# Challenge 2024-05-31 : Classement par moyennes

Table des matières

[Challenge 2024-05-31 : Classement par moyennes 1](#_Toc168172500)

[1) Enoncé 3](#_Toc168172501)

[2) Logigramme des modules 4](#_Toc168172502)

[a. Logigramme simple du code 4](#_Toc168172503)

[3) Variables du code 5](#_Toc168172504)

[4) Code 6](#_Toc168172505)

[b. constantes.py 6](#_Toc168172506)

[c. main.py 7](#_Toc168172507)

[d. moyennes.py 8](#_Toc168172508)

[e. classement.py 9](#_Toc168172509)

[1) Résulta 10](#_Toc168172510)

## 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 |
| main.py  moyennes.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: 2024-06-01  
# ------------------------------------------  
# version: 1.0  
#-------------------------------------------  
  
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**]]  
 ]  
  
"""  
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  
"""

### main.py

# main.py  
# But:  
# Contient la logique du programme  
# -----------------------------------  
# Date de création: 2024-06-01  
# Date de dernière modification: 2024-06-01  
# ------------------------------------------  
# version: 1.0  
#-------------------------------------------  
  
# Appelle des modules externes  
  
# Appelle des modules internes  
from constantes import LISTE\_ELEVES  
from moyenne import moyenne\_eleves  
from classement import classement\_eleves  
  
if \_\_name\_\_ == '\_\_main\_\_':  
  
 dict\_eleves = moyenne\_eleves(LISTE\_ELEVES)  
 sorted\_dict\_eleves = classement\_eleves(dict\_eleves)  
 for index\_numb**,**infos\_eleves in enumerate(sorted\_dict\_eleves): #Afficher le classement des élèves  
 print(f"{index\_numb+**1**}: {infos\_eleves[**0**]} avec une moyenne de {infos\_eleves[**1**]}/20")

### moyennes.py

# # moyenne.py  
# But:  
# Contient la fonction moyenne\_eleves  
# => Renvoyer la valeur moyenne de chaque élève sous forme de dico  
# -----------------------------------  
# Date de création: 2024-06-01  
# Date de dernière modification: 2024-06-01  
# ------------------------------------------  
# version: 1.0  
#-------------------------------------------  
  
def moyenne\_eleves(LISTE\_ELEVES):  
 dict\_eleves={}  
 for eleves in LISTE\_ELEVES:  
 moyenne\_float = sum(eleves[**1**])/len(eleves[**1**])  
 moyenne = int(round(moyenne\_float**, 0**)) # Retourne la moyenne arrondi à l'entier le + proche  
 dict\_eleves[eleves[**0**]] = moyenne # Sauvegarde de la moyenne avec l'élève associé  
 return dict\_eleves

### classement.py

# classement.py  
# But:  
# Contient la fonction classement\_eleves  
# => Renvoyer le classement des élèves par rapport à leur moyenne sous forme de dico  
# -----------------------------------  
# Date de création: 2024-06-01  
# Date de dernière modification: 2024-06-01  
# ------------------------------------------  
# version: 1.0  
#-------------------------------------------  
  
def classement\_eleves(dict\_eleves):  
 return sorted(dict\_eleves.items()**,** key=lambda x:x[**1**]**,** reverse=True) #Trie par ordre décroissant

1. Résultat

