

Travail pratique # 1

Une vie pleine d'essence !

Enseignant : Cem Subakan

Préparé par : Honoré Hounwanou

Date de remise : au plus tard le 05 Juin 2025 à 23h59

Pondération de la note finale : 6%

1 Objectifs

Ce travail a comme principaux objectifs de vous aider à assimiler les concepts vus dans les premières semaines de cours : l'affectation de variables, les types de données de base, les entrées/sorties, les structures conditionnelles et les structures itératives.

- Ce travail pratique est à faire **OBLIGATOIREMENT** de façon **INDIVIDUELLE**.
- Interdiction de se servir de l'IA générative pour vous assister dans la réalisation de ce travail.
- Interdiction d'utiliser des fonctions natives (« *built-in* ») Python, à l'exception de celles vues en classe durant les trois premiers modules (vous pouvez par exemple utiliser les fonctions `print`, `input`, `range`, `int`, `str`, `float`, `abs`, etc). L'utilisation de la fonction `round` est également autorisée.
- Interdiction d'utiliser les listes Python (`list`), les dictionnaires Python (`dict`), ainsi que leurs méthodes respectives.
- Interdiction d'utiliser les mots-clés `break` et `continue`.
- Autre que cela, vous avez le droit d'utiliser toutes les notions vues en classe durant les trois premiers modules : les *f-strings*, les structures conditionnelles `if-elif-else`, la boucle `while`, la boucle `for`, les tuples, etc.
- Bien que cela soit abordé au module 4, vous êtes libre de créer vos propres fonctions si cela peut améliorer la lisibilité de votre programme. Cependant, il est important de noter que cela ne vous accordera aucun point supplémentaire.
- Il se peut que la description du TP1 présentée ci-dessous manque de précision sur certains aspects. Si c'est le cas, il est de votre responsabilité de le remarquer et de demander des précisions sur le forum du cours.

2 Le problème à résoudre

Il y a de cela deux (2) ans, vous vous êtes acheté une auto flambant neuve modèle *SUBAKAN-FLASH-1004*. Cependant, votre auto est un peu spéciale dans le sens où vous êtes obligé de refaire le plein à chaque début de journée pour vous rendre au boulot.

Malheureusement pour vous, vous travaillez tous les jours de la semaine (7j/7) et avec la fluctuation du prix de l'essence, tout ceci requiert un sacré budget.

Comme un(e) grand(e), vous décidez de prendre une décision formelle : ne pas aller travailler si le prix de l'essence excède un certain montant que vous vous êtes fixé comme maximum.

Le programme que vous allez écrire vous permettra de savoir exactement :

1. quand est-ce que vous allez devoir aller travailler (i.e. lorsque le prix de l'essence sera inférieur ou égal à la limite que vous vous êtes fixée) ;
2. quand est-ce que vous allez devoir rester à la maison (i.e. lorsque le prix de l'essence sera supérieur à la limite que vous vous êtes fixée) ;
3. la consommation totale de votre auto (en litres) sur la période évaluée ;
4. le montant total dépensé (en dollars) sur la période évaluée ;
5. le nombre total de jours passés à la maison sur la période évaluée.
6. le nombre total de jours de travail sur la période évaluée.

Un exemple d'exécution du programme final devrait ressembler à cela :

Entrez votre prénom : Marie-Frédéric

Entrez la capacité maximale de votre réservoir (en litres) : 50

Entrez la période d'évaluation (en semaines) : 2

Entrez le montant maximum que vous êtes prêt à payer par litre (en dollars) : 1.17

Semaine 1

=====

Jour 1 : Le prix sur l'afficheur est de 107.9 cents.

Marie-Frédéric vient de faire le plein pour aller au boulot.

Jour 2 : Le prix sur l'afficheur est de 103.8 cents.

Marie-Frédéric vient de faire le plein pour aller au boulot.

Jour 3 : Le prix sur l'afficheur est de 131.1 cents.

Marie-Frédéric a décidé de rester à la maison.

Jour 4 : Le prix sur l'afficheur est de 108.6 cents.

Marie-Frédéric vient de faire le plein pour aller au boulot.

Jour 5 : Le prix sur l'afficheur est de 127.4 cents.

Marie-Frédéric a décidé de rester à la maison.

Jour 6 : Le prix sur l'afficheur est de 137.1 cents.
Marie-Frédéric a décidé de rester à la maison.

Jour 7 : Le prix sur l'afficheur est de 143.5 cents.
Marie-Frédéric a décidé de rester à la maison.

Semaine 2

=====

Jour 1 : Le prix sur l'afficheur est de 136.6 cents.
Marie-Frédéric a décidé de rester à la maison.

Jour 2 : Le prix sur l'afficheur est de 123.0 cents.
Marie-Frédéric a décidé de rester à la maison.

Jour 3 : Le prix sur l'afficheur est de 148.4 cents.
Marie-Frédéric a décidé de rester à la maison.

Jour 4 : Le prix sur l'afficheur est de 126.2 cents.
Marie-Frédéric a décidé de rester à la maison.

Jour 5 : Le prix sur l'afficheur est de 107.4 cents.
Marie-Frédéric vient de faire le plein pour aller au boulot.

Jour 6 : Le prix sur l'afficheur est de 107.9 cents.
Marie-Frédéric vient de faire le plein pour aller au boulot.

Jour 7 : Le prix sur l'afficheur est de 109.6 cents.
Marie-Frédéric vient de faire le plein pour aller au boulot.

=====

Consommation totale : 300.00 litres.

Montant total dépensé : 322.60\$.

Nombre de jours de travail : 6.

Nombre de jours passés à la maison : 8.

