

PRG. POO. Exercices. Corrigé

Exercice 1 :

```
class Fraction :
    def __init__(self,num,den) :
        if den == 0 :
            print("le dénominateur ne peut pas être nul")

        self.numérateur = num
        self.dénominateur = den

    def affiche(self) :
        if self.dénominateur == 1 :
            print(self.numérateur)
        else : # Utilisation du formatage pour l'affichage
            print(f"{self.numérateur} / {self.dénominateur}")

    def __eq__(self,frac) : # Renvoie 'True' si égalité
        return self.numérateur*frac.dénominateur == self.dénominateur*frac.numérateur

    def __lt__(self,frac) :
        val = self.numérateur*frac.dénominateur - self.dénominateur*frac.numérateur
        if val < 0 :
            return True
        else :
            return False
```

Jeu de tests :

```
# f1 = f2
f1 = Fraction(2,3)
f2 = Fraction(4,6)

# f3 < f1 et f4 > f1
f3 = Fraction(2,5)
f4 = Fraction(3,4)

f1.affiche() # Affiche 2 / 3
print(f1 == f2) # 'True' attendu
print(f1 == f3) # 'False' attendu
print(f1 < f3) # 'False' attendu
print(f1 < f4) # 'True' attendu

fnul = Fraction(6,0) # Message attendu

2 / 3
True
False
False
True
le dénominateur ne peut pas être nul
```

Remarque : on verra une bien meilleure façon de prendre en charge les erreurs (comme ici le dénominateur nul).

Exercice 2 :

```
class Date :
    def __init__(self,jo,mo,an) :
        self.annee = an
        self.mois = mo
        self.jour = jo

    def affiche(self) :
        print(f"{self.jour} {self.mois} {self.annee}")

    def __lt__(self,dat) :
        # Comparaison sur Les années
        if self.annee < dat.annee :
            return True
        if self.annee > dat.annee :
            return False

        # Si on arrive ici, c'est que Les années sont Les mêmes
        # on compare donc sur Les mois
        if mois[self.mois] < mois[dat.mois] :
            return True
        if mois[self.mois] > mois[dat.mois] :
            return False

        # Si on arrive ici, c'est que Les années et Les mois sont Les mêmes
        # on compare donc sur Les jours
        if self.jour < dat.jour :
            return True
        else :
            return False # On prend aussi le cas de L'égalité des dates

# Dictionnaire de mois
mois = { "janvier" : 1, "février" : 2, "mars" : 3, "avril" : 4, "mai" : 5, \
        "juin" : 6, "juillet" : 7, "août" : 8, "septembre" : 9, \
        "octobre" : 10, "novembre" : 11, "décembre" : 12 }

# Jeu de tests
date1 = Date(21,"mars",2020)
date2 = Date(2,"avril",2021)
date3 = Date(2,"avril",2000)
date4 = Date(2,"mai",2021)
date5 = Date(1,"avril",2021)
date1.affiche() # Affiche la date
print(date2 < date1) # 'False' attendu
print(date2 < date2) # 'False' attendu
print(date2 < date3) # 'False' attendu
print(date2 < date4) # 'True' attendu
print(date5 < date2) # 'True' attendu
```

```
21 mars 2020
False
False
False
True
True
```

Remarque : dans le dictionnaire *mois*, les antislashs « \ » en fin de ligne permettent un retour à la ligne (pratique).