie une chaîne de caractère de la forme `"12 / 7"` ou simplement `"12"` si le dénominateur est égal à 1.
3. Définir les méthodes `__eq__` et `__lt__` qui reçoivent une deuxième fraction en argument et renvoient `True` si la première fraction représente respectivement un nombre égal ou strictement inférieur à la deuxième fraction.
4. Définir les méthodes `__add__` et `__mul__` qui reçoivent une deuxième fraction en argument et renvoient une nouvelle fraction représentant respectivement la somme et le produit des deux fractions.

Exercice 3 : On définit une classe *Date* pour représenter une date avec trois attributs *jour*, *mois* et *annee*.

1. Écrire son constructeur.
2. Définir la méthode `__str__` qui renvoie une chaîne de la forme `"8 mai 1945"`.
3. Définir la méthode `__lt__` qui permet de comparer deux dates.
4. Tester ces méthodes.