

Fusionner des tables

Réunion (ou Concaténation)

Reprenons notre table de pays, pour y ajouter 2 pays, l'Espagne et le Portugal.

```
>>> pays
[['France', 'Paris', '68'],
 ['Espagne', 'Madrid', '48'],
 ['Italie', 'Rome', '60']]
```

Et un tableau de nouveaux pays.

```
>>> nouveau_pays = [['Allemagne', 'Berlin', '82'], ['Portugal', 'Lisbonne', None]]
```

Remarque : Pour un tableau de tableaux ou de p-uplets, il faut veiller à ce que les données sur une même colonne correspondent à leur descripteur. S'il manque une donnée, on complète avec '' ou None¹.

On peut simplement utiliser l'opérateur + qui crée un nouveau tableau :

```
>>> pays + nouveau_pays
[['France', 'Paris', '68'],
 ['Espagne', 'Madrid', '48'],
 ['Italie', 'Rome', '60'],
 ['Allemagne', 'Berlin', '82'],
 ['Portugal', 'Lisbonne', None]]
```

Ou encore la méthode .extend() qui modifie la variable pays :

```
>>> pays.extend(nouveau_pays)
>>> pays
[['France', 'Paris', '68'], ['Espagne', 'Madrid', '48'], ['Italie', 'Rome', '60'],
 ['Allemagne', 'Berlin', '82'], ['Portugal', 'Lisbonne', None]]
```

Si on ne veut ajouter qu'un seul élément (= un seul tableau pour un pays), on peut utiliser .append() :

```
>>> pays.append(['Belgique', 'Bruxelles', '11'])
```

⚠ Attention, la concaténation peut mener à des doublons dans une table ou des incohérences :

```
>>> pays.append(['Pays-Bas', '17', 'Amsterdam'])
>>> pays
[['France', 'Paris', '68'],
 ['Espagne', 'Madrid', '48'],
 ['Italie', 'Rome', '60'],
 ['Allemagne', 'Berlin', '82'],
 ['Portugal', 'Lisbonne', None],
 ['Pays-Bas', '17', 'Amsterdam']]
```

Note: On peut supprimer la ligne avec `del(pays[6])` et pour éviter les doublons créés par `pays + pays_sud` on préférera faire :

```
>>> pays_sud = [['Espagne', 'Madrid', '47'], ['Portugal', 'Lisbonne', None], ['Italie', 'Rome', '60'], ['Grece', 'Athenes', 10]]
>>> pays = pays + [p for p in pays_sud if p not in pays]
```

Et éviter des incohérences par exemple en vérifiant les **domaines** de valeur. Pour les pays, on peut vérifier que les données de population contiennent bien des chiffres.

```
>>> pays.extend([p for p in pays_sud if p[2].isdigit()])
```

Fusion (ou Jointure)

On peut aussi vouloir réunir les informations concernant les mêmes pays à travers plusieurs tables, par exemple :

```
pib = [['France', '2900'], ['Espagne', '1600']]
```

il faut alors rajouter le nouveau champs ligne par ligne (avec `None` pour les informations manquantes) :

```
pays = [['France', 'Paris', '68'],
        ['Espagne', 'Madrid', '48'],
        ['Italie', 'Rome', '60']]

pib = [('France', '2900'),
       ('Espagne', '1600')]

for ligne_pays in pays:      # pour chaque ligne de pays
    for ligne_pib in pib:    # parcourir les lignes de pib
        if ligne_pays[0] == ligne_pib[0]: # si c'est le meme pays
            ligne_pays.append(ligne_pib[1]) # alors on ajoute le pib à la fin
            break                        # inutile de continuer à parcourir pib
    if len(ligne_pays) == 3: # si on n'a pas trouvé de pib pour ce pays
        ligne_pays.append(None) # on renseigne avec None
```

A noter la dernière ligne qui permet de conserver l'intégrité de la table en ajoutant un champ `None` si le PIB d'un pays n'est pas présent.

Pour un tableau de p-uplets, les p-uplets sont immuables et la méthode `.append()` ne fonctionne pas, il faut créer un nouveau p-uplet pour chaque ligne, par exemple:

```
pays = [('France', 'Paris', 68),
        ('Espagne', 'Madrid', 48),
        ('Italie', 'Rome', 60)]

pib = [('France', 2900), ('Allemagne', 1600)]

for ligne_pays in range(len(pays)):      # pour chaque ligne de pays
    for ligne_pib in pib:                # parcourir les lignes de pib
        if pays[ligne_pays][0] == ligne_pib[0]: # si c'est le meme pays
            pays[ligne_pays] = pays[ligne_pays] + (ligne_pib[1], ) # on ajoute le pib à la fin
            break                        # inutile de continuer à parcourir pib
    if len(pays[ligne_pays]) == 3 :      # on n'a pas trouvé de pib
        pays[ligne_pays] = pays[ligne_pays] + (None, ) # on complete le puplet avec None
```

Et pour un tableau de dictionnaire, on rajoute un couple de clé-valeur à chaque dictionnaire du tableau :

```
pays = [{'Capitale': 'Paris', 'Pays': 'France', 'Population (ml)': '68'}, \
        {'Capitale': 'Madrid', 'Pays': 'Espagne', 'Population (ml)': '48'}, \
        {'Capitale': 'Rome', 'Pays': 'Italie', 'Population (ml)': '60'}]

pib = [('France', 2900), ('Espagne', 1600)]

for ligne_pays in pays:                # pour chaque ligne de pays
    for ligne_pib in pib:                # parcourir les lignes de pib
        if ligne_pays['Pays'] == ligne_pib[0]:    # si c'est le meme pays
            ligne_pays['PIB'] = ligne_pib[1] # alors on ajoute le pib à la clé PIB
            break                                # inutile de continuer à parcourir pib
        if len(ligne_pays) == 3 : ligne_pays['PIB'] = None    # on n'a pas trouvé de pib
```

1. Ce n'est pas nécessaire pour un tableau de dictionnaires. [←](#)