

TP1 : Un pèse personnes intelligent

1. Objectif final :

- Analyser un problème,
- Pratiquer les structures élémentaires de programmation en python,
- Réaliser un premier projet.

2. Objectifs intermédiaires :

- Résoudre un problème en utilisant une méthodologie de développement,
- Savoir extraire les données utiles à la résolution,
- Élaboration d'algorithmes,
- Utiliser les structures séquentielles,
- Utiliser les structures conditionnelles et les structures de répétition et boucles
- Utiliser des fonctions pour le métabolisme de base et Indice de Masse Corporelle

3. Projet :

Vous avez été désigné comme développeur dans une équipe pour réaliser une études sur le suivi du poids d'une population. L'objectif est de les aider à gérer leur poids et se prendre en charge en mangeant santé. Le produit final sera déployé dans des hôpitaux, cliniques, etc. **L'application doit calculer l'IMC et le métabolisme de base (MB) pour toute les personnes de l'étude.**

Nom	Olivier	Claire	Damien
Genre	h	f	H
Age	35	28	65
Poids (Kg)	78	64	89
Taille (m)	1.85	1.74	1.71

4. Métabolisme de base :

Le métabolisme de base (MB) est la quantité d'énergie dont l'organisme a besoin au repos. En d'autres termes, c'est l'énergie requise par l'organisme pour effectuer ses fonctions de base, comme respirer, faire battre le cœur et maintenir une température corporelle stable. Pour une personne sédentaire, il permet de connaître leur besoin calorique quotidien.

$MB(\text{Homme}) = 13,707 * \text{Poids (kg)} + 492,3 * \text{Taille (m)} - 6,673 * \text{Age (an)} + 77,607$

$MB(\text{Femme}) = 9,740 * \text{Poids (kg)} + 172,9 * \text{Taille (m)} - 4,737 * \text{Age (an)} + 667,051$

5. Indice de masse corporel :

IMC (Indice de Masse Corporelle) est une formule qui permet de mesurer l'indice de masse corporelle.

$$IMC = \frac{\text{poids}}{\text{taille}^2} \quad (\text{poids} / (\text{taille} * \text{taille}))$$

L'IMC se calcule en divisant le poids exprimée en kilogrammes par la taille (exprimée en mètres) au carré. l'indice de masse corporelle et risque de maladie selon l'Organisation mondiale de la santé :

Classification	Indice de masse corporelle (kg/m2)	Risque de maladies
Maigreur	Moins de 18,99	Accru
Poids normal	19,0 à 24,99	Faible
Embonpoint	25,0 à 29,99	Accru
Obésité modérée	30,0 ou plus	Élevé

Attention encore, cette classification reste statistique, elle ne s'applique pas forcément à tous les adultes, notamment les sportifs ou les seniors. Ainsi, nous allons utiliser l'indice de masse grasse qui est plus précis.

6. Remise :

Un fichier word avec le code en python sur LÉA

7. Date de remise :

29/11/2022 23h59

Looking in indexes: <https://pypi.org/simple>, <https://us-python.pkg.dev/colab-wheels/public/simple/>
Requirement already satisfied: tabulate in /usr/local/lib/python3.7/dist-packages (0.8.10)

```
1 import csv
2
3 def get_data():
4     dataset = []
5
6     with open('dataset.csv', newline='') as csvfile:
7         reader = csv.DictReader(csvfile)
8         for row in reader:
9             dct = dict(nom = row['nom'], age = int(row['age']), poids_kg = float(row['poids_kg']), taille_m = float(row['taille_m']), genre :
10             dataset.append(dct)
11     return dataset
12
13 def get_metabolisme_base(dict_item):
14     if dict_item.get('genre') == 'h':
15         return (13.707 * dict_item.get('poids_kg') + 492.3 * dict_item.get('taille_m') - 6.673 * dict_item.get('taille_m') + 77.607)
16     else:
17         return (9.740 * dict_item.get('poids_kg') + 172.9 * dict_item.get('taille_m') - 4.737 * dict_item.get('taille_m') + 667.051)
18
19 def get_imc(dict_item):
20     return (dict_item.get('poids_kg')/(dict_item.get('taille_m')**2))
21
22 def get_classification(imc):
23     if imc <= 19:
24         return dict(classificateur = "Maigreur", risque_maladies = "Accru")
25     elif imc >= 19 and imc < 25:
26         return dict(classificateur = "Poids normal", risque_maladies = "Faible")
27     elif imc >= 25 and imc < 30:
28         return dict(classificateur = "Embonpoint", risque_maladies = "Accru")
29     elif imc >= 30:
30         return dict(classificateur = "Obésité modérée", risque_maladies = "Élevé")
31
32 dataset = get_data()
33
34 for it in dataset:
35     mb = get_metabolisme_base(it)
36     imc = get_imc(it)
37     it.update({"metabolisme_base": int(mb)})
38     it.update({"imc": int(imc)})
39     it.update(get_classification(int(imc)))
```

```
1 from tabulate import tabulate
2 print(tabulate(dataset, headers="keys"))
```

nom	age	poids_kg	taille_m	genre	mtabolisme_base	imc	classificateur	risque_maladies
Olivier	35	78	1.85	h	2045	22	Poids normal	Faible
Claire	28	64	1.74	f	1583	21	Poids normal	Faible
Damien	65	89	1.71	h	2127	30	Obésité modérée	Élevé

1