

I Mise en place

1.1 But du TD

Le but de ce TD est de créer un petit script qui permet de choisir un élève au hasard dans la classe, de manière plus ou moins élaborée, avec comme objectif de ramasser des copies de devoir.

Pour cela le fichier `elevés.py` vous est fourni, et il contient une liste nommée `elevés`. Pour y accéder dans un script ou dans la console :

```
1 import élevés # le nom du module
2 élevés.elevés # la liste élevés du module élevés
```

ou plus simplement, en utilisant une autre syntaxe de la commande d'import

```
1 from élevés import élevés # import seulement de la liste, pas du module
2 élevés # fait référence au nom importé : la liste
```

Dans les deux cas, python doit être capable de trouver le fichier `elevés.py`. Cela ne posera pas de problème si vous utilisez F5 pour envoyer directement le script que vous créez dans la console, et si ce fichier créé est dans le même répertoire que `elevés.py`.

1.2 Tester le code produit

Il est très important de toujours tester les fonctions que l'on vient de créer. Rappelons que pour ce faire vous devrez d'abord exécuter votre feuille de script, puis appeler les fonctions créées dans la console avec différentes valeurs du ou des paramètres.

1.2.1 Docstrings

Dans le fichier que vous allez devoir compléter (`tirage.py`), les fonctions à créer ont déjà une docstring, c'est à dire une description du comportement attendu, avec en plus des exemples d'exécution. Ces descriptions font partie intégrante de l'énoncé.

Ne supprimez pas ces docstrings, et écrivez votre code après (en dessous).

Les 3 dernières lignes du fichier permettent de vérifier que les exemples d'exécutions se déroulent comme prévu lorsque que vous exécutez le script via F5. Pour le moment vous obtenez des messages d'erreur. Pour supprimer les messages d'erreur, il suffit de commenter ces 3 lignes (en insérant un dièse au début de la ligne).

II Code

2.1 Approche naïve

La première approche est de choisir au hasard (chaque élève a autant de chances de passer à chaque fois). Dans l'aide de python (un lien vers la documentation python est disponible dans Spyder) chercher l'aide du module `random`. A vous de trouver l'outil pour compléter la fonction `tire_au_hasard` du fichier `tirage.py`.

2.2 Compter les passages

Une méthode peut-être plus juste consiste à compter les passages au tableau de chaque élève. On décide de créer une liste `passages` qui contient le nombre de passage de chaque élève. Plus précisément, le nombre d'indice i dans `passages` est le nombre de fois où l'élèves d'indice i dans `elevés` a déjà été choisi.

Complétez la fonction `choisi_eleve`. Pour les tests, il faudra créer une liste `passages` dans la console à passer comme paramètre. Une construction possible est donnée à la fin du fichier `elevés.py`

2.3 Tenir compte des passages

On décide dans un premier temps d'équilibrer le nombre de passages :

- Quand passage est rempli de 1, on remet tout ses éléments à 0.
- On ne peut choisir qu'un élève dont le nombre de passage courant est 0.

Complétez les fonction `remise_a_zero` et `choix_sans_remise`.

Testez la fonction de remise à zéro.

```
1 L = [2, 7, 9]
2 remise_a_zero(L) # pas de valeur de retour
3 L == [0, 0, 0]
```

2.4 Pour aller plus loin

On peut décider de raffiner le processus de choix. En particulier pour éviter la trop grande prévisibilité du système précédent. Voici une manière parmi d'autres.

- On ne remet plus passages à zéro.
- La probabilité d'être choisi dépend du nombre de passage déjà effectué : on calcule la moyenne de passages, et les élèves ayant un nombre de passage supérieur à cette moyenne ont moins de chances d'être de nouveau envoyés au tableau.
- Plus précisément, on décide qu'un élève étant passé au tableau en moyenne une fois de plus que les autres aura 2 fois moins de chances d'être re-choisi, et réciproquement qu'un élève étant passé 1 fois de moins en moyenne aura 2 fois plus de chances d'être choisi.

2.4.1 Lemmes

Pour cela il nous faut des outils techniques : complétez les fonction `moyenne` et `sommes_cumulees`.

2.4.2 On se lance

Votre rôle est de traduire en Python dans la fonction `choix_pondere` l'algorithme suivant.

1. Calculer la moyenne m de la liste `passages`.
2. Créer la liste **chances** des chances respectives d'être choisi (en proportion) : c'est la liste des 2^{m-p} où p prend toutes les valeurs de la liste `passages`. Evidemment, **chances[i]** correspond à **eleves[i]**.
Remarque : on voit que si $p = m$, alors la proportion de chance d'être choisi est 1, c'est à dire qu'un élève étant passé au tableau exactement autant de fois que la moyenne est notre élève de référence.
Si $p = m + 1$ (un passage de plus déjà effectué), alors $2^{m-p} = \frac{1}{2}$. Un élève étant passé une fois de plus que la moyenne à 2 fois moins de chances d'être re-choisi.
3. Créer la liste des probabilités d'être choisi (**probas** par exemple) en divisant chaque élément de **chances** par la somme des éléments de **chances**.
La somme des éléments de la liste **probas** doit être 1 aux erreurs d'arrondis près.
4. Créer la liste **cumul** des sommes cumulées de **probas**. Son dernier élément est 1 d'après la remarque précédente.
5. Tirer un nombre au hasard entre 0 et 1. On le note p .
6. On considère **cumul** comme une liste d'intervalle : Si p est entre 0 et **cumul[0]**, c'est l'élève 0 qui a été choisi, si p est entre **cumul[0]** et **cumul[1]**, c'est l'élève 1, etc. On convient d'exclure la borne supérieure de chaque intervalle ainsi considéré, sauf la dernière.
Trouver quel élève a été choisi.
7. Mettre à jour **passages**
8. Retourner le nom de l'élève choisi.