# Challenge 2024-05-31 : Classement par moyennes

Table des matières

[Challenge 2024-05-31 : Classement par moyennes 1](#_Toc168519328)

[1) Enoncé 3](#_Toc168519329)

[2) Logigramme des modules 4](#_Toc168519330)

[a. Logigramme simple du code 4](#_Toc168519331)

[3) Variables du code 5](#_Toc168519332)

[4) Code 6](#_Toc168519333)

[b. constantes.py 6](#_Toc168519334)

[c. main.py 7](#_Toc168519335)

[1) Résulta **Erreur ! Signet non défini.**](#_Toc168519336)

## Enoncé

Ce challenge est un niveau débutant, mais si vous avez plus d'expérience, vous pouvez essayer de trouver de belles astuces pour un code propre, rapide et concis. Voici la liste des élèves et leurs notes respectives:

Tao -> 18, 12, 3, 5, 19

Josette -> 20, 2, 12, 18, 14

Patrick -> 2, 4, 6, 18, 17

Pema -> 3, 19, 15, 3, 12

Jean -> 0, 9, 8, 8, 4

Bixente -> 14, 20, 10, 12, 4

Paco -> 16, 1, 1, 1, 20

Chuluun -> 15, 6, 17, 20, 15

Marie -> 16, 4, 16, 20, 12

Mohamed -> 16, 19, 17, 6, 20

Conditions :

Créez une fonction moyenne\_eleves() qui retourne un dictionnaire contenant en clé le nom de l'élève et en valeur sa moyenne arrondie à l'entier le plus proche.

Créez une fonction classement\_eleves() qui permettra d'afficher dans le terminal le classement complet de la classe avec la position de chaque élève par rapport aux autres.

Exemples :

{'Tao': 11, 'Josette': 13, 'Patrick': 9, 'Pema': 10, 'Jean': 6, 'Bixente': 12, 'Paco': 8, 'Chuluun': 15, 'Marie': 14, 'Mohamed': 16}

1 : Mohamed avec une moyenne de 16/20

2 : Chuluun avec une moyenne de 15/20

3 : Marie avec une moyenne de 14/20

4 : Josette avec une moyenne de 13/20

5 : Bixente avec une moyenne de 12/20

6 : Tao avec une moyenne de 11/20

7 : Pema avec une moyenne de 10/20

8 : Patrick avec une moyenne de 9/20

9 : Paco avec une moyenne de 8/20

10 : Jean avec une moyenne de 6/20

## Logigramme des modules

### Logigramme simple du code

Début

Main.py

Contient la logique du programme

Moyenne.py

Contient la fonction **moyennes\_eleves**

Classement.py

Contient la fonction **classement\_eleves**

Fin

Envoi la variable

LISTE\_ELEVES

à la fonction

Retourne la variable dict\_eleves

Envoi la variable dict\_eleves à la fonction

1

2

3

5

Retourne la variable sorted\_dict\_eleves

4

Afficher le classement comme l’énoncé

Constantes.py

Contient les variables globales

## Variables du code

|  |  |  |  |
| --- | --- | --- | --- |
| Modules | Nom de la variable | Type | Commentaires |
| constantes.py | LISTE\_ELEVES | list | Liste contenant le nom des élèves ainsi que leurs notes |
| EOR | Str | Permet de faire le retour à la ligne |
| main.py | dict\_eleves | Dict | Dictionnaire contenant le nom et la moyenne des élèves |
| main.py | sorted\_dict\_eleves | list | Liste classant par ordre décroissant la moyenne de chaque élève avec leur nom |

## Code

### constantes.py

# constantes.py  
# But:  
# Contient les constantes du code  
# -----------------------------------  
# Date de création: 2024-06-01  
# Date de dernière modification: 2024-06-02  
# ------------------------------------------  
# version: 2.0  
# - Changed LISTE\_ELEVES from list type to tuple type  
# - Added EOR  
#-------------------------------------------  
  
LISTE\_ELEVES = (  
 ("Tao"**,** (**18, 12, 3, 5, 19**))**,** ("Josette"**,** (**20, 2, 12, 18, 14**))**,** ("Patrick"**,** (**2, 4, 6, 18, 17**))**,** ("Pema"**,** (**3, 19, 15, 3, 12**))**,** ("Jean"**,** (**0, 9, 8, 8, 4**))**,** ("Bixente"**,** (**14, 20, 10, 12, 4**))**,** ("Paco"**,** (**16, 1, 1, 1, 20**))**,** ("Chuluun"**,** (**15, 6, 17, 20, 15**))**,** ("Marie"**,** (**16, 4, 16, 20, 12**))**,** ("Mohamed"**,** (**16, 19, 17, 6, 20**))  
)  
  
EOR = "\n"

### main.py

# main.py  
# But:  
# Contient la logique du programme  
# -----------------------------------  
# Date de création: 2024-06-01  
# Date de dernière modification: 2024-06-05  
# ----------------------------------------------------------------  
# version: 5.0  
# - Added EOR (V2)  
# - Compréhension liste pour l'affichage + fonction 'join' (V2)  
# - Déplacement de l'affichage du classement dans le module 'classement' (V3)  
# - Ajout de l'EOR + join dans la fonction 'classement\_eleves' (V4)  
# - Optimisation du code et conformité PEP8 (V5)  
#-------------------------------------------------------------------  
  
# Appel des modules externes  
  
# Appel des modules internes  
from constantes import LISTE\_ELEVES**,** EOR  
  
  
def moyenne\_eleves(LISTE\_ELEVES):  
 return {nom: int(round(sum(notes)/len(notes))) for nom**,** notes in LISTE\_ELEVES}  
  
def classement\_eleves(dict\_eleves):  
 sorted\_dict\_eleves = sorted(dict\_eleves.items()**,** key = lambda x:x[**1**]**,** reverse=True) #Trie par ordre décroissant  
 return EOR.join(  
 f"{index\_numb:2} : {nom\_eleve} avec une moyenne de {moy\_eleve}/20"  
 for index\_numb**,**(nom\_eleve**,**moy\_eleve) in enumerate(sorted\_dict\_eleves**, 1**)  
 )  
  
  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 dict\_eleves = moyenne\_eleves(LISTE\_ELEVES)  
 print(classement\_eleves(dict\_eleves))

## Résultat

