

# Quizz\_2022\_09

April 1, 2022

On veut “crypter” une chaîne de caractères en remplaçant chacun de ses caractères par un autre. Pour cela, on suppose que l’on dispose d’une liste de “correspondances” : c’est une liste de paires de la forme (a,b) où a est le caractère initial et b le caractère de remplacement. Écrire une fonction `cryptage()` qui prend en entrée une chaîne de caractères et une liste de correspondances, et renvoie (par `return`) la chaîne cryptée.

```
[1]: def cryptage(s,c):
      new_s = ""
      for lettre in s:
          for (i,o) in c:
              lettre_conv = lettre
              if lettre == i:
                  lettre_conv = o
                  break
          new_s += lettre_conv
      return new_s
```

```
[2]: C = [('a', '0'), ('c', '-'), ('d', 'P'), ('e', '9'), ('i', '*'), ('s', 'x'), ('_
      →', 'm'), ('u', ' ')]
      new_s = cryptage("Sois sage ô ma douleur", C)
      print(new_s)
```

So\*xmx0g9mômm0mPo 19 r

Pour illustrer le principe de programmation modulaire, voici une version où la partie “recherche de correspondance” est séparée de la partie “construction de la chaîne résultante”. L’intérêt de procéder ainsi est que les deux fonctions peuvent être testées indépendamment l’une de l’autre, et que la lisibilité est augmentée.

```
[9]: def correspondance(l,C):
      for (i,o) in C:
          if l == i:
              return o
      return l
      def cryptage_mod(s,c):
          new_s = ""
          for lettre in s:
              new_s += correspondance(lettre,c)
          return new_s
```