

Algorithmique / Python / Structures de Données

Rappels...

Manuel Munier

Version 18 septembre 2025 (10:36)

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 résultats d'équipes de foot et de championnats. Nous nous limiterons à des structures de données "classiques" (tableaux, tuples, listes, dictionnaires, etc.) ; il n'est pas nécessaire dans cet exercice d'utiliser des objets, classes et méthodes.

Voici dans les grandes lignes les principaux points à développer dans cet exercice :

1. Écrire une fonction `creerEquipe(nom)` pour définir la structure de données que vous allez utiliser pour représenter une équipe avec :
 - son nom (`nom`)
 - le nombre de matchs joués (`MJ`)
 - le nombre de victoires (`V`)
 - le nombre de matchs nuls (`N`)
 - le nombre de défaites (`D`)
 - le nombre de points (`PTS`)
2. Écrire une fonction `enregistrerResultat(equipe1, equipe2, buts1, buts2)` qui met à jour les statistiques des 2 équipes après un match. Rappel des règles :
 - victoire = 3 points
 - match nul = 1 point
 - défaite = 0 points
3. Écrire une fonction `pointsEquipe(equipe)` qui retourne le nombre de points de l'équipe passée en paramètre.
4. Écrire la fonction `creerChampionnat(nom)` pour définir la structure de données que vous allez utiliser pour représenter un championnat avec son nom (`nom`) et la liste des équipes sous forme d'un tableau (`equipes`).
5. Écrire la fonction `ajouterEquipe(champ, equipe)` permettant d'ajouter une équipe dans un championnat.
6. Écrire la fonction `meilleureEquipe(champ)` qui retourne la meilleure équipe du championnat indiqué en paramètre.
7. Écrire la fonction `classement(champ)` qui retourne liste des équipes triées par points décroissants (sous forme d'un tableau).
8. Plus, bien évidemment, des programmes de test en tout genre, des fonctions diverses et variées pour déboguer (ex : affichage), etc.