

Implémenter des conceptions simples

```
def trier_aliments():
    # Liste d'aliments avec leurs dates de péremption
    aliments = [("Lait", 5), ("Yaourt", 2)]

def trier_aliments_par_urgence(aliments):
    def calculer_urgence(aliment):
        jours_restants = (aliment['date_peremption'] - datetime.now()).days # Calculer Le nombre de jours avant expiration
        return (jours_restants * 100) + (100 - aliment['quantite']) # On multiplie par 100 pour éviter les problèmes de précision

    aliments_tries = sorted(aliments, key=calculer_urgence)
    return aliments_tries

# Exemple d'utilisation
if __name__ == "__main__":
    from datetime import datetime, timedelta

    # Exemple de Liste d'aliments
    aliments = [
        {'nom': 'Lait', 'date_peremption': datetime.now() + timedelta(days=2), 'quantite': 20},
        {'nom': 'Pain', 'date_peremption': datetime.now() + timedelta(days=5), 'quantite': 15},
        {'nom': 'Yaourt', 'date_peremption': datetime.now() + timedelta(days=1), 'quantite': 30},
        {'nom': 'Fromage', 'date_peremption': datetime.now() + timedelta(days=7), 'quantite': 10}
    ]

    # Trier Les aliments
    aliments_tries = trier_aliments_par_urgence(aliments)

    # Afficher Le résultat
    print("\nAliments triés par urgence (du plus urgent au moins urgent):")
    for aliment in aliments_tries:
        jours_restants = (aliment['date_peremption'] - datetime.now()).days
        print(f"{aliment['nom']}: {jours_restants} jours restants, {aliment['quantite']}% restant")
```

```
Aliments triés par urgence (du plus urgent au moins urgent):
Yaourt: 1 jours restants, 30% restant
Lait: 2 jours restants, 20% restant
Pain: 5 jours restants, 15% restant
Fromage: 7 jours restants, 10% restant
```

Je suis capable d'implémenter des conceptions simples. Lors des cours de « Tutorat » avec les étudiants de 3^{ème} année, nous avons codé un petit script simple en Python qui permet de trier une liste d'aliments selon leur urgence de consommation (qui va bientôt périmer).