

# Travail pratique 2

IFT 1004 - Introduction à la programmation

Date de remise: au plus tard le 27 octobre 2017 à 23h50.

Pondération de la note finale: 6%

**Objectifs** Pratiquer la décomposition fonctionnelle, l’usage de libraires utilitaires, des listes, des dictionnaires, des chaînes de caractères et des fichiers.

## 1 Travail à faire

Vous devez programmer un gestionnaire d’agenda. Un agenda sera représenté dans votre programme par un dictionnaire ayant trois clés: *proprio*, *evenements* et *max\_id*. La valeur liée à *proprio* sera le nom du propriétaire de l’agenda. La valeur liée à *evenements* sera une liste contenant des événements, où chaque événement est aussi un dictionnaire avec les champs *id* (l’identifiant de l’évènement), *date* (la date de l’évènement), *heure\_debut* (l’heure à laquelle l’évènement débute), *heure\_fin* (l’heure à laquelle l’évènement se termine), *titre* (le titre ou le nom de l’évènement) et *lieu* (le lieu où se déroule l’évènement). La valeur liée à *max\_id* sera un entier qui permettra de garder le compte de l’identifiant le plus grand des événements.

Pour vous aider à bien faire votre travail, nous avons effectué une décomposition fonctionnelle de votre programme. Les fonctions issues de la décomposition sont documentées et fournies dans le fichier *tp2.py* fourni avec cet énoncé mais leur code est manquant. Vous aurez besoin de faire usage des classes **date** et **time** du module **datetime** dans ce travail. Nous vous renvoyons à la documentation disponible (<https://docs.python.org/3/library/datetime.html>) pour prendre connaissance de la manière de créer et manipuler des dates et des heures au travers de ce module.

## 2 Remise:

Votre code doit être rédigé dans le même fichier Python que nous vous avons fourni *tp2.py*. Le fichier doit être compressé dans une archive Zip (tp2.zip). Si vous avez du mal à y arriver ou si vous êtes confus, nous vous invitons à consulter les auxiliaires lors des TDs ou encore le CARÉ. Vous pouvez également demander de l’aide aux autres étudiants sur le forum du cours. Assurez-vous que vous remettez le bon fichier. Nous ne pourrions pas donner de points à votre travail si vous remettez le mauvais fichier. **Aussi noter qu’un programme qui n’est pas fonctionnel (qui ne s’exécute pas ou qui plante à l’exécution) pourrait recevoir une note de 0.**

## 3 Évaluation

Votre travail sera noté sur 100 comme suit:

Éléments évalués	Pondération
str_to_date	6 points
str_to_heure	6 points
creer_agenda	6 points
creer_evenement	6 points
evenement_to_str	6 points
sauvegarder_agenda	8 points
charger_agenda	12 points
saisir_evenement	12 points
evenements_en_conflit	8 points
ajouter_evenement	12 points
supprimer_evenement	8 points
Qualité du code(noms de variables, indentations, commentaires)	10 points

## 4 Remarques additionnelles

**Plagiat** : Tel que décrit dans le plan de cours, le plagiat est strictement interdit. Ne partagez pas votre code source à quiconque. Une politique stricte de tolérance zéro est appliquée en tout temps et sous toute circonstance. Tous les cas détectés seront référés à la direction de la faculté. Des logiciels comparant chaque paire de TPs pourraient être utilisés pour détecter les cas de plagiat.

**Retards** : Tout travail remis en retard peut être envoyé par courriel à l'enseignant si le portail des cours n'accepte plus les remises. Voir le plan de cours pour les pénalités.

**Remises multiples** : Il vous est possible de remettre votre TP plusieurs fois sur le portail des cours. La dernière version sera considérée pour la correction.

**Respect des normes de programmation** : Nous vous demandons de prêter attention au respect des normes de programmation établies pour le langage Python, notamment de nommer vos variables et fonctions en utilisant la convention suivante : `ma_variable`, `fichier_entree`, etc. Utiliser PyCharm s'avère être d'une très grande utilité, car celui-ci nous donne des indications sur la qualité de notre code (en marge à droite, et souligné).