

# Exercice - Programmation Objet en Python

Objectif : Se familiariser avec la programmation Objet en Python.

Source: Frédéric Amblard, Philippe Muller, Maxime Chupin, Benoit Gaudou

## ***Exercice - Réseau social***

### ***A. Génération du réseau***

Nous souhaitons représenter et manipuler un objet représentant un réseau social, c'est-à-dire un ensemble d'utilisateurs reliés entre eux par des liens d'amitié.

La structure principale du réseau social sera un dictionnaire, contenant comme clé le nom de l'utilisateur et comme valeur sa liste d'amis.

Initialement, le dictionnaire est vide.

1. Définir la classe réseau social et son constructeur (il ne prend aucun paramètre et crée un réseau vide).
2. Ajouter une méthode `adduser(self, user)` qui ajoute un utilisateur dans le réseau (c'est-à-dire dans les clés du dictionnaire). Un user sera une chaîne de caractère.
3. Redéfinissez la méthode prédéfinie permettant d'afficher le réseau social.
4. Testez en créant un réseau, ajoutant 3 utilisateur et afficher le résultat.
5. Ajouter la méthode `addlink(self, user1, user2)` qui ajoute une lien d'amitié en `user1` et `user2` : l'`user1` est ajouté à la liste des amis de `user2` et réciproquement. Si un des utilisateurs n'est pas déjà présent dans le réseau, il faut l'ajouter.  
Attention, il ne doit pas y avoir de doublon dans la liste d'amis.
6. Ajouter la méthode qui renvoie tous les utilisateurs du réseau

### ***B. Analyses***

1. Définir une méthode qui vérifie si un utilisateur est ami avec un autre utilisateur.
2. Définir une méthode qui renvoie l'utilisateur qui a le plus d'amis.
3. Définir une méthode qui renvoie la liste des amis communs entre deux utilisateurs.
4. Définir une méthode qui prend un utilisateur et qui renvoie l'utilisateur qui a le plus d'ami commun avec lui (et qui n'est pas déjà son ami).
5. [Difficile] Définir une méthode qui calcule la distance relationnelle entre deux utilisateurs.