

# TP1 - Le problème du rendu de monnaie

Le but de ce TP est de programmer un appareil qui rend automatiquement la monnaie avec des pièces.  
Pour ce faire, on aura recours à l'algorithme de rendu de monnaie vu la séance précédente.

On considère dans les parties I et II que l'on dispose d'autant de pièces que l'on veut.

## I Une première approche

On se propose de coder en Python l'algorithme de rendu de monnaie vu la séance précédente. Pour faciliter les calculs, on va exprimer les valeurs de toutes les pièces en centimes que l'on stockera dans une liste.

1. Ecrire sous forme d'une liste nommée **caisse** contenant les valeurs par ordre décroissant.
2. Ecrire une fonction **rendmonnaie** qui prend en paramètre la somme à rendre **s** en € et une caisse **c** de type list et qui renvoie une liste contenant la quantité de monnaie à rendre pour chaque valeur.  
Par exemple, `rendmonnaie(2.34, caisse)` doit afficher : `[1, 0, 0, 1, 1, 0, 2, 0]`  
où `[1, 0, 0, 1, 1, 0, 2, 0]` indique que l'on a rendu une pièce de 2 €, une pièce de 20 centimes, une pièce de 10 centimes et deux pièces de 2 centimes.

## II Une première amélioration

Ecrire une fonction **rendmonnaie2** qui prend en paramètre la somme à rendre **s** en € et une caisse **c** de type list et qui renvoie une liste constituée par les couples [pièces, quantité rendue]

Par exemple, `rendmonnaie2(2.34, caisse)` doit afficher : `[[200, 1], [100, 0], [50, 0], [20, 1], [10, 1], [5, 0], [2, 2], [1, 0]]`

A partir de la partie III, on disposera pour chaque pièce d'une certaine quantité.

## III Avec un dictionnaire

1. On représente maintenant le contenu contenu dans le monnayeur par un dictionnaire :
  - la valeur faciale de la pièce comme cle
  - le nombre de pièces de cette valeur contenue dans le monnayeur

Créer un dictionnaire nommé **caisse2** répondant à ces conditions.

La quantité prise par la pièce sera générée aléatoirement avec le module random.

2. Ecrire une fonction **rendmonnaie3** qui prend en paramètre la somme à rendre **s** en € et une caisse **c** de type dictionnaire et qui renvoie un dictionnaire constituée par les couples [pièces, quantité rendue] si c'est possible et le message 'il n'y a plus de monnaie disponible' si le rendu est impossible  
Par exemple, `rendmonnaie3(2.34, caisse2)` doit afficher :  
`{200 : 1, 100 : 0, 50 : 0, 20 : 1, 10 : 1, 5 : 0, 2 : 2, 1 : 0}`  
si la caisse avait la répartition `{200 : 6, 100 : 0, 50 : 1, 20 : 9, 10 : 1, 5 : 8, 2 : 9, 1 : 6}`  
il n'y a plus assez de monnaie si la caisse avait la répartition suivante `{200 : 0, 100 : 1, 50 : 1, 20 : 1, 10 : 2, 5 : 2, 2 : 0, 1 : 1}`.

Pour aller plus loin ...

Faire l'exercice 4 de la feuille d'exercices