

Liste des combinaisons d'une chaîne de n caractères pris k à k avec répétition

On va simplement utiliser la fonction précédente:

```
def combinaisonsreps(n,k):
    return [''.join(s) for s in combinations_with_replacement(list(chr(i)) for i in range(n)),k)]
```

Exemple:

```
ch = '123'
print combinaisonsreps(ch,2)
['11', '12', '13', '22', '23', '33']
```

Combinaisons de n objets non tous distincts pris k à k avec ou sans répétition

Ici, on va généraliser tout ce qu'on a vu jusqu'à présent, pour faire à peu près tout ce qu'on veut!

Par exemple, on a [1,2,3,4,4,4,5], et on va chercher les combinaisons de ces objets pris 3 à 3.

On voit qu'on a 8 objets, avec 5 objets distincts (le 2 apparaît 2 fois, et le 4 apparaît 3 fois).

Le fait que certains objets se retrouvent à plusieurs exemplaires va entraîner 2 conséquences dans la liste des résultats:

- présence de répétitions d'éléments dans chaque séquence. Exemple: [1,2,2]
- présence de répétition de séquences dans la liste. Exemple: [2,4,5] et [2,4,5]

On va donc créer du code pour, selon le problème à résoudre, supprimer des résultats l'une ou l'autre de ces répétitions (ou les 2).

Suppression de la liste résultat des séquences comportant des répétitions d'éléments (exemple: [1,2,2])

```
def sansrepetitione(s):
    """Supprime de la liste p toutes les séquences avec des doublons comme [1,2,2]"""
    r = []
    for seq in p:
        for elem in seq:
            ok = True
            if seq.count(elem)>1:
                ok = False
                break
            if ok:
                r.append(seq)
    return r
```

Exemple:

```
p = [[1,2,3],[1,2,3],[1,3,1],[1,3,2]]
print sansrepetitione(p)
[[1], [2], [3]]
```

Dans la liste résultat, suppression des séquences répétées de sorte que chaque séquence n'apparaisse qu'une seule fois (exemple: [[1,2,4],[1,2,4]] -> [[1,2,4]])

```
def sansrepetition(s):
    """Supprime de la liste p toutes les répétitions de séquences comme [1,2,4] et [1,2,4]"""
    r = []
    for seq in p:
        if seq not in r:
            r.append(seq)
    return r
```

Exemple:

```
p = [[1,2,4],[1,2,3],[1,3,2],[1,3,4]]
print sansrepetition(s)
[[1], [2], [4], [1,2,3], [3]]
```

On peut même supprimer les 2 type de répétitions d'un seul coup:

```
def sansrep(s):
    return sansrepetition(sansrepetitione(s))
```

On peut aussi vouloir calculer le nombre d'éléments distincts d'une séquence:

```
def elandiff(seq):
    """Récupère le nombre d'éléments différents de la liste seq"""
    r = []
    for elem in seq:
        if elem not in r:
            r.append(elem)
    return len(r)
```

Reprenons maintenant notre exemple plus haut: seq=[1,2,2,3,4,4,4,5] et cherchons les combinaisons de ces objets non tous distincts, pris 3 à 3:

(à terminer)

Amusez-vous bien!