PS : J'avoue qu'il aurait été plus simple pour vous de changer d'auto. Mais croyez-moi, vous aurez tout le temps pour y réfléchir après avoir rendu votre TP.

3 Quelques précisions

Le prix de l'essence sera une valeur réelle entre 102 et 150 générée de façon aléatoire. Cette valeur devra avoir une unique décimale (voir exemple d'exécution).

Le fichier **tp1.py** qui vous a été fourni contient déjà du code vous permettant de générer cette valeur aléatoire. Vous êtes donc libre de l'utiliser ou non.

```
from random import uniform

ESSENCE_PRIX_MIN = 102
ESSENCE_PRIX_MAX = 150

prix_essence_du_jour = round(uniform(ESSENCE_PRIX_MIN, ESSENCE_PRIX_MAX), 1)

print(prix_essence_du_jour)
```

4 Modalités d'évaluation

Ce travail sera évalué sur 11 points, et la note sera ramenée sur 6. Voici la grille de correction :

Élément	Pondération
Le programme affiche le texte pour faire les entrées d'informations demandées	1 point
Le programme calcule la bonne quantité de litres consommée	1 point
Le programme calcule le bon montant total dépensé	1 point
Le programme affiche le montant total dépensé en dollars et non en cents	1 point
Le programme calcule et affiche le bon nombre de jours passés à la maison	1 point
Le programme calcule et affiche le bon nombre de jours de travail	1 point
Le programme affiche les valeurs avec 2 décimales	1 point
Qualité du code (noms de variables, style, commentaires, documentation)	2 points

Notez 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.

Votre programme doit être rédigé **à même le fichier Python fourni**, que vous devez compresser dans une archive Zip (fichier avec extension **.zip**) avec les autres fichiers à rendre. Si vous avez du mal à y arriver ou si vous êtes confus, nous vous invitons à réécouter la vidéo fournie en amont, où nous avons discuté de ces détails de fond en comble. Vous pouvez également demander de l'aide sur le forum du cours. **Assurez-vous que vous remettez tous les fichiers nécessaires à l'exécution de votre TP.** Nous vous recommandons fortement de créer un **dossier** dans lequel vous mettrez les fichiers relatifs à votre TP, et **rien d'autre** (certains d'entre vous ont la fâcheuse habitude de ne pas organiser leurs fichiers dans des dossiers). Nous ne pourrions pas donner de points à votre travail si vous remettez le mauvais fichier.

Utilisation de Git (2 points)

Vous devez compléter la partie intitulée « *Lien vers le dépôt GitHub PRIVÉ de votre TP :* » dans le fichier `correction.txt` fourni avec cet énoncé. Un total de 2 points sera attribué pour tout dépôt GitHub **privé** illustrant une utilisation de Git tout au long du développement de ce premier travail pratique.

5 Stratégie suggérée

- Résolvez les problèmes dans un premier temps en utilisant du papier et un crayon. Une bonne pratique de programmation consiste à formuler une ébauche de solution avant de se lancer dans l'écriture de code. Réfléchissez donc à un pseudo-code.
- Écrivez ensuite une version simple de votre programme en Python. Exécutez le programme fréquemment et trouvez les sources d'erreur. Ajoutez une fonctionnalité à la fois en répétant le cycle exécution/débogage.

Si vous avez du mal à programmer votre solution, voici quelques questions que vous devez vous poser et ensuite vérifier votre programme :

1. Quelles sont les informations que notre programme doit stocker en mémoire, et comment pouvons-nous les conserver pour les réutiliser plus tard ?
2. Quel devrait être le type de ces données ?
3. Quelle structure étudiée en classe nous permettra d'effectuer les opérations demandées ?

6 Remarques

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 : Les travaux pratiques doivent être impérativement remis via le portail de cours. Aucune remise par courriel n'est acceptée. Un travail qui ne respecte pas les directives de remise se verra pénalisé de 20%. Tout travail remis en retard se verra pénalisé de 25% par jour de retard. Chaque journée de retard débute dès la limite de remise dépassée (dès la première minute). Un retard excédant 2 jours provoquera le rejet du travail pour la correction et la note de 0 pour ce travail.

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`, `mon_autre_variable`, etc. Utiliser **PyCharm** s'avère être une très bonne idée ici, car celui-ci nous donne des indications sur la qualité de notre code (en marge à droite, et souligné).

7 Ce que vous devez rendre

Vous devez remettre une archive `.zip` d'un **dossier**, contenant **uniquement** les fichiers suivants :

- **tp1.py** : Un fichier Python contenant votre programme. Les correcteurs doivent être en mesure d'exécuter ce fichier sans lui apporter de modifications.
- **correction.txt** : Le fichier de correction fourni avec l'énoncé, que vous devez compléter en y indiquant votre nom, votre NI, le lien vers le dépôt GitHub **PRIVÉ** de votre TP, et le nombre d'heures que vous avez consacré à compléter votre TP.

Cette archive doit être remise via le site Web du cours. Une pénalité de 5% pourra être appliquée si vous remettez des fichiers inutiles.

8 Pour ceux qui veulent en faire plus

Votre travail sera évalué selon les modalités énoncées ci-dessus. Toutefois, vous pouvez ajouter les fonctionnalités optionnelles suivantes :

1. Donner la possibilité de relancer le programme : *Souhaitez-vous faire une autre évaluation ? (O/N)*.
2. Afficher le message *"Les factures ne vont pas se payer toutes seules **NOM DE LA PERSONNE!**"* s'il s'avère que la personne a eu à passer plus de la moitié de la période d'évaluation à la maison.
3. Valider les différentes informations et les redemander en cas d'erreur de validation.

Bon travail !