

## Algorithmique / Python / Structures de Données

### Rappels...

Manuel Munier

*Version 12 septembre 2022 (12:53)*

## Objectifs

L'objectif de ce premier TP est de faire un rappel de vos compétences en algorithmique et en programmation acquises au cours de votre première année de BUT. Vous travaillerez individuellement afin que chacun d'entre vous puisse prendre conscience de ses (éventuelles) lacunes, rattraper son (éventuel) retard et progresser en programmation sans devoir attendre son binôme/groupe/promo. Nous vous conseillons donc vivement de prendre ce premier TP très au sérieux car ces bases de la programmation vous seront essentielles pour la suite de votre cursus (et pas qu'en BUT!) :

- variables, structures de données (tableaux, tuples, listes)
- structures de contrôle (tests, boucles, structures imbriquées, ...)
- sous-programmes (fonctions, avec passage de paramètres et retour de résultat, visibilité des variables)
- concevoir un algo à partir d'un énoncé, le traduire en Python
- compiler, corriger et tester un programme

## Consignes de programmation

Voici quelques consignes (de bon sens) pour prendre de bonnes habitudes en programmation :

- Pensez à commenter votre code.
- Au début de chaque fonction indiquez en commentaire le rôle de chaque variable passée en paramètre et le type du résultat qui sera retourné.
- Une fonction ne doit pas effectuer d'entrées/sorties, c'est-à-dire ne doit pas demander à l'utilisateur de saisir des valeurs au clavier ou lui afficher des informations dans le terminal. Sauf, bien évidemment, les fonctions dont le rôle est explicitement de saisir une valeur ou d'afficher une information. ...
- Prenez l'habitude que vos fonctions s'exécutent jusqu'à la fin de leur code dont la dernière instruction est le `return`. Évitez à tout prix de faire un `return` dans une boucle ou dans un test ; ça fonctionne, certes, mais c'est source de gros problèmes lors du debugage. ...

## Travail à réaliser

Dans cet exercice on se propose de programmer en Python un ensemble de fonctions nécessaire à la réalisation d'une application de gestion des étudiants et de leurs notes. Nous nous limiterons à des structures de données "classiques" (tableaux, tuples, listes, etc.) ; il n'est pas question dans cet exercice d'utiliser des objets, classes et méthodes.

Voici dans les grandes lignes les principaux points à développer dans ce premier TP :

1. Définir la structure de données que vous allez utiliser pour représenter un étudiant avec son nom, son prénom, la liste de ses notes avec les coefficients de chaque note. Coder quelques exemples pour tester et valider cette structure.
2. Définir la structure de données que vous allez utiliser pour représenter une promotion avec son intitulé et sa liste d'étudiants (cf. structure définie ci-dessus). Coder quelques exemples pour tester et valider cette structure.
3. Fonctionnalités sur les étudiants :
  - (a) Écrire une fonction qui permet d'insérer une nouvelle note avec son coefficient à un étudiant passé en paramètre (on passe la structure, pas uniquement le nom de l'étudiant).
  - (b) Écrire une fonction qui retourne le nombre de notes d'un étudiant passé en paramètre.
  - (c) Écrire une fonction qui calcule la moyenne d'un étudiant passé en paramètre.
4. Fonctionnalités sur les promotions :
  - (a) Écrire une fonction qui permet d'insérer un nouvel étudiant dans une promotion.
  - (b) Écrire une fonction qui retourne le nombre d'étudiants d'une promotion passée en paramètre.
  - (c) Écrire une fonction qui calcule la moyenne d'une promotion passée en paramètre, en excluant bien évidemment les étudiants qui ne disposeraient encore d'aucune note.
5. Plus, bien évidemment, des programmes de test en tout genre, des fonctions diverses et variées pour déboguer (ex : affichage), etc.