

# SAE Python 1

Lors de cette SAE il nous était demandé de compléter diverses fonctions permettant de retrouver des informations à partir des résultats de matchs internationaux se trouvant dans des fichiers CSV. Et également de faire un affichage permettant d'interroger ces fonctions, en plus d'autres fonctions que nous étions libre de créer. Toutes les fonctions devaient être testées via des tests unitaires, comporter des docstrings et des commentaires.

J'ai réussi à implémenter toutes les fonctions demandées initialement de manière à ce qu'elles respectent les tests initiaux. J'ai également complété les tests manquants. Pour certaines fonctions qui avaient des buts similaires comme `plus_gros_scores` et `matchs_spectaculaires`, qui sont des recherches de maximum mais qui retourne une liste plutôt qu'une valeur. J'ai implémenté une fonction `max_liste` qui prend en paramètre une liste, et une clé qui est une fonction qui récupère l'attribut que l'on souhaite comparer (par exemple `lambda match: match[3] + match[4]` pour `matchs_spectaculaires`)

Enfin j'ai fait l'interface de consultation des CSV dans le terminal. Cette interface permet de choisir un fichier CSV se trouvant dans un dossier nommé "csv" (la gestion des chemins est effectuée avec le module "pathlib" afin que cela marche sur Windows tout comme sur Linux). Et d'effectuer diverses consultations:

1. Consulter les statistiques d'un match
2. Consulter les statistiques d'une équipe
3. Consulter les statistiques d'une compétition
4. Consulter les statistiques d'une ville
5. Consulter les statistiques globales

Pour chaque action, des exemples de consultations sont données à l'utilisateur. Cependant les fonctions fournies initialement ne permettaient pas de consulter chaque information de cette liste. C'est pourquoi j'ai dû rajouter des fonctions en plus (qui sont également soumises à des tests): `ensemble_des_compétitions`, `ensemble_des_villes`, `rechercher_par_date`, `nb_but`, et `est_un_entier` (fonction utilisé pour la navigation dans les menus).

L'interface utilise des listes numérotées pour l'affichage, mais aussi des tableaux pour afficher les matchs.

Je n'ai globalement pas eu de difficulté à effectuer le travail demandé, j'ai utilisé beaucoup de notions vus en cours tel que le try except, l'open, etc... Mais également d'autres notions comme le fonctionnel, que j'ai mentionné plus haut. Ainsi que d'autres structures de données comme les sets, et les dictionnaires pour la fonction `meilleures_equipes`. Que j'avais eu besoin de refaire lors du développement de

l'interface car elle prenait plus d'une minute à traiter `histoire1.csv`, contre quelques secondes maintenant.

J'ai mis en œuvre la compétence "Développer des applications informatiques simples" en implémentant les diverses fonctions demandées du fichier "`histoire2foot.py`". Par exemple la fonction `nombre_moyen_buts` qui a pour but de calculer le nombre moyen de buts marqués lors d'une compétition. Elle est documentée avec une docstring, les types des paramètres et du retour, et est commentée avec les invariants de boucles. La fonction est testée avec différents tests qui prennent plusieurs cas en compte, déjà avec les listes d'exemples données, pour s'assurer que le résultat est bon. Mais aussi avec des cas plus "extrêmes" comme avec la liste vide, ou avec des noms de compétitions qui n'existent pas, pour s'assurer qu'on ne fasse pas de division par zéro. Celle-ci respecte ce qui était demandé, en prenant en paramètre une liste de matchs et le nom d'une compétition et retourne un float. Et malgré le fait que j'ai eu besoin de la modifier afin qu'elle puisse également donner la moyenne peu importe la compétition, j'ai fait en sorte qu'elle marche quand même comme elle était originalement pensé en ajoutant un paramètre par défaut au nom de la compétition